Osnove statistike

Kolegij za studente komunikologije

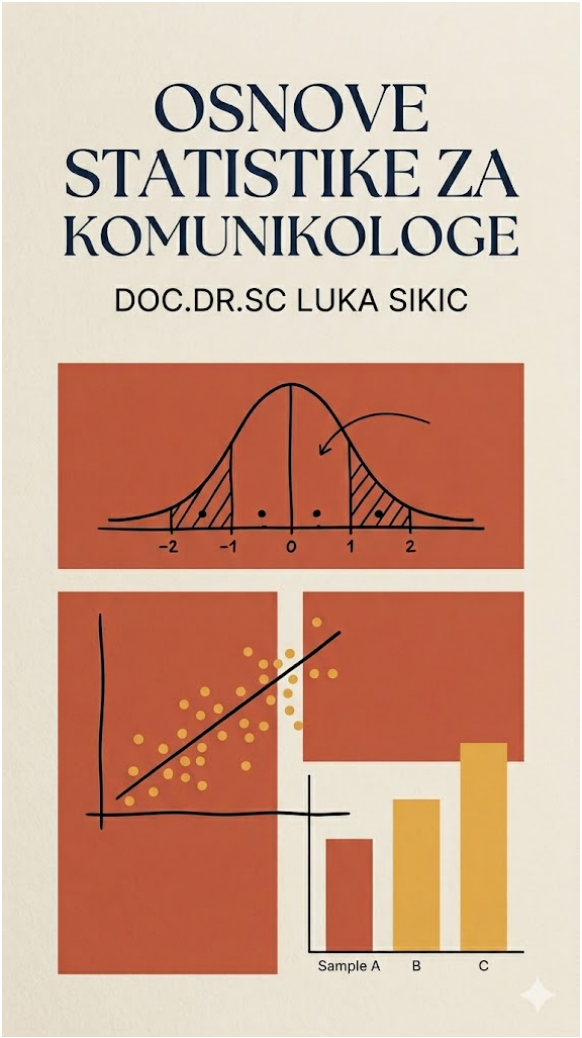
doc. dr. sc. Luka Šikić

2026-03-02

Table of contents

1 Osnove statistike

Kolegij za studente komunikologije



Statistika je jezik podataka, a podaci su danas svugdje. Ovaj kolegij pruža temeljna znanja iz statistike prilagođena studentima komunikologije, s naglaskom na praktičnu primjenu u analizi medijskih podataka. Kroz 13 tjedana naučit ćete prikupljati, analizirati i interpretirati

podatke koristeći programski jezik R i tidyverse ekosustav. Svi primjeri dolaze iz svijeta medija i komunikacija.

Osnovni podaci o kolegiju

Studij	Komunikologija, Hrvatsko katoličko sveučilište
Semestar	Ljetni semestar 2024/2025
ECTS	6
Opterećenje	30P + 30S
Nositelj	doc. dr. sc. Luka Šikić
Kontakt	luka.sikic@unicath.hr
Web	lukasikic.info · GitHub · LinkedIn
Konzultacije	Prema dogovoru putem emaila

1.1 Struktura kolegija

Kolegij je organiziran u četiri tematske cjeline koje prate logiku postupnog građenja znanja:

1.1.1 Cjelina 1

Uvod i osnove programiranja (tjedni 1 do 4)

Statističko razmišljanje, istraživački dizajn, programiranje u R-u, tidyverse ekosustav za učitavanje, čišćenje i transformaciju podataka.

1.1.2 Cjelina 2

Deskriptivna statistika i vizualizacija (tjedni 5 i 6)

Mjere centralne tendencije i raspršenosti, vizualizacija podataka pomoću ggplot2.

1.1.3 Cjelina 3

Statistička teorija (tjedni 7 do 9)

Teorija vjerojatnosti, distribucije, uzorkovanje, procjena parametara, intervali pouzdanosti i testiranje hipoteza.

1.1.4 Cjelina 4

Inferencijalna statistika (tjedni 10 do 13)

Hi-kvadrat testovi, t-testovi, ANOVA i linearna regresija na primjerima iz komunikologije.

1.2 Preuzmi cijelu knjigu

Sva predavanja u jednom dokumentu

Cijeli sadržaj kolegija dostupan je za preuzimanje kao jedinstveni dokument:
Preuzmi PDF

1.3 Brzi pristup

Resurs	Opis
Silabus	Detaljan plan kolegija s opisima svih 13 tjedana
Raspored	Tjedni raspored predavanja i seminara
Popis literature	Obavezna i preporučena literatura s anotacijama
Alati	Softver, R paketi i platforme za kolegij
Praktični projekt	Upute, scenariji i rubrika za završni projekt

1.4 Obavijesti

Kolegij je u pripremi. Sadržaj predavanja bit će objavljen postupno kako kolegij bude napredovao. Pratite ovu stranicu za ažuriranja.

Dio I

Uvod i osnove programiranja

2 Tjedan 1: Zašto statistika? Uvod u istraživački dizajn

Kako podaci mijenjaju način na koji razumijemo medije i komunikaciju

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti zašto su statističke metode neophodne u istraživanjima komunikacije i medija, čak i kad se problemi čine intuitivno jasnima.
2. Prepoznati situacije u kojima zdravorazumsko zaključivanje vodi na krivi trag i objasniti Simpsonov paradoks na primjeru medijskih podataka.
3. Razlikovati četiri razine mjerenja (nominalna, ordinalna, intervalna, omjerna) i prepoznati kojoj razini pripadaju tipične varijable u komunikološkim istraživanjima.
4. Objasniti pojmove pouzdanosti i valjanosti mjerenja te navesti primjere za svaki tip.
5. Razlikovati eksperimentalni i neeksperimentalni istraživački dizajn te objasniti prednosti i ograničenja svakog pristupa u kontekstu medijskih istraživanja.
6. Definirati ključne pojmove poput varijable, nezavisne i zavisne varijable, operacionalizacije, konfundirajuće varijable te interne i eksterne valjanosti.
7. Kritički procijeniti jednostavna istraživačka izvješća i prepoznati potencijalne prijetnje valjanosti zaključaka.

2.1 Imate li uopće izbora?

Zamislite da radite u novinarskoj redakciji i da vam šef kaže da naš portal gubi čitatelje i da trebamo promijeniti pristup. Vi pitate što točno ne valja, a on odgovara: ljudi ne klikaju na naše članke. I onda doda: mislim da su naslovi previše dosadni. Promijenite stil naslova, stavite više clickbaita, i sve će biti u redu.

Zvuči uvjerljivo. Ali razmislite na trenutak. Otkud šefu to znanje? Možda je u pravu. Možda su naslovi zaista problem. Ali možda je problem nešto sasvim drugo, poput vremena objave, duljine članaka, tematike, konkurentskih portala koji su lansirali novu aplikaciju, ili

činjenica da je ljeto i ljudi su na moru umjesto za računalom. Bez podataka, ova rasprava je čisto nagađanje. Svatko ima svoju teoriju, svi su uvjereni u svoje, a odluka se na kraju donosi na temelju toga tko je najglasnije govorio na sastanku.

Statistika postoji upravo zato da bismo izbjegli ovakve situacije. Njezin temeljni cilj nije komplicirati vam život formulama i grčkim slovima (iako se ponekad čini tako). Cilj je pružiti alate kojima se od gomile podataka dolazi do zaključaka u koje se možemo pouzdati, barem više nego u nagađanje šefa na jutarnjem sastanku. I tu dolazimo do pitanja iz naslova — imate li uopće izbora? Trebate li vi, kao budući komunikolog, zaista učiti statistiku?

Kratak odgovor je da. Duži odgovor zaslužuje objašnjenje.

2.2 Zašto komunikolog treba statistiku

Komunikologija je empirijska znanost. To znači da se njezine tvrdnje moraju temeljiti na dokazima, a ne samo na teoretskim razmatranjima ili osobnom iskustvu. Kad istraživač tvrdi da TikTok negativno utječe na pažnju adolescenata, ili da negativne vijesti generiraju više klikova od pozitivnih, ili da je povjerenje u medije palo nakon pandemije, te tvrdnje moraju biti podržane podacima. A podaci bez statističke analize su samo hrpa brojki.

Ali potreba za statistikom u komunikologiji ide dublje od akademskog istraživanja. Moderna medijska industrija je u potpunosti prožeta podacima. Svaki klik na web stranici, svako otvaranje newslettera, svaka sekunda provedena na video sadržaju se bilježi i analizira. Društvene mreže generiraju enormne količine podataka o ponašanju korisnika. Oglašivačka industrija troši milijarde na temelju statističkih modela koji predviđaju koji će oglas izazvati reakciju kod koje publike. Ako ne razumijete kako ti podaci nastaju, kako se analiziraju i što njihovi rezultati zapravo znače (i ne znače!), ne možete se ravnopravno uključiti u rasprave koje oblikuju medijski krajolik.

Navarro u svojoj knjizi, koja služi kao temelj za ovaj kolegij, koristi zgodnu analogiju. Kaže da je statistika pomalo poput učenja kuharskog recepta. Na prvu, čini se kao skup mehaničkih koraka u stilu dodaj ovo, izmiješaj ono, stavi u pećnicu na toliko stupnjeva. Ali kad počnete razumjeti zašto svaki korak postoji, zašto se maslac dodaje hladan a ne topao, zašto se tijesto ostavlja da odmara, odjednom prestajete pratiti recept i počinjete kuhati. Isto je sa statistikom: na početku izgleda kao skup formula i pravila, ali kad shvatite logiku iza toga, dobivate alat za razmišljanje koji mijenja način na koji gledate na bilo koju tvrdnju, u akademskom radu ili u svakodnevnom životu.

2.3 Kad nas intuicija iznevjeri

Jedan od najvažnijih razloga za učenje statistike jest taj da je naša intuicija o podacima i vjerojatnostima nevjerojatno nepouzdana. Ljudski mozak je izvanredno dobar u prepoznavanju uzoraka, ali taj isti mehanizam nas navodi da vidimo uzorke tamo gdje ih nema i da donosimo zaključke na temelju nedovoljnih ili pristrano odabranih informacija.

Razmislite o sljedećim primjerima.

Slučajnost ne izgleda onako kako mislimo. Zamislite da bacate novčić deset puta i dobijete ovakav niz: P G P G P G P G P G (pismo i grb se savršeno izmjenjuju). Većina ljudi bi rekla da ovo ne izgleda slučajno, da je previše pravilno. I bili bi u pravu, taj niz bi bio prilično neobičan. Ali pogledajte ovaj niz: P P P G P G G P P G. Većina ljudi bi rekla da ovo izgleda slučajno. A zapravo, svaki specifični niz od deset bacanja ima potpuno istu vjerojatnost. Ono što naš mozak radi jest to da uspoređuje niz s mentalnim modelom slučajnosti koji je netočan. Mi očekujemo da slučajnost izgleda ravnomjerno, ali slučajnost je zapravo neuredna.

Anegdota nije dokaz. Vaš susjed je počeo koristiti novu aplikaciju za vijesti i kaže da je sada puno bolje informiran. Čuli ste sličnu priču od dva kolege na fakultetu. Je li to dokaz da aplikacija zaista poboljšava informiranost? Naravno da nije. Tri osobe nisu reprezentativni uzorak. Možda su te tri osobe ionako bile natprosječno zainteresirane za vijesti. Možda bi se jednako dobro osjećale da su počele koristiti bilo koju novu aplikaciju. Možda je to jednostavno placebo efekt noviteta. Ali ljudski mozak, suočen s tri konzistentne priče, automatski zaključuje da mora biti nečeg u tome. To se u psihologiji zove heuristika dostupnosti: informacije koje su nam lako dostupne u sjećanju (poput živopisnih anegdota) percipiramo kao reprezentativnije nego što jesu.

Korelacija zavodi. Portali koji objavljuju više članaka dnevno imaju više ukupnih klikova. Znači li to da bismo trebali objavljivati više? Možda. Ali možda veći portali imaju i više novinara, veći budžet za marketing, stariju i lojalniju čitateljsku bazu, i jednostavno više resursa koji privlače čitatelje neovisno o broju članaka. Korelacija između broja članaka i broja klikova ne znači da jedno uzrokuje drugo. Ovo je toliko čest logički skok da se u statistici za njega koristi posebna fraza: korelacija nije kauzalnost. Do ovog pojma ćemo se vraćati tijekom cijelog kolegija jer je toliko važan i toliko se često zanemaruje.

2.3.1 Simpsonov paradoks: kad podaci lažu

Postoji jedan fenomen koji savršeno ilustrira zašto nam je statistika potrebna, a zove se **Simpsonov paradoks**. To je situacija u kojoj trend koji vidite u ukupnim podacima potpuno nestane ili se čak preokrene kad podatke razbijete po grupama. Zvuči apstraktno, pa pogledajmo konkretan primjer iz svijeta medija.

Zamislite da analizirate podatke o tome koji format vijesti generira više angažmana na dva medijska portala. Portal A i Portal B oba objavljuju vijesti u dva formata — tekst i video. Gledate ukupne podatke i vidite da Portal A ima veći prosječni angažman po članku nego Portal B. Zaključak je, čini se, očigledan: Portal A radi nešto bolje.

Ali onda razbijete podatke po formatu. I otkrijete nešto zapanjujuće — za tekstualne članke, Portal B ima veći prosječni angažman. Za video sadržaje, Portal B opet ima veći prosječni angažman. Kako je to moguće? Kako Portal A može biti bolji ukupno, a Portal B bolji u svakoj pojedinoj kategoriji?

Odgovor leži u proporcijama. Portal A objavljuje pretežno video sadržaj (koji općenito ima veći angažman), dok Portal B objavljuje pretežno tekstualni sadržaj (koji općenito ima manji angažman). Kad gledate ukupni prosjek, Portal A izgleda bolje jer ima veći udio visoko angažirajućeg formata (videa), ne zato što je zapravo bolji u bilo čemu.

Evo istog principa s brojevima. Portal A objavi 100 videa s prosječnim angažmanom 200 i 10 tekstualnih članaka s prosječnim angažmanom 50. Ukupni prosjek Portala A je (100 puta 200 plus 10 puta 50) podijeljeno sa 110, dakle oko 186. Portal B objavi 10 videa s prosječnim angažmanom 220 i 100 tekstualnih članaka s prosječnim angažmanom 60. Ukupni prosjek Portala B je (10 puta 220 plus 100 puta 60) podijeljeno sa 110, dakle oko 75.

Portal A ima ukupni prosjek 186, Portal B ima 75. Ali Portal B je bolji i u videu (220 naprema 200) i u tekstu (60 naprema 50). Ukupni prosjek zavodi jer ne uzima u obzir drastičnu razliku u proporcijama formata.

Simpsonov paradoks nije egzotična statistička kurioznost. Pojavljuje se u stvarnom životu češće nego što biste očekivali. Klasičan primjer iz akademskog svijeta je slučaj pristupa na sveučilište Berkeley iz 1970ih, gdje su ukupni podaci sugerirali diskriminaciju žena, ali kad su se podaci razbili po odjelima, žene su zapravo imale veću stopu prijema u većini pojedinih odjela. Problem je bio u tome što su se žene više prijavljivale na kompetitivnije odjele s nižom stopom prijema za sve.

Za komunikologe, Simpsonov paradoks je posebno relevantan jer medijska istraživanja redovito uključuju podatke koji se mogu raščlaniti na više načina — po platformi, po dobnoj skupini publike, po vremenu dana, po tipu sadržaja. Svaki put kad gledate agregirane podatke bez raščlambe, riskirate da donesete zaključak koji je ne samo netočan, nego dijametralno suprotan stvarnosti.

> Statistika nije skup trikova za impresioniranje publike brojevima. Statistika je disciplinirani način razmišljanja koji vas štiti od donošenja krivih zaključaka, uključujući zaključke koji se na prvi pogled čine savršeno logičnima.

2.4 Što je uopće mjerenje?

Prije nego što uopće počnemo razmišljati o statističkim metodama, moramo se zapitati nešto fundamentalnije: što znači mjeriti nešto? Ovo pitanje zvuči trivijalno kad govorimo o fizičkim veličinama. Mjeriti visinu osobe znači staviti metar uz nju i pročitati broj. Ali što znači mjeriti povjerenje u medije? Ili angažman korisnika na društvenim mrežama? Ili kvalitetu novinarstva?

U komunikologiji, kao i u drugim društvenim znanostima, većina stvari koje želimo mjeriti su apstraktni koncepti (teoretski konstrukti) koje ne možemo izravno vidjeti ili dotaknuti. Povjerenje u medije je mentalno stanje. Kvaliteta novinarstva je procjena koja ovisi o kriterijima. Angažman korisnika je složeni fenomen koji se može manifestirati na mnogo načina, od klika i komentara do dijeljenja i vremena provedenog na stranici.

Proces pretvaranja apstraktnog koncepta u nešto mjerljivo zove se **operacionalizacija**. Kad kažemo da ćemo mjeriti povjerenje u medije, moramo točno definirati kako ćemo to napraviti. Hoćemo li pitati ljude koliko vjeruju pojedinim medijima na skali od 1 do 10? Ili ćemo im dati tvrdnje poput "vjerujem informacijama koje pročitam na portalu X" i zamoliti ih da izraze stupanj slaganja? Ili ćemo mjeriti ponašanje, na primjer koliko često dijele članke s određenog portala?

Svaka od ovih operacionalizacija zahvaća nešto malo drugačije. Skala od 1 do 10 daje grubu procjenu općeg osjećaja. Tvrdnje sa stupnjem slaganja daju precizniju sliku o specifičnim aspektima povjerenja. Dijeljenje članaka mjeri ponašanje, a ne stav, i na njega utječu mnogi drugi faktori (poput toga koliko su članci zanimljivi, koliko je osoba aktivna na društvenim mrežama općenito, i tako dalje).

> **!** Važna napomena
>
> Operacionalizacija je jedan od najkritičnijih koraka u svakom istraživanju. Dva istraživanja koja istražuju isti koncept (recimo, utjecaj društvenih mreža na političku polarizaciju) mogu doći do potpuno suprotnih zaključaka jednostavno zato što su koristila različite operacionalizacije. Jedno je mjerilo polarizaciju kao razliku u stavovima između pristaša različitih stranaka, a drugo kao učestalost negativnih izjava o političkim protivnicima. Kad čitate istraživačke rezultate, uvijek provjerite kako su ključni pojmovi operacionalizirani. To vam govori što je istraživanje zapravo mjerilo, a ne samo što tvrdi da je mjerilo.

2.4.1 Varijable i opažanja

Sve što mjerimo u istraživanju nazivamo **varijablom**. Varijabla je bilo koje svojstvo koje se može razlikovati od jednog mjerenja do drugog, od jednog ispitanika do drugog. Dob je varijabla jer ljudi imaju različite godine. Spol je varijabla jer se ispitanici razlikuju po spolu. Dnevno vrijeme provedeno na TikToku je varijabla jer svatko provodi različitu količinu vremena. Čak i tip medija kojeg netko najčešće koristi je varijabla jer su odgovori različiti.

Svako pojedinačno mjerenje varijable nazivamo **opažanjem** (observation). Ako ispitamo 300 ljudi o tome koliko minuta dnevno provode na TikToku, imamo 300 opažanja jedne varijable. Ako svakom ispitaniku postavimo 10 pitanja, imamo 300 opažanja za svaku od 10 varijabli.

Navarro u knjizi naglašava jednu stvar koja se studentima čini očitom ali je zapravo duboka: razlikovanje između onoga što nas zanima (teoretski konstrukt) i onoga što zapravo mjerimo (opažanja). Povjerenje u medije je teoretski konstrukt. Odgovor na pitanje "na skali od 1 do 10, koliko vjerujete portalu Index.hr" je opažanje. Te dvije stvari nisu iste, i kvaliteta vašeg

istraživanja uvelike ovisi o tome koliko dobro vaša opažanja odražavaju konstrukt koji vas zapravo zanima.

2.5 Razine mjerenja

Nije svaka varijabla iste prirode, i način na koji varijablu mjerimo određuje što smijemo raditi s njom statistički. Američki psiholog S. S. Stevens je 1946. godine predložio klasifikaciju razina mjerenja koja se koristi do danas. Postoje četiri razine, i svaka dopušta drugačije statističke operacije.

2.5.1 Nominalna razina

Na nominalnoj razini, brojevi (ili oznake) služe samo za imenovanje kategorija. Nema nikakve prirodne hijerarhije ili redoslijeda između kategorija. U komunikološkim istraživanjima tipične nominalne varijable uključuju spol (muški, ženski, ostalo), tip medija (televizija, radio, web portal, društvena mreža, podcast), političku opciju, državu prebivališta i ime platforme.

Jedino što možemo smisleno napraviti s nominalnim podacima jest prebrojati koliko opažanja pripada svakoj kategoriji i izračunati proporcije. Ne možemo ih zbrajati, oduzimati ili izračunavati prosjek. "Prosječni spol" nema smisla. "Prosječni tip medija" također ne.

2.5.2 Ordinalna razina

Ordinalna razina dodaje informaciju o redoslijedu. Kategorije imaju prirodan poredak, ali razmaci između njih nisu nužno jednaki. Klasičan primjer su odgovori na Likertovoj skali poput "uopće se ne slažem" (1), "ne slažem se" (2), "niti se slažem niti ne" (3), "slažem se" (4), "potpuno se slažem" (5). Znamo da je 4 više od 3 i 3 više od 2, ali ne možemo tvrditi da je razlika između "ne slažem se" i "niti se slažem niti ne" jednaka razlici između "slažem se" i "potpuno se slažem".

Drugi primjeri ordinalnih varijabli u komunikologiji uključuju rangiranje omiljenih medija (prvi, drugi, treći izbor), obrazovnu razinu (osnovna škola, srednja škola, prvostupnik, magistar, doktor) i rang na ljestvici najčitanijih članaka.

S ordinalnim podacima možemo raditi sve što i s nominalnim (prebrojavanje, proporcije), plus možemo govoriti o redoslijedu i izračunavati medijan. Ali prosjek je, strogo govoreći, problematičan jer pretpostavlja jednake razmake. U praksi se prosjeci Likertovih skala ipak računaju gotovo svugdje, i to je jedna od onih situacija u kojima se stroga metodološka čistoća sukobljava s praktičnošću. Vrijedi biti svjestan da je to kompromis.

2.5.3 Intervalna razina

Na intervalnoj razini, razmaci između vrijednosti su jednaki i smisleni, ali nulta točka je proizvoljno određena. Klasičan primjer je temperatura u Celzijevim stupnjevima, gdje je razlika između 20 i 30 stupnjeva jednaka je razlici između 30 i 40 stupnjeva, ali 0 stupnjeva ne znači "nema temperature" (to je samo točka na kojoj se voda ledi). Zbog toga ne možemo reći da je 40 stupnjeva dvostruko toplije od 20 stupnjeva.

U komunikologiji su prave intervalne skale relativno rijetke. Mnogi istraživači tretiraju Likertove skale kao intervalne (pod pretpostavkom da su razmaci između odgovora psihološki jednaki), ali to je pretpostavka, ne činjenica. Standardizirani testovi znanja o medijskoj pismenosti ili indeksi poput nekih kompozitnih mjera medijskog pluralizma mogu se opravdano tretirati kao intervalni.

S intervalnim podacima možemo raditi sve dosad navedeno plus izračunavati aritmetičku sredinu i standardnu devijaciju, što otvara vrata većini statističkih metoda.

2.5.4 Omjerna razina

Omjerna (ratio) razina ima sve karakteristike intervalne, plus ima apsolutnu nultu točku koja znači potpuno odsustvo mjerne veličine. Vrijeme provedeno na platformi je omjerna varijabla jer 0 minuta zaista znači da osoba uopće nije koristila platformu, a 60 minuta je zaista dvostruko više od 30 minuta. Broj dijeljenja objave, broj pratitelja, prihod od oglašavanja, stopa klikanja (click through rate), sve su to omjerne varijable.

S omjernim podacima možemo raditi apsolutno sve — zbrajati, oduzimati, množiti, dijeliti, izračunavati omjere. Izjava "Portal A ima dvostruko više čitatelja od Portala B" ima smisla jer je broj čitatelja omjerna varijabla.

2.5.5 Diskretne i kontinuirane varijable

Osim razina mjerenja, postoji još jedna važna razlika — varijable mogu biti **diskretne** ili **kontinuirane**. Diskretna varijabla može poprimiti samo određene, odvojene vrijednosti. Broj komentara na članak je diskretan jer ne možete imati 3.7 komentara. Kontinuirana varijabla može teoretski poprimiti bilo koju vrijednost unutar nekog raspona. Vrijeme provedeno na stranici je kontinuirano jer možete provesti 3 minute i 27.438 sekundi.

Ova razlika je važna jer utječe na izbor grafičkih prikaza (histogrami za kontinuirane, stupičasti grafikoni za diskretne) i na izbor statističkih testova. U praksi, mnoge varijable koje su tehnički diskretne tretiramo kao kontinuirane kad imaju dovoljno velik raspon mogućih vrijednosti. Na primjer, broj bodova na testu od 0 do 100 je tehnički diskretan, ali razlika između susjednih vrijednosti je toliko mala da ga bez problema tretiramo kao kontinuiranog.

Praktični savjet

Kad kreirate anketu ili instrument za istraživanje, razina mjerenja varijabli nije nešto što se naknadno određuje. Morate unaprijed razmisliti o tome kakve podatke želite i kakve statističke analize planirate provoditi. Ako želite izračunavati prosjeke i korelacije, trebate barem intervalne podatke. Ako samo želite znati distribuciju po kategorijama, nominalni podaci su dovoljni. Ovaj korak planiranja vas štedi od situacije u kojoj imate prikupljene podatke ali ne možete s njima napraviti analizu koja vas zanima.

2.6 Pouzdanost mjerenja

Zamislite da imate vagu koja svaki put kad stanete na nju pokazuje drugačiji broj: jednom 72 kg, sljedeći put 78 kg, pa opet 65 kg. Očito, ta vaga je beskorisna. Ne zato što ne mjeri težinu, nego zato što njezina mjerenja nisu **pouzdana** (reliable). Pouzdanost mjerenja odnosi se na stupanj u kojem dobivamo konzistentne rezultate kad ponovimo isto mjerenje pod istim uvjetima.

U komunikologiji je pouzdanost posebno važna jer mjerimo apstraktne koncepte koje ne možemo izravno vidjeti. Kad mjerimo povjerenje u medije putem upitnika, moramo se pitati: bi li ispitanik dao iste odgovore da smo ga ispitali tjedan dana kasnije? Bi li dva različita istraživača koja čitaju isti članak i kodiraju ga prema istim kriterijima dodijelila iste kodove?

Navarro u knjizi opisuje pet tipova pouzdanosti, a za komunikologe su relevantna osobito tri.

Pouzdanost ponovljenog mjerenja (test-retest reliability) mjeri konzistentnost rezultata kroz vrijeme. Ako danas mjerite stavove studenata o pouzdanosti televizijskih vijesti i ponovite isto mjerenje za dva tjedna, rezultati bi trebali biti slični (pretpostavljajući da se u međuvremenu nije dogodilo ništa dramatično). Ako se rezultati drastično razlikuju bez očitog razloga, vaš instrument nije pouzdan.

Problem s ovim tipom pouzdanosti u komunikologiji jest to da se stavovi prema medijima zaista mogu brzo promijeniti. Ako se između dva mjerenja dogodi veliki medijski skandal, promjena u odgovorima nije znak nepouzdanog instrumenta, nego stvarne promjene u stavovima. Zato je važno razlikovati nepouzdanost instrumenta od stvarne varijabilnosti u onome što mjerite.

Pouzdanost između procjenjivača (inter-rater reliability) posebno je važna u analizi sadržaja, jednoj od temeljnih metoda u komunikologiji. Kad dva ili više kodera analiziraju isti medijski sadržaj, moraju se slagati u svojim procjenama. Ako jedan koder ocijeni članak kao “neutralan”, a drugi isti članak kao “negativno pristrasan”, imate problem s pouzdanošću. Mjere poput Cohenove kappa i Krippendorffove alpha koriste se za kvantificiranje razine slaganja između kodera, i standardni su dio metodološkog izvještaja u analizi sadržaja.

Interna konzistentnost mjeri koliko su pitanja u upitniku konzistentna jedno s drugim. Ako imate upitnik o medijskoj pismenosti s 15 pitanja, sva bi trebala mjeriti isti konstrukt. Ako jedno pitanje ne korelira s ostalima, možda mjeri nešto drugo, ili je loše formulirano. Cronbachov alfa koeficijent je daleko najčešća mjera interne konzistentnosti u društvenim znanostima, i gotovo sigurno ćete ga susresti u svakom metodološkom dijelu akademskog rada koji koristi upitnike. Vrijednosti iznad 0.70 se obično smatraju prihvatljivima, iako je i ta granica predmet rasprave.

> Praktični savjet
>
> Kad čitate akademski rad iz komunikologije, obratite pažnju na dio u kojem autori opisuju pouzdanost svojih mjera. Ako koriste upitnik, trebali bi navesti Cronbachov alfa za svaku skalu. Ako rade analizu sadržaja, trebali bi navesti mjeru slaganja između kodera. Odsutnost ovih informacija je ozbiljan metodološki propust i razlog za oprez pri tumačenju rezultata.

2.7 Valjanost mjerenja

Pouzdanost je nužan, ali ne i dovoljan uvjet za dobro mjerenje. Mjerenje može biti savršeno pouzdano (daje isti rezultat svaki put), a potpuno besmisleno. Zamislite da mjerite kvalitetu novinarstva brojem zareza u članku. Ovaj instrument bi bio izuzetno pouzdan (zareze je lako prebrojati i dva procjenjivača će gotovo uvijek dati isti rezultat), ali potpuno nevaljan, jer broj zareza očito nema nikakve veze s kvalitetom novinarstva.

Valjanost (validity) mjerenja odnosi se na pitanje mjeri li instrument zaista ono što bi trebao mjeriti. To je konceptualno jednostavno ali praktički vrlo zahtjevno pitanje, osobito u društvenim znanostima.

Postoji nekoliko tipova valjanosti koji su relevantni za komunikološka istraživanja.

Valjanost sadržaja (content validity) odnosi se na to pokriva li instrument sve relevantne aspekte konstrukta koji mjerimo. Ako mjerite medijsku pismenost, a vaš upitnik sadrži samo pitanja o televiziji i ništa o digitalnim medijima, valjanost sadržaja je niska jer ste propustili važan aspekt suvremene medijske pismenosti.

Konstruktna valjanost (construct validity) odnosi se na to mjeri li instrument zaista teoretski konstrukt koji namjeravamo mjeriti, a ne nešto drugo. Ovo je najkompleksniji tip valjanosti jer zahtijeva dokaze iz više izvora. Jedanaest mjera medijskog angažmana (klikovi, dijeljenja, komentari, vrijeme na stranici) moglo bi se sve nazvati mjerama angažmana, ali zapravo mjere različite stvari. Klikovi mjere početni interes, vrijeme na stranici mjeri dubinu čitanja, komentari mjere motivaciju za javno izražavanje mišljenja. Sve su to aspekti angažmana, ali nisu međusobno zamjenjivi.

Ekološka valjanost (ecological validity) pita koliko se rezultati iz kontroliranih uvjeta (laboratorija, online eksperimenta) mogu generalizirati na stvarni svijet. Ako testirate kako ljudi reagiraju na lažne vijesti tako da im u laboratorijskom okruženju pokažete članke na bijelom ekranu bez ikakvih drugih distrakcija, vaši rezultati možda ne vrijede za situaciju u kojoj ista osoba na svom telefonu prolazi kroz feed na Facebooku s desecima konkurentskih sadržaja, notifikacijama i porukama prijatelja.

2.8 Varijable u istraživanju: nezavisne i zavisne

U svakom istraživanju postoje varijable koje nas primarno zanimaju i varijable kojima manipuliramo ili ih promatramo da bismo razumjeli one prve. Standardna terminologija koristi dva ključna pojma.

Nezavisna varijabla (independent variable, IV) je varijabla za koju pretpostavljamo (ili testiramo hipotezu) da ima utjecaj na nešto. U eksperimentu, to je varijabla kojom istraživač manipulira. Na primjer, ako testirate dva različita formata naslova (neutralni naprema senzacionalistički) da vidite koji generira više klikova, format naslova je nezavisna varijabla.

Zavisna varijabla (dependent variable, DV) je varijabla koja mjeri ishod ili rezultat. To je ono što promatramo da bismo vidjeli je li nezavisna varijabla imala efekt. U primjeru s naslovima, broj klikova je zavisna varijabla.

Ova terminologija može zbuniti jer u neeksperimentalnim istraživanjima nema prave manipulacije. Kad proučavate vezu između dobi i korištenja društvenih mreža, ne manipulirate ničijom dobi. U takvim situacijama neki autori preferiraju termine **prediktor** (predictor) umjesto nezavisne varijable i **kriterij** ili **ishod** (outcome) umjesto zavisne varijable. U ovom kolegiju koristit ćemo oba skupa termina, ovisno o kontekstu.

2.8.1 Konfundirajuće varijable

Jedna od najopasnijih prijetnji valjanosti istraživanja su **konfundirajuće varijable** (confounding variables, ili kraće konfaunderi). To su varijable koje utječu i na nezavisnu i na zavisnu varijablu, stvarajući lažni dojam da između njih postoji uzročna veza.

Zamislite da istražujete vezu između korištenja Instagrama i samopoštovanja adolescenata. Rezultati pokazuju negativnu korelaciju, odnosno oni koji više koriste Instagram imaju niže samopoštovanje. Ali pričekajte. Adolescenti koji imaju niže samopoštovanje možda provode više vremena na društvenim mrežama jer traže potvrdu. Ili pak treća varijabla, poput socijalne izolacije, uzrokuje i više korištenja Instagrama i niže samopoštovanje. Bez kontrole konfundirajućih varijabli, ne možete znati što uzrokuje što.

Navarro u knjizi koristi termin “treća varijabla” (third variable problem) i naglašava da je to razlog zbog kojeg korelacija ne implicira kauzalnost. Mi ćemo se na ovu temu vraćati mnogo puta tijekom kolegija. Zapamtite je od prvog dana: kad god vidite da netko tvrdi

da X uzrokuje Y na temelju korelacijskih podataka, pitajte se postoji li varijabla Z koja bi mogla objasniti vezu.

2.9 Eksperimentalni istraživački dizajn

Kako onda možemo utvrditi da jedna stvar zaista uzrokuje drugu, a ne samo da su povezane? Odgovor je eksperiment. Eksperiment je istraživački dizajn u kojem istraživač aktivno **manipulira** nezavisnom varijablom i promatra učinak te manipulacije na zavisnu varijablu, uz kontrolu svih ostalih relevantnih čimbenika.

Zamislite klasičan primjer iz komunikologije — A/B test naslova na web portalu. Istraživač pripremi dva naslova za isti članak, jedan neutralan ("Rezultati istraživanja o utjecaju društvenih mreža na mlade") i jedan senzacionalistički ("Šokantan nalaz: društvene mreže uništavaju generaciju Z"). Zatim nasumično odabere koji čitatelji će vidjeti koji naslov. Nakon određenog vremena, mjeri stopu klikanja za svaki naslov.

Ovaj dizajn ima tri ključna svojstva pravog eksperimenta.

Manipulacija nezavisnom varijablom. Istraživač odlučuje koji ispitanici dobivaju koji naslov. Ne promatra što se prirodno događa, nego aktivno intervenira u situaciju.

Nasumična dodjela (randomizacija). Ispitanici su nasumično dodijeljeni grupama. Ovo je apsolutno ključno jer nasumična dodjela osigurava da se grupe ne razlikuju sustavno ni po jednoj varijabli osim nezavisne. Ako biste umjesto nasumične dodjele pustili da čitatelji sami odaberu naslov, oni koji biraju senzacionalistički naslov možda su ionako skloniji klikanju, i ne biste mogli znati je li razlika u klikanju posljedica naslova ili predispozicije čitatelja.

Kontrola ostalih varijabli. Sve ostalo osim nezavisne varijable treba biti jednako za obje grupe. Isti članak, isto vrijeme objave, ista pozicija na stranici. Ako se grupe razlikuju po nečemu osim naslova, taj nešto postaje konfundirajuća varijabla.

Kad su ova tri uvjeta ispunjena, logika eksperimenta je jednostavna i snažna. Ako se grupe razlikuju u zavisnoj varijabli, a jedina razlika između grupa je nezavisna varijabla, onda nezavisna varijabla mora biti uzrok te razlike. Ovo je jedini istraživački dizajn koji dopušta kauzalne zaključke.

> **!** Važna napomena
>
> U komunikologiji, pravi eksperimenti su mogući ali imaju ograničenja. Mnoge stvari koje nas zanimaju ne možemo manipulirati — ne možemo nasumično dodijeliti ljude različitim razinama medijske pismenosti, ne možemo kontrolirati koliko netko koristi društvene mreže u svakodnevnom životu, ne možemo manipulirati ničijim socioekonomskim statusom. Za takva pitanja moramo koristiti neeksperimentalne dizajne, uz puno veći oprez pri donošenju zaključaka.

2.9.1 Interna valjanost eksperimenta

Interna valjanost je stupanj u kojem možemo biti sigurni da je opažena razlika između grupa zaista uzrokovana nezavisnom varijablom, a ne nečim drugim. Prijetnje internoj valjanosti su sve one situacije u kojima se u istraživanje uvlači neka neplanirana sustavna razlika između grupa.

Navarro u knjizi navodi nekoliko prijetnji internoj valjanosti. U kontekstu medijskih istraživanja, neke su posebno relevantne.

Efekti povijesti nastaju kad se između početka i kraja istraživanja dogodi nešto što utječe na rezultate. Ako mjerite povjerenje u medije prije i poslije neke intervencije, a u međuvremenu se dogodi veliki medijski skandal, promjena u povjerenju može biti posljedica skandala, ne vaše intervencije.

Gubitak ispitanika (attrition) je problem kad ispitanici napuštaju istraživanje, i to neravnomjerno po grupama. Ako u vašem eksperimentu o utjecaju clickbait naslova čitatelji koji su dobili neutralne naslove češće napuštaju stranicu (jer ih naslovi nisu privukli), preostali ispitanici u toj grupi možda nisu reprezentativni, i usporedba postaje iskrivljena.

Efekt očekivanja (demand characteristics) nastaje kad ispitanici shvate (ili misle da shvaćaju) što istraživač očekuje i ponašaju se u skladu s tim. U anketama o medijskoj pismenosti, ispitanici možda daju odgovore koji ih prikazuju kao kritičnije i medijski pismenije nego što zaista jesu, jer osjećaju da je to "pravi" odgovor.

2.10 Neeksperimentalni istraživački dizajn

U praksi, većina istraživanja u komunikologiji nije eksperimentalna. Razlog je jednostavan: mnoge varijable koje nas zanimaju ne možemo kontrolirati niti njima manipulirati. Ne možemo nasumično odrediti tko će gledati televiziju a tko čitati portale. Ne možemo kontrolirati koji će sadržaj postati viralan. Ne možemo manipulirati kulturalnim kontekstom u kojem netko konzumira medije.

2.10.1 Opažajne (korelacijske) studije

Najčešći tip neeksperimentalnog istraživanja u komunikologiji je **opažajna studija** u kojoj istraživač mjeri varijable onakve kakve jesu, bez ikakve intervencije. Primjer: anketno ispitivanje 500 studenata o tome koliko koriste društvene mreže, koliko vjeruju vijestima koje tamo pronalaze i koliko su politički informirani. Istraživač mjeri tri varijable i analizira veze između njih.

Problem s opažajnim studijama, koji smo već nagovijestili, jest nemogućnost kauzalnog zaključivanja. Ako nađete negativnu korelaciju između korištenja društvenih mreža i političke informiranosti, to može značiti tri stvari: (1) društvene mreže smanjuju informiranost, (2)

manje informirani ljudi više koriste društvene mreže, ili (3) treća varijabla (poput obrazovanja ili dobi) utječe na oboje. Bez eksperimenta, ne možete razlikovati ove mogućnosti.

Ipak, opažajne studije su izuzetno korisne i čine osnovu komunikološkog istraživanja. One nam omogućuju proučavanje fenomena u njihovom prirodnom kontekstu, s velikim i reprezentativnim uzorcima, i bez etičkih ograničenja koja bi nam onemogućila eksperimentalnu manipulaciju.

2.10.2 Kvazieksperimenti

Kvazieksperiment zauzima srednje mjesto između eksperimenta i opažajne studije. U kvazieksperimentu, istraživač uspoređuje grupe koje se prirodno razlikuju po nekoj varijabli, ali tu varijablu nije sam manipulirao i ispitanike nije nasumično dodijelio grupama.

Na primjer, istraživač uspoređuje stavove o privatnosti podataka između korisnika koji su doživjeli curenje osobnih podataka (data breach) i onih koji nisu. Naravno, istraživač nije uzrokovao curenje podataka i nije nasumično odredio tko će ga doživjeti. Ali može usporediti dvije grupe i pokušati kontrolirati što više konfundirajućih varijabli (dob, spol, razina digitalne pismenosti) da dobije što čistiju sliku.

Kvazieksperimenti su korisni kad eksperiment nije moguć, ali kauzalni zaključci iz njih moraju biti formulirani puno opreznije. Umjesto "X uzrokuje Y" kažemo "X je povezan s Y, čak i nakon kontrole varijabli Z1, Z2 i Z3, što sugerira mogući uzročni odnos".

2.10.3 Studije slučaja i kvalitativna istraživanja

Na drugom kraju spektra nalaze se **studije slučaja** (case studies) koje detaljno proučavaju jedan ili mali broj primjera. Studija slučaja o tome kako je jedna medijska kuća provela digitalnu transformaciju može pružiti bogat uvid u procese i mehanizme koji stoje iza promjene, ali ne može generalizirati na sve medijske kuće. U komunikologiji, studije slučaja se često koriste za istraživanje novih fenomena (poput pojave novog tipa medijskog sadržaja ili platforme) gdje još nemamo dovoljno znanja za postavljanje hipoteza koje bi testirali kvantitativno.

Kvalitativna istraživanja općenito (dubinski intervjui, fokus grupe, etnografija medijskih praksi) pružaju dubinu koju kvantitativna istraživanja ne mogu postići, ali ne koriste statistiku u tradicionalnom smislu. Na ovom kolegiju fokusiramo se na kvantitativne metode, ali vrijedi znati da su kvalitativne i kvantitativne metode komplementarne, a ne konkurentne.

2.11 Eksterna valjanost: možemo li generalizirati?

Čak i kad imamo savršeno dizajniran eksperiment s visokom internom valjanošću, ostaje pitanje — vrijede li naši rezultati izvan specifičnog konteksta u kojem smo ih dobili? To je pitanje **eksterne valjanosti**.

Eksterna valjanost se pojavljuje u nekoliko oblika.

Generalizacija na populaciju. Ako ste testirali učinak naslova na studente komunikologije u Zagrebu, vrijede li rezultati i za čitatelje u Splitu? Za starije čitatelje? Za čitatelje u Srbiji koji čitaju iste portale?

Generalizacija na kontekst. Eksperiment je proveden u kontroliranim uvjetima na web stranici. Vrijede li rezultati i na mobilnoj aplikaciji? U situaciji kad čitatelj prolazi kroz feed društvene mreže umjesto da namjerno posjećuje portal?

Generalizacija kroz vrijeme. Medijski krajolik se brzo mijenja. Rezultat iz 2020. o tome kako ljudi konzumiraju vijesti možda ne vrijedi 2025. jer su se u međuvremenu pojavile nove platforme, promijenile navike i dogodile velike društvene promjene.

U komunikologiji je eksterna valjanost poseban izazov jer su medijska ponašanja izuzetno kontekstualno ovisna. Algoritmi društvenih mreža se mijenjaju, nove platforme nastaju i nestaju, kulturalne norme oko medijske konzumacije variraju između zemalja i generacija. Rezultati jednog istraživanja provedenog na jednoj platformi, u jednoj zemlji, u jednom trenutku, ne bi se trebali nekritički generalizirati na sve ostale situacije.

> Praktični savjet
>
> Kad čitate istraživačke rezultate u medijima (ili u akademskim radovima), uvijek se pitajte — tko su bili ispitanici? Gdje i kada je istraživanje provedeno? Na kojoj platformi ili u kojem kontekstu? Ova pitanja vam pomažu procijeniti koliko je opravdano generalizirati nalaze na situaciju koja vas zanima. Izjava “istraživanje je pokazalo da…” gotovo nikad ne znači “ovo vrijedi uvijek i svugdje”.

2.12 Pregled istraživačkih dizajna u komunikologiji

Da rezimiramo dizajne koje smo obradili, vrijedi ih sagledati kao spektar od veće do manje kontrole, s pripadajućim prednostima i ograničenjima.

Na jednom kraju spektra stoje **eksperimenti** (uključujući A/B testove u digitalnim medijima). Oni pružaju najveću kontrolu i dopuštaju kauzalne zaključke, ali su ograničeni u tome što mogu testirati (jer mnoge varijable ne mogu biti manipulirane) i mogu imati nisku ekološku valjanost (jer se provode u kontroliranim uvjetima).

U sredini su **kvazieksperimenti** i **opažajne studije s kontrolom varijabli**. Oni proučavaju fenomene bliže prirodnom kontekstu, ali kauzalni zaključci su slabiji jer uvijek postoji mogućnost konfundirajućih varijabli.

Na drugom kraju su **deskriptivne studije**, **studije slučaja** i **kvalitativna istraživanja**. Ona pružaju bogat kontekstualni uvid i mogu generirati hipoteze, ali ne testiraju kauzalne odnose.

U praksi, najbolja istraživanja kombiniraju više pristupa. Na primjer, možete započeti kvalitativnim intervjuima da identificirate relevantne varijable, zatim provesti opažajnu studiju na velikom uzorku da utvrdite korelacije, i na kraju dizajnirati eksperiment da testirate kauzalni odnos za najvažnije nalaze. Ovaj pristup, poznat kao mješovite metode (mixed methods), sve je popularniji u suvremenoj komunikologiji.

! Ključni zaključci

1. Statistika je disciplinirani način razmišljanja koji nas štiti od krivih zaključaka temeljenih na intuiciji, anegdotama i prividnim korelacijama.
2. Simpsonov paradoks pokazuje da agregirani podaci mogu potpuno zavarati i da je raščlamba po relevantnim grupama neophodna za ispravne zaključke.
3. Operacionalizacija je proces pretvaranja apstraktnog koncepta u mjerljivu varijablu. Različite operacionalizacije istog koncepta mogu dati različite rezultate.
4. Četiri razine mjerenja (nominalna, ordinalna, intervalna, omjerna) određuju koje statističke operacije smijemo primjenjivati na podatke.
5. Pouzdanost mjerenja (konzistentnost rezultata) je nužan ali ne i dovoljan uvjet za valjanost (mjeri li instrument ono što bi trebao).
6. Eksperimentalni dizajn s nasumičnom dodjelom ispitanika grupama jedini dopušta kauzalne zaključke, ali mnoge komunikološke varijable ne mogu biti eksperimentalno manipulirane.
7. Neeksperimentalni dizajni (opažajne studije, kvazieksperimenti) omogućuju proučavanje fenomena u prirodnom kontekstu, ali zahtijevaju oprez pri kauzalnom zaključivanju zbog mogućih konfundirajućih varijabli.
8. Eksterna valjanost (mogućnost generalizacije rezultata) poseban je izazov u komunikologiji jer su medijska ponašanja visoko ovisna o platformi, kulturi i vremenu.

3 Tjedan 2: Uvod u R i tidyverse

Vaš novi najdraži alat za rad s podacima

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti zašto je R bolji izbor od Excela i SPSS-a za statističku analizu u komunikologiji i prepoznati situacije u kojima je svaki alat prikladan.
2. Koristiti R kao kalkulator i razumjeti osnovne aritmetičke i logičke operacije.
3. Kreirati objekte u R-u, razumjeti pravila imenovanja i razlikovati tipove podataka (numerički, tekstualni, logički).
4. Konstruirati i indeksirati vektore te primijeniti vektorizirane operacije.
5. Razumjeti razliku između tibble i data.frame te koristiti tibble za organizaciju podataka.
6. Koristiti pipe operator (`|>`) za čitljivo ulančavanje operacija.
7. Učitati CSV datoteku pomoću `read_csv()`, pregledati strukturu podataka s `glimpse()` i `head()`, te identificirati tipove stupaca.
8. Instalirati i učitati R pakete te razumjeti ulogu tidyverse ekosustava.

3.1 Zašto R, a ne nešto drugo?

Ovo je pitanje koje studenti postavljaju na prvom satu, i to je potpuno legitimno pitanje. Zašto biste učili programski jezik kad postoje alati s grafičkim sučeljem koji rade iste stvari? Zašto ne ostati u Excelu, koji već znate koristiti, ili naučiti SPSS koji ima menije i gumbe za svaku analizu?

Odgovor ima nekoliko slojeva i vrijedi ih proći redom jer razumijevanje prednosti R-a mijenja način na koji pristupate cijelom kolegiju.

Ponovljivost. Kad radite analizu u Excelu, vaš rad je nevidljiv jer klikate po ćelijama, povlačite formule, sortirate stupce, i na kraju imate rezultat, ali nemate zapis toga kako ste do njega došli. Ako vas kolega pita kako ste izračunali nešto, morate mu pokazati i

objasniti svaki korak. Ako trebate ponoviti istu analizu s novim podacima, morate sve raditi ispočetka. U R-u, vaša analiza je skripta, dakle tekstualna datoteka koja sadrži svaki korak od učitavanja podataka do konačnog rezultata. Tu skriptu možete pokrenuti ponovo jednim klikom, poslati je kolegi, objaviti uz akademski rad, ili je modificirati za novi dataset. Ovo nije trivijalna prednost. U vremenu kad se sve više govori o krizi repliciranja u društvenim znanostima, ponovljivost analize nije luksuz nego nužnost.

Fleksibilnost. SPSS ima meni za t-test. Ima meni za ANOVA-u. Ima meni za regresiju. Ali što kad trebate nešto što nije u meniju? Što kad trebate kombinirati podatke iz tri različite ankete, izračunati prilagođenu mjeru angažmana koja uključuje i klikove i vrijeme na stranici i komentare, filtrirati samo ispitanike koji zadovoljavaju tri uvjeta istovremeno, i onda vizualizirati rezultat na način koji SPSS ne podržava? U R-u, granica je samo vaša mašta (i znanje, ali znanje raste). SPSS je poput restorana s fiksnim menijem. R je kuhinja u kojoj možete pripremiti bilo što.

Vizualizacija. Ovaj argument sam po sebi opravdava učenje R-a. Paket ggplot2, koji ćemo intenzivno koristiti od petog tjedna nadalje, proizvodi grafike profesionalne kvalitete koje možete izravno staviti u akademski rad, poslovni izvještaj ili medijsku prezentaciju. Excel grafovi izgledaju kao Excel grafovi. ggplot2 grafovi izgledaju kao da ih je dizajnirao grafički dizajner.

Besplatnost i zajednica. R je besplatan i open-source. Ne trebate licencu, ne trebate institucionalnu pretplatu, ne trebate brinuti hoćete li imati pristup nakon diplome. SPSS licenca košta stotine eura godišnje. Osim toga, R ima ogromnu zajednicu korisnika i tisuće paketa za svaku zamisivu analizu. Ako imate pitanje, gotovo sigurno je netko prije vas imao isto pitanje i odgovor postoji na internetu.

Zapošljivost. Ovo je pragmatičan argument ali važan. Analitičke vještine u R-u (ili Pythonu, koji je sličan po filozofiji) sve su traženije na tržištu rada, ne samo u akademiji nego i u medijskoj industriji, marketingu, oglaševanju i PR-u. Znanje Excela se podrazumijeva. Znanje R-a vas izdvaja.

Navarro u svojoj knjizi otvoreno priznaje da je učenje R-a teže od učenja softvera s grafičkim sučeljem. Početna krivulja učenja je strmija. Ali isto tako kaže da se ta investicija višestruko isplati jer jednom kad savladate osnove, možete raditi stvari koje su u drugim alatima nemoguće ili zahtijevaju enorman ručni rad. Mi ćemo slijediti njezin pristup i ići polako, objašnjavajući svaki korak bez pretpostavke o prethodnom iskustvu s programiranjem.

> **Praktični savjet**
>
> Ako vas uhvati frustracija dok učite R (a uhvatit će vas, to je normalno), prisjetite se da ste u istoj situaciji kao kad ste prvi put otvorili Photoshop ili Premiere. Prvi sat je bio zbunjujući. Nakon tjedan dana ste znali osnove. Nakon mjesec dana više niste mogli zamisliti rad bez tog alata. S R-om je isto, samo što je nagrada na kraju veća jer R možete koristiti za analizu bilo čega.

3.2 R i Positron: vaš radni prostor

Prije nego što počnemo pisati kod, trebamo dva programa.

R je sam programski jezik i sustav za statističko računanje. Kad instalirate R, dobivate motor koji izvršava naredbe, ali sučelje je minimalistično, gotovo spartan. Možete koristiti R izravno u terminalu (naredbenom retku), ali to je poput pisanja romana u Notepadu. Tehnički moguće, ali nitko razuman to ne radi.

Positron je integrirano razvojno okruženje (IDE) koje olakšava rad s R-om. Daje vam uređivač teksta s bojanjem sintakse (ključne riječi R-a prikazuje u boji da kod bude čitljiviji), prozor za pregled rezultata, prozor za grafike, prozor za pomoć, i mnoštvo drugih alata koji čine rad ugodnijim. Positron je relativno novi IDE razvijen od strane Posit tima (isti ljudi koji su stvorili RStudio), i koristi Visual Studio Code arhitekturu, što znači da je moderan, brz i proširiv.

> **!** Važna napomena
>
> Ako ste već koristili RStudio, Positron će vam biti intuitivan jer dijeli istu filozofiju, samo s modernijim sučeljem. Ako niste koristili niti jedan IDE, ne brinite. Mi ćemo koristiti samo osnovne funkcionalnosti poput pisanja koda u skripti (gornji lijevi panel), izvršavanja koda u konzoli (donji panel) i pregleda rezultata. Sve ostalo ćete naučiti usput, kad bude potrebno.

3.2.1 Kako izgleda Positron

Kad otvorite Positron i kreirate novu R datoteku, vidjet ćete sučelje podijeljeno u nekoliko panela. Gornji lijevi panel je **editor** u kojem pišete kod. Donji panel je **konzola** u kojoj se kod izvršava i prikazuju rezultati. Desna strana ima panele za pregled varijabli, grafika i pomoći.

Radni tok u Positronu je jednostavan. Pišete kod u editoru. Označite redak (ili više redova) koji želite izvršiti. Pritisnete Ctrl+Enter (ili Cmd+Enter na Macu). Kod se pošalje u konzolu, izvrši se i rezultat se prikaže. Ovo ćemo raditi stotine puta tijekom kolegija, pa će vam vrlo brzo postati automatsko.

3.3 Prve naredbe: R kao kalkulator

Najjednostavniji način da počnete koristiti R jest da ga tretirate kao kalkulator. I to ne kao obični kalkulator, nego kao vrlo moćan kalkulator koji može raditi s cijelim skupovima brojeva odjednom. Pogledajmo neke osnovne operacije.

```r
# Zbrajanje
10 + 25
```

```
[1] 35
```

```r
# Oduzimanje
100 - 37
```

```
[1] 63
```

```r
# Množenje
12 * 8
```

```
[1] 96
```

```r
# Dijeljenje
144 / 12
```

```
[1] 12
```

```r
# Potenciranje
2^10
```

```
[1] 1024
```

Ništa revolucionarno za sada. Ali primijetite par stvari. Prvo, linije koje počinju s **#** su **komentari**. R ih potpuno ignorira. Komentari služe vama (i vašim kolegama) da objasnite što kod radi. Pisanje komentara je navika koju biste trebali razviti od prvog dana jer vam štedi mnogo vremena kad se mjesec dana kasnije vraćate na vlastiti kod i pokušavate shvatiti što ste radili.

Drugo, R poštuje standardni redoslijed matematičkih operacija (množenje i dijeljenje prije zbrajanja i oduzimanja, potenciranje prije svega), ali za svaki slučaj koristite zagrade kad želite biti sigurni.

```r
# Bez zagrada: množi se prvo
5 + 3 * 2
```

```
[1] 11
```

```r
# Sa zagradama: zbraja se prvo
(5 + 3) * 2
```

```
[1] 16
```

R ima i nekoliko korisnih matematičkih funkcija koje ćemo trebati tijekom kolegija.

```r
# Kvadratni korijen
sqrt(144)
```

```
[1] 12
```

```r
# Apsolutna vrijednost
abs(-42)
```

```
[1] 42
```

```r
# Zaokruživanje
round(3.14159, 2)
```

```
[1] 3.14
```

```r
# Logaritam (prirodni)
log(100)
```

```
[1] 4.60517
```

```r
# Logaritam baze 10
log10(100)
```

```
[1] 2
```

Funkcija `sqrt()` računa kvadratni korijen, `abs()` vraća apsolutnu vrijednost, `round()` zaokružuje na zadani broj decimala. Funkcija `log()` bez drugog argumenta računa prirodni logaritam (baza e), a `log10()` računa logaritam baze 10. Logaritme ćemo koristiti kasnije kad budemo radili s podacima koji imaju ekstremno asimetričnu distribuciju, poput broja pratitelja na društvenim mrežama.

3.4 Objekti: pohranjivanje vrijednosti

Kalkulator je koristan, ali prava snaga programiranja dolazi od mogućnosti da rezultate pohranite i koristite kasnije. U R-u, vrijednosti pohranjujete u **objekte** (neki ih zovu varijablama, ali da izbjegnemo zabunu sa statističkim varijablama, koristit ćemo termin objekti).

Objekt se kreira operatorom pridruživanja `<-` (strelica lijevo). Čita se kao "dodijeli vrijednost desne strane objektu na lijevoj strani".

```
# Pohranjujemo broj ispitanika
n_ispitanika <- 500

# Pohranjujemo prosječno dnevno korištenje (u minutama)
prosjek_minuta <- 87.3

# Sada možemo koristiti te objekte
n_ispitanika
```

```
[1] 500
```

```
prosjek_minuta
```

```
[1] 87.3
```

Kad kreirate objekt, R ga tiho pohrani u memoriju bez ikakvog ispisa. Da biste vidjeli što objekt sadrži, jednostavno upišite njegovo ime. Objekte možete koristiti u izračunima baš kao i brojeve.

```
# Ukupno vrijeme svih ispitanika (u minutama)
ukupno_minuta <- n_ispitanika * prosjek_minuta
ukupno_minuta
```

```
[1] 43650
```

```
# Pretvaranje u sate
ukupno_sati <- ukupno_minuta / 60
ukupno_sati
```

```
[1] 727.5
```

```r
# Prosječno korištenje u satima
prosjek_sati <- prosjek_minuta / 60
round(prosjek_sati, 1)
```

```
[1] 1.5
```

Svih 500 ispitanika u našem imaginarnom uzorku zajedno provede gotovo 44 tisuće minuta dnevno na društvenim mrežama. To je oko 727 sati, ili nešto više od 30 punih dana. Svaki dan. Statistika, čak i ovako jednostavna, pomaže vizualizirati razmjere fenomena koji istražujemo.

3.4.1 Pravila imenovanja objekata

R ima nekoliko pravila i mnogo dobrih praksi za imenovanje objekata. Pravila su stroga i traže da ime mora počinjati slovom (ne brojem), ne smije sadržavati razmake ni specijalne znakove osim točke i podvlake, i R razlikuje velika i mala slova (`prosjek` i `Prosjek` su dva različita objekta).

Dobre prakse su mekše ali važne za čitljivost. Na ovom kolegiju koristimo konvenciju **snake_case**, u kojoj su riječi odvojene podvlakom i sve je malim slovima (npr. `prosjek_minuta`, `n_ispitanika`, `dnevno_koristenje`). Ova konvencija je standard u tidyverse zajednici i čini kod čitljivijim od alternativa poput `prosjekMinuta` (camelCase) ili `prosjek.minuta` (dot notation).

```r
# Dobra imena (snake_case, opisna)
broj_portala <- 15
prosjecni_ctr <- 0.034
ime_studije <- "medijske navike 2025"

# Funkcionalna ali loša imena (nejasna ili nekonzistentna)
x <- 15
bp <- 15
BrojPortala <- 15
```

Sva četiri donja primjera rade, ali samo `broj_portala` odmah komunicira što objekt sadrži. Kad budete imali skriptu s 50 objekata, razlika između opisnih i kriptičnih imena postaje enormna. Navarro u knjizi koristi lijep savjet: imenujte objekte tako da ih možete razumjeti kad se vratite na kod nakon dva tjedna bez spavanja.

> **Praktični savjet**
>
> R ima neke rezervirane riječi koje ne smijete koristiti kao imena objekata jer imaju posebno značenje u jeziku. Na primjer, `TRUE`, `FALSE`, `NULL`, `NA`, `if`, `else`, `for`, `function`. Ako pokušate, dobit ćete grešku. Također, izbjegavajte imena koja se poklapaju s

postojećim R funkcijama, poput `mean`, `sum`, `data` ili `c`. Tehnički možete kreirati objekt nazvan `mean`, ali to će pregaziti funkciju `mean()` i uzrokovati zbunjujuće greške. Zato je dobra praksa koristiti opisna imena poput `prosjek_dobi` umjesto generičnog `mean`.

3.4.2 Prepisivanje objekata

Važno je razumjeti da kad dodijelite novu vrijednost postojećem objektu, stara vrijednost nestaje bez upozorenja.

```
temperatura <- 22
temperatura
```

```
[1] 22
```

```
# Nova dodjela prepisuje staru vrijednost
temperatura <- 35
temperatura
```

```
[1] 35
```

R vas neće pitati jeste li sigurni. Neće vam reći da ste upravo izgubili prethodnu vrijednost. Jednostavno će to napraviti. Ovo je razlog zašto je dobra praksa koristiti opisna imena i ne reciklirati isti objekt za različite svrhe. Ako vam trebaju temperatura zraka i temperatura vode, napravite `temp_zraka` i `temp_vode`, nemojte koristiti isti objekt `temp` za oboje.

3.5 Vektori: rad s više vrijednosti odjednom

Do sada smo pohranjivali pojedinačne brojeve, ali u statistici gotovo nikad ne radimo s jednim brojem. Radimo sa skupovima podataka. Najjednostavnija struktura za pohranjivanje više vrijednosti u R-u je **vektor**.

Vektor je uređeni niz vrijednosti istog tipa. Kreiramo ga funkcijom `c()` (od "combine" ili "concatenate").

```
# Dnevno korištenje TikToka za 8 ispitanika (u minutama)
dnevno_tiktok <- c(95, 22, 112, 45, 78, 8, 135, 55)
dnevno_tiktok
```

```
[1]  95  22 112  45  78   8 135  55
```

```r
# Dobne skupine istih ispitanika
dobne_skupine <- c("18-24", "45+", "18-24", "25-34", "18-24", "55+", "18-24", "35-44")
dobne_skupine
```

```
[1] "18-24" "45+"   "18-24" "25-34" "18-24" "55+"   "18-24" "35-44"
```

Primijetite da se tekstualne vrijednosti stavljaju u navodnike, a numeričke ne. Ovo je razlika između tipova podataka o kojoj ćemo govoriti detaljnije za trenutak.

Snaga vektora je u tome što R automatski primjenjuje operacije na sve elemente odjednom. Ovo se zove **vektorizacija** i jedna je od najvažnijih karakteristika R-a.

```r
# Pretvorba u sate (dijeli SVAKI element sa 60)
dnevno_sati <- dnevno_tiktok / 60
round(dnevno_sati, 1)
```

```
[1] 1.6 0.4 1.9 0.8 1.3 0.1 2.2 0.9
```

```r
# Tjedno korištenje (množi SVAKI element sa 7)
tjedno_tiktok <- dnevno_tiktok * 7
tjedno_tiktok
```

```
[1] 665 154 784 315 546  56 945 385
```

```r
# Koliko minuta iznad prosjeka?
prosjek <- mean(dnevno_tiktok)
iznad_prosjeka <- dnevno_tiktok - prosjek
round(iznad_prosjeka, 1)
```

```
[1]  26.2 -46.8  43.2 -23.8   9.2 -60.8  66.2 -13.8
```

Kad napišete `dnevno_tiktok / 60`, R ne dijeli vektor kao cjelinu sa 60 (to ne bi imalo smisla), nego dijeli svaki element vektora sa 60. Rezultat je novi vektor iste duljine. Isto vrijedi za sve aritmetičke operacije. Ovo je enormno praktično jer nam omogućuje da jednom naredbom transformiramo stotine ili tisuće vrijednosti.

Na vektore možemo primijeniti i funkcije koje sažimaju podatke u jednu vrijednost.

```r
# Broj elemenata
length(dnevno_tiktok)
```

```
[1] 8
```

```r
# Prosjek
mean(dnevno_tiktok)
```

```
[1] 68.75
```

```r
# Medijan
median(dnevno_tiktok)
```

```
[1] 66.5
```

```r
# Standardna devijacija
sd(dnevno_tiktok)
```

```
[1] 44.18064
```

```r
# Minimum i maksimum
min(dnevno_tiktok)
```

```
[1] 8
```

```r
max(dnevno_tiktok)
```

```
[1] 135
```

```r
# Zbroj svih vrijednosti
sum(dnevno_tiktok)
```

```
[1] 550
```

Ove funkcije uzimaju čitav vektor i vraćaju jednu vrijednost. `mean()` računa aritmetičku sredinu, `median()` srednju vrijednost, `sd()` standardnu devijaciju, `min()` i `max()` najmanju i najveću vrijednost, `sum()` zbroj svih elemenata. Detaljno ćemo objasniti svaku od ovih mjera u tjednu o deskriptivnoj statistici. Za sada je dovoljno znati da postoje i da rade na vektorima.

3.5.1 Indeksiranje vektora

Ponekad trebamo pristupiti samo jednom ili nekoliko elemenata vektora. To radimo uglatim zagradama `[]`.

```r
# Treći element
dnevno_tiktok[3]
```

```
[1] 112
```

```r
# Elementi od drugog do petog
dnevno_tiktok[2:5]
```

```
[1]  22 112  45  78
```

```r
# Elementi koji zadovoljavaju uvjet
dnevno_tiktok[dnevno_tiktok > 100]
```

```
[1] 112 135
```

```r
# Koliko ispitanika koristi TikTok više od 100 minuta dnevno?
sum(dnevno_tiktok > 100)
```

```
[1] 2
```

Posebno je korisna mogućnost filtriranja po uvjetu. Izraz `dnevno_tiktok > 100` proizvodi logički vektor (niz TRUE i FALSE vrijednosti), a kad ga stavimo u uglate zagrade, R vraća samo one elemente za koje je uvjet TRUE. Funkcija `sum()` primijenjena na logički vektor broji koliko je TRUE vrijednosti, jer R tretira TRUE kao 1 i FALSE kao 0.

> **!** Važna napomena
>
> R indeksira od 1, ne od 0. Prvi element vektora je `vektor[1]`, ne `vektor[0]`. Ako ste učili Python ili JavaScript, ovo je važna razlika. Većina početničkih grešaka s indeksiranjem u R-u dolazi od zaboravljanja da R počinje brojati od 1.

3.6 Tipovi podataka

Svaka vrijednost u R-u ima tip koji određuje što možete s njom raditi. Četiri osnovna tipa koja ćete koristiti su numerički, tekstualni, logički i faktorski.

3.6.1 Numerički tip (numeric / double)

Svaki broj u R-u je po defaultu tipa `double`, što znači da se pohranjuje kao decimalni broj čak i kad izgleda kao cijeli broj. Postoji i podtip `integer` (cijeli broj) koji se kreira dodavanjem slova L: `42L`. U praksi, razlika rijetko bitna i R se uglavnom sam snalazi.

```
x <- 42
class(x)
```

```
[1] "numeric"
```

```
y <- 42L
class(y)
```

```
[1] "integer"
```

```
# Oboje radi jednako u većini situacija
x == y
```

```
[1] TRUE
```

3.6.2 Tekstualni tip (character)

Tekst u R-u se označava navodnicima, bilo jednostrukim ili dvostrukim. U ovom kolegiju koristimo dvostruke navodnike jer je to konvencija u tidyverse zajednici.

```
platforma <- "TikTok"
class(platforma)
```

```
[1] "character"
```

```
poruka <- "Ispitanik koristi platformu 95 minuta dnevno"
poruka
```

```
[1] "Ispitanik koristi platformu 95 minuta dnevno"
```

S tekstualnim vrijednostima ne možete raditi aritmetiku. Ako pokušate zbrojiti dva teksta pomoću `+`, dobit ćete grešku. Za spajanje tekstova koristi se funkcija `paste()` ili `paste0()`.

```r
ime <- "Portal"
broj <- "Index.hr"

# paste() spaja s razmakom (po defaultu)
paste(ime, broj)
```

```
[1] "Portal Index.hr"
```

```r
# paste0() spaja bez razmaka
paste0("n = ", n_ispitanika)
```

```
[1] "n = 500"
```

3.6.3 Logički tip (logical)

Logičke vrijednosti su samo dvije. Kako se pojavljuju: `TRUE` i `FALSE` nastaju kad R evaluira uvjete.

```r
# Usporedbe vraćaju logičke vrijednosti
10 > 5
```

```
[1] TRUE
```

```r
10 < 5
```

```
[1] FALSE
```

```r
10 == 10   # jednako (dva znaka jednakosti!)
```

```
[1] TRUE
```

```r
10 != 5    # nije jednako
```

```
[1] TRUE
```

```r
# Logički vektor
minuta <- c(95, 22, 112, 45, 78)
visoko_koristenje <- minuta > 60
visoko_koristenje
```

```
[1]  TRUE FALSE  TRUE FALSE  TRUE
```

Obratite pažnju na razliku između = i ==. Jedan znak jednakosti (=) je operator pridruživanja (isto kao <-). Dva znaka jednakosti (==) je operator usporedbe koji provjerava jesu li dvije vrijednosti jednake i vraća TRUE ili FALSE. Zamjena jednog s drugim je jedna od najčešćih pogrešaka u R-u.

3.6.4 Faktorski tip (factor)

Faktori su poseban tip za kategorijalene podatke. Iznad smo vidjeli da razine mjerenja određuju što smijemo raditi s varijablom. Faktori su način na koji R implementira kategorijalne varijable, posebno nominalne i ordinalne.

```
# Kreiranje faktora
platforme <- factor(c("Instagram", "TikTok", "YouTube", "TikTok", "Instagram", "YouTube",
platforme
```

```
[1] Instagram TikTok    YouTube   TikTok    Instagram YouTube   TikTok
Levels: Instagram TikTok YouTube
```

```
# Razine faktora (unique kategorije)
levels(platforme)
```

```
[1] "Instagram" "TikTok"    "YouTube"
```

```
# Uređeni faktor (ordinalni)
obrazovanje <- factor(
  c("srednja", "prvostupnik", "magistar", "srednja", "magistar"),
  levels = c("srednja", "prvostupnik", "magistar", "doktor"),
  ordered = TRUE
)
obrazovanje
```

```
[1] srednja     prvostupnik magistar    srednja     magistar
Levels: srednja < prvostupnik < magistar < doktor
```

Faktori postaju bitni kad počnemo raditi vizualizacije i statističke testove. Na primjer, ako želite da se kategorije na grafikonu pojave u specifičnom redoslijedu (ne abecednom), morate koristiti faktore s definiranim razinama. Za sada je dovoljno znati da postoje, a detaljnije ćemo ih koristiti od tjedna vizualizacije.

3.6.5 Provjera i pretvorba tipova

R ima funkcije za provjeru tipa (`class()`, `is.numeric()`, `is.character()`) i za pretvorbu (`as.numeric()`, `as.character()`).

```
# Provjera tipa
tekst_broj <- "42"
class(tekst_broj)
```

```
[1] "character"
```

```
# Ovo je tekst, ne broj! Ne možemo računati s njim.
# tekst_broj + 10 bi dalo grešku

# Pretvorba u broj
pravi_broj <- as.numeric(tekst_broj)
class(pravi_broj)
```

```
[1] "numeric"
```

```
pravi_broj + 10
```

```
[1] 52
```

Razumijevanje tipova podataka postaje ključno kad učitavate podatke iz CSV datoteka. Ponekad R pogrešno protumači stupac (na primjer, stupac s brojevima koji sadrži jedno slovo "N/A" umjesto prazne ćelije bit će učitan kao tekst umjesto broja). Znanje o tipovima i pretvorbama pomaže vam dijagnosticirati i popraviti takve probleme.

3.7 Tibble: moderna tablica podataka

Vektor može sadržavati samo vrijednosti istog tipa. Ali u stvarnim podacima imamo i brojeve (dob, minuta korištenja) i tekst (ime platforme, spol) i logičke vrijednosti, sve za istog ispitanika. Za organizaciju takvih podataka koristimo **tablicu** ili, u R žargonu, **data frame**.

U tidyverse ekosustavu koristimo poboljšanu verziju data framea koja se zove **tibble** (iz paketa tibble koji je dio tidyverse). Tibble je tablica u kojoj svaki stupac može biti drugog tipa, svaki redak predstavlja jedno opažanje i svaki stupac predstavlja jednu varijablu. Ovo

odgovara onome što Wickham naziva **tidy data** (uredni podaci), i to je filozofija oko koje je cijeli tidyverse izgrađen.

Kreirajmo mali tibble s podacima o našim zamišljenim ispitanicima.

```
anketa <- tibble(
  id = 1:8,
  dob = c(19, 52, 21, 35, 23, 61, 20, 42),
  spol = c("ženski", "muški", "ženski", "muški", "ženski", "muški", "muški", "ženski"),
  platforma = c("TikTok", "Facebook", "Instagram", "LinkedIn", "TikTok", "Facebook", "TikT
  dnevno_min = c(95, 22, 112, 45, 78, 8, 135, 55)
)

anketa
```

```
# A tibble: 8 x 5
     id   dob spol   platforma dnevno_min
  <int> <dbl> <chr>  <chr>          <dbl>
1     1    19 ženski TikTok            95
2     2    52 muški  Facebook          22
3     3    21 ženski Instagram        112
4     4    35 muški  LinkedIn          45
5     5    23 ženski TikTok            78
6     6    61 muški  Facebook           8
7     7    20 muški  TikTok           135
8     8    42 ženski Instagram         55
```

Nekoliko stvari koje vrijedi primijetiti. Kad ispišete tibble, R automatski prikazuje tip svakog stupca ispod imena (`<int>` za cijele brojeve, `<dbl>` za decimalne, `<chr>` za tekst). Ovo je enormno korisno jer na prvi pogled vidite kakve su varijable u vašim podacima. Tibble također automatski prikazuje samo prvih 10 redova, što sprječava da vam konzola bude poplavljena tisućama redova kad radite s velikim datasetima.

Usporedite ovo s klasičnim data frameom.

```
# Klasični data.frame
df <- data.frame(
  id = 1:3,
  ime = c("Ana", "Marko", "Petra"),
  dob = c(22, 35, 28)
)

# tibble
tb <- tibble(
  id = 1:3,
  ime = c("Ana", "Marko", "Petra"),
```

```
  dob = c(22, 35, 28)
)

# Razlika u ispisu
class(df)
```

```
[1] "data.frame"
```

```
class(tb)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

Razlika postaje očitija s većim podacima, ali ključna poanta je da je tibble modernija, čistija verzija data framea i da je standard u tidyverse ekosustavu. Na ovom kolegiju ćemo uvijek koristiti tibble.

3.7.1 Pristupanje stupcima

Pojedinom stupcu tibble pristupamo operatorom `$` ili pomoću funkcije `pull()`.

```
# Dolar operator
anketa$dnevno_min
```

```
[1]  95  22 112  45  78   8 135  55
```

```
# Izračun prosjeka jednog stupca
mean(anketa$dnevno_min)
```

```
[1] 68.75
```

```
# Provjera koliko ispitanika koristi TikTok
sum(anketa$platforma == "TikTok")
```

```
[1] 3
```

Operator `$` je brz i praktičan za pristup jednom stupcu. U tidyverse pristupu ćemo češće koristiti `select()` i `pull()`, ali `$` je savršeno ispravan i često najbrži način da dohvatite jedan stupac.

3.8 Pipe operator: čitljivo ulančavanje

Sada dolazimo do jednog od najvažnijih koncepata u tidyverse pristupu: **pipe operatora** `|>`. Pipe je jednostavan ali transformativan koncept koji čini R kod dramatično čitljivijim.

Zamislite da želite napraviti sljedeće: trebate uzeti naš tibble `anketa`, filtrirati samo ispitanike mlađe od 30 godina i izračunati prosjek njihovog dnevnog korištenja. Bez pipea, ovo možete napisati na dva načina, i oba su neelegantna.

```
# Pristup 1: ugniježđene funkcije (čitanje iznutra prema van)
mean(filter(anketa, dob < 30)$dnevno_min)
```

```
[1] 105
```

```
# Pristup 2: međuobjekti (stvara nepotrebne objekte)
mladi <- filter(anketa, dob < 30)
prosjek_mladi <- mean(mladi$dnevno_min)
prosjek_mladi
```

```
[1] 105
```

Prvi pristup je kompaktan ali nečitljiv. Trebate čitati iznutra prema van, pa najprije vidite `mean()` pa se trebate probiti do `filter()` unutra da shvatite na što se mean primjenjuje. Drugi pristup je čitljiviji ali stvara objekt `mladi` koji nam zapravo ne treba i zatrpava radni prostor.

Pipe operator rješava oba problema. Čita se kao "uzmi ovo i onda napravi ono".

```
anketa |>
  filter(dob < 30) |>
  pull(dnevno_min) |>
  mean()
```

```
[1] 105
```

Čitate ovaj kod odozgo prema dolje, s lijeva na desno, baš kao tekst. Uzmi `anketa`, **zatim** filtriraj retke gdje je dob manja od 30, **zatim** izvuci stupac `dnevno_min`, **zatim** izračunaj prosjek. Svaki `|>` znači "uzmi rezultat prethodnog koraka i proslijedi ga kao prvi argument sljedećoj funkciji".

Evo još jednog primjera koji pokazuje zašto je pipe tako koristan.

```r
anketa |>
  filter(dob < 40) |>
  select(id, platforma, dnevno_min) |>
  arrange(desc(dnevno_min))
```

```
# A tibble: 5 x 3
     id platforma dnevno_min
  <int> <chr>          <dbl>
1     7 TikTok           135
2     3 Instagram        112
3     1 TikTok            95
4     5 TikTok            78
5     4 LinkedIn          45
```

Uzmi anketu, zadrži samo ispitanike mlađe od 40, odaberi tri stupca i sortiraj po dnevnom korištenju od najvećeg prema najmanjem. Svaki korak je jasan i čitljiv. Bez pipea, trebate napisati: `arrange(select(filter(anketa, dob < 40), id, platforma, dnevno_min), desc(dnevno_min))`. Isti rezultat, ali mozak se muči dok ga parsira.

> **Praktični savjet**
>
> Tipkovnička kratica za pipe operator `|>` u Positronu je Ctrl+Shift+M (ili Cmd+Shift+M na Macu). Budući da ćete pipe koristiti u gotovo svakom redu koda od sada pa nadalje, ova kratica će vam uštedjeti mnogo tipkanja. Provjerite u postavkama Positrona da je kratica podešena na native pipe `|>`, a ne na magrittr pipe `%>%`. Oba rade gotovo identično, ali `|>` je noviji i preporučen.

> Pipe operator je poput veznika "i onda" u rečenici. Bez njega, R kod se čita kao telegram. S njim, čita se kao priča.

3.9 Paketi: proširivanje R-a

R sam po sebi dolazi s osnovnim funkcijama (base R), ali prava snaga leži u **paketima** koje je zajednica korisnika razvila za specifične zadatke. Paket je kolekcija funkcija, podataka i dokumentacije koju netko drugi napisao i koju vi možete koristiti u svom radu.

Na ovom kolegiju dominira jedan paket, zapravo kolekcija paketa, koji se zove **tidyverse**.

3.9.1 Što je tidyverse?

Tidyverse nije jedan paket nego skup od osam paketa koji dijele zajedničku filozofiju dizajna i besprijekorno surađuju. Kad učitate tidyverse naredbom `library(tidyverse)`, zapravo učitavate sljedeće pakete.

ggplot2 za vizualizaciju podataka, **dplyr** za manipulaciju podacima (filter, select, mutate, summarise, group_by), **tidyr** za preoblikovanje podataka (pivot_longer, pivot_wider), **readr** za učitavanje podataka (read_csv), **tibble** za moderne tablice podataka, **stringr** za rad s tekstom, **forcats** za rad s faktorima, te **purrr** za funkcionalno programiranje.

Od ovih osam, na ovom kolegiju ćemo najintenzivnije koristiti dplyr, ggplot2, tidyr i readr. S ostalima ćemo se susresti po potrebi.

3.9.2 Instalacija i učitavanje paketa

Paketi se instaliraju jednom, a učitavaju svaki put kad pokrenete R sesiju. Razmislite na analógiju gdje je instalacija poput kupnje knjige (radite to jednom), dok je učitavanje poput otvaranja knjige (radite svaki put kad ju trebate).

```
# Instalacija (samo jednom, u konzoli)
install.packages("tidyverse")

# Učitavanje (na početku svake skripte)
library(tidyverse)
```

Naredbu `install.packages()` pokrećete u konzoli, ne u skripti, jer ne želite da se paket reinstalira svaki put kad pokrenete skriptu. Naredbu `library()` stavljate na početak svake skripte jer R mora znati koje pakete koristite.

> **!** Važna napomena
>
> Kad prvi put instalirate tidyverse, proces može trajati nekoliko minuta jer se instalira mnogo paketa i njihovih ovisnosti. To je normalno. Kad instalacija završi, ne morate ju ponavljati osim ako ne želite ažurirati na noviju verziju. Ako dobijete grešku tijekom instalacije, najčešći uzrok je nedostatak sistemskih biblioteka na Linuxu ili zastarjela verzija R-a. U tom slučaju, ažurirajte R na najnoviju verziju i pokušajte ponovo.

3.10 Učitavanje podataka: read_csv()

Teorija je lijepa, ali prava zabava počinje kad počnemo raditi sa stvarnim (ili barem realistično simuliranim) podacima. Najčešći format za podatke je CSV (comma-separated values), običan tekstualni fajl u kojem su vrijednosti odvojene zarezima. CSV možete otvoriti u bilo čemu, od Excela do Notepada, i gotovo svaki softver ga može izvesti.

Za učitavanje CSV datoteka koristimo funkciju `read_csv()` iz paketa readr (dio tidyverse). Učitajmo dataset o korištenju društvenih mreža koji ćemo koristiti na ovom predavanju.

```
social <- read_csv("../resources/datasets/social_media_survey.csv")
```

Funkcija `read_csv()` čita datoteku i automatski pogađa tipove stupaca. Vraća tibble koji smo pohranili u objekt nazvan `social`. Primijetite da smo koristili `read_csv()` (s podvlakom), a ne `read.csv()` (s točkom). Ovo nije kozmetička razlika. `read_csv()` je brža, automatski stvara tibble (ne data.frame), bolje pogađa tipove stupaca i daje informativnije poruke.

3.10.1 Prvi pogled na podatke

Kad učitate novi dataset, prva stvar koju uvijek radite jest pogledati što se unutra nalazi. Nekoliko funkcija je korisno za to.

```
# Struktura podataka sa stupcima, tipovima i prvim vrijednostima
glimpse(social)
```

```
Rows: 500
Columns: 12
$ respondent_id       <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
$ age                 <dbl> 35, 43, 40, 21, 36, 23, 45, 52, 59, 33, 32, 20, 49~
$ age_group           <chr> "35-44", "35-44", "35-44", "18-24", "35-44", "18-
2~
$ gender              <chr> "male", "female", "female", "female", "male", "fem~
$ education           <chr> "srednja_skola", "prvostupnik", "srednja_skola", "~
$ primary_platform    <chr> "Instagram", "Facebook", "Twitter", "TikTok", "You~
$ daily_minutes       <dbl> 92, 70, 79, 158, 14, 79, 38, 23, 41, 173, 153, 100~
$ num_platforms       <dbl> 4, 3, 3, 5, 3, 1, 2, 1, 2, 2, 2, 4, 2, 2, 2, 4, 3,~
$ trust_social_news   <dbl> 2, 6, 3, 4, 5, 9, 6, 2, 3, 2, 3, 5, 4, 5, 2, 4, 2,~
$ primary_news_source <chr> "portal", "drustvene_mreze", "portal", "portal", "~
$ weekly_posts        <dbl> 2, 9, 6, 14, 6, 0, 0, 0, 0, 3, 3, 19, 0, 2, 0, 13,~
$ privacy_concern     <dbl> 7, 6, 8, 7, 5, 5, 5, 5, 1, 5, 7, 7, 8, 8, 9, 5, 7,~
```

Funkcija `glimpse()` je jedna od najkorisnijih u tidyverse. Na jednom ekranu vidite broj redova i stupaca, ime svakog stupca, tip svakog stupca i prvih nekoliko vrijednosti. To je dovoljno da stvorite mentalnu sliku dataseta.

```r
# Prvih 10 redova
head(social, 10)
```

```
# A tibble: 10 x 12
   respondent_id   age age_group gender education primary_platform daily_minutes
           <dbl> <dbl> <chr>     <chr>  <chr>     <chr>                      <dbl>
 1             1    35 35-44     male   srednja_~ Instagram                     92
 2             2    43 35-44     female prvostup~ Facebook                      70
 3             3    40 35-44     female srednja_~ Twitter                       79
 4             4    21 18-24     female srednja_~ TikTok                       158
 5             5    36 35-44     male   magistar  YouTube                       14
 6             6    23 18-24     female magistar  Instagram                     79
 7             7    45 45-54     female srednja_~ Facebook                      38
 8             8    52 45-54     female srednja_~ Twitter                       23
 9             9    59 55+       male   srednja_~ Instagram                     41
10            10    33 25-34     male   srednja_~ Twitter                      173
# i 5 more variables: num_platforms <dbl>, trust_social_news <dbl>,
#   primary_news_source <chr>, weekly_posts <dbl>, privacy_concern <dbl>
```

Funkcija `head()` prikazuje prvih N redova (po defaultu 6, ali možete zadati drugi broj). Korisna je kad želite vidjeti kako stvarni redovi izgledaju.

```r
# Broj redova i stupaca
nrow(social)
```

```
[1] 500
```

```r
ncol(social)
```

```
[1] 12
```

```r
# Imena stupaca
names(social)
```

```
 [1] "respondent_id"       "age"                 "age_group"
 [4] "gender"              "education"           "primary_platform"
 [7] "daily_minutes"       "num_platforms"       "trust_social_news"
[10] "primary_news_source" "weekly_posts"        "privacy_concern"
```

Naš dataset sadrži 500 ispitanika i 12 varijabli. Varijable uključuju demografske podatke (dob, spol, obrazovanje), podatke o korištenju društvenih mreža (primarna platforma, dnevne minute, broj platformi, tjedno objavljenih postova) i stavove (povjerenje u vijesti na društvenim mrežama, briga za privatnost). Također imamo varijablu o primarnom izvoru vijesti.

3.10.2 Provjera tipova stupaca

Ponekad `read_csv()` ne protumači stupac onako kako bismo željeli. Dobra praksa je provjeriti tipove i napraviti eventualne korekcije.

```
# Pregled prvih redova odabranih stupaca
social |>
  select(respondent_id, age, gender, primary_platform, daily_minutes) |>
  head()
```

```
# A tibble: 6 x 5
  respondent_id   age gender primary_platform daily_minutes
          <dbl> <dbl> <chr>  <chr>                    <dbl>
1             1    35 male   Instagram                   92
2             2    43 female Facebook                    70
3             3    40 female Twitter                     79
4             4    21 female TikTok                     158
5             5    36 male   YouTube                     14
6             6    23 female Instagram                   79
```

Vidimo da je `respondent_id` učitan kao broj (`<dbl>`), `age` kao broj, `gender` kao tekst (`<chr>`), `primary_platform` kao tekst i `daily_minutes` kao broj. Ovo je razumno za naše podatke. U kasnijim tjednima naučit ćemo kako pretvoriti tekstualne stupce u faktore kad nam to bude trebalo za analizu ili vizualizaciju.

3.11 Istraživanje podataka: prvi uvidi

Sad kad imamo podatke učitane, napravimo nekoliko osnovnih istraživanja da stvorimo osjećaj za ono s čime radimo. Ovo je korak koji biste uvijek trebali napraviti prije ikakve ozbiljne analize.

```
# Osnovna deskriptivna statistika za numeričke varijable
summary(social$daily_minutes)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00   51.00   86.50   93.58  136.00  272.00
```

```
summary(social$age)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   23.00   32.00   34.05   42.00   71.00
```

Funkcija `summary()` daje brzi pregled distribucije, uključujući minimum, prvi kvartil, medijan, prosjek, treći kvartil i maksimum. Detaljno ćemo objasniti sve ove mjere u tjednu o deskriptivnoj statistici. Za sada je dovoljno vidjeti da prosječni ispitanik provodi otprilike 90 minuta dnevno na društvenim mrežama i da su dobi raspoređene od 18 do 71 godinu.

Pogledajmo distribuciju kategoričkih varijabli.

```
# Koliko ispitanika po platformi?
social |>
  count(primary_platform, sort = TRUE)
```

```
# A tibble: 8 x 2
  primary_platform     n
  <chr>            <int>
1 Instagram          103
2 Facebook            93
3 TikTok              90
4 YouTube             86
5 LinkedIn            49
6 Twitter             41
7 Snapchat            20
8 Reddit              18
```

Funkcija `count()` je iz paketa dplyr i radi nešto vrlo jednostavno ali korisno. Evo što radi — prebrojava koliko redova pripada svakoj kategoriji. Argument `sort = TRUE` sortira rezultat po frekvenciji od najveće prema najmanjoj. Vidimo da su Instagram, Facebook i TikTok najzastupljenije platforme u našem uzorku.

```
# Odakle ispitanici dobivaju vijesti?
social |>
  count(primary_news_source, sort = TRUE)
```

```
# A tibble: 5 x 2
  primary_news_source     n
  <chr>               <int>
1 drustvene_mreze       199
2 portal                132
3 TV                    102
4 print                  39
5 radio                  28
```

Zanimljivo je da najveći broj ispitanika navodi društvene mreže kao primarni izvor vijesti, što je konzistentno s trendom koji vidimo u istraživanjima diljem svijeta, osobito kod mlađih dobnih skupina.

Kombinirajmo pipe operator s nečim složenijim. Pogledajmo prosječno dnevno korištenje po dobnim skupinama.

```
social |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_min = round(mean(daily_minutes), 1),
    medijan_min = median(daily_minutes)
  )
```

```
# A tibble: 5 x 4
  age_group     n prosjek_min medijan_min
  <chr>     <int>       <dbl>       <dbl>
1 18-24       167       146.        148
2 25-34       130        95.9        94.5
3 35-44       107        61.2        62
4 45-54        58        40.1        38
5 55+          38        26.8        27
```

Ovo je prvi primjer obrasca `group_by() |> summarise()` koji će postati vaš najvažniji alat u tjednima koji dolaze. Logika je jednostavna. `group_by()` dijeli podatke u grupe po varijabli `age_group`, a `summarise()` izračunava statistike za svaku grupu zasebno. Vidimo jasnu razliku u korištenju društvenih mreža između dobnih skupina, s mladima koji provode daleko više vremena na platformama.

> **Praktični savjet**
>
> Funkcija `n()` unutar `summarise()` vraća broj opažanja u svakoj grupi. Uvijek je dobra praksa uključiti `n = n()` u svaki `summarise()` poziv jer vam govori koliko podataka stoji iza svake izračunate statistike. Prosjek izračunat na 5 opažanja je puno manje pouzdan od prosjeka izračunatog na 500 opažanja, i bez `n()` to ne biste znali.

3.12 Logički operatori: kombiniranje uvjeta

U prvom dijelu predavanja vidjeli smo jednostavne usporedbe poput `dob < 30` ili `platforma == "TikTok"`. Ali u stvarnoj analizi rijetko vas zanima samo jedan uvjet. Tipičnije je da

tražite ispitanike koji su mlađi od 25 **i** koriste TikTok, ili ispitanike koji koriste Instagram **ili** Facebook, ili ispitanike koji **ne** pripadaju najstarijoj dobnoj skupini. Za kombiniranje uvjeta koristimo logičke operatore.

I operator (`&`) vraća TRUE samo kad su oba uvjeta ispunjena.

```
# Ispitanici mlađi od 25 koji koriste TikTok
social |>
  filter(age < 25 & primary_platform == "TikTok") |>
  head(8)
```

```
# A tibble: 8 x 12
  respondent_id   age age_group gender education  primary_platform daily_minutes
          <dbl> <dbl> <chr>     <chr>  <chr>      <chr>                     <dbl>
1             4    21 18-24     female srednja_s~ TikTok                      158
2            31    21 18-24     female magistar   TikTok                      218
3            33    24 18-24     female magistar   TikTok                       77
4            44    22 18-24     female srednja_s~ TikTok                      112
5            45    24 18-24     female magistar   TikTok                      115
6            70    23 18-24     male   prvostupn~ TikTok                       67
7            77    24 18-24     female srednja_s~ TikTok                      182
8            83    20 18-24     male   magistar   TikTok                      131
# i 5 more variables: num_platforms <dbl>, trust_social_news <dbl>,
#   primary_news_source <chr>, weekly_posts <dbl>, privacy_concern <dbl>
```

ILI operator (`|`) vraća TRUE kad je barem jedan uvjet ispunjen.

```
# Ispitanici koji koriste Instagram ili TikTok
social |>
  filter(primary_platform == "Instagram" | primary_platform == "TikTok") |>
  count(primary_platform)
```

```
# A tibble: 2 x 2
  primary_platform     n
  <chr>            <int>
1 Instagram          103
2 TikTok              90
```

NE operator (`!`) preokretne logičku vrijednost. Evo kako to radi: TRUE postaje FALSE i obrnuto.

```
# Ispitanici koji NE koriste Facebook
social |>
  filter(!primary_platform == "Facebook") |>
  count(primary_platform, sort = TRUE)
```

```
# A tibble: 7 x 2
  primary_platform     n
  <chr>            <int>
1 Instagram          103
2 TikTok              90
3 YouTube             86
4 LinkedIn            49
5 Twitter             41
6 Snapchat            20
7 Reddit              18
```

Kad trebate provjeriti pripada li vrijednost jednoj od više kategorija, umjesto dugačkog niza ILI uvjeta koristite operator `%in%`.

```
# Ispitanici koji koriste jednu od tri platforme
vizualne_platforme <- c("Instagram", "TikTok", "Snapchat")

social |>
  filter(primary_platform %in% vizualne_platforme) |>
  count(primary_platform, sort = TRUE)
```

```
# A tibble: 3 x 2
  primary_platform     n
  <chr>            <int>
1 Instagram          103
2 TikTok              90
3 Snapchat            20
```

Operator `%in%` je ekvivalentan pisanju `primary_platform == "Instagram" | primary_platform == "TikTok" | primary_platform == "Snapchat"`, ali je dramatično čitljiviji i manje podložan greškama. Kad imate pet ili više kategorija, `%in%` je jedini razuman izbor.

Logičke operatore možete kombinirati i u kontekstu vektora izvan tibbleova.

```
dobi <- c(19, 52, 21, 35, 23, 61, 20, 42)

# Koliko ispitanika je između 20 i 30 godina?
sum(dobi >= 20 & dobi <= 30)
```

```
[1] 3
```

```
# Koliko ih je mlađe od 20 ILI starije od 50?
sum(dobi < 20 | dobi > 50)
```

```
[1] 3
```

Praktični savjet

Česta greška je pisati `platforma == "Instagram" | "TikTok"` umjesto `platforma == "Instagram" | platforma == "TikTok"`. Prva verzija ne radi jer R interpretira `"TikTok"` kao samostalnu logičku vrijednost (neprazan tekst je uvijek TRUE), pa uvjet uvijek vraća TRUE. Koristite `%in%` da izbjegnete ovakve zamke.

3.13 Nedostajuće vrijednosti: NA

U stvarnom svijetu podaci gotovo nikad nisu potpuni. Ispitanik preskoči pitanje u anketi, senzor ne zabilježi podatak, sistem zapiše grešku. R koristi posebnu oznaku **NA** (not available) za nedostajuće vrijednosti i ove vrijednosti zahtijevaju posebnu pažnju od prvog dana jer se ponašaju drugačije od svega ostalog.

Temeljno pravilo je jednostavno i nemilosrdno. Svaka operacija koja uključuje NA vraća NA.

```
# Vektor s nedostajućom vrijednošću
ocjene <- c(4, 5, NA, 3, 4)

# Prosjek vektora s NA
mean(ocjene)
```

```
[1] NA
```

```
# Zbroj vektora s NA
sum(ocjene)
```

```
[1] NA
```

Rezultat je NA, ne broj. R ne pretpostavlja da možete ignorirati nedostajuću vrijednost jer ne znate što bi ta vrijednost bila. Možda je nedostajuća ocjena bila 1, možda 5, a možda nešto između. Prosjek s tom vrijednošću i bez nje bio bi različit. R vas prisiljava da svjesno odlučite što ćete učiniti.

Najčešće rješenje je argument `na.rm = TRUE` koji govori R-u da ignorira NA vrijednosti.

```
# Prosjek bez NA vrijednosti
mean(ocjene, na.rm = TRUE)
```

```
[1] 4
```

```r
# Zbroj bez NA vrijednosti
sum(ocjene, na.rm = TRUE)
```

```
[1] 16
```

```r
# Medijan, SD, min, max - svi imaju na.rm argument
median(ocjene, na.rm = TRUE)
```

```
[1] 4
```

```r
sd(ocjene, na.rm = TRUE)
```

```
[1] 0.8164966
```

3.13.1 Provjera i prepoznavanje NA

Za otkrivanje NA vrijednosti koristimo funkciju `is.na()`, nikad usporedbu s `==`.

```r
# ISPRAVNO: is.na()
is.na(ocjene)
```

```
[1] FALSE FALSE  TRUE FALSE FALSE
```

```r
# Koliko je NA vrijednosti?
sum(is.na(ocjene))
```

```
[1] 1
```

```r
# NEISPRAVNO: usporedba s == ne radi za NA
ocjene == NA
```

```
[1] NA NA NA NA NA
```

Usporedba `ocjene == NA` vraća niz NA vrijednosti, ne TRUE/FALSE. To je zato što je NA nepoznata vrijednost, a usporedba nečega nepoznatog s nepoznatim daje nepoznat rezultat. Ovo je logično kad se zamisli — ako ne znate koliko je Ana visoka i ne znate koliko je Marko visok, ne možete reći jesu li jednako visoki. Odgovor je “ne znam”, dakle NA.

3.13.2 NA u tibbleovima

Kad učitavate podatke, prazne ćelije i tekstualne oznake poput "N/A" ili "missing" automatski se pretvaraju u NA (ili bi se trebale, ovisno o formatu). U tidyverse okruženju, provjera NA u cijelom datasetu izgleda ovako.

```
# Provjera NA za svaki stupac
social |>
  summarise(across(everything(), ~sum(is.na(.x))))
```

```
# A tibble: 1 x 12
  respondent_id   age age_group gender education primary_platform daily_minutes
          <int> <int>     <int>  <int>     <int>            <int>         <int>
1             0     0         0      0         0                0             0
# i 5 more variables: num_platforms <int>, trust_social_news <int>,
#   primary_news_source <int>, weekly_posts <int>, privacy_concern <int>
```

Naš simulirani dataset nema nedostajućih vrijednosti, ali u stvarnim podacima ih gotovo uvijek ima. Navikavanje na provjeru NA od prvog kontakta s podacima je navika koja vam štedi sate frustracije. Detaljno ćemo obraditi strategije za rad s nedostajućim vrijednostima u tjednu o deskriptivnoj statistici, uključujući razliku između podataka koji nedostaju nasumično i onih koji nedostaju sustavno.

> **!** Važna napomena
>
> Nikada nemojte pretpostaviti da vaši podaci nemaju NA. Čak i kad su podaci "čisti", funkcije poput `read_csv()` ponekad stvore NA na neočekivanim mjestima (prazna ćelija, razmak umjesto broja, tekst u numeričkom stupcu). Pravilo je jednostavno — uvijek provjerite, nikad ne pretpostavljajte.

3.14 Korisne funkcije za vektore

Prije nego prijeđemo na pisanje skripti, vrijedi proći još nekoliko funkcija koje ćete često koristiti. Sve rade na vektorima i pojavljuju se u gotovo svakoj analizi.

3.14.1 Generiranje nizova

```
# Niz cijelih brojeva od 1 do 10
1:10
```

```
 [1]  1  2  3  4  5  6  7  8  9 10
```

```r
# Niz s zadanim korakom
seq(from = 0, to = 100, by = 10)
```

```
 [1]   0  10  20  30  40  50  60  70  80  90 100
```

```r
# Niz zadane duljine
seq(from = 0, to = 1, length.out = 5)
```

```
[1] 0.00 0.25 0.50 0.75 1.00
```

```r
# Ponavljanje
rep("kontrolna", times = 5)
```

```
[1] "kontrolna" "kontrolna" "kontrolna" "kontrolna" "kontrolna"
```

```r
rep(c("A", "B"), times = 3)
```

```
[1] "A" "B" "A" "B" "A" "B"
```

```r
rep(c("A", "B"), each = 3)
```

```
[1] "A" "A" "A" "B" "B" "B"
```

Funkcija `seq()` stvara pravilne nizove. Koristit ćemo je kad budemo trebali osi za grafike ili sekvence za simulacije. Funkcija `rep()` ponavlja vrijednosti i korisna je kad kreirate testne podatke ili oznake za eksperimentalne grupe. Obratite pažnju na razliku između `times` (ponavlja cijeli vektor) i `each` (ponavlja svaki element).

3.14.2 Sortiranje i redoslijed

```r
minute <- c(95, 22, 112, 45, 78, 8, 135, 55)

# Sortirano uzlazno
sort(minute)
```

```
[1]   8  22  45  55  78  95 112 135
```

```r
# Sortirano silazno
sort(minute, decreasing = TRUE)
```

```
[1] 135 112  95  78  55  45  22   8
```

```r
# Rang (pozicija u sortiranom nizu)
rank(minute)
```

```
[1] 6 2 7 3 5 1 8 4
```

```r
# Indeksi koji bi sortirali vektor
order(minute)
```

```
[1] 6 2 4 8 5 1 3 7
```

Funkcija `sort()` vraća sortirane vrijednosti. Funkcija `rank()` vraća rang svakog elementa (najmanji dobiva rang 1). Funkcija `order()` vraća indekse koji bi sortirali vektor, što je korisno za sortiranje jednog vektora prema redoslijedu drugog. U tidyverse pristupu češće koristimo `arrange()` za sortiranje tibbleova, ali `sort()` i `rank()` ostaju korisni za rad s pojedinačnim vektorima.

3.14.3 Jedinstvene vrijednosti i tablice frekvencija

```r
platforme <- c("TikTok", "Instagram", "TikTok", "YouTube", "Instagram", "TikTok", "Faceboo

# Jedinstvene vrijednosti
unique(platforme)
```

```
[1] "TikTok"    "Instagram" "YouTube"   "Facebook"
```

```r
# Broj jedinstvenih vrijednosti
length(unique(platforme))
```

```
[1] 4
```

```r
# Tablica frekvencija (base R)
table(platforme)
```

```
platforme
 Facebook Instagram    TikTok   YouTube
        1         2         3         1
```

Funkcija `unique()` vraća sve različite vrijednosti u vektoru. `table()` prebrojava koliko se puta svaka vrijednost pojavljuje. U tidyverse pristupu, `count()` radi isto ali elegantnije i vraća tibble umjesto tablice. Ipak, `unique()` i `length(unique())` su toliko korisni da ih vrijedi znati neovisno o tidyverse.

3.14.4 Zaokruživanje i formatiranje

```
x <- 3.14159265

# Zaokruživanje na N decimala
round(x, 2)
```

```
[1] 3.14
```

```
round(x, 4)
```

```
[1] 3.1416
```

```
# Zaokruživanje prema gore i dolje
ceiling(2.3)
```

```
[1] 3
```

```
floor(2.9)
```

```
[1] 2
```

```
# Značajne znamenke
signif(x, 3)
```

```
[1] 3.14
```

Funkcija `round()` se pojavljuje konstantno jer je rezultate statističkih izračuna gotovo uvijek potrebno zaokružiti prije prikazivanja. Konvencija u akademskim radovima je obično 2 decimale za korelacije i p-vrijednosti, 1 decimala za prosjeke i standardne devijacije. Na ovom kolegiju ćemo se držati tih konvencija.

3.15 Pisanje čistih R skripti

Do sada smo pisali kod redak po redak, ali u praksi ćete pisati **skripte**, datoteke koje sadrže sav kod za jednu analizu od početka do kraja. Čista skripta je nešto što možete dati kolegi, i kolega može pokrenuti vaš kod i dobiti identične rezultate. To je suština ponovljivosti o kojoj smo govorili na početku.

Dobra R skripta ima jasnu strukturu.

```
# ============================================
# Analiza korištenja društvenih mreža
# Kolegij: Statistika za komunikologe
# Datum: 2025-03-01
# Autor: Ime Prezime
# ============================================

# 1. Učitavanje paketa ----
library(tidyverse)

# 2. Učitavanje podataka ----
social <- read_csv("resources/datasets/social_media_survey.csv")

# 3. Pregled podataka ----
glimpse(social)

# 4. Deskriptivna statistika po dobnim skupinama ----
social |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_min = round(mean(daily_minutes), 1),
    sd_min = round(sd(daily_minutes), 1),
    .groups = "drop"
  )

# 5. Najpopularnije platforme ----
social |>
  count(primary_platform, sort = TRUE)
```

Primijetite nekoliko stvari o ovoj skripti. Na vrhu je zaglavlje u komentarima koje objašnjava što skripta radi, za koji kolegij je, tko ju je napisao i kad. Sekcije su označene komentarima s četiri crtice na kraju (`# Naslov ----`), što Positron prepoznaje i prikazuje kao navigacijske točke u bočnom panelu. Svaka sekcija ima jasan opis. Kod teče logički — najprije paketi, pa podaci, pa pregled, pa analiza.

3.15.1 Radni direktorij i putanje do datoteka

Jedna od najčešćih frustracija za početnike je problem s putanjama do datoteka. Kad napišete `read_csv("social_media_survey.csv")`, R traži tu datoteku u **radnom direktoriju** (working directory). Ako datoteka nije tamo, dobit ćete grešku.

```
# Koji je trenutni radni direktorij?
getwd()
```

```
[1] "C:/Users/lsikic/Dropbox/HKS/Kolegiji/Osnove statistike/GHub/lectures"
```

U Positronu, radni direktorij se obično automatski postavlja na mapu u kojoj je otvorena R datoteka ili projekt. To znači da ako je vaša skripta u mapi `projekt/analize/` i dataset u mapi `projekt/podaci/`, putanja u skripti bi bila `"../podaci/social_media_survey.csv"` (dvije točke znače "idi jednu mapu gore").

> **Praktični savjet**
>
> Najbolja praksa je koristiti **R projekte** (ili Quarto projekte) jer automatski postavljaju radni direktorij na korijensku mapu projekta. Kad otvorite projekt u Positronu, svi putovi su relativni prema toj mapi, i nikad ne morate razmišljati o apsolutnim putanjama poput `C:/Users/Ana/Documents/faks/statistika/podaci/...`. Apsolutne putanje su problem jer ne rade na tuđem računalu (kolega nema mapu `Ana` na svom disku). Relativne putanje rade svugdje jer polaze od mape projekta.

3.16 Spremanje podataka: write_csv()

Jednako važno kao učitavanje podataka jest njihovo spremanje. Nakon što očistite podatke ili izračunate nove varijable, želite pohraniti rezultat da ne morate ponavljati iste korake svaki put. Funkcija `write_csv()` sprema tibble u CSV datoteku.

```
# Kreiranje sažetka
sazetak_po_dobi <- social |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_min = round(mean(daily_minutes), 1),
    sd_min = round(sd(daily_minutes), 1),
    prosjek_platformi = round(mean(num_platforms), 1),
    .groups = "drop"
```

```
  )

# Spremanje u CSV
write_csv(sazetak_po_dobi, "rezultati/sazetak_po_dobi.csv")
```

Funkcija `write_csv()` prima dva argumenta: tibble koji želite spremiti i putanju s imenom datoteke. Mapa `rezultati/` mora postojati prije nego pozovete funkciju, inače ćete dobiti grešku. Možete je kreirati ručno u datotečnom pregledniku ili iz R-a naredbom `dir.create("rezultati")`.

3.17 Traženje pomoći

Čak i iskusni R korisnici redovito trebaju pomoć. R ima ugrađeni sustav dokumentacije koji je izuzetno detaljan, i postoji nekoliko načina da mu pristupite.

```
# Pomoć za specifičnu funkciju
?mean
help(mean)

# Pretraživanje pomoći po ključnoj riječi
??correlation

# Primjeri korištenja funkcije
example(mean)
```

Upitnik ispred imena funkcije (`?mean`) otvara stranicu pomoći za tu funkciju. Stranica sadrži opis, listu argumenata, detalje o ponašanju, povratnu vrijednost i primjere. Na početku stranice pomoći djeluju zastrašujuće jer su pisane tehničkim jezikom, ali brzo ćete naučiti preskočiti na sekciju **Examples** na dnu, koja gotovo uvijek postoji i pokazuje kako se funkcija koristi u praksi.

3.17.1 Kad pomoć ne pomaže: internet

Realno, za većinu problema ćete koristiti internet. Tri resursa su daleko najkorisnija.

Stack Overflow je forum za programerska pitanja. Gotovo svako pitanje o R-u koje možete zamisliti već je postavljeno i odgovoreno na Stack Overflowu. Ključ je znati kako formulirati pitanje za pretragu. Umjesto “moj kod ne radi”, tražite “r dplyr filter multiple conditions” ili “r ggplot change axis labels”.

Posit Community (community.rstudio.com) je forum specifičan za R i tidyverse. Atmosfera je prijateljska i odgovori su obično vrlo detaljni.

R dokumentacija i vinjete. Mnogi paketi dolaze s vinjetama (vignettes), dugačkim dokumentima koji objašnjavaju filozofiju paketa i pokazuju tipične radne tokove. Vinjete za dplyr (`vignette("dplyr")`) i ggplot2 su izvrsni resursi.

> Praktični savjet
>
> Kad tražite pomoć na internetu, uvijek uključite "tidyverse" ili ime paketa u pretragu. Bez toga, odgovori će često biti u base R sintaksi koja je drugačija od onoga što koristimo na kolegiju. Na primjer, tražite "tidyverse filter rows by condition" umjesto "R filter rows".

3.18 Česte greške i kako ih popraviti

Greške su normalan i neizbježan dio programiranja. Čak i nakon godina iskustva, R korisnici redovito dobivaju poruke o greškama. Razlika između početnika i iskusnog korisnika nije u tome koliko grešaka prave, nego u tome koliko brzo ih prepoznaju i poprave. Pogledajmo najčešće greške s kojima ćete se susresti.

3.18.1 Greška: objekt nije pronađen

```
# Error: object 'prosjek' not found
prosjek_minuta <- mean(social$daily_minutes)
prosjek   # krivi naziv, nedostaje "_minuta"
```

Ova greška znači da R ne može pronaći objekt s tim imenom. Najčešći uzroci su pogrešno ime (tipfeler), zaboravljeno pokretanje koda koji kreira objekt, ili pokretanje koda izvan redoslijeda (pokušavate koristiti objekt prije nego ste ga kreirali). Što trebate učiniti: provjerite ime i provjerite jeste li pokrenuli sve prethodne retke koda.

3.18.2 Greška: neočekivani simbol

```
# Error: unexpected symbol in "social |> filter(age < 30 daily_minutes > 60)"
social |> filter(age < 30 daily_minutes > 60)   # nedostaje &
```

R je naišao na nešto što ne očekuje. Najčešće nedostaje operator (`&`, `,`, `|>`), nedostaje zatvarajuća zagrada, ili ste zaboravili zarez između argumenata. Rješenje: pažljivo pregledajte redak i usporedite s ispravnom sintaksom.

3.18.3 Greška: datoteka nije pronađena

```
# Error: 'podaci.csv' does not exist in current working directory
read_csv("podaci.csv")
```

R ne može pronaći datoteku na zadanoj putanji. Provjerite je li naziv datoteke ispravan (uključujući velika i mala slova), je li datoteka u radnom direktoriju, i je li putanja ispravna. Koristite `getwd()` da vidite gdje R traži datoteke.

3.18.4 Upozorenje naspram greške

Važno je razlikovati **greške** (errors) od **upozorenja** (warnings). Greška zaustavlja izvršavanje koda jer R ne može nastaviti. Upozorenje ne zaustavlja izvršavanje ali vas obavještava da se nešto neobično dogodilo. Na primjer, kad pretvarate tekst u broj, a tekst sadrži slova.

```
# Ovo daje upozorenje ali ne grešku
as.numeric(c("10", "20", "trideset"))
```

```
Warning: NAs introduced by coercion
```

```
[1] 10 20 NA
```

R je uspio pretvoriti "10" i "20" u brojeve, ali "trideset" nije mogao pretvoriti i stavio je NA. Upozorenje vam govori da je nešto pošlo po krivu, ali kod se izvršio do kraja. Uvijek čitajte upozorenja jer vam govore o potencijalnim problemima s podacima.

> **!** Važna napomena
>
> Kad dobijete grešku, pročitajte poruku o grešci. Zvuči očito, ali većina početnika reagira panikom umjesto čitanjem. R poruke o greškama su obično informativne i govore vam što je pošlo po krivu. Tekst "object 'x' not found" vam doslovno govori da objekt x ne postoji. Tekst "unexpected symbol" govori da je nešto krivo sa sintaksom. Čitanje poruke je prvi i najvažniji korak u rješavanju problema.

3.19 Sve zajedno: kompletna mini analiza

Zaokružimo ovo predavanje tako da povežemo sve što smo naučili u jednu koherentnu analizu. Istražit ćemo koji izvor vijesti dominira u različitim dobnim skupinama i kako je korištenje društvenih mreža povezano s povjerenjem u vijesti na tim platformama.

```
# Primarni izvor vijesti po dobnim skupinama
social |>
  group_by(age_group, primary_news_source) |>
  summarise(n = n(), .groups = "drop") |>
  group_by(age_group) |>
  mutate(postotak = round(n / sum(n) * 100, 1)) |>
  arrange(age_group, desc(postotak))
```

```
# A tibble: 25 x 4
# Groups:   age_group [5]
   age_group primary_news_source     n postotak
   <chr>     <chr>                <int>    <dbl>
 1 18-24     drustvene_mreze         80     47.9
 2 18-24     portal                  50     29.9
 3 18-24     TV                      25     15
 4 18-24     print                    7      4.2
 5 18-24     radio                    5      3
 6 25-34     drustvene_mreze         55     42.3
 7 25-34     portal                  37     28.5
 8 25-34     TV                      18     13.8
 9 25-34     print                   10      7.7
10 25-34     radio                   10      7.7
# i 15 more rows
```

Ovaj kod radi nešto složenije nego što smo do sada vidjeli. Najprije grupira podatke po dobnoj skupini i izvoru vijesti te broji ispitanike. Zatim, unutar svake dobne skupine, računa postotak. Funkciju `mutate()` koristimo za kreiranje novog stupca, a detalje ćemo objasniti sljedeći tjedan. Za sada je dovoljno vidjeti obrazac i rezultat.

Vidimo jasnu razliku između generacija. Mladi (18 do 24) dominantno koriste društvene mreže kao primarni izvor vijesti, dok stariji ispitanici (55+) preferiraju televiziju. Ovo je konzistentno s istraživanjima diljem svijeta i ilustrira zašto je raščlamba po dobnim skupinama toliko važna, tema koju smo obradili i u prvom tjednu kad smo govorili o Simpsonovom paradoksu.

Pogledajmo sada vezu između dnevnog korištenja i povjerenja u vijesti na društvenim mrežama.

```
social |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_min = round(mean(daily_minutes), 1),
    prosjek_trust = round(mean(trust_social_news), 1),
    korelacija = round(cor(daily_minutes, trust_social_news), 2),
    .groups = "drop"
  )
```

```
# A tibble: 5 x 5
  age_group     n prosjek_min prosjek_trust korelacija
  <chr>     <int>       <dbl>         <dbl>      <dbl>
1 18-24       167       146.            5.4      -0.06
2 25-34       130        95.9           4.5       0.12
3 35-44       107        61.2           3.7      -0.04
4 45-54        58        40.1           3         0.04
5 55+          38        26.8           2.5       0.33
```

Ovo je tablica koja komunicira mnogo informacija. Za svaku dobnu skupinu vidimo koliko ispitanika imamo, koliko u prosjeku koriste društvene mreže, koliko im vjeruju kao izvoru vijesti i kakva je korelacija između korištenja i povjerenja unutar svake skupine. Ovakve tablice čine okosnicu svakog deskriptivnog izvještaja u komunikologiji.

Na kraju, pogledajmo koliko platformi u prosjeku koriste različite dobne skupine i kako se to razlikuje po spolu.

```
social |>
  group_by(age_group, gender) |>
  summarise(
    n = n(),
    prosjek_platformi = round(mean(num_platforms), 1),
    .groups = "drop"
  ) |>
  filter(gender != "non-binary") |>
  arrange(age_group, gender)
```

```
# A tibble: 10 x 4
   age_group gender     n prosjek_platformi
   <chr>     <chr>  <int>             <dbl>
 1 18-24     female    72               4
 2 18-24     male      89               4
 3 25-34     female    71               3.2
 4 25-34     male      55               3.1
```

```
 5 35-44     female    44              3
 6 35-44     male      63              2.7
 7 45-54     female    34              1.7
 8 45-54     male      23              1.7
 9 55+       female    17              1.5
10 55+       male      21              1.7
```

Mladi koriste više platformi od starijih ispitanika, što je logično. Razlika između spolova je relativno mala u usporedbi s razlikom između dobnih skupina. Ovo je opet ilustracija važnog principa. Kad gledate ukupne prosjeke, gubite informaciju o tome koja varijabla zapravo objašnjava razlike.

> Svaka analiza podataka počinje s pitanjem. Dobar analitičar ne otvara dataset i "vidi što će pronaći". Dobar analitičar ima pitanje, prevede ga u kod i interpretira rezultat u kontekstu tog pitanja.

! Ključni zaključci

1. R je programski jezik za statističko računanje koji nudi ponovljivost, fleksibilnost i profesionalnu vizualizaciju. Početna krivulja učenja je strmija od softvera s grafičkim sučeljem, ali dugoročna isplativost je znatno veća.
2. Positron je moderno razvojno okruženje (IDE) koje čini rad s R-om ugodnijim. Radni tok uključuje pisanje koda u editoru, izvršavanje u konzoli i pregled rezultata.
3. Objekti pohranjuju vrijednosti za kasniju upotrebu. Koristite opisna imena u snake_case konvenciji i ne reciklirajte objekte za različite svrhe.
4. Vektori su uređeni nizovi vrijednosti istog tipa. R automatski primjenjuje operacije na sve elemente vektora (vektorizacija), što omogućuje efikasan rad s podacima.
5. Četiri osnovna tipa podataka su numerički (`numeric`), tekstualni (`character`), logički (`logical`) i faktorski (`factor`). Razumijevanje tipova ključno je za dijagnosticiranje problema s podacima.
6. Tibble je moderna tablica podataka u kojoj svaki stupac može biti drugog tipa. Standard je u tidyverse ekosustavu.
7. Pipe operator (`|>`) čini kod čitljivijim jer omogućuje ulančavanje operacija u prirodnom redoslijedu, odozgo prema dolje. Koristite ga uvijek kad imate više od jednog koraka.
8. Funkcija `read_csv()` učitava CSV datoteke u tibble. Nakon učitavanja, uvijek pregledajte podatke s `glimpse()`, `head()` i `count()`.

9. Logički operatori (`&`, `|`, `!`, `%in%`) omogućuju kombiniranje uvjeta za precizno filtriranje i odabir podataka.
10. NA (nedostajuće vrijednosti) zahtijevaju svjesnu odluku o tretmanu. Uvijek provjerite ima li ih u podacima i koristite `na.rm = TRUE` kad je prikladno.
11. Greške su normalan dio programiranja. Čitajte poruke o greškama, provjerite imena objekata i zagrade, i koristite internet kao resurs za rješavanje problema.
12. Čista R skripta ima jasnu strukturu s zaglavljem, učitavanjem paketa, učitavanjem podataka, pregledom podataka i analizom. Koristite komentare i relativne putanje.

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan nastavljamo s radom s podacima u tidyverse. Naučit ćemo temeljne funkcije za manipulaciju podacima — `filter()` za odabir redova po uvjetu, `select()` za odabir stupaca, `mutate()` za kreiranje novih varijabli, `arrange()` za sortiranje, te `pivot_longer()` i `pivot_wider()` za preoblikovanje podataka. Ove funkcije, u kombinaciji s `group_by()` i `summarise()` koje smo već upoznali, čine okosnicu svake analize podataka u R-u.
Za pripremu napravite sljedeće:

1. Provjerite da vam R i Positron rade ispravno. Otvorite Positron, stvorite novu R datoteku i pokrenite `library(tidyverse)`. Ako ne dobijete grešku, spremni ste.
2. Ponovite sve primjere iz ovog predavanja. Nemojte samo čitati kod, nego ga upišite i pokrenite. Trebate eksperimentirati primjenom promjena brojeva, dodavanja novih vektora i testiranja što se dogodi kad nešto napravite krivo.
3. Učitajte dataset `social_media_survey.csv` i pokušajte odgovoriti na sljedeća pitanja koristeći `filter()`, `count()` i `group_by() |> summarise()`:
   - Koja je najčešća platforma među ispitanicima starijim od 45 godina?
   - Koliki je prosječni broj dnevnih minuta za korisnike Instagrama u usporedbi s korisnicima Facebooka?
   - Koliko ispitanika koristi 5 ili više platformi?
4. Pročitajte poglavlja 3 i 4 iz knjige R for Data Science (besplatno online na r4ds.hadley.nz). Poglavlje 3 pokriva osnove tidyverse radnog toka, a poglavlje 4 transformaciju podataka s dplyr.

3.20 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 3: Getting Started with R. Besplatno dostupno na learningstatisticswithr.com. Poglavlje pokriva iste teme kao ovo predavanje, ali koristi base R pristup umjesto tidyverse. Korisno za razumijevanje osnova jezika, ali kod iz knjige ne koristimo izravno na kolegiju.

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 1, 3 i 4. Besplatno dostupno na r4ds.hadley.nz. Poglavlje 1 daje motivaciju za tidyverse pristup, poglavlje 3 pokriva transformaciju podataka, a poglavlje 4 organizaciju radnog toka.

Preporučeno

Ismay, C. & Kim, A. (2020). *Statistical Inference via Data Science: A ModernDive into R and the Tidyverse*. Besplatno dostupno na moderndive.com. Alternativni udžbenik koji od početka koristi tidyverse i naglašava vizualno razmišljanje.

Bryan, J. & Hester, J. *What They Forgot to Teach You About R*. Besplatno dostupno na rstats.wtf. Pokriva praktične aspekte rada s R-om: projekte, putanje, radne direktorije, organizaciju datoteka. Čitanje za one koji žele profesionalizirati svoj radni tok.

3.21 Pojmovnik

Pojam	Objašnjenje
R	Programski jezik i okruženje za statističko računanje i vizualizaciju. Besplatan i open-source.
Positron	Moderno integrirano razvojno okruženje (IDE) za rad s R-om, razvijeno od strane Posit tima.
IDE (integrirano razvojno okruženje)	Program koji kombinira editor teksta, konzolu, pregled varijabli i druge alate u jednom sučelju.
Objekt	Pohranjena vrijednost u R-u kojoj se pristupa putem imena. Kreira se operatorom `<-`.
Vektor	Uređeni niz vrijednosti istog tipa. Temeljna struktura podataka u R-u. Kreira se funkcijom `c()`.
Vektorizacija	Svojstvo R-a da automatski primjenjuje operacije na sve elemente vektora odjednom.

Pojam	Objašnjenje
Tip podataka	Klasifikacija vrijednosti koja određuje moguće operacije. Osnovni tipovi: numeric, character, logical, factor.
Faktor (factor)	Poseban tip podataka za kategorijalne varijable. Sadrži unaprijed definirane razine (levels).
Tibble	Moderna verzija data framea iz tidyverse ekosustava. Prikazuje tipove stupaca i ograničava ispis na prvih 10 redova.
Data frame	Tablica u R-u u kojoj svaki stupac može biti drugog tipa. Tibble je poboljšana verzija.
Pipe operator (`|>`)	Operator koji proslijeđuje rezultat jednog izraza kao prvi argument sljedećoj funkciji. Čini kod čitljivijim.
Tidyverse	Kolekcija R paketa za rad s podacima koji dijele zajedničku filozofiju dizajna. Uključuje ggplot2, dplyr, tidyr, readr, tibble i druge.
Paket (package)	Kolekcija R funkcija, podataka i dokumentacije koja proširuje mogućnosti R-a. Instalira se s `install.packages()`, učitava s `library()`.
`read_csv()`	Funkcija iz paketa readr za učitavanje CSV datoteka. Vraća tibble i automatski pogađa tipove stupaca.
`write_csv()`	Funkcija iz paketa readr za spremanje tibble u CSV datoteku. Korisna za izvoz obrađenih podataka.
`glimpse()`	Funkcija iz tidyverse koja prikazuje strukturu dataseta: stupce, tipove i prvih nekoliko vrijednosti.
`count()`	Funkcija iz dplyr koja prebrojava opažanja po kategorijama.
`group_by()`	Funkcija iz dplyr koja dijeli podatke u grupe po jednoj ili više varijabli. Koristi se u kombinaciji sa `summarise()`.
`summarise()`	Funkcija iz dplyr koja izračunava sažetke (prosjek, medijan, SD i sl.) za svaku grupu ili za cijeli dataset.
`filter()`	Funkcija iz dplyr koja odabire retke koji zadovoljavaju zadani uvjet.
`mutate()`	Funkcija iz dplyr koja kreira nove stupce ili mijenja postojeće.
`arrange()`	Funkcija iz dplyr koja sortira retke po vrijednostima jednog ili više stupaca.

Pojam	Objašnjenje
`select()`	Funkcija iz dplyr koja odabire stupce po imenu.
NA (not available)	Oznaka za nedostajuću vrijednost u R-u. Svaka operacija s NA vraća NA osim ako se eksplicitno kaže `na.rm = TRUE`.
`is.na()`	Funkcija koja provjerava jesu li vrijednosti NA. Jedini ispravan način provjere (nikad koristiti `== NA`).
Logički operatori	Operatori za kombiniranje uvjeta: `&` (i), `|` (ili), `!` (ne), `%in%` (pripada skupu).
Skripta	Tekstualna datoteka (.R) koja sadrži R kod od početka do kraja analize. Omogućuje ponovljivost.
Radni direktorij	Mapa u kojoj R traži datoteke i sprema rezultate. Provjerava se s `getwd()`.
snake_case	Konvencija imenovanja: riječi odvojene podvlakom, sve malim slovima (npr. `dnevno_koristenje`). Standard u tidyverse zajednici.
CSV (comma-separated values)	Tekstualni format za pohranu tabličnih podataka u kojem su vrijednosti odvojene zarezima.
Komentar	Tekst u kodu koji počinje s `#` i koji R ignorira. Služi za objašnjavanje koda.

4 Tjedan 3: Rad s podacima u tidyverse

Od sirovih podataka do analizi spremnog dataseta

```
library(tidyverse)
library(janitor)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti zašto je čišćenje i transformacija podataka najvažniji (i najdugotrajniji) korak u svakoj analizi.
2. Koristiti `clean_names()` za standardizaciju imena stupaca i prepoznati zašto je to važno za ponovljivost.
3. Koristiti `filter()` za odabir redova po jednom ili više uvjeta, uključujući kombinacije logičkih operatora i rad s nedostajućim vrijednostima.
4. Koristiti `select()` za odabir, preimenovanje i preuređivanje stupaca, uključujući pomoćne funkcije poput `starts_with()`, `ends_with()` i `contains()`.
5. Koristiti `mutate()` za kreiranje novih varijabli, transformaciju postojećih i rekodiranje vrijednosti pomoću `case_when()` i `if_else()`.
6. Koristiti `arrange()` za sortiranje podataka po jednom ili više stupaca u uzlaznom i silaznom redoslijedu.
7. Kombinirati dplyr glagole u pipeline koristeći pipe operator za složene transformacije podataka.
8. Prepoznati tipične probleme u sirovim podacima (nekonzistentno kodiranje, mješoviti tipovi, nedostajuće vrijednosti) i primijeniti odgovarajuće strategije čišćenja.

4.1 Prljava tajna analize podataka

Postoji jedna stvar o kojoj vam udžbenici statistike rijetko govore. Otvorite bilo koji udžbenik i vidjet ćete poglavlje o t-testu, poglavlje o regresiji, poglavlje o ANOVA-i. Sve lijepo i uredno. Ali nitko vam ne kaže da ćete 80% vremena u bilo kojoj analizi provesti na nečemu što se ne pojavljuje ni u jednom od tih poglavlja. Radi se o čišćenju i pripremi podataka.

Ovo nije pretjerivanje. Stvarni podaci su gotovo uvijek neuredni. Anketa prikupljena putem Google Formsa dolazi s imenima stupaca poput "Koliko često pratite vijesti na društvenim mrežama? (odaberite jedan odgovor)". Ispitanici u polje za spol upisuju "Ženski", "ženski", "Ž", "female" i "Zensko", a sve to treba biti ista kategorija. Stupac koji bi trebao sadržavati brojeve sadrži i tekst poput "ne gledam" ili prazne ćelije. Neki ispitanici imaju 19 godina, a jedan ima 199 jer mu je prst skliznuo na tipkovnici.

Sve ovo morate riješiti prije nego što možete izračunati ijedan prosjek ili napraviti ijedno testiranje hipoteza. I upravo zato je ovaj tjedan posvećen manipulaciji podacima. Naučit ćemo pet temeljnih funkcija iz paketa dplyr (`filter()`, `select()`, `mutate()`, `summarise()`, `group_by()`) plus alate za čišćenje i preoblikovanje iz paketa tidyr i janitor. Ove funkcije, spojene pipe operatorom u elegantne pipeline, čine okosnicu svake analize podataka u R-u.

Navarro u knjizi (poglavlja 4 i 7) pokriva sličan teren, ali u base R sintaksi. Mi ćemo koristiti tidyverse pristup koji je čitljiviji i konzistentniji. Kad ste jednom naučili logiku dplyr glagola, ista logika se primjenjuje na svaki dataset, svaki problem, svaku analizu.

4.2 Naši podaci: anketa o medijskim navikama studenata

Na ovom predavanju koristit ćemo simulirani dataset koji oponaša ono što biste zaista dobili kad biste proveli online anketu među studentima. Dataset je namjerno neuredan jer želimo vježbati čišćenje podataka na realističnom primjeru.

Učitajmo podatke i pogledajmo s čime se suočavamo.

```
raw <- read_csv("../resources/datasets/media_habits_raw.csv")
glimpse(raw)
```

```
Rows: 250
Columns: 17
$ `ID respondenta`                  <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,~
$ Timestamp                         <dttm> 2025-03-28 17:05:00, 2025-04-
20 2~
$ Dob                               <dbl> 20, 27, 27, 18, 25, 26, 28, 26, 22~
$ Spol                              <chr> "ženski", "Muški", "muški", "femal~
$ Grad                              <chr> "Zagreb", "Zadar", "Zagreb", "Spli~
$ `Godina studija`                  <chr> "2", "3", "1", "1", "2", "2.", "2"~
$ `TV (min/dan)`                    <chr> "0", "0", "65", NA, NA, "91", "91"~
$ `Portali (min/dan)`               <dbl> 40, 20, 0, 11, 32, 25, 81, 28, 37,~
$ `Društvene mreže (min/dan)`       <dbl> 59, 101, 177, 71, 161, 155, 114, 1~
$ `Radio (min/dan)`                 <dbl> 49, NA, 0, NA, 26, NA, 0, 0, 17, 0~
$ `Podcast (min/dan)`               <dbl> 89, 0, 49, 0, 0, NA, NA, 31, 0, 19~
$ `Povjerenje TV (1-10)`            <dbl> 2, 3, 4, 5, 5, 4, 3, 6, 6, 7, 2, 7~
```

```
$ `Povjerenje portali (1-10)`         <dbl> 6, 5, 6, 6, 3, 7, 7, 1, 7, 5, 6, 5~
$ `Povjerenje društvene mreže (1-10)` <dbl> 4, 3, 1, 4, 4, 7, 2, 3, 4, 6, 1, 2~
$ `Broj platformi`                    <dbl> 9, 5, 7, 6, 5, 2, 1, 8, 5, 7, 6, 6~
$ `Koje platforme koristi`            <chr> "Snapchat, WhatsApp, Facebook", "F~
$ `Koliko često prati vijesti`        <chr> "više puta dnevno", "nekoliko puta~
```

Već na prvi pogled vidimo nekoliko problema. Imena stupaca sadrže razmake, zagrade i dijakritičke znakove, što otežava rad u R-u. Stupci poput `Spol` imaju nekonzistentne vrijednosti. Stupac `TV (min/dan)` sadrži i brojeve i tekst ("ne gledam") i prazne ćelije, pa ga je R učitao kao tekst umjesto broja.

Pogledajmo prvih nekoliko redova detaljnije.

```
raw |>
  head(10)
```

```
# A tibble: 10 x 17
   `ID respondenta` Timestamp             Dob Spol   Grad     `Godina studija`
              <dbl> <dttm>              <dbl> <chr>  <chr>    <chr>
 1                1 2025-03-28 17:05:00    20 ženski Zagreb   2
 2                2 2025-04-20 21:11:00    27 Muški  Zadar    3
 3                3 2025-04-18 14:48:00    27 muški  Zagreb   1
 4                4 2025-03-28 12:54:00    18 female Split    1
 5                5 2025-03-21 18:06:00    25 Ženski Zagreb   2
 6                6 2025-04-14 20:26:00    26 M      Zagreb   2.
 7                7 2025-04-22 15:48:00    28 m      Zagreb   2
 8                8 2025-03-04 19:04:00    26 ženski Karlovac 2
 9                9 2025-03-12 12:17:00    22 female Split    2
10               10 2025-03-19 18:17:00    21 ž      Osijek   1
# i 11 more variables: `TV (min/dan)` <chr>, `Portali (min/dan)` <dbl>,
#   `Društvene mreže (min/dan)` <dbl>, `Radio (min/dan)` <dbl>,
#   `Podcast (min/dan)` <dbl>, `Povjerenje TV (1-10)` <dbl>,
#   `Povjerenje portali (1-10)` <dbl>,
#   `Povjerenje društvene mreže (1-10)` <dbl>, `Broj platformi` <dbl>,
#   `Koje platforme koristi` <chr>, `Koliko često prati vijesti` <chr>
```

Ovo je tipičan izgled sirovih podataka iz ankete. Prije bilo kakve analize, moramo napraviti čišćenje. Krenimo redom.

4.3 Korak nula: čišćenje imena stupaca

Prva stvar koju radimo sa svakim novim datasetom je standardizacija imena stupaca. Imena poput `TV (min/dan)` i `Povjerenje društvene mreže (1-10)` su problematična jer sadrže razmake, zagrade i specijalne znakove. Kad ih želite koristiti u kodu, morate ih stavljati u obrnute navodnike (poput `` `TV (min/dan)` ``). To je neugodno, nečitljivo i podložno greškama.

Paket janitor ima funkciju `clean_names()` koja automatski pretvara sva imena u snake_case format, uključujući mala slova, zamjenu razmaka podvlakama i uklanjanje specijalnih znakova.

```
raw <- raw |>
  clean_names()

names(raw)
```

```
 [1] "id_respondenta"                  "timestamp"
 [3] "dob"                             "spol"
 [5] "grad"                            "godina_studija"
 [7] "tv_min_dan"                      "portali_min_dan"
 [9] "drustvene_mreze_min_dan"         "radio_min_dan"
[11] "podcast_min_dan"                 "povjerenje_tv_1_10"
[13] "povjerenje_portali_1_10"         "povjerenje_drustvene_mreze_1_10"
[15] "broj_platformi"                  "koje_platforme_koristi"
[17] "koliko_cesto_prati_vijesti"
```

Usporedite ova imena s originalnima. Umjesto `Povjerenje društvene mreže (1-10)` sada imamo `povjerenje_drustvene_mreze_1_10`. Duže jest, ali potpuno funkcionalno u R kodu bez ikakvih navodnika ili zagrada. Ovo je mala investicija koja štedi mnogo frustracije.

> **Praktični savjet**
>
> Navikajte se da `clean_names()` bude prva stvar koju pozovete nakon `read_csv()`. Možete to čak staviti u isti pipeline: `raw <- read_csv("datoteka.csv") |> clean_names()`. Ovo je toliko standardna praksa da mnogi R korisnici to rade automatski za svaki dataset, čak i kad su imena stupaca već uredna. Bolje spriječiti nego liječiti.

Sad kad imamo čista imena, možemo krenuti s pravim poslom. Pogledajmo strukturu nakon čišćenja.

```
glimpse(raw)
```

```
Rows: 250
Columns: 17
$ id_respondenta                   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,~
$ timestamp                        <dttm> 2025-03-28 17:05:00, 2025-04-
20 21:11~
$ dob                              <dbl> 20, 27, 27, 18, 25, 26, 28, 26, 22, 21~
$ spol                             <chr> "ženski", "Muški", "muški", "female", ~
$ grad                             <chr> "Zagreb", "Zadar", "Zagreb", "Split", ~
$ godina_studija                   <chr> "2", "3", "1", "1", "2", "2.", "2", "2~
$ tv_min_dan                       <chr> "0", "0", "65", NA, NA, "91", "91", "0~
$ portali_min_dan                  <dbl> 40, 20, 0, 11, 32, 25, 81, 28, 37, 5, ~
$ drustvene_mreze_min_dan          <dbl> 59, 101, 177, 71, 161, 155, 114, 119, ~
$ radio_min_dan                    <dbl> 49, NA, 0, NA, 26, NA, 0, 0, 17, 0, 0,~
$ podcast_min_dan                  <dbl> 89, 0, 49, 0, 0, NA, NA, 31, 0, 19, 0,~
$ povjerenje_tv_1_10               <dbl> 2, 3, 4, 5, 5, 4, 3, 6, 6, 7, 2, 7, 7,~
$ povjerenje_portali_1_10          <dbl> 6, 5, 6, 6, 3, 7, 7, 1, 7, 5, 6, 5, 5,~
$ povjerenje_drustvene_mreze_1_10 <dbl> 4, 3, 1, 4, 4, 7, 2, 3, 4, 6, 1, 2, 3,~
$ broj_platformi                   <dbl> 9, 5, 7, 6, 5, 2, 1, 8, 5, 7, 6, 6, 2,~
$ koje_platforme_koristi           <chr> "Snapchat, WhatsApp, Facebook", "Faceb~
$ koliko_cesto_prati_vijesti       <chr> "više puta dnevno", "nekoliko puta tje~
```

4.4 filter() za odabir redova po uvjetu

Funkcija `filter()` je dplyr glagol za odabir redova koji zadovoljavaju jedan ili više uvjeta. Rezultat je tibble koji sadrži samo retke za koje su svi uvjeti TRUE. Redovi za koje je uvjet FALSE ili NA se odbacuju.

4.4.1 Osnovni uvjeti

```
# Samo ispitanici iz Zagreba
raw |>
  filter(grad == "Zagreb") |>
  nrow()
```

```
[1] 100
```

```
# Ispitanici mlađi od 21
raw |>
  filter(dob < 21) |>
  head(5)
```

```
# A tibble: 5 x 17
  id_respondenta timestamp             dob spol  grad  godina_studija tv_min_dan
           <dbl> <dttm>              <dbl> <chr> <chr> <chr>          <chr>
1              1 2025-03-28 17:05:00    20 žens~ Zagr~ 2              0
2              4 2025-03-28 12:54:00    18 fema~ Split 1              <NA>
3             13 2025-03-25 15:16:00    20 muški Zagr~ 1              0
4             14 2025-04-20 11:09:00    20 Žens~ Zagr~ 2              0
5             16 2025-04-07 18:06:00    19 muški Split 1              0
# i 10 more variables: portali_min_dan <dbl>, drustvene_mreze_min_dan <dbl>,
#   radio_min_dan <dbl>, podcast_min_dan <dbl>, povjerenje_tv_1_10 <dbl>,
#   povjerenje_portali_1_10 <dbl>, povjerenje_drustvene_mreze_1_10 <dbl>,
#   broj_platformi <dbl>, koje_platforme_koristi <chr>,
#   koliko_cesto_prati_vijesti <chr>
```

Svaki poziv `filter()` zapravo evaluira logički izraz za svaki redak. Za prvi primjer, R prolazi kroz svaki od 250 redova i provjerava je li vrijednost u stupcu `grad` jednaka "Zagreb". Retci za koje je odgovor TRUE ostaju, ostali nestaju.

4.4.2 Kombiniranje uvjeta

Snaga `filter()` dolazi do izražaja kad kombinirate više uvjeta. Unutar jednog `filter()` poziva, uvjeti odvojeni zarezom automatski se kombiniraju s I operatorom (`&`).

```
# Ispitanici iz Zagreba mlađi od 22
# Zarez između uvjeta je ekvivalentan &
raw |>
  filter(grad == "Zagreb", dob < 22) |>
  nrow()
```

```
[1] 41
```

```
# Isto kao:
raw |>
  filter(grad == "Zagreb" & dob < 22) |>
  nrow()
```

```
[1] 41
```

Oba pristupa daju identičan rezultat. Zarez je kraći za pisanje, `&` je eksplicitniji. Koristite što vam je čitljivije.

Za ILI uvjete, morate eksplicitno koristiti `|` operator.

```r
# Ispitanici iz Zagreba ILI Splita
raw |>
  filter(grad == "Zagreb" | grad == "Split") |>
  count(grad)
```

```
# A tibble: 2 x 2
  grad       n
  <chr>  <int>
1 Split     44
2 Zagreb   100
```

```r
# Elegantnije s %in%
raw |>
  filter(grad %in% c("Zagreb", "Split", "Rijeka")) |>
  count(grad, sort = TRUE)
```

```
# A tibble: 3 x 2
  grad       n
  <chr>  <int>
1 Zagreb   100
2 Split     44
3 Rijeka    18
```

Operator `%in%` smo upoznali prošli tjedan. U kontekstu `filter()` je izuzetno koristan jer zamjenjuje dugačke nizove ILI uvjeta jednim kompaktnim izrazom. Kad imate više od dvije kategorije, uvijek koristite `%in%`.

4.4.3 Filtriranje numeričkih raspona

```r
# Ispitanici koji koriste društvene mreže između 60 i 180 minuta dnevno
raw |>
  filter(drustvene_mreze_min_dan >= 60, drustvene_mreze_min_dan <= 180) |>
  nrow()
```

```
[1] 202
```

```r
# Alternativa s between()
raw |>
  filter(between(drustvene_mreze_min_dan, 60, 180)) |>
  nrow()
```

```
[1] 202
```

Funkcija `between(x, left, right)` je kratica za `x >= left & x <= right`. Oba pristupa daju isti rezultat, ali `between()` je čitljiviji kad filtrirate po rasponu.

4.4.4 Filtriranje teksta

Za tekstualne stupce, osim točnog podudaranja (`==`) i pripadnosti skupu (`%in%`), koristimo funkciju `str_detect()` iz paketa stringr (dio tidyverse) za pretraživanje po uzorku.

```
# Ispitanici čije platforme uključuju "Instagram" (bilo gdje u tekstu)
raw |>
  filter(str_detect(koje_platforme_koristi, "Instagram")) |>
  nrow()
```

```
[1] 39
```

```
# Ispitanici koji prate vijesti barem jednom dnevno
raw |>
  filter(str_detect(koliko_cesto_prati_vijesti, "dnevno")) |>
  count(koliko_cesto_prati_vijesti)
```

```
# A tibble: 2 x 2
  koliko_cesto_prati_vijesti     n
  <chr>                      <int>
1 jednom dnevno                 60
2 više puta dnevno              81
```

Funkcija `str_detect()` vraća TRUE ako tekstualni uzorak postoji bilo gdje u vrijednosti. Ovo je mnogo fleksibilnije od `==` jer ne zahtijeva točno podudaranje. Na primjer, `str_detect(x, "dnevno")` hvata i "više puta dnevno" i "jednom dnevno".

4.4.5 filter() i nedostajuće vrijednosti

Važno svojstvo `filter()` je da **automatski odbacuje retke s NA u uvjetu**. Ovo je uglavnom poželjno ponašanje, ali morate biti svjesni da se događa.

```
# Koliko redova imamo ukupno?
nrow(raw)
```

```
[1] 250
```

```r
# Koliko ima NA u stupcu radio_min_dan?
sum(is.na(raw$radio_min_dan))
```

```
[1] 32
```

```r
# filter s numeričkim uvjetom na stupcu s NA
raw |>
  filter(radio_min_dan > 0) |>
  nrow()
```

```
[1] 92
```

Rezultat ne uključuje retke s NA u stupcu `radio_min_dan`. Ako želite eksplicitno zadržati retke s NA, morate to navesti.

```r
# Zadrži retke gdje je radio > 0 ILI je NA
raw |>
  filter(radio_min_dan > 0 | is.na(radio_min_dan)) |>
  nrow()
```

```
[1] 124
```

```r
# Zadrži SAMO retke s NA
raw |>
  filter(is.na(radio_min_dan)) |>
  nrow()
```

```
[1] 32
```

```r
# Izbaci retke s NA (zadrži samo kompletne)
raw |>
  filter(!is.na(radio_min_dan)) |>
  nrow()
```

```
[1] 218
```

Kombinacija `filter(!is.na(stupac))` je način da zadržite samo retke s poznatim vrijednostima u tom stupcu. Alternativno, funkcija `drop_na()` iz paketa tidyr uklanja retke koji imaju NA u bilo kojem stupcu (ili u specificiranim stupcima).

```
# Ukloni retke s NA u specifičnom stupcu
raw |>
  drop_na(radio_min_dan) |>
  nrow()
```

```
[1] 218
```

```
# Ukloni retke s NA u BILO KOJEM stupcu (agresivno!)
raw |>
  drop_na() |>
  nrow()
```

```
[1] 149
```

Primijetite drastičnu razliku. Kad koristimo `drop_na()` bez argumenata, gubimo mnogo redova jer se uklanjaju svi retci koji imaju NA u ijednom stupcu. U praksi, `drop_na()` bez argumenata se rijetko koristi jer je previše agresivan. Bolje je ciljano raditi s NA u stupcima koji vas zapravo zanimaju.

> **!** Važna napomena
>
> Svaki put kad koristite `filter()` ili `drop_na()`, dokumentirajte koliko redova ste izgubili i zašto. Ako ste od 250 ispitanika zadržali samo 150, to je informacija koju morate navesti u metodološkom dijelu rada. Čitatelj mora znati na koliko se opažanja vaši rezultati temelje i zašto su neka isključena.

4.5 select() za odabir i preimenovanje stupaca

Dok `filter()` radi s redovima, `select()` radi sa stupcima. Koristi se za tri svrhe, a to su odabir stupaca koji vam trebaju, uklanjanje stupaca koji vam ne trebaju i preimenovanje stupaca.

4.5.1 Odabir po imenu

```
# Odabir specifičnih stupaca
raw |>
  select(id_respondenta, dob, spol, grad) |>
  head(5)
```

```
# A tibble: 5 x 4
  id_respondenta   dob spol   grad
           <dbl> <dbl> <chr>  <chr>
1              1    20 ženski Zagreb
2              2    27 Muški  Zadar
3              3    27 muški  Zagreb
4              4    18 female Split
5              5    25 Ženski Zagreb
```

```
# Odabir raspona stupaca (od do)
raw |>
  select(id_respondenta:godina_studija) |>
  head(5)
```

```
# A tibble: 5 x 6
  id_respondenta timestamp             dob spol   grad   godina_studija
           <dbl> <dttm>              <dbl> <chr>  <chr>  <chr>
1              1 2025-03-28 17:05:00    20 ženski Zagreb 2
2              2 2025-04-20 21:11:00    27 Muški  Zadar  3
3              3 2025-04-18 14:48:00    27 muški  Zagreb 1
4              4 2025-03-28 12:54:00    18 female Split  1
5              5 2025-03-21 18:06:00    25 Ženski Zagreb 2
```

Stupce navodite po imenu, bez navodnika. Operator `:` bira sve stupce između dva navedena, uključujući oba krajnja. Ovo je praktično kad su relevantni stupci jedan do drugoga u datasetu.

4.5.2 Uklanjanje stupaca

Minus ispred imena stupca znači "sve osim ovoga".

```
# Sve osim timestampa i ID-a
raw |>
  select(-timestamp, -id_respondenta) |>
  names()
```

```
 [1] "dob"                              "spol"
 [3] "grad"                             "godina_studija"
 [5] "tv_min_dan"                       "portali_min_dan"
 [7] "drustvene_mreze_min_dan"          "radio_min_dan"
 [9] "podcast_min_dan"                  "povjerenje_tv_1_10"
[11] "povjerenje_portali_1_10"          "povjerenje_drustvene_mreze_1_10"
[13] "broj_platformi"                   "koje_platforme_koristi"
[15] "koliko_cesto_prati_vijesti"
```

```
# Uklanjanje raspona
raw |>
  select(-(povjerenje_tv_1_10:povjerenje_drustvene_mreze_1_10)) |>
  names()
```

```
 [1] "id_respondenta"              "timestamp"
 [3] "dob"                         "spol"
 [5] "grad"                        "godina_studija"
 [7] "tv_min_dan"                  "portali_min_dan"
 [9] "drustvene_mreze_min_dan"     "radio_min_dan"
[11] "podcast_min_dan"             "broj_platformi"
[13] "koje_platforme_koristi"      "koliko_cesto_prati_vijesti"
```

4.5.3 Pomoćne funkcije za odabir

Kad imate mnogo stupaca, ručno nabrajanje postaje nepraktično. dplyr nudi pomoćne funkcije za pametni odabir.

```
# Stupci čije ime počinje s "povjerenje"
raw |>
  select(starts_with("povjerenje")) |>
  head(3)
```

```
# A tibble: 3 x 3
  povjerenje_tv_1_10 povjerenje_portali_1_10 povjerenje_drustvene_mreze_1_10
               <dbl>                   <dbl>                            <dbl>
1                  2                       6                                4
2                  3                       5                                3
3                  4                       6                                1
```

```
# Stupci čije ime završava s "dan"
raw |>
  select(ends_with("dan")) |>
  head(3)
```

```
# A tibble: 3 x 5
  tv_min_dan portali_min_dan drustvene_mreze_min_dan radio_min_dan
  <chr>                <dbl>                   <dbl>         <dbl>
1 0                       40                      59            49
2 0                       20                     101            NA
3 65                       0                     177             0
# i 1 more variable: podcast_min_dan <dbl>
```

```r
# Stupci čije ime sadrži "min"
raw |>
  select(contains("min")) |>
  head(3)
```

```
# A tibble: 3 x 5
  tv_min_dan portali_min_dan drustvene_mreze_min_dan radio_min_dan
  <chr>                <dbl>                   <dbl>         <dbl>
1 0                       40                      59            49
2 0                       20                     101            NA
3 65                       0                     177             0
# i 1 more variable: podcast_min_dan <dbl>
```

```r
# Samo numerički stupci
raw |>
  select(where(is.numeric)) |>
  head(3)
```

```
# A tibble: 3 x 10
  id_respondenta   dob portali_min_dan drustvene_mreze_min_dan radio_min_dan
           <dbl> <dbl>           <dbl>                   <dbl>         <dbl>
1              1    20              40                      59            49
2              2    27              20                     101            NA
3              3    27               0                     177             0
# i 5 more variables: podcast_min_dan <dbl>, povjerenje_tv_1_10 <dbl>,
#   povjerenje_portali_1_10 <dbl>, povjerenje_drustvene_mreze_1_10 <dbl>,
#   broj_platformi <dbl>
```

Funkcija `starts_with()` bira stupce čije ime počinje zadanim tekstom. `ends_with()` bira po završetku. `contains()` traži tekst bilo gdje u imenu. `where()` prima funkciju za provjeru tipa i bira stupce koji zadovoljavaju taj uvjet. Ove funkcije postaju neprocjenjive kad radite s datasetima koji imaju 50 ili 100 stupaca (što nije neuobičajeno u anketnim istraživanjima).

4.5.4 Preimenovanje stupaca

Unutar `select()` možete preimenovati stupac sintaksom `novo_ime = staro_ime`. Ili koristite zasebnu funkciju `rename()` koja preimenu stupce ali zadrži sve ostale.

```r
# Preimenovanje unutar select (odabire SAMO navedene stupce)
raw |>
  select(
    id = id_respondenta,
    dob,
```

```
    spol,
    sm_minuta = drustvene_mreze_min_dan
  ) |>
  head(3)
```

```
# A tibble: 3 x 4
     id   dob spol    sm_minuta
  <dbl> <dbl> <chr>       <dbl>
1     1    20 ženski         59
2     2    27 Muški         101
3     3    27 muški         177
```

```
# rename() mijenja imena ali zadržava sve stupce
raw |>
  rename(
    id = id_respondenta,
    sm_minuta = drustvene_mreze_min_dan
  ) |>
  names()
```

```
 [1] "id"                                "timestamp"
 [3] "dob"                               "spol"
 [5] "grad"                              "godina_studija"
 [7] "tv_min_dan"                        "portali_min_dan"
 [9] "sm_minuta"                         "radio_min_dan"
[11] "podcast_min_dan"                   "povjerenje_tv_1_10"
[13] "povjerenje_portali_1_10"           "povjerenje_drustvene_mreze_1_10"
[15] "broj_platformi"                    "koje_platforme_koristi"
[17] "koliko_cesto_prati_vijesti"
```

Razlika je važna. `select()` s preimenovanjem zadržava samo stupce koje ste naveli. `rename()` zadržava sve stupce i samo mijenja imena onih koje ste specificirali. U praksi, `rename()` je sigurniji izbor kad želite samo promijeniti ime jednog ili dva stupca bez gubitka ostalih.

4.5.5 Preuređivanje stupaca

Funkcija `relocate()` premješta stupce na drugu poziciju u datasetu.

```
# Premjesti grad na početak (odmah nakon ID-a)
raw |>
  relocate(grad, .after = id_respondenta) |>
  head(3)
```

```
# A tibble: 3 x 17
  id_respondenta grad  timestamp             dob spol  godina_studija tv_min_dan
           <dbl> <chr> <dttm>              <dbl> <chr> <chr>          <chr>
1              1 Zagr~ 2025-03-28 17:05:00    20 žens~ 2              0
2              2 Zadar 2025-04-20 21:11:00    27 Muški 3              0
3              3 Zagr~ 2025-04-18 14:48:00    27 muški 1              65
# i 10 more variables: portali_min_dan <dbl>, drustvene_mreze_min_dan <dbl>,
#   radio_min_dan <dbl>, podcast_min_dan <dbl>, povjerenje_tv_1_10 <dbl>,
#   povjerenje_portali_1_10 <dbl>, povjerenje_drustvene_mreze_1_10 <dbl>,
#   broj_platformi <dbl>, koje_platforme_koristi <chr>,
#   koliko_cesto_prati_vijesti <chr>
```

```
# Premjesti sve numeričke stupce na kraj
raw |>
  relocate(where(is.numeric), .after = last_col()) |>
  head(3)
```

```
# A tibble: 3 x 17
  timestamp           spol   grad   godina_studija tv_min_dan
  <dttm>              <chr>  <chr>  <chr>          <chr>
1 2025-03-28 17:05:00 ženski Zagreb 2              0
2 2025-04-20 21:11:00 Muški  Zadar  3              0
3 2025-04-18 14:48:00 muški  Zagreb 1              65
# i 12 more variables: koje_platforme_koristi <chr>,
#   koliko_cesto_prati_vijesti <chr>, id_respondenta <dbl>, dob <dbl>,
#   portali_min_dan <dbl>, drustvene_mreze_min_dan <dbl>, radio_min_dan <dbl>,
#   podcast_min_dan <dbl>, povjerenje_tv_1_10 <dbl>,
#   povjerenje_portali_1_10 <dbl>, povjerenje_drustvene_mreze_1_10 <dbl>,
#   broj_platformi <dbl>
```

`relocate()` ne dodaje niti uklanja stupce, samo ih premješta. Ovo je korisno za organizaciju dataseta kad želite da relevantni stupci budu jedni do drugih.

4.6 mutate() za kreiranje i transformaciju varijabli

Funkcija `mutate()` je najsvestraniji dplyr glagol. Služi za kreiranje novih stupaca na temelju postojećih, transformaciju postojećih stupaca i rekodiranje vrijednosti. Rezultat je tibble s istim brojem redova ali potencijalno novim ili izmijenjenim stupcima.

4.6.1 Kreiranje novih varijabli

```
# Ukupno dnevno korištenje medija (portal + društvene mreže)
raw |>
  mutate(
    ukupno_digital = portali_min_dan + drustvene_mreze_min_dan
  ) |>
  select(id_respondenta, portali_min_dan, drustvene_mreze_min_dan, ukupno_digital) |>
  head(8)
```

```
# A tibble: 8 x 4
  id_respondenta portali_min_dan drustvene_mreze_min_dan ukupno_digital
           <dbl>           <dbl>                   <dbl>          <dbl>
1              1              40                      59             99
2              2              20                     101            121
3              3               0                     177            177
4              4              11                      71             82
5              5              32                     161            193
6              6              25                     155            180
7              7              81                     114            195
8              8              28                     119            147
```

`mutate()` evaluira izraz na desnoj strani znaka jednakosti za svaki redak i rezultat pohranjuje u novi stupac nazvan imenom na lijevoj strani. Kao i kod vektoriziranih operacija, R automatski primjenjuje operaciju redak po redak.

Možete kreirati više stupaca u jednom `mutate()` pozivu, i kasniji stupci mogu koristiti ranije definirane.

```
raw |>
  mutate(
    ukupno_digital = portali_min_dan + drustvene_mreze_min_dan,
    ukupno_sati = ukupno_digital / 60,
    iznad_2_sata = ukupno_sati > 2
  ) |>
  select(id_respondenta, ukupno_digital, ukupno_sati, iznad_2_sata) |>
  head(8)
```

```
# A tibble: 8 x 4
  id_respondenta ukupno_digital ukupno_sati iznad_2_sata
           <dbl>          <dbl>       <dbl> <lgl>
1              1             99        1.65 FALSE
2              2            121        2.02 TRUE
3              3            177        2.95 TRUE
```

```
4              4              82        1.37 FALSE
5              5             193        3.22 TRUE
6              6             180        3    TRUE
7              7             195        3.25 TRUE
8              8             147        2.45 TRUE
```

Primijetite da smo u istom `mutate()` pozivu najprije izračunali `ukupno_digital`, zatim ga koristili za izračun `ukupno_sati`, a onda `ukupno_sati` za logički stupac `iznad_2_sata`. Ova mogućnost referiranja na upravo kreirane stupce čini `mutate()` izuzetno moćnim.

4.6.2 Transformacija postojećih stupaca

`mutate()` može i prepisati postojeći stupac.

```
# Zaokruži portal minute na desetice (prepisuje stupac)
raw |>
  mutate(
    portali_min_dan = round(portali_min_dan, -1)
  ) |>
  select(id_respondenta, portali_min_dan) |>
  head(8)
```

```
# A tibble: 8 x 2
  id_respondenta portali_min_dan
           <dbl>           <dbl>
1              1              40
2              2              20
3              3               0
4              4              10
5              5              30
6              6              20
7              7              80
8              8              30
```

Kad date mutate stupcu isto ime kao postojeći stupac, novi vrijednosti zamjenjuju stare. Ovo je korisno za čišćenje podataka (na primjer, pretvorbu teksta u mala slova), ali budite oprezni jer originalne vrijednosti nestaju. Dobra praksa je raditi transformacije na kopiji dataseta, ne na originalu.

4.6.3 Čišćenje stupca spol pomoću str_to_lower() i case_when()

Pogledajmo koliko je neuredan stupac `spol` u našim podacima.

```r
raw |>
  count(spol, sort = TRUE)
```

```
# A tibble: 12 x 2
   spol       n
   <chr>  <int>
 1 Muški     48
 2 Ženski    45
 3 muški     42
 4 ženski    31
 5 Ž         16
 6 male      14
 7 M         11
 8 Musko     11
 9 ž         10
10 m          9
11 Zensko     7
12 female     6
```

Imamo dvanaestak varijanti istih dviju kategorija. “Ženski”, “ženski”, “Ž”, “ž”, “Zensko”, “female” bi sve trebalo biti jedna kategorija. Ovo je klasičan problem u anketnim podacima i jedan od najčešćih razloga za čišćenje.

Funkcija `case_when()` je najfleksibilniji alat za rekodiranje. Radi kao niz IF-THEN pravila. Za svaki redak, R provjerava uvjete redom i dodjeljuje vrijednost prvog uvjeta koji je ispunjen.

```r
raw <- raw |>
  mutate(
    spol_clean = case_when(
      str_to_lower(spol) %in% c("ženski", "ž", "zensko", "female") ~ "ženski",
      str_to_lower(spol) %in% c("muški", "m", "musko", "male") ~ "muški",
      .default = "ostalo"
    )
  )

raw |>
  count(spol_clean)
```

```
# A tibble: 2 x 2
  spol_clean     n
  <chr>      <int>
1 muški        135
2 ženski       115
```

Raščlanimo ovaj kod. Funkcija `str_to_lower()` pretvara tekst u mala slova, čime eliminiramo razliku između "Ženski" i "ženski". Zatim `%in%` provjerava pripada li vrijednost jednom od navedenih oblika. Ako da, dodjeljuje standardizirani oblik. Argument `.default` hvata sve što ne odgovara nijednom uvjetu.

Rezultat je čist stupac `spol_clean` s tri konzistentne kategorije umjesto dvanaest neujednačenih varijanti.

> Praktični savjet
>
> Kad čistite tekstualne podatke, uvijek najprije pretvorite u mala slova pomoću `str_to_lower()`. Ovo odmah eliminira najčešći izvor nekonzistentnosti (razliku u kapitalizaciji) i smanjuje broj slučajeva koje morate pokriti u `case_when()`. Redoslijed je važan. Najprije trebate koristiti `str_to_lower()`, pa tek onda provjerite uvjete.

4.6.4 Rekodiranje numeričkih varijabli u kategorije

Čest zadatak u komunikologiji je pretvaranje kontinuirane varijable u kategorije. Na primjer, umjesto točne dobi, želimo dobne skupine.

```
raw <- raw |>
  mutate(
    dobna_skupina = case_when(
      dob < 20 ~ "18-19",
      dob < 22 ~ "20-21",
      dob < 24 ~ "22-23",
      dob >= 24 ~ "24+"
    )
  )

raw |>
  count(dobna_skupina)
```

```
# A tibble: 4 x 2
  dobna_skupina     n
  <chr>         <int>
1 18-19            73
2 20-21            61
3 22-23            47
4 24+              69
```

Redoslijed uvjeta u `case_when()` je bitan. R provjerava uvjete odozgo prema dolje i dodjeljuje vrijednost prvog ispunjenog uvjeta. Ako osoba ima 19 godina, prvi uvjet (`dob < 20`) je TRUE i dodjeljuje se "18-19". R ne provjerava preostale uvjete. Zato uvjete postavljamo od najspecifičnijeg prema najopćenitijem.

4.6.5 if_else(): binarno rekodiranje

Za jednostavne da/ne situacije, `if_else()` je kraći od `case_when()`.

```
raw <- raw |>
  mutate(
    visoko_koristenje_sm = if_else(drustvene_mreze_min_dan > 120, "visoko", "nisko/umjeren
    prati_vijesti_cesto = if_else(
      koliko_cesto_prati_vijesti %in% c("više puta dnevno", "jednom dnevno"),
      TRUE,
      FALSE
    )
  )

raw |>
  count(visoko_koristenje_sm)
```

```
# A tibble: 2 x 2
  visoko_koristenje_sm     n
  <chr>                <int>
1 nisko/umjereno         117
2 visoko                 133
```

Funkcija `if_else()` prima tri argumenta, uključujući uvjet, vrijednost za TRUE i vrijednost za FALSE. Prednost nad base R `ifelse()` je što `if_else()` strogo provjerava tipove i daje razumljivije greške kad nešto ne štima.

4.6.6 Čišćenje stupca godina studija

Pogledajmo još jedan neuredan stupac.

```
raw |>
  count(godina_studija, sort = TRUE)
```

```
# A tibble: 11 x 2
   godina_studija     n
   <chr>          <int>
 1 2                 54
 2 1                 52
 3 3                 38
 4 3.                22
 5 1.                19
 6 4                 19
```

```
 7 druga             14
 8 5                 12
 9 2.                 7
10 treća              7
11 prva               6
```

Imamo “1”, “1.”, “prva”, “2”, “2.”, “druga” i tako dalje. Sve to treba svesti na konzistentne brojeve.

```
raw <- raw |>
  mutate(
    godina_clean = case_when(
      str_to_lower(godina_studija) %in% c("1", "1.", "prva") ~ 1,
      str_to_lower(godina_studija) %in% c("2", "2.", "druga") ~ 2,
      str_to_lower(godina_studija) %in% c("3", "3.", "treća", "treca") ~ 3,
      str_to_lower(godina_studija) %in% c("4", "4.", "četvrta", "cetvrta") ~ 4,
      str_to_lower(godina_studija) %in% c("5", "5.", "peta") ~ 5,
      .default = NA_real_
    )
  )

raw |>
  count(godina_clean)
```

```
# A tibble: 5 x 2
  godina_clean     n
         <dbl> <int>
1            1    77
2            2    75
3            3    67
4            4    19
5            5    12
```

Ovaj put smo neprepoznate vrijednosti kodirali kao `NA_real_` (NA numeričkog tipa) umjesto tekstualne kategorije. To je ispravniji pristup kad očekujemo numerički rezultat. Ako neka vrijednost ne odgovara nijednom poznatom obrascu, bolje je eksplicitno reći “ne znam” (NA) nego nagađati.

4.6.7 Rad s problematičnim numeričkim stupcima

Prisjetimo se da je stupac `tv_min_dan` učitan kao tekst jer sadrži i brojeve i tekst (“ne gledam”) i prazne ćelije. Moramo ga pretvoriti u broj.

```r
# Pogledajmo problematične vrijednosti
raw |>
  count(tv_min_dan, sort = TRUE) |>
  head(15)
```

```
# A tibble: 15 x 2
   tv_min_dan     n
   <chr>      <int>
 1 0             85
 2 <NA>          41
 3 ne gledam      6
 4 71             4
 5 10             3
 6 112            3
 7 119            3
 8 26             3
 9 48             3
10 49             3
11 51             3
12 7              3
13 76             3
14 82             3
15 104            2
```

```r
# "ne gledam" tretiramo kao 0, prazne kao NA
raw <- raw |>
  mutate(
    tv_minuta = case_when(
      tv_min_dan == "ne gledam" ~ 0,
      tv_min_dan == "" ~ NA_real_,
      .default = as.numeric(tv_min_dan)
    )
  )

# Provjera
raw |>
  select(tv_min_dan, tv_minuta) |>
  filter(is.na(tv_minuta) | tv_minuta == 0) |>
  head(10)
```

```
# A tibble: 10 x 2
   tv_min_dan tv_minuta
   <chr>          <dbl>
 1 0                  0
```

```
 2 0                   0
 3 <NA>               NA
 4 <NA>               NA
 5 0                   0
 6 <NA>               NA
 7 0                   0
 8 0                   0
 9 0                   0
10 <NA>               NA
```

Sada imamo čist numerički stupac `tv_minuta` u kojem je "ne gledam" pretvoreno u 0 (jer osoba zaista ne gleda TV, dakle 0 minuta), a prazne ćelije su NA (jer ne znamo koliko ta osoba gleda TV).

Ova razlika između 0 i NA je konceptualno važna i vraća nas na tipove nedostajućih vrijednosti koje smo spominjali u tjednu 2. Nula znači "znamo odgovor, i odgovor je ništa". NA znači "ne znamo odgovor".

4.7 arrange(): sortiranje podataka

Funkcija `arrange()` sortira retke po vrijednostima jednog ili više stupaca. Po defaultu sortira uzlazno (od najmanjeg prema najvećem ili abecedno). Za silazno sortiranje koristimo `desc()`.

```
# Sortirano po dobi (najmlađi prvi)
raw |>
  select(id_respondenta, dob, grad, drustvene_mreze_min_dan) |>
  arrange(dob) |>
  head(8)
```

```
# A tibble: 8 x 4
  id_respondenta   dob grad    drustvene_mreze_min_dan
           <dbl> <dbl> <chr>                     <dbl>
1              4    18 Split                        71
2             23    18 Split                       145
3             25    18 Osijek                      184
4             31    18 Zagreb                       97
5             39    18 Šibenik                     154
6             44    18 Zagreb                      152
7             45    18 Pula                         86
8             57    18 Zagreb                      192
```

```r
# Sortirano po korištenju društvenih mreža (najviše korištenja prvo)
raw |>
  select(id_respondenta, dob, grad, drustvene_mreze_min_dan) |>
  arrange(desc(drustvene_mreze_min_dan)) |>
  head(8)
```

```
# A tibble: 8 x 4
  id_respondenta   dob grad           drustvene_mreze_min_dan
           <dbl> <dbl> <chr>                            <dbl>
1            149    22 Zadar                              296
2             21    21 Slavonski Brod                     271
3             42    25 Zagreb                             246
4             87    18 Zagreb                             246
5            141    23 Split                              222
6            181    22 Split                              217
7             16    19 Split                              214
8             79    22 Rijeka                             206
```

4.7.1 Sortiranje po više stupaca

Kad sortirate po više stupaca, prvi stupac ima prioritet. Unutar istih vrijednosti prvog stupca, koristi se drugi za razrješenje.

```r
# Sortiraj po gradu (abecedno), unutar grada po dobi (silazno)
raw |>
  select(id_respondenta, grad, dob, drustvene_mreze_min_dan) |>
  arrange(grad, desc(dob)) |>
  head(12)
```

```
# A tibble: 12 x 4
   id_respondenta grad        dob drustvene_mreze_min_dan
            <dbl> <chr>     <dbl>                   <dbl>
 1            241 Dubrovnik    27                     165
 2            198 Dubrovnik    25                      99
 3             34 Dubrovnik    23                     151
 4            222 Dubrovnik    21                     109
 5            243 Dubrovnik    20                     117
 6            129 Dubrovnik    19                     203
 7             75 Karlovac     28                     115
 8              8 Karlovac     26                     119
 9            229 Karlovac     24                     134
10             15 Karlovac     23                      35
11             73 Karlovac     21                      20
12             95 Karlovac     20                     147
```

Vidimo da su najprije svi ispitanici iz Dubrovnika (abecedno prvi), unutar kojih je najstariji na vrhu. Zatim dolazi Karlovac, pa dalje.

4.7.2 Gdje se NA pojavljuje pri sortiranju?

```
# NA vrijednosti uvijek idu na kraj, bez obzira na smjer
raw |>
  select(id_respondenta, radio_min_dan) |>
  arrange(radio_min_dan) |>
  tail(8)
```

```
# A tibble: 8 x 2
  id_respondenta radio_min_dan
           <dbl>         <dbl>
1            207            NA
2            211            NA
3            217            NA
4            218            NA
5            219            NA
6            220            NA
7            233            NA
8            244            NA
```

R stavlja NA na kraj sortiranog dataseta, neovisno o tome sortirate li uzlazno ili silazno. Ovo je korisno znati jer ćete ponekad htjeti vidjeti retke s nedostajućim vrijednostima, a oni su uvijek na dnu.

4.8 Kombiniranje glagola u pipeline

Prava snaga dplyr-a nije u pojedinačnim glagolima nego u njihovoj kombinaciji. Pipe operator (`|>`) omogućuje ulančavanje operacija u jednu koherentnu sekvencu koja čita dataseta od početka do kraja, korak po korak.

Pogledajmo realistični primjer. Želimo odgovoriti na pitanje te saznati koji gradovi imaju studente koji najviše koriste društvene mreže.

```
raw |>
  filter(dob >= 18, dob <= 25) |>
  select(grad, drustvene_mreze_min_dan) |>
  group_by(grad) |>
  summarise(
    n = n(),
    prosjek = round(mean(drustvene_mreze_min_dan), 1),
    .groups = "drop"
  ) |>
  filter(n >= 5) |>
  arrange(desc(prosjek))
```

```
# A tibble: 11 x 3
   grad               n prosjek
   <chr>          <int>   <dbl>
 1 Slavonski Brod     8   141.
 2 Zadar             11   137.
 3 Dubrovnik          5   136.
 4 Pula               9   134.
 5 Split             40   133.
 6 Šibenik            9   127.
 7 Osijek            18   124.
 8 Zagreb            79   116.
 9 Rijeka            18   112.
10 Varaždin           5    79.6
11 Karlovac           6    79.3
```

Čitamo odozgo prema dolje. Uzmi sirove podatke. Zadrži samo ispitanike između 18 i 25 godina. Odaberi samo stupce za grad i minute korištenja. Grupiraj po gradu. Izračunaj broj ispitanika i prosječno korištenje za svaki grad. Zadrži samo gradove s barem 5 ispitanika (da prosjeci budu smisleni). Sortiraj po prosječnom korištenju od najvišeg prema najnižem.

Svaki korak je sam za sebe jasan, a zajedno tvore kompletnu analizu. Ovo je radni obrazac koji ćete koristiti stotine puta.

Pogledajmo drugi primjer. Trebamo profil tipičnog korisnika koji često prati vijesti.

```
raw |>
  filter(prati_vijesti_cesto == TRUE) |>
  summarise(
    n = n(),
    prosjek_dob = round(mean(dob), 1),
    prosjek_sm_min = round(mean(drustvene_mreze_min_dan), 1),
    prosjek_portal_min = round(mean(portali_min_dan), 1),
    prosjek_trust_portal = round(mean(povjerenje_portali_1_10), 1),
```

```
    prosjek_trust_sm = round(mean(povjerenje_drustvene_mreze_1_10), 1)
  )
```

```
# A tibble: 1 x 6
      n prosjek_dob prosjek_sm_min prosjek_portal_min prosjek_trust_portal
  <int>       <dbl>          <dbl>              <dbl>                 <dbl>
1   141        21.9           122.               44.2                   4.9
# i 1 more variable: prosjek_trust_sm <dbl>
```

Ljudi koji prate vijesti barem jednom dnevno provode određen broj minuta na portalima i društvenim mrežama te imaju specifičan profil povjerenja u različite medije. Ova tablica daje bogat uvid u jednom pipeline.

4.8.1 Pipeline za čišćenje podataka

Uobičajena praksa je napisati jedan veliki pipeline za čišćenje koji pretvara sirove podatke u analizi spreman dataset. Evo kako bi to izgledalo za naše podatke.

```
clean <- raw |>
  # Preimenovanje stupaca za čitljivost
  rename(
    id = id_respondenta,
    sm_min = drustvene_mreze_min_dan,
    portal_min = portali_min_dan,
    trust_tv = povjerenje_tv_1_10,
    trust_portal = povjerenje_portali_1_10,
    trust_sm = povjerenje_drustvene_mreze_1_10,
    n_platformi = broj_platformi,
    vijesti_frekvencija = koliko_cesto_prati_vijesti
  ) |>
  # Korištenje već očišćenih stupaca
  select(
    id, dob, spol_clean, grad, godina_clean,
    tv_minuta, portal_min, sm_min, radio_min_dan, podcast_min_dan,
    trust_tv, trust_portal, trust_sm,
    n_platformi, vijesti_frekvencija,
    dobna_skupina, visoko_koristenje_sm, prati_vijesti_cesto
  ) |>
  # Završno preimenovanje čistih stupaca
  rename(
    spol = spol_clean,
    godina = godina_clean,
    radio_min = radio_min_dan,
```

```
    podcast_min = podcast_min_dan
  )

glimpse(clean)
```

```
Rows: 250
Columns: 18
$ id                   <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15~
$ dob                  <dbl> 20, 27, 27, 18, 25, 26, 28, 26, 22, 21, 22, 27, 2~
$ spol                 <chr> "ženski", "muški", "muški", "ženski", "ženski", "~
$ grad                 <chr> "Zagreb", "Zadar", "Zagreb", "Split", "Zagreb", "~
$ godina               <dbl> 2, 3, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 1~
$ tv_minuta            <dbl> 0, 0, 65, NA, NA, 91, 91, 0, 76, 66, NA, 109, 0, ~
$ portal_min           <dbl> 40, 20, 0, 11, 32, 25, 81, 28, 37, 5, 38, 44, 26,~
$ sm_min               <dbl> 59, 101, 177, 71, 161, 155, 114, 119, 56, 40, 129~
$ radio_min            <dbl> 49, NA, 0, NA, 26, NA, 0, 0, 17, 0, 0, 13, 0, 0, ~
$ podcast_min          <dbl> 89, 0, 49, 0, 0, NA, NA, 31, 0, 19, 0, 29, 22, 25~
$ trust_tv             <dbl> 2, 3, 4, 5, 5, 4, 3, 6, 6, 7, 2, 7, 7, 5, 4, 2, 5~
$ trust_portal         <dbl> 6, 5, 6, 6, 3, 7, 7, 1, 7, 5, 6, 5, 5, 5, 4, 6, 4~
$ trust_sm             <dbl> 4, 3, 1, 4, 4, 7, 2, 3, 4, 6, 1, 2, 3, 5, 2, 2, 4~
$ n_platformi          <dbl> 9, 5, 7, 6, 5, 2, 1, 8, 5, 7, 6, 6, 2, 7, 4, 3, 6~
$ vijesti_frekvencija  <chr> "više puta dnevno", "nekoliko puta tjedno", "više~
$ dobna_skupina        <chr> "20-21", "24+", "24+", "18-19", "24+", "24+", "24~
$ visoko_koristenje_sm <chr> "nisko/umjereno", "nisko/umjereno", "visoko", "ni~
$ prati_vijesti_cesto  <lgl> TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, FALSE, TRU~
```

Sada imamo čist dataset `clean` s razumljivim imenima stupaca, konzistentnim kodiranjem spola i godine studija, numeričkim stupcem za TV minute i binarnim varijablama za visoko korištenje i praćenje vijesti. Ovaj dataset je spreman za deskriptivnu statistiku i vizualizaciju.

> U svakom projektu analize podataka, trebali biste imati jasnu granicu između sirovih podataka (koje nikad ne mijenjate) i čistih podataka (koje kreirate skriptom iz sirovih). Skripta za čišćenje je vaš zapis svakog koraka, i svaki korišten uvjet mora biti dokumentiran komentarima.

4.9 Brzi pregled očišćenog dataseta

Provjerimo da je čišćenje uspjelo i iskoristimo priliku da povežemo sve naučene glagole.

```r
# Distribucija po spolu
clean |>
  count(spol)
```

```
# A tibble: 2 x 2
  spol       n
  <chr>  <int>
1 muški    135
2 ženski   115
```

```r
# Distribucija po gradu (top 5)
clean |>
  count(grad, sort = TRUE) |>
  head(5)
```

```
# A tibble: 5 x 2
  grad       n
  <chr>  <int>
1 Zagreb   100
2 Split     44
3 Osijek    23
4 Rijeka    18
5 Zadar     15
```

```r
# Prosječno korištenje medija po dobnim skupinama
clean |>
  group_by(dobna_skupina) |>
  summarise(
    n = n(),
    sm_prosjek = round(mean(sm_min), 1),
    portal_prosjek = round(mean(portal_min), 1),
    tv_prosjek = round(mean(tv_minuta, na.rm = TRUE), 1),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 5
  dobna_skupina     n sm_prosjek portal_prosjek tv_prosjek
  <chr>         <int>      <dbl>          <dbl>      <dbl>
1 18-19            73       122            42.3       33
2 20-21            61       120            43.6       34.4
3 22-23            47       120.           42.9       35.6
4 24+              69       122.           42.9       36.1
```

Tablica pokazuje jasne razlike. Studenti različitih dobnih skupina imaju različite obrasce korištenja medija. Najmlađi (18 do 19) provode najviše vremena na društvenim mrežama, dok je korištenje portala ravnomjernije raspoređeno. TV je konzistentno najniži oblik medijske konzumacije u svim skupinama, što je očekivano za studentsku populaciju.

```
# Povjerenje u medije - tko kome vjeruje
clean |>
  summarise(
    trust_tv_prosjek = round(mean(trust_tv), 1),
    trust_portal_prosjek = round(mean(trust_portal), 1),
    trust_sm_prosjek = round(mean(trust_sm), 1)
  )
```

```
# A tibble: 1 x 3
  trust_tv_prosjek trust_portal_prosjek trust_sm_prosjek
             <dbl>                <dbl>            <dbl>
1              4.5                    5              3.4
```

```
# Povjerenje po spolu
clean |>
  group_by(spol) |>
  summarise(
    n = n(),
    trust_sm = round(mean(trust_sm), 1),
    trust_portal = round(mean(trust_portal), 1),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 4
  spol       n trust_sm trust_portal
  <chr>  <int>    <dbl>        <dbl>
1 muški    135      3.3          4.9
2 ženski   115      3.5          5.1
```

Studenti u prosjeku najviše vjeruju portalima, zatim televiziji, a najmanje društvenim mrežama. Ovo je zanimljiv nalaz jer istovremeno na društvenim mrežama provode daleko najviše vremena. Provode li ljudi najviše vremena na medijima kojima najmanje vjeruju? Ili se povjerenje gradi korištenjem? Ovo su pitanja na koja ćemo se vraćati kad budemo radili korelacije i regresiju u kasnijim tjednima.

4.10 group_by() i summarise() za statistike po grupama

Kombinaciju `group_by()` i `summarise()` smo već koristili u dosadašnjim primjerima, ali zaslužuje detaljniju obradu jer je ovo daleko najvažniji obrazac u cijelom tidyverse radnom toku. Gotovo svaka analiza u komunikologiji uključuje usporedbu između grupa: razlikuju li se muškarci i žene po korištenju medija? Razlikuju li se gradovi po povjerenju? Razlikuju li se generacije po izvorima vijesti?

4.10.1 Osnovna logika

`group_by()` dijeli tibble na nevidljive podskupove prema vrijednostima jednog ili više stupaca. Sam po sebi ne proizvodi nikakav vidljiv rezultat. Ali kad nakon njega pozovete `summarise()`, izračun se ponavlja zasebno za svaki podskup.

```
# Prosječno korištenje društvenih mreža po spolu
clean |>
  group_by(spol) |>
  summarise(
    n = n(),
    prosjek_sm = round(mean(sm_min), 1),
    sd_sm = round(sd(sm_min), 1),
    medijan_sm = median(sm_min),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 5
  spol       n prosjek_sm sd_sm medijan_sm
  <chr>  <int>      <dbl> <dbl>      <dbl>
1 muški    135       117.  49.2        115
2 ženski   115       126.  47.3        129
```

Argument `.groups = "drop"` na kraju govori R-u da ukloni grupiranje nakon izračuna. Bez njega, rezultirajući tibble bi ostao grupiran, što može uzrokovati neočekivano ponašanje u kasnijim operacijama. Dobra praksa je uvijek eksplicitno navesti `.groups = "drop"`.

4.10.2 Grupiranje po više varijabli

```
# Korištenje po spolu i dobnoj skupini
clean |>
  group_by(spol, dobna_skupina) |>
  summarise(
    n = n(),
```

```
    prosjek_sm = round(mean(sm_min), 1),
    prosjek_portal = round(mean(portal_min), 1),
    .groups = "drop"
  ) |>
  filter(spol != "ostalo") |>
  arrange(dobna_skupina, spol)
```

```
# A tibble: 8 x 5
  spol   dobna_skupina     n prosjek_sm prosjek_portal
  <chr>  <chr>         <int>      <dbl>          <dbl>
1 muški  18-19            39       122.           40.2
2 ženski 18-19            34       122.           44.7
3 muški  20-21            36       114.           41.4
4 ženski 20-21            25       128.           46.8
5 muški  22-23            24       110.           44.1
6 ženski 22-23            23       130.           41.7
7 muški  24+              36       119.           42.9
8 ženski 24+              33       126.           43
```

Kad grupirate po više varijabli, `summarise()` izračunava statistike za svaku kombinaciju tih varijabli. S dva spola i četiri dobne skupine dobivate osam grupa (ili manje, ako neke kombinacije nemaju opažanja). Filtrirali smo kategoriju "ostalo" jer s malim brojem opažanja statistike nisu pouzdane.

4.10.3 count() kao kratica

Funkcija `count()` je zapravo kratica za `group_by() |> summarise(n = n()) |> ungroup()`. Koristite je kad vam treba samo prebrojavanje.

```
# Ovo:
clean |>
  count(grad, sort = TRUE) |>
  head(5)
```

```
# A tibble: 5 x 2
  grad       n
  <chr>  <int>
1 Zagreb   100
2 Split     44
3 Osijek    23
4 Rijeka    18
5 Zadar     15
```

```r
# Je ekvivalentno ovome:
clean |>
  group_by(grad) |>
  summarise(n = n(), .groups = "drop") |>
  arrange(desc(n)) |>
  head(5)
```

```
# A tibble: 5 x 2
  grad       n
  <chr>  <int>
1 Zagreb   100
2 Split     44
3 Osijek    23
4 Rijeka    18
5 Zadar     15
```

Obje verzije daju identičan rezultat, ali `count()` štedi tri reda koda. Za jednostavno prebrojavanje uvijek koristite `count()`.

4.10.4 group_by() s mutate()

Manje poznata ali izuzetno korisna kombinacija je `group_by()` s `mutate()`. Umjesto da sažima podatke u jednu vrijednost po grupi (kao `summarise()`), `mutate()` dodaje novu kolonu svakom retku, ali izračun se radi unutar grupe.

```r
# Z-score korištenja društvenih mreža UNUTAR svake dobne skupine
clean |>
  group_by(dobna_skupina) |>
  mutate(
    sm_prosjek_grupe = mean(sm_min),
    sm_z = round((sm_min - mean(sm_min)) / sd(sm_min), 2)
  ) |>
  ungroup() |>
  select(id, dob, dobna_skupina, sm_min, sm_prosjek_grupe, sm_z) |>
  head(10)
```

```
# A tibble: 10 x 6
      id   dob dobna_skupina sm_min sm_prosjek_grupe  sm_z
   <dbl> <dbl> <chr>          <dbl>            <dbl> <dbl>
 1     1    20 20-21             59             120. -1.24
 2     2    27 24+              101             122. -0.53
 3     3    27 24+              177             122.  1.33
 4     4    18 18-19             71             122. -0.98
```

```
 5     5    25 24+             161            122.  0.94
 6     6    26 24+             155            122.  0.79
 7     7    28 24+             114            122. -0.21
 8     8    26 24+             119            122. -0.09
 9     9    22 22-23            56            120. -1.2
10    10    21 20-21            40            120. -1.62
```

Primijetite `ungroup()` na kraju. Kad koristite `group_by()` s `mutate()`, grupiranje ostaje aktivno nakon mutatea (za razliku od `summarise()` koji ga automatski smanjuje). Uvijek dodajte `ungroup()` kad završite s grupiranim operacijama da izbjegnete iznenađenja.

4.11 across() za istu operaciju na više stupaca

Do sada smo u `summarise()` ručno pisali svaku statistiku za svaki stupac. Kad imate pet ili deset numeričkih stupaca, to postaje zamorno. Funkcija `across()` rješava ovaj problem jer primjenjuje istu funkciju (ili više funkcija) na više stupaca odjednom.

```
# Prosjek za sve stupce koji sadrže "trust" u imenu
clean |>
  summarise(
    across(starts_with("trust"), ~round(mean(.x), 1))
  )
```

```
# A tibble: 1 x 3
  trust_tv trust_portal trust_sm
     <dbl>        <dbl>    <dbl>
1      4.5            5      3.4
```

Sintaksa `~round(mean(.x), 1)` koristi lambda notaciju (tilda formula). `.x` je placeholder za svaki stupac na koji se `across()` primjenjuje. Ovo se čita kao "za svaki stupac koji počinje s trust, izračunaj zaokruženi prosjek".

4.11.1 Više funkcija odjednom

```
# Prosjek i SD za stupce s minutama
clean |>
  summarise(
    across(
      c(sm_min, portal_min, tv_minuta),
```

```
      list(
        prosjek = ~round(mean(.x, na.rm = TRUE), 1),
        sd = ~round(sd(.x, na.rm = TRUE), 1)
      ),
      .names = "{.col}_{.fn}"
    )
  )
```

```
# A tibble: 1 x 6
  sm_min_prosjek sm_min_sd portal_min_prosjek portal_min_sd tv_minuta_prosjek
           <dbl>     <dbl>              <dbl>         <dbl>              <dbl>
1           121.      48.5               42.9          22.3               34.7
# i 1 more variable: tv_minuta_sd <dbl>
```

Kad proslijedite imenovu listu funkcija, `across()` kreira zasebne stupce za svaku kombinaciju stupca i funkcije. Argument `.names = "{.col}_{.fn}"` kontrolira kako se novi stupci imenuju — `{.col}` je ime izvornog stupca, `{.fn}` je ime funkcije iz liste.

4.11.2 across() s group_by()

Kombinacija `across()` i `group_by()` omogućuje izračun više statistika za više stupaca po grupama, u jednom kompaktnom pozivu.

```
clean |>
  group_by(dobna_skupina) |>
  summarise(
    n = n(),
    across(
      c(sm_min, portal_min, trust_sm, trust_portal),
      list(M = ~round(mean(.x, na.rm = TRUE), 1)),
      .names = "{.col}_{.fn}"
    ),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 6
  dobna_skupina     n sm_min_M portal_min_M trust_sm_M trust_portal_M
  <chr>         <int>    <dbl>        <dbl>      <dbl>          <dbl>
1 18-19            73      122         42.3        3.5            4.9
2 20-21            61      120         43.6        3.4            4.9
3 22-23            47      120.        42.9        3.5            5
4 24+              69      122.        42.9        3.3            5.2
```

U jednom pozivu dobivamo prosjeke četiri varijable za svaku dobnu skupinu, plus broj opažanja. Ovo je obrazac koji ćete koristiti za izradu tablica deskriptivnih statistika u akademskim radovima.

4.11.3 across() s mutate()

`across()` radi i unutar `mutate()` za transformaciju više stupaca odjednom.

```
# Centriranje svih trust varijabli (oduzimanje prosjeka)
clean |>
  mutate(
    across(
      starts_with("trust"),
      ~.x - mean(.x),
      .names = "{.col}_cent"
    )
  ) |>
  select(id, starts_with("trust")) |>
  head(5)
```

```
# A tibble: 5 x 7
     id trust_tv trust_portal trust_sm trust_tv_cent trust_portal_cent
  <dbl>    <dbl>        <dbl>    <dbl>         <dbl>              <dbl>
1     1        2            6        4        -2.48             0.992
2     2        3            5        3        -1.48            -0.00800
3     3        4            6        1        -0.476            0.992
4     4        5            6        4         0.524            0.992
5     5        5            3        4         0.524           -2.01
# i 1 more variable: trust_sm_cent <dbl>
```

Ovo je osobito korisno za standardizaciju ili transformaciju velikog broja varijabli u jednom koraku.

> **Praktični savjet**
>
> Funkcija `across()` čini kod kompaktnijim ali i teže čitljivim za početnike. Ako vam lambda notacija (`~mean(.x)`) izgleda zbunjujuće, nema ništa loše u tome da najprije pišete svaku statistiku ručno, a `across()` počnete koristiti kad se osjećate ugodno s osnovnim glagolima. Cilj je čitljivost, ne kratkoća.

4.12 pivot_longer() i pivot_wider() za preoblikovanje podataka

Ponekad podaci dolaze u obliku koji nije pogodan za analizu ili vizualizaciju i moramo ih preoblikovati. Dva najčešća slučaja su pretvaranje širokog formata u dugački i obrnuto.

4.12.1 Tidy data — princip urednih podataka

Wickham (2014) definira **uredne podatke** (tidy data) kao tablicu u kojoj svaki redak predstavlja jedno opažanje, svaki stupac jednu varijablu i svaka ćelija jednu vrijednost. Zvuči jednostavno, ali mnogi dataseti ne zadovoljavaju ovaj princip.

Pogledajmo konkretan primjer. Naši podaci o povjerenju imaju tri zasebna stupca: `trust_tv`, `trust_portal`, `trust_sm`. Za neke analize (posebno vizualizaciju), bilo bi korisnije imati jedan stupac `medij` s vrijednostima "TV", "portal" i "društvene mreže" i jedan stupac `povjerenje` s numeričkom ocjenom.

4.12.2 pivot_longer() za pretvaranje od širokog prema dugačkom formatu

```
trust_long <- clean |>
  select(id, dob, spol, dobna_skupina, trust_tv, trust_portal, trust_sm) |>
  pivot_longer(
    cols = starts_with("trust"),
    names_to = "medij",
    values_to = "povjerenje",
    names_prefix = "trust_"
  )

trust_long |>
  head(12)
```

```
# A tibble: 12 x 6
      id   dob spol   dobna_skupina medij  povjerenje
   <dbl> <dbl> <chr>  <chr>         <chr>       <dbl>
 1     1    20 ženski 20-21         tv              2
 2     1    20 ženski 20-21         portal          6
 3     1    20 ženski 20-21         sm              4
 4     2    27 muški  24+           tv              3
 5     2    27 muški  24+           portal          5
 6     2    27 muški  24+           sm              3
 7     3    27 muški  24+           tv              4
 8     3    27 muški  24+           portal          6
 9     3    27 muški  24+           sm              1
10     4    18 ženski 18-19         tv              5
```

```
11     4    18 ženski 18-19         portal          6
12     4    18 ženski 18-19         sm              4
```

Funkcija `pivot_longer()` pretvara stupce u redove. Ključni argumenti funkcije su sljedeći.

`cols` specificira koje stupce pretvaramo (ovdje sve koji počinju s "trust").

`names_to` je ime novog stupca koji će sadržavati imena izvornih stupaca.

`values_to` je ime novog stupca koji će sadržavati vrijednosti iz izvornih stupaca.

`names_prefix` uklanja zajednički prefiks iz imena (bez njega bismo imali "trust_tv" umjesto "tv").

Iz originalnih 250 redova (jedan po ispitaniku) dobili smo 750 redova (tri po ispitaniku, jedan za svaki tip medija). Ovo je dugački format.

Sad možemo lako izračunati prosječno povjerenje po tipu medija.

```
trust_long |>
  group_by(medij) |>
  summarise(
    prosjek = round(mean(povjerenje), 2),
    sd = round(sd(povjerenje), 2),
    .groups = "drop"
  )
```

```
# A tibble: 3 x 3
  medij  prosjek    sd
  <chr>    <dbl> <dbl>
1 portal    5.01  1.73
2 sm        3.41  1.72
3 tv        4.48  1.99
```

Ili po tipu medija i dobnoj skupini.

```
trust_long |>
  group_by(dobna_skupina, medij) |>
  summarise(
    prosjek = round(mean(povjerenje), 1),
    .groups = "drop"
  ) |>
  arrange(dobna_skupina, medij)
```

```
# A tibble: 12 x 3
   dobna_skupina medij  prosjek
   <chr>         <chr>    <dbl>
 1 18-19         portal     4.9
 2 18-19         sm         3.5
 3 18-19         tv         4.6
 4 20-21         portal     4.9
 5 20-21         sm         3.4
 6 20-21         tv         4.8
 7 22-23         portal     5
 8 22-23         sm         3.5
 9 22-23         tv         4.1
10 24+           portal     5.2
11 24+           sm         3.3
12 24+           tv         4.4
```

Ova tablica jasno pokazuje obrasce koje bi bilo teško vidjeti u širokom formatu. Dugački format je posebno koristan za vizualizaciju jer ggplot2 (koji ćemo učiti sljedeći tjedan) radi prirodno s dugačkim podacima.

4.12.3 pivot_wider() za pretvaranje od dugačkog prema širokom formatu

Obrnuta operacija, `pivot_wider()`, pretvara redove u stupce. Korisna je kad želite tablicu u obliku koji je čitljiv za ljude (široki format), a ne za računalo (dugački format).

```
# Prosječno povjerenje po dobnoj skupini i mediju, u širokom formatu
trust_long |>
  group_by(dobna_skupina, medij) |>
  summarise(prosjek = round(mean(povjerenje), 1), .groups = "drop") |>
  pivot_wider(
    names_from = medij,
    values_from = prosjek
  )
```

```
# A tibble: 4 x 4
  dobna_skupina portal    sm    tv
  <chr>          <dbl> <dbl> <dbl>
1 18-19            4.9   3.5   4.6
2 20-21            4.9   3.4   4.8
3 22-23            5     3.5   4.1
4 24+              5.2   3.3   4.4
```

Rezultat je tablica s jednim retkom po dobnoj skupini i jednim stupcem po tipu medija. Ovo je format koji biste stavili u izvještaj ili akademski rad jer je lako čitljiv.

Argumenti su zrcalni u odnosu na `pivot_longer()`:

`names_from` je stupac čije će vrijednosti postati imena novih stupaca.

`values_from` je stupac čije će vrijednosti popuniti nove stupce.

4.12.4 Primjer s minutama korištenja

Isti obrazac primjenjujemo i na podatke o korištenju medija.

```
# Pretvorba minuta korištenja u dugački format
koristenje_long <- clean |>
  select(id, dob, spol, dobna_skupina, tv_minuta, portal_min, sm_min) |>
  pivot_longer(
    cols = c(tv_minuta, portal_min, sm_min),
    names_to = "medij",
    values_to = "minuta"
  ) |>
  mutate(
    medij = case_when(
      medij == "tv_minuta" ~ "TV",
      medij == "portal_min" ~ "Portali",
      medij == "sm_min" ~ "Društvene mreže"
    )
  )

# Prosječno korištenje po tipu medija
koristenje_long |>
  group_by(medij) |>
  summarise(
    prosjek = round(mean(minuta, na.rm = TRUE), 1),
    medijan = median(minuta, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(desc(prosjek))
```

```
# A tibble: 3 x 3
  medij           prosjek medijan
  <chr>             <dbl>   <dbl>
1 Društvene mreže   121.      124
2 Portali            42.9      43
3 TV                 34.7      17
```

Društvene mreže dominiraju s velikim razmakom. TV je daleko na dnu. Ovi podaci su za studentsku populaciju, pa ne iznenađuju, ali upravo ovakve tablice čine temelj svakog izvještaja o medijskim navikama.

> **!** Važna napomena
>
> Zapamtite sljedeće pravilo. `pivot_longer()` koristite kad želite pretvoriti podatke iz oblika čitljivog za ljude u oblik pogodan za analizu i vizualizaciju. `pivot_wider()` koristite kad želite rezultate pretvoriti natrag u oblik čitljiv za ljude (za tablice u izvještajima). Tipičan radni tok uključuje sljedeće korake: učitajte podatke, pretvorite u dugački format, analizirajte i pretvorite rezultate u široki format za prezentaciju.

4.13 Spajanje tablica pomoću left_join()

U stvarnim istraživanjima, podaci rijetko dolaze u jednoj tablici. Možda imate jednu tablicu s demografskim podacima ispitanika i drugu s rezultatima eksperimenta. Ili jednu tablicu s podacima o člancima i drugu s podacima o komentarima. Da biste ih analizirali zajedno, morate ih spojiti.

Kreirajmo pomoćnu tablicu za demonstraciju.

```
# Tablica s informacijama o gradovima
gradovi_info <- tibble(
  grad = c("Zagreb", "Split", "Rijeka", "Osijek", "Zadar", "Dubrovnik",
           "Slavonski Brod", "Pula", "Karlovac", "Varaždin", "Šibenik", "Sisak"),
  regija = c("Središnja", "Dalmacija", "Primorje", "Slavonija", "Dalmacija", "Dalmacija",
             "Slavonija", "Istra", "Središnja", "Sjever", "Dalmacija", "Središnja"),
  populacija_tis = c(770, 160, 108, 96, 70, 41, 50, 52, 46, 41, 34, 33)
)

gradovi_info
```

```
# A tibble: 12 x 3
   grad           regija    populacija_tis
   <chr>          <chr>              <dbl>
 1 Zagreb         Središnja            770
 2 Split          Dalmacija            160
 3 Rijeka         Primorje             108
 4 Osijek         Slavonija             96
 5 Zadar          Dalmacija             70
 6 Dubrovnik      Dalmacija             41
 7 Slavonski Brod Slavonija             50
 8 Pula           Istra                 52
 9 Karlovac       Središnja             46
10 Varaždin       Sjever                41
```

```
11 Šibenik        Dalmacija             34
12 Sisak          Središnja             33
```

Sada možemo spojiti ovu tablicu s našim čistim podacima da svakom ispitaniku dodamo informaciju o regiji i populaciji grada.

```
clean_s_regijom <- clean |>
  left_join(gradovi_info, by = "grad")

clean_s_regijom |>
  select(id, grad, regija, populacija_tis, sm_min) |>
  head(10)
```

```
# A tibble: 10 x 5
      id grad     regija    populacija_tis sm_min
   <dbl> <chr>    <chr>              <dbl>  <dbl>
 1     1 Zagreb   Središnja            770     59
 2     2 Zadar    Dalmacija             70    101
 3     3 Zagreb   Središnja            770    177
 4     4 Split    Dalmacija            160     71
 5     5 Zagreb   Središnja            770    161
 6     6 Zagreb   Središnja            770    155
 7     7 Zagreb   Središnja            770    114
 8     8 Karlovac Središnja             46    119
 9     9 Split    Dalmacija            160     56
10    10 Osijek   Slavonija             96     40
```

Funkcija `left_join()` spaja dvije tablice po zajedničkom stupcu (ovdje `grad`). Za svaki redak u lijevoj tablici (`clean`), traži podudarajući redak u desnoj tablici (`gradovi_info`) i dodaje stupce iz desne tablice. Ako nema podudaranja (na primjer, grad koji nije u tablici `gradovi_info`), dobivamo NA.

Argument `by = "grad"` specificira koji stupac koristimo za podudaranje. Ako se stupac za spajanje različito zove u dvjema tablicama, koristimo sintaksu `by = c("ime_lijevo" = "ime_desno")`.

Sad možemo analizirati podatke po regijama.

```
clean_s_regijom |>
  group_by(regija) |>
  summarise(
    n = n(),
    prosjek_sm = round(mean(sm_min), 1),
    prosjek_trust_sm = round(mean(trust_sm), 1),
    .groups = "drop"
```

```
  ) |>
  filter(!is.na(regija)) |>
  arrange(desc(prosjek_sm))
```

```
# A tibble: 6 x 4
  regija        n prosjek_sm prosjek_trust_sm
  <chr>     <int>      <dbl>            <dbl>
1 Istra        10       140                3
2 Dalmacija    74       132.               3.2
3 Slavonija    31       127.               3.5
4 Središnja   110       115.               3.6
5 Primorje     18       112.               3.4
6 Sjever        7        79                3.3
```

Ovo je moć spajanja tablica — informacija koja je bila u zasebnoj tablici sada je dio naše analize i omogućuje grupiranje po varijabli koja nije postojala u izvornim podacima.

4.13.1 Vrste joinova

`left_join()` je daleko najčešći join i jedini koji ćete trebati u većini situacija. Ali vrijedi znati da postoje i drugi.

`left_join(a, b)` zadržava sve retke iz a, dodaje podudarajuće iz b. Ako nema podudaranja, NA.

`inner_join(a, b)` zadržava samo retke koji postoje u obje tablice.

`full_join(a, b)` zadržava sve retke iz obje tablice, s NA gdje nema podudaranja.

`anti_join(a, b)` zadržava retke iz a koji nemaju podudaranje u b. Korisno za pronalaženje nepodudarajućih zapisa.

```
# Ima li ispitanika iz gradova koji nisu u našoj tablici?
clean |>
  anti_join(gradovi_info, by = "grad") |>
  count(grad)
```

```
# A tibble: 0 x 2
# i 2 variables: grad <chr>, n <int>
```

`anti_join()` je odličan dijagnostički alat jer otkriva retke koji se ne mogu spojiti. U ovom slučaju vidimo gradove koji postoje u anketi ali ne u našoj tablici gradova.

4.14 Stringovi — osnove rada s tekstom

U komunikologiji se često radi s tekstualnim podacima, kao što su imena platformi, naslovi članaka i otvoreni odgovori u anketama. Paket stringr (dio tidyverse) pruža konzistentan skup funkcija za rad s tekstom.

Već smo koristili `str_to_lower()` i `str_detect()`. Pogledajmo još nekoliko korisnih funkcija.

```
# Stupac s platformama je slobodan tekst s više unosa
raw |>
  select(id_respondenta, koje_platforme_koristi) |>
  head(5)
```

```
# A tibble: 5 x 2
  id_respondenta koje_platforme_koristi
           <dbl> <chr>
1              1 Snapchat, WhatsApp, Facebook
2              2 Facebook, YouTube
3              3 WhatsApp
4              4 Pinterest, LinkedIn
5              5 Viber, Snapchat, Reddit
```

```
# Koliko ispitanika koristi Instagram (bilo gdje u tekstu)?
raw |>
  mutate(koristi_instagram = str_detect(koje_platforme_koristi, "Instagram")) |>
  count(koristi_instagram)
```

```
# A tibble: 2 x 2
  koristi_instagram     n
  <lgl>             <int>
1 FALSE               211
2 TRUE                 39
```

4.14.1 Brojanje i izdvajanje uzoraka

```
# Koliko platformi svaki ispitanik navodi (brojeći zareze + 1)?
raw |>
  mutate(
    navedeno_platformi = str_count(koje_platforme_koristi, ",") + 1
  ) |>
  select(id_respondenta, koje_platforme_koristi, navedeno_platformi) |>
  head(8)
```

```
# A tibble: 8 x 3
  id_respondenta koje_platforme_koristi                navedeno_platformi
           <dbl> <chr>                                              <dbl>
1              1 Snapchat, WhatsApp, Facebook                           3
2              2 Facebook, YouTube                                      2
3              3 WhatsApp                                               1
4              4 Pinterest, LinkedIn                                    2
5              5 Viber, Snapchat, Reddit                                3
6              6 Pinterest, YouTube                                     2
7              7 WhatsApp                                               1
8              8 Reddit, Pinterest, WhatsApp, Twitter/X                 4
```

Funkcija `str_count()` broji koliko se puta uzorak pojavljuje u tekstu. Budući da su platforme odvojene zarezima, broj zareza plus jedan daje broj navedenih platformi. Ovo je primjer kako tekstualne operacije pomažu u izvlačenju numeričkih informacija iz nestrukturiranih podataka.

4.14.2 Zamjena i čišćenje teksta

```
# Zamjena "Twitter/X" s "X" za konzistentnost
raw |>
  mutate(
    platforme_clean = str_replace(koje_platforme_koristi, "Twitter/X", "X")
  ) |>
  filter(str_detect(koje_platforme_koristi, "Twitter")) |>
  select(koje_platforme_koristi, platforme_clean) |>
  head(5)
```

```
# A tibble: 5 x 2
  koje_platforme_koristi                  platforme_clean
  <chr>                                   <chr>
1 Reddit, Pinterest, WhatsApp, Twitter/X  Reddit, Pinterest, WhatsApp, X
2 YouTube, Twitter/X, Pinterest, WhatsApp YouTube, X, Pinterest, WhatsApp
3 Pinterest, Instagram, Telegram, Twitter/X Pinterest, Instagram, Telegram, X
4 Reddit, Twitter/X, Pinterest            Reddit, X, Pinterest
5 LinkedIn, Telegram, Snapchat, Twitter/X LinkedIn, Telegram, Snapchat, X
```

Funkcija `str_replace()` zamjenjuje prvo pojavljivanje uzorka, a `str_replace_all()` zamjenjuje sva pojavljivanja. Funkcija `str_trim()` uklanja razmake s početka i kraja teksta, što je korisno kad ispitanici slučajno unesu razmak.

4.15 Sve zajedno — kompletna analiza od sirovih do gotovih podataka

Zaokružimo ovo predavanje tako da napišemo kompletnu analizu koja prolazi kroz sve faze, a to su učitavanje, čišćenje, transformacija, analiza i prezentacija rezultata. Ovo je obrazac koji ćete ponavljati u svakom projektu.

```
# FAZA 1: Učitavanje i čišćenje
anketa_clean <- read_csv("../resources/datasets/media_habits_raw.csv") |>
  clean_names() |>
  mutate(
    # Čišćenje spola
    spol = case_when(
      str_to_lower(spol) %in% c("ženski", "ž", "zensko", "female") ~ "ženski",
      str_to_lower(spol) %in% c("muški", "m", "musko", "male") ~ "muški",
      .default = NA_character_
    ),
    # Čišćenje godine studija
    godina = case_when(
      str_to_lower(godina_studija) %in% c("1", "1.", "prva") ~ 1L,
      str_to_lower(godina_studija) %in% c("2", "2.", "druga") ~ 2L,
      str_to_lower(godina_studija) %in% c("3", "3.", "treća", "treca") ~ 3L,
      str_to_lower(godina_studija) %in% c("4", "4.") ~ 4L,
      str_to_lower(godina_studija) %in% c("5", "5.") ~ 5L,
      .default = NA_integer_
    ),
    # TV minute: text -> broj
    tv_min = case_when(
      tv_min_dan == "ne gledam" ~ 0,
      tv_min_dan == "" ~ NA_real_,
      .default = as.numeric(tv_min_dan)
    ),
    # Dobna skupina
    dobna_sk = case_when(
      dob < 20 ~ "18-19",
      dob < 22 ~ "20-21",
      dob < 24 ~ "22-23",
      dob >= 24 ~ "24+"
    )
  ) |>
  # Odabir i preimenovanje konačnih stupaca
  select(
    id = id_respondenta,
    dob, spol, grad, godina,
    tv_min,
```

```
    portal_min = portali_min_dan,
    sm_min = drustvene_mreze_min_dan,
    trust_tv = povjerenje_tv_1_10,
    trust_portal = povjerenje_portali_1_10,
    trust_sm = povjerenje_drustvene_mreze_1_10,
    vijesti = koliko_cesto_prati_vijesti,
    dobna_sk
  )

glimpse(anketa_clean)
```

```
Rows: 250
Columns: 13
$ id           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
$ dob          <dbl> 20, 27, 27, 18, 25, 26, 28, 26, 22, 21, 22, 27, 20, 20, 2~
$ spol         <chr> "ženski", "muški", "muški", "ženski", "ženski", "muški", ~
$ grad         <chr> "Zagreb", "Zadar", "Zagreb", "Split", "Zagreb", "Zagreb",~
$ godina       <int> 2, 3, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 1, 1, 2, ~
$ tv_min       <dbl> 0, 0, 65, NA, NA, 91, 91, 0, 76, 66, NA, 109, 0, 0, 56, 0~
$ portal_min   <dbl> 40, 20, 0, 11, 32, 25, 81, 28, 37, 5, 38, 44, 26, 65, 35,~
$ sm_min       <dbl> 59, 101, 177, 71, 161, 155, 114, 119, 56, 40, 129, 95, 72~
$ trust_tv     <dbl> 2, 3, 4, 5, 5, 4, 3, 6, 6, 7, 2, 7, 7, 5, 4, 2, 5, 3, 4, ~
$ trust_portal <dbl> 6, 5, 6, 6, 3, 7, 7, 1, 7, 5, 6, 5, 5, 5, 4, 6, 4, 5, 6, ~
$ trust_sm     <dbl> 4, 3, 1, 4, 4, 7, 2, 3, 4, 6, 1, 2, 3, 5, 2, 2, 4, 3, 6, ~
$ vijesti      <chr> "više puta dnevno", "nekoliko puta tjedno", "više puta dn~
$ dobna_sk     <chr> "20-21", "24+", "24+", "18-19", "24+", "24+", "24+", "24+~
```

```
# FAZA 2: Provjera
# Koliko NA po stupcu?
anketa_clean |>
  summarise(across(everything(), ~sum(is.na(.)))) |>
  pivot_longer(everything(), names_to = "stupac", values_to = "n_NA") |>
  filter(n_NA > 0)
```

```
# A tibble: 1 x 2
  stupac  n_NA
  <chr>  <int>
1 tv_min    41
```

```
# FAZA 3: Deskriptivna analiza
# Korištenje medija po dobnoj skupini
anketa_clean |>
  group_by(dobna_sk) |>
  summarise(
```

```
    n = n(),
    across(
      c(sm_min, portal_min, tv_min),
      list(M = ~round(mean(.x, na.rm = TRUE), 1)),
      .names = "{.col}_{.fn}"
    ),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 5
  dobna_sk     n sm_min_M portal_min_M tv_min_M
  <chr>    <int>    <dbl>        <dbl>    <dbl>
1 18-19       73      122          42.3     33
2 20-21       61      120          43.6     34.4
3 22-23       47      120.         42.9     35.6
4 24+         69      122.         42.9     36.1
```

```
# Povjerenje po tipu medija (dugački format za lakšu usporedbu)
anketa_clean |>
  pivot_longer(
    cols = starts_with("trust"),
    names_to = "medij",
    values_to = "povjerenje",
    names_prefix = "trust_"
  ) |>
  mutate(
    medij = case_when(
      medij == "tv" ~ "Televizija",
      medij == "portal" ~ "Web portali",
      medij == "sm" ~ "Društvene mreže"
    )
  ) |>
  group_by(medij) |>
  summarise(
    M = round(mean(povjerenje), 2),
    SD = round(sd(povjerenje), 2),
    Med = median(povjerenje),
    .groups = "drop"
  ) |>
  arrange(desc(M))
```

```
# A tibble: 3 x 4
  medij               M    SD   Med
  <chr>           <dbl> <dbl> <dbl>
```

```
1 Web portali      5.01  1.73     5
2 Televizija       4.48  1.99     4
3 Društvene mreže  3.41  1.72     3
```

```r
# Tko prati vijesti, a tko ne?
anketa_clean |>
  mutate(
    cesto_prati = vijesti %in% c("više puta dnevno", "jednom dnevno")
  ) |>
  group_by(cesto_prati) |>
  summarise(
    n = n(),
    prosjek_dob = round(mean(dob), 1),
    prosjek_sm = round(mean(sm_min), 1),
    prosjek_trust_portal = round(mean(trust_portal), 1),
    prosjek_trust_sm = round(mean(trust_sm), 1),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 6
  cesto_prati     n prosjek_dob prosjek_sm prosjek_trust_portal prosjek_trust_sm
  <lgl>       <int>       <dbl>      <dbl>                <dbl>            <dbl>
1 FALSE         109        21.6       120.                  5.2              3.3
2 TRUE          141        21.9       122.                  4.9              3.5
```

Ova kompletna analiza, od učitavanja sirovih podataka do gotovih tablica, stane u manje od 80 redova koda. Svaki korak je dokumentiran, ponovljiv i čitljiv. Ako sutra dobijete ažurirane podatke s još 100 ispitanika, pokrenete istu skriptu i dobijete ažurirane rezultate. To je suština ponovljive analize.

> Cilj čišćenja podataka nije savršenstvo. Cilj je da od neurednog, nekonzistentnog i djelomično nepoznatog skupa podataka stvorite skup koji je dovoljno uredan i dokumentiran da možete s povjerenjem raditi statističku analizu i transparentno komunicirati svaki izbor koji ste napravili.

! Ključni zaključci

1. Čišćenje i priprema podataka oduzima 80% vremena u bilo kojoj analizi. Stvarni podaci su gotovo uvijek neuredni i zahtijevaju sistematično čišćenje prije ikakve statističke analize.

2. Funkcija `clean_names()` iz paketa janitor standardizira imena stupaca u

snake_case format. Koristite je odmah nakon učitavanja svakog dataseta.

3. `filter()` odabire retke po uvjetu. Automatski odbacuje retke s NA. Koristite `%in%` za provjeru pripadnosti skupu, `between()` za raspone i `str_detect()` za pretraživanje teksta.

4. `select()` odabire, uklanja i preuređuje stupce. Pomoćne funkcije `starts_with()`, `ends_with()`, `contains()` i `where()` omogućuju pametan odabir. `rename()` mijenja imena bez gubitka stupaca.

5. `mutate()` kreira nove stupce i transformira postojeće. `case_when()` je alat za složeno rekodiranje. `if_else()` za binarno. Razlika između 0 i NA je konceptualno važna.

6. `arrange()` sortira retke. `desc()` za silazni smjer. NA uvijek na kraj.

7. `group_by() |> summarise()` je temeljni obrazac za izračun statistika po grupama. Uvijek dodajte `.groups = "drop"`. `count()` je kratica za prebrojavanje.

8. `across()` primjenjuje istu operaciju na više stupaca odjednom. Kombinira se i sa `summarise()` i s `mutate()`.

9. `pivot_longer()` pretvara stupce u redove (široki u dugački format). `pivot_wider()` pretvara redove u stupce. Dugački format je pogodan za analizu i vizualizaciju, široki za prezentaciju.

10. `left_join()` spaja dvije tablice po zajedničkom stupcu. Koristite ga kad trebate kombinirati podatke iz više izvora.

11. stringr funkcije (`str_detect()`, `str_to_lower()`, `str_replace()`, `str_count()`) omogućuju rad s tekstualnim podacima. Bitne za čišćenje anketnih podataka.

12. Svaka analiza ima jasne faze, a to su učitavanje, čišćenje, provjera, analiza i prezentacija. Dokumentirajte svaki korak i svaki izbor (osobito koliko redova gubite filtiranjem).

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan bavimo se **deskriptivnom statistikom**: mjerama centralne tendencije (prosjek, medijan, mod), mjerama varijabilnosti (varijanca, standardna devijacija, IQR), korelacijama i standardnim rezultatima (z-scores). Sve ćemo raditi kroz `summarise()` i `group_by()` koje ste upravo naučili.
Za pripremu napravite sljedeće:

1. Ponovite kompletni pipeline čišćenja iz ovog predavanja. Pokrenite ga red po red

i provjerite da razumijete svaki korak.
2. Pokušajte odgovoriti na pitanje: razlikuje li se prosječno povjerenje u društvene mreže između ispitanika koji prate vijesti često i onih koji ne prate? (Hint: `mutate()` za kreiranje binarne varijable, `group_by() |> summarise()` za usporedbu.)
3. Pretvorite podatke o korištenju (TV, portali, društvene mreže) u dugački format pomoću `pivot_longer()` i izračunajte prosječno korištenje po tipu medija i spolu.
4. Pročitajte poglavlje 5 iz knjige Navarro (Learning Statistics with R) o deskriptivnoj statistici. Fokusirajte se na koncepte, ne na R kod (jer knjiga koristi base R).

4.16 Dodatno čitanje

Obavezno

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 4, 5 i 6. Besplatno dostupno na r4ds.hadley.nz. Poglavlje 4 pokriva transformaciju podataka, poglavlje 5 organizaciju radnog toka, poglavlje 6 preoblikovanje podataka s pivot funkcijama.

Navarro, D. (2018). *Learning Statistics with R*, Chapters 4 i 7. Besplatno dostupno na learningstatisticswithr.com. Pokrivaju sličan teren u base R sintaksi.

Preporučeno

Wickham, H. (2014). Tidy Data. *Journal of Statistical Software*, 59(10). Besplatno dostupno na vita.had.co.nz/papers/tidy-data.pdf. Klasičan rad koji definira princip urednih podataka.

Firke, S. (2023). *janitor: Simple Tools for Examining and Cleaning Dirty Data*. Dokumentacija paketa na sfirke.github.io/janitor. Osim `clean_names()`, paket sadrži i `tabyl()` za brze tablice frekvencija i `remove_empty()` za uklanjanje praznih redova i stupaca.

4.17 Pojmovnik

Pojam	Objašnjenje
dplyr	R paket iz tidyverse ekosustava za manipulaciju podacima. Sadrži glagole `filter()`, `select()`, `mutate()`, `summarise()`, `arrange()`, `group_by()` i druge.
`filter()`	dplyr glagol za odabir redova koji zadovoljavaju logički uvjet. Automatski odbacuje retke s NA u uvjetu.
`select()`	dplyr glagol za odabir, uklanjanje i preuređivanje stupaca. Podržava pomoćne funkcije poput `starts_with()`, `contains()` i `where()`.
`mutate()`	dplyr glagol za kreiranje novih stupaca ili transformaciju postojećih. Novi stupci mogu referirati na upravo kreirane.
`arrange()`	dplyr glagol za sortiranje redova po vrijednostima stupaca. `desc()` za silazno sortiranje.
`summarise()`	dplyr glagol za sažimanje podataka u jednu vrijednost po grupi (ili za cijeli dataset). Koristi se s agregatnim funkcijama poput `mean()`, `sd()`, `n()`.
`group_by()`	dplyr glagol koji dijeli podatke u grupe po jednoj ili više varijabli. Sve naknadne operacije se izvršavaju zasebno za svaku grupu.
`ungroup()`	dplyr glagol koji uklanja grupiranje. Koristite nakon `group_by() |> mutate()` da izbjegnete neočekivano ponašanje.
`count()`	Kratica za `group_by() |> summarise(n = n()) |> ungroup()`. Prebrojava opažanja po kategorijama.
`across()`	Funkcija za primjenu iste operacije na više stupaca odjednom. Radi unutar `summarise()` i `mutate()`.
`rename()`	dplyr glagol za preimenovanje stupaca bez gubitka ostalih. Sintaksa: `rename(novo = staro)`.
`relocate()`	dplyr glagol za premještanje stupaca na drugu poziciju u datasetu.
`case_when()`	Funkcija za složeno rekodiranje s više uvjeta. Svaki uvjet ima oblik `uvjet ~ vrijednost`. Provjerava uvjete redom.

Pojam	Objašnjenje
`if_else()`	Funkcija za binarno rekodiranje. Prima uvjet, vrijednost za TRUE i vrijednost za FALSE.
`between()`	Pomoćna funkcija: provjera je li vrijednost unutar raspona. Kratica za `x >= left & x <= right`.
`str_detect()`	stringr funkcija: provjerava sadrži li tekst zadani uzorak. Vraća TRUE/FALSE.
`str_to_lower()`	stringr funkcija: pretvara tekst u mala slova. Korisna za standardizaciju.
`str_replace()`	stringr funkcija: zamjenjuje prvo pojavljivanje uzorka u tekstu. `str_replace_all()` zamjenjuje sva.
`str_count()`	stringr funkcija: broji pojavljivanja uzorka u tekstu.
`pivot_longer()`	tidyr funkcija za pretvaranje stupaca u redove (široki u dugački format). Ključni argumenti: `cols`, `names_to`, `values_to`.
`pivot_wider()`	tidyr funkcija za pretvaranje redova u stupce (dugački u široki format). Ključni argumenti: `names_from`, `values_from`.
`left_join()`	dplyr funkcija za spajanje dviju tablica po zajedničkom stupcu. Zadržava sve retke iz lijeve tablice.
`inner_join()`	Spajanje koje zadržava samo retke koji postoje u obje tablice.
`anti_join()`	Spajanje koje zadržava retke iz lijeve tablice koji nemaju podudaranje u desnoj. Dijagnostički alat.
`clean_names()`	janitor funkcija: pretvara imena stupaca u snake_case. Uklanja razmake, zagrade i specijalne znakove.
`drop_na()`	tidyr funkcija za uklanjanje redova s NA. Može se primijeniti na cijeli dataset ili specifične stupce.
Tidy data (uredni podaci)	Princip organizacije podataka: svaki redak je opažanje, svaki stupac varijabla, svaka ćelija vrijednost.
Široki format	Organizacija podataka u kojoj su različita mjerenja iste varijable raspoređena u zasebne stupce. Čitljiv za ljude.

Pojam	Objašnjenje
Dugački format	Organizacija podataka u kojoj su različita mjerenja u zasebnim redovima s identifikacijskim stupcem. Pogodan za analizu i vizualizaciju.
Pipeline	Niz operacija spojenih pipe operatorom (`|>`) koji transformira podatke korak po korak.
Sirovi podaci (raw data)	Podaci u izvornom obliku, prije čišćenja. Ne bi ih trebalo mijenjati izravno.
Čisti podaci (clean data)	Podaci nakon standardizacije, rekodiranja i provjere. Spremni za analizu.
Rekodiranje	Pretvaranje vrijednosti varijable u standardizirani oblik (npr. svih varijanti spola u "ženski"/"muški").

5 Tjedan 4: Programiranje u R-u

Funkcije, uvjeti i ponovljive analize

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti zašto su vlastite funkcije korisne za izbjegavanje ponavljanja koda i smanjenje grešaka.
2. Napisati vlastitu R funkciju s argumentima i podrazumijevanim (default) vrijednostima.
3. Koristiti uvjetne naredbe (`if`, `else`, `if_else()`, `case_when()`) za kontrolu toka programa.
4. Koristiti `for` petlje za ponavljanje operacija nad skupom elemenata.
5. Koristiti `map()` funkcije iz paketa purrr kao modernu alternativu petljama.
6. Primijeniti principe DRY (Don't Repeat Yourself) na pisanje analitičkih skripti.
7. Organizirati analitičku skriptu s jasnom strukturom, uključujući učitavanje, čišćenje, analizu, vizualizaciju i izvoz.
8. Prepoznati kada je pisanje vlastite funkcije isplativije od kopiranja koda.

5.1 Koliko programiranja treba komunikolog?

Ovo pitanje zaslužuje iskren odgovor. Ne trebate postati softverski inženjer. Ne trebate znati pisati web aplikacije, baze podataka ili algoritme strojnog učenja. Ali trebate znati dovoljno programiranja da vaše analize budu **ponovljive**, **prilagodljive** i **manje podložne greškama**.

Zamislite sljedeću situaciju. Radite analizu medijskih navika za klijenta. Napravili ste čišćenje podataka, deskriptivnu statistiku, osam grafova i izvještaj. Klijent je zadovoljan, ali tjedan dana kasnije kaže: "Dobili smo još 200 odgovora na anketu, možete li ponoviti analizu s novim podacima?" Ako ste sve radili ručno u Excelu, to znači ponoviti svaki korak od nule. Ako ste napisali R skriptu, to znači promijeniti jednu liniju koda (putanju do nove datoteke) i pokrenuti skriptu. Pet sekundi umjesto pet sati.

Programiranje u kontekstu analize podataka nije apstraktno akademsko znanje. To je praktična vještina koja vas čini bržima, preciznijima i profesionalnijima. U ovom tjednu naučit ćemo tri temeljne programerske koncepte (funkcije, uvjetne naredbe i iteraciju) i pokazati kako ih koristiti u kontekstu koji je relevantan za komunikologe.

5.2 Naši podaci: newsletter kampanje

Ovaj tjedan koristimo dataset o 50 newsletter kampanja jednog informativnog portala. Za svaku kampanju imamo podatke o tipu, stilu naslova, vremenu slanja, broju pretplatnika, open rateu (postotak otvaranja), click rateu (postotak klikova) i drugim metrikama.

```
nl <- read_csv("../resources/datasets/newsletter_campaigns.csv")
glimpse(nl)
```

```
Rows: 50
Columns: 13
$ campaign_id      <chr> "NL-001", "NL-002", "NL-003", "NL-004", "NL-
005", "NL~
$ campaign_type    <chr> "special_report", "weekly_digest", "special_report", ~
$ subject_style    <chr> "personalizirani", "hitno", "upitni", "informativni",~
$ day_sent         <chr> "petak", "petak", "utorak", "ponedjeljak", "utorak", ~
$ send_hour        <dbl> 8, 11, 8, 9, 20, 16, 13, 6, 11, 19, 11, 19, 9, 15, 18~
$ subscribers      <dbl> 11770, 14266, 10652, 23113, 9847, 9150, 23450, 12798,~
$ open_rate        <dbl> 0.2410, 0.2696, 0.3023, 0.2134, 0.2887, 0.1921, 0.288~
$ click_rate       <dbl> 0.0858, 0.0065, 0.0309, 0.0656, 0.0273, 0.0651, 0.015~
$ unsubscribe_rate <dbl> 0.00447, 0.00283, 0.00519, 0.00000, 0.00082, 0.00391,~
$ word_count       <dbl> 499, 378, 437, 545, 559, 146, 428, 210, 309, 519, 376~
$ n_links          <dbl> 2, 9, 4, 1, 3, 1, 10, 3, 8, 4, 9, 3, 8, 7, 5, 7, 1, 5~
$ has_image        <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, FALS~
$ revenue          <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 4659.53, 0.~
```

```
nl |>
  count(campaign_type, sort = TRUE)
```

```
# A tibble: 5 x 2
  campaign_type      n
  <chr>          <int>
1 special_report    17
2 weekly_digest     12
3 sponsored          8
4 breaking_news      7
5 event_promo        6
```

Ovo je manji dataset od prethodnih tjedana, ali upravo to ga čini pogodnim za učenje programiranja. S 50 redova možete vidjeti svaki korak i razumjeti što vaš kod radi.

5.3 Zašto funkcije? Problem kopiranja koda

Krenimo od konkretnog problema. Recimo da za svaki tip kampanje želite izračunati sažetak s prosjekom, medijanom i standardnom devijacijom open ratea. Jedan pristup je kopiranje koda.

```
# Sažetak za weekly digest
nl |>
  filter(campaign_type == "weekly_digest") |>
  summarise(
    n = n(),
    or_prosjek = round(mean(open_rate), 3),
    or_medijan = round(median(open_rate), 3),
    or_sd = round(sd(open_rate), 3)
  )
```

```
# A tibble: 1 x 4
      n or_prosjek or_medijan or_sd
  <int>      <dbl>      <dbl> <dbl>
1    12      0.289       0.29 0.066
```

```
# Sažetak za breaking news (kopiran kod s jednom promjenom)
nl |>
  filter(campaign_type == "breaking_news") |>
  summarise(
    n = n(),
    or_prosjek = round(mean(open_rate), 3),
    or_medijan = round(median(open_rate), 3),
    or_sd = round(sd(open_rate), 3)
  )
```

```
# A tibble: 1 x 4
      n or_prosjek or_medijan or_sd
  <int>      <dbl>      <dbl> <dbl>
1     7      0.198      0.195 0.055
```

Ovo radi, ali ima tri problema. Prvo, ako želite promijeniti izračun (recimo dodati trimmed mean), morate to napraviti na svakom mjestu gdje ste kopirali kod. Drugo, svako kopiranje je prilika za grešku. Možda zaboravite promijeniti ime kampanje na jednom mjestu. Treće, kad imate pet ili deset tipova kampanja, kod postaje nepregledano dugačak.

Naravno, za ovaj specifični problem znamo elegantno rješenje s `group_by()`.

```
nl |>
  group_by(campaign_type) |>
  summarise(
    n = n(),
    or_prosjek = round(mean(open_rate), 3),
    or_medijan = round(median(open_rate), 3),
    or_sd = round(sd(open_rate), 3),
    .groups = "drop"
  )
```

```
# A tibble: 5 x 5
  campaign_type      n or_prosjek or_medijan or_sd
  <chr>          <int>      <dbl>      <dbl> <dbl>
1 breaking_news      7      0.198      0.195 0.055
2 event_promo        6      0.25       0.23  0.086
3 special_report    17      0.259      0.255 0.067
4 sponsored          8      0.248      0.246 0.043
5 weekly_digest     12      0.289      0.29  0.066
```

Ali `group_by()` ne rješava svaki problem. Kad trebate ponoviti složeniju analizu (koja uključuje čišćenje, više izračuna, graf i tablicu) za različite podskupove podataka, vlastite funkcije postaju nezamjenjive.

5.4 Pisanje vlastite funkcije

Funkcija u R-u je objekt koji prima ulazne podatke (argumente), izvršava niz operacija i vraća rezultat. Već koristite funkcije svaki dan; na primjer, `mean()`, `filter()` i `ggplot()` su sve funkcije koje je netko napisao. Sad ćete naučiti pisati vlastite.

5.4.1 Anatomija funkcije

```r
# Funkcija koja pretvara postotke u razlomke
postotak_u_razlomak <- function(postotak) {
  postotak / 100
}

postotak_u_razlomak(25)
```

```
[1] 0.25
```

```r
postotak_u_razlomak(73.5)
```

```
[1] 0.735
```

Raščlanimo sintaksu. `postotak_u_razlomak` je ime funkcije (kao ime bilo kojeg objekta, dodjeljujemo ga s `<-`). Ključna riječ `function()` govori R-u da kreiramo funkciju. Unutar zagrada su argumenti (ulazni podatci). Unutar vitičastih zagrada `{}` je tijelo funkcije (operacije koje se izvršavaju). Zadnji izraz u tijelu je povratna vrijednost (ono što funkcija vraća).

5.4.2 Funkcija s više argumenata

```r
# Funkcija za izračun engagement ratea
engagement_rate <- function(clicks, opens) {
  rate <- clicks / opens
  round(rate, 4)
}

engagement_rate(clicks = 150, opens = 1200)
```

```
[1] 0.125
```

```r
engagement_rate(clicks = 80, opens = 500)
```

```
[1] 0.16
```

Funkcija prima dva argumenta i vraća zaokruženi omjer. Kad pozivate funkciju, argumente možete navesti po imenu (`clicks = 150`) ili po poziciji. Po imenu je sigurnije jer nije bitno kojim redoslijedom ih navedete.

5.4.3 Default vrijednosti argumenata

Ponekad želite da argument ima podrazumijevanu (default) vrijednost koju korisnik može promijeniti ako želi.

```
# Funkcija za sažetak numeričke varijable
sazetak_varijable <- function(x, decimale = 2) {
  tibble(
    n = length(x),
    n_NA = sum(is.na(x)),
    prosjek = round(mean(x, na.rm = TRUE), decimale),
    medijan = round(median(x, na.rm = TRUE), decimale),
    sd = round(sd(x, na.rm = TRUE), decimale),
    min = round(min(x, na.rm = TRUE), decimale),
    max = round(max(x, na.rm = TRUE), decimale)
  )
}

# Korištenje s default decimala (2)
sazetak_varijable(nl$open_rate)
```

```
# A tibble: 1 x 7
      n  n_NA prosjek medijan    sd   min   max
  <int> <int>   <dbl>   <dbl> <dbl> <dbl> <dbl>
1    50     0    0.25    0.25  0.07  0.11  0.41
```

```
# Korištenje s 4 decimale
sazetak_varijable(nl$open_rate, decimale = 4)
```

```
# A tibble: 1 x 7
      n  n_NA prosjek medijan     sd   min   max
  <int> <int>   <dbl>   <dbl>  <dbl> <dbl> <dbl>
1    50     0   0.255   0.252 0.0676 0.115 0.408
```

Argument `decimale = 2` ima default vrijednost 2. Ako ga ne navedete pri pozivu, koristi se 2. Ako ga eksplicitno navedete, koristi se vaša vrijednost. Ovo čini funkciju fleksibilnom bez opterećivanja korisnika nepotrebnim odlukama.

5.4.4 Funkcija koja radi s tibbleom

Funkcije koje primaju cijeli tibble i koriste dplyr glagole unutar sebe su izuzetno korisne u praksi.

```r
# Funkcija za sažetak kampanje po tipu
sazetak_kampanje <- function(data, tip) {
  data |>
    filter(campaign_type == tip) |>
    summarise(
      tip = tip,
      n = n(),
      prosjek_or = round(mean(open_rate), 3),
      prosjek_ctr = round(mean(click_rate), 4),
      ukupni_doseg = sum(subscribers),
      .groups = "drop"
    )
}

sazetak_kampanje(nl, "weekly_digest")
```

```
# A tibble: 1 x 5
  tip               n prosjek_or prosjek_ctr ukupni_doseg
  <chr>         <int>      <dbl>       <dbl>        <dbl>
1 weekly_digest    12      0.289       0.029       217303
```

```r
sazetak_kampanje(nl, "breaking_news")
```

```
# A tibble: 1 x 5
  tip               n prosjek_or prosjek_ctr ukupni_doseg
  <chr>         <int>      <dbl>       <dbl>        <dbl>
1 breaking_news     7      0.198      0.0382       120123
```

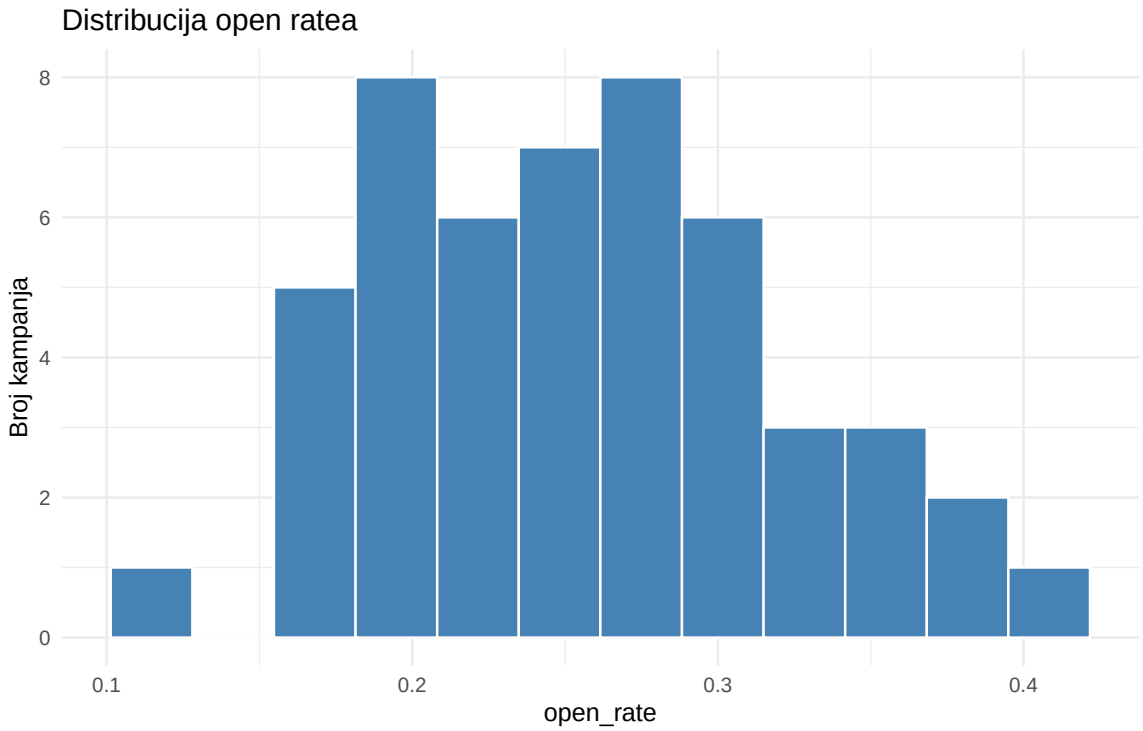
Sad umjesto kopiranja pet blokova koda, pozivamo jednu funkciju s različitim argumentom. Ako želite promijeniti izračun (dodati novu metriku), mijenjate na jednom mjestu i promjena se automatski primjenjuje svugdje.

5.4.5 Funkcija koja vraća graf

Funkcije mogu vraćati bilo koji R objekt, uključujući ggplot grafove.

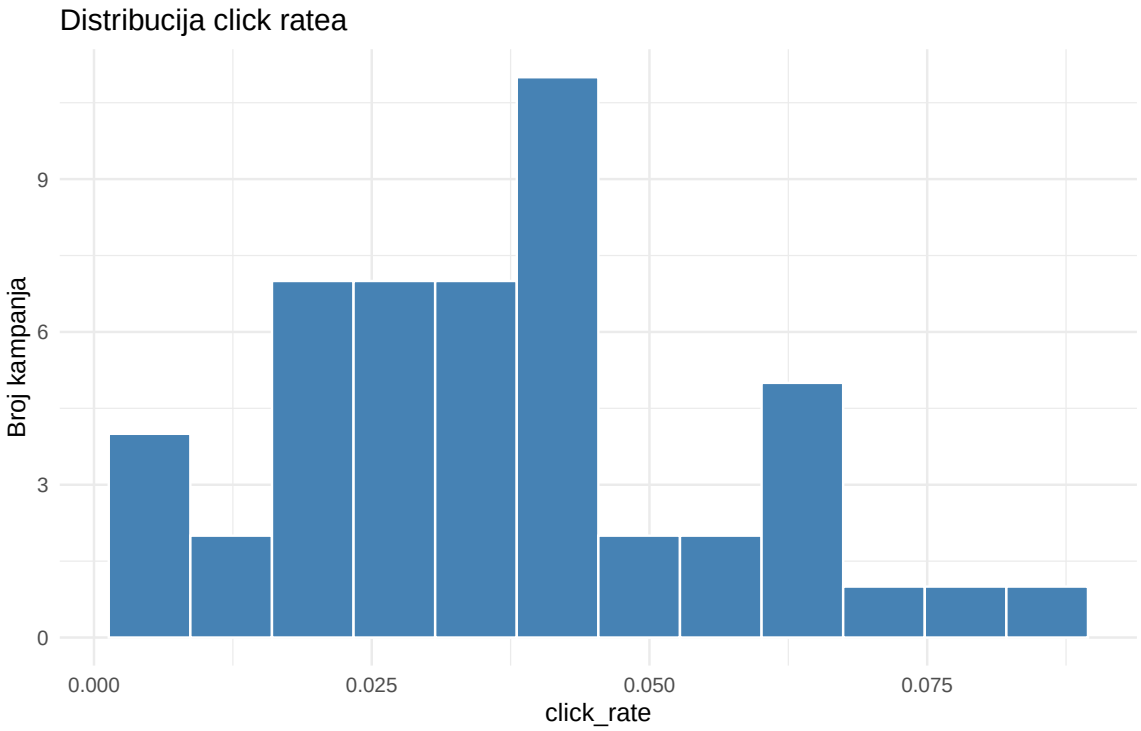
```r
graf_distribucije <- function(data, varijabla, naslov) {
  data |>
    ggplot(aes(x = .data[[varijabla]])) +
    geom_histogram(fill = "steelblue", color = "white", bins = 12) +
    labs(title = naslov, x = varijabla, y = "Broj kampanja") +
    theme_minimal()
```

```
}

graf_distribucije(nl, "open_rate", "Distribucija open ratea")
```



Konstrukcija `.data[[varijabla]]` omogućuje prosljeđivanje imena stupca kao teksta. Ovo je tehnički detalj tidyverse programiranja koji je koristan kad pišete funkcije koje rade s različitim stupcima.

```
graf_distribucije(nl, "click_rate", "Distribucija click ratea")
```



Ista funkcija, druga varijabla, novi graf. Ovo je suština DRY principa—napišete logiku jednom i koristite je koliko god puta trebate.

> **Praktični savjet**
>
> Prema Pravilu tri, ako ste kopirali isti blok koda tri puta ili više, vrijeme je da ga pretvorite u funkciju. Dva kopiranja su još prihvatljiva (ponekad je brže kopirati nego pisati funkciju), ali tri signaliziraju obrazac koji će se ponavljati i dalje. Funkcija vam štedi vrijeme dugoročno i smanjuje rizik od grešaka pri kopiranju.

5.5 Uvjetne naredbe: if i else

Uvjetne naredbe omogućuju R-u da donese odluku—ako je uvjet ispunjen, napravi jedno, inače napravi drugo. Već smo koristili `if_else()` i `case_when()` unutar `mutate()` za rekodiranje varijabli. Sad učimo klasične `if`/`else` naredbe koje rade izvan tibble konteksta.

5.5.1 Osnovna sintaksa

```r
prosjek_or <- mean(nl$open_rate)

if (prosjek_or > 0.25) {
  cat("Prosječni open rate je iznad 25%, što je odličan rezultat.\n")
} else {
  cat("Prosječni open rate je ispod 25%, ima prostora za poboljšanje.\n")
}
```

```
Prosječni open rate je iznad 25%, što je odličan rezultat.
```

R evaluira uvjet u zagradi. Ako je TRUE, izvršava kod u prvom bloku. Ako je FALSE, izvršava kod u else bloku. Funkcija `cat()` ispisuje tekst u konzolu (slično `print()`, ali bez dodatnih oznaka).

5.5.2 if, else if, else

Za više od dva ishoda, koristite `else if`.

```r
ocijeni_kampanju <- function(open_rate) {
  if (open_rate > 0.30) {
    "izvrsna"
  } else if (open_rate > 0.20) {
    "dobra"
  } else if (open_rate > 0.10) {
    "prosječna"
  } else {
    "loša"
  }
}

ocijeni_kampanju(0.35)
```

```
[1] "izvrsna"
```

```r
ocijeni_kampanju(0.22)
```

```
[1] "dobra"
```

```r
ocijeni_kampanju(0.08)
```

```
[1] "loša"
```

Uvjeti se provjeravaju redom, od vrha prema dnu. Čim je jedan uvjet TRUE, pripadajuća vrijednost se vraća i R ne provjerava preostale uvjete. Zato uvjete postavljamo od najstrožeg prema najblažem.

5.5.3 Razlika između if/else i if_else()/case_when()

Ovo je česta točka zbunjenosti. Postoje dva različita sustava uvjetnog izvršavanja u R-u i svaki ima svoje mjesto.

Klasični `if/else` radi s jednom vrijednošću. Koristi se u funkcijama, skriptama i kontroli toka programa. Nije vektoriziran, što znači da ne može obrađivati cijeli stupac odjednom.

`if_else()` i `case_when()` su vektorizirane funkcije. Rade s cijelim vektorom (stupcem) odjednom i koriste se unutar `mutate()` za rekodiranje varijabli u tibbleu.

```
# if_else() unutar mutate: radi na cijelom stupcu
nl |>
  mutate(
    ocjena = if_else(open_rate > 0.25, "iznad prosjeka", "ispod prosjeka")
  ) |>
  count(ocjena)
```

```
# A tibble: 2 x 2
  ocjena             n
  <chr>          <int>
1 ispod prosjeka    25
2 iznad prosjeka    25
```

```
# Klasični if/else: radi s jednom vrijednošću
# (koristili smo ga u funkciji ocijeni_kampanju)
```

Pravilo je jednostavno. Unutar `mutate()` koristite `if_else()` ili `case_when()`. Izvan `mutate()`, u funkcijama i skriptama, koristite klasični `if/else`.

5.5.4 Uvjeti u funkcijama: validacija ulaza

Praktična primjena `if/else` u funkcijama je provjera jesu li ulazni podaci ispravni.

```
izracunaj_ctr <- function(clicks, impressions) {
  if (impressions <= 0) {
    warning("Broj impresija mora biti pozitivan. Vraćam NA.")
    return(NA_real_)
  }
```

```
  if (clicks < 0) {
    warning("Broj klikova ne može biti negativan. Vraćam NA.")
    return(NA_real_)
  }

  round(clicks / impressions, 4)
}

izracunaj_ctr(150, 5000)
```

```
[1] 0.03
```

```
izracunaj_ctr(150, 0)
```

```
[1] NA
```

```
izracunaj_ctr(-10, 5000)
```

```
[1] NA
```

Funkcija `warning()` ispisuje upozorenje ali ne zaustavlja izvršavanje. Funkcija `return()` eksplicitno vraća vrijednost i izlazi iz funkcije. Bez `return()`, funkcija bi nastavila izvršavanje i pokušala podijeliti s nulom.

Validacija ulaza je ono što razdvaja robusne funkcije od krhkih. Kad pišete funkciju za sebe, možda znate da nikad nećete unijeti negativan broj. Ali kad tu funkciju koristi netko drugi (ili vi za šest mjeseci, kad ste zaboravili detalje), validacija sprečava tihe greške.

5.6 For petlje: ponavljanje operacija

Petlja je naredba koja ponavlja blok koda za svaki element u skupu. `for` petlja u R-u ima jednostavnu sintaksu.

5.6.1 Osnovna for petlja

```r
tipovi <- unique(nl$campaign_type)

for (tip in tipovi) {
  n <- nl |> filter(campaign_type == tip) |> nrow()
  cat(tip, ":", n, "kampanja\n")
}
```

```
special_report : 17 kampanja
weekly_digest : 12 kampanja
breaking_news : 7 kampanja
sponsored : 8 kampanja
event_promo : 6 kampanja
```

R prolazi kroz svaki element vektora `tipovi`, dodjeljuje ga varijabli `tip`, i izvršava kod u tijelu petlje. Kad se tijelo izvrši za zadnji element, petlja završava.

5.6.2 For petlja za generiranje rezultata

Čest obrazac je korištenje petlje za prikupljanje rezultata u listu ili tibble.

```r
# Inicijalizirajte praznu listu za rezultate
rezultati <- list()

for (tip in tipovi) {
  saz <- nl |>
    filter(campaign_type == tip) |>
    summarise(
      tip = tip,
      n = n(),
      prosjek_or = round(mean(open_rate), 3),
      prosjek_ctr = round(mean(click_rate), 4)
    )

  rezultati[[tip]] <- saz
}

# Spojite sve rezultate u jedan tibble
bind_rows(rezultati)
```

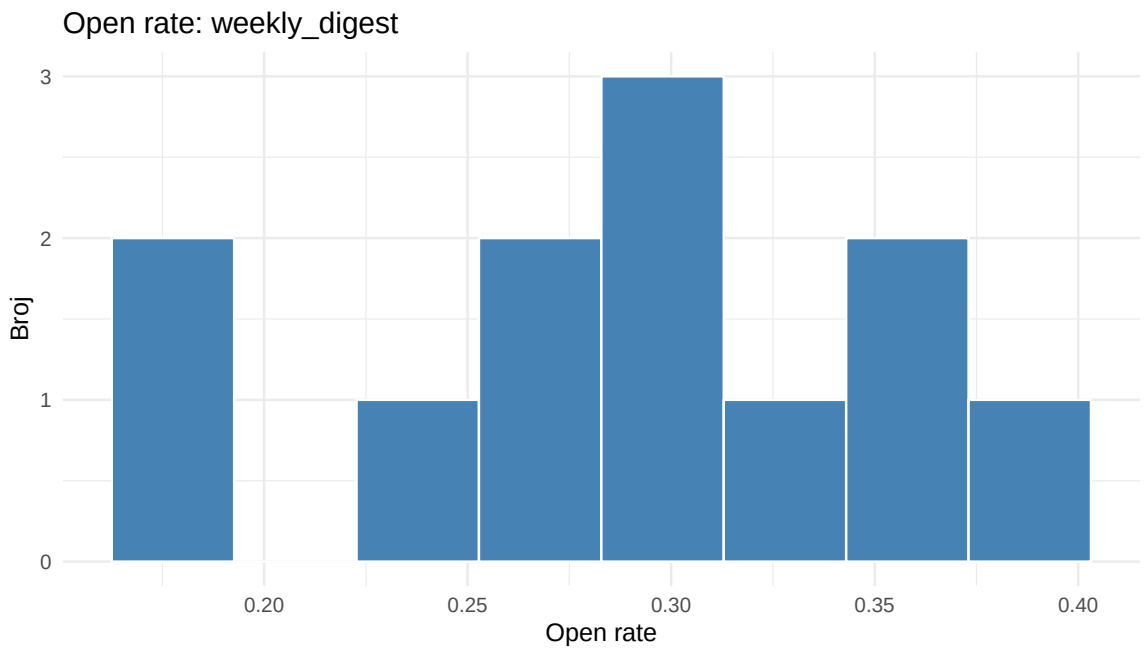
```
# A tibble: 5 x 4
  tip                n prosjek_or prosjek_ctr
  <chr>          <int>      <dbl>       <dbl>
1 special_report    17      0.259      0.0519
```

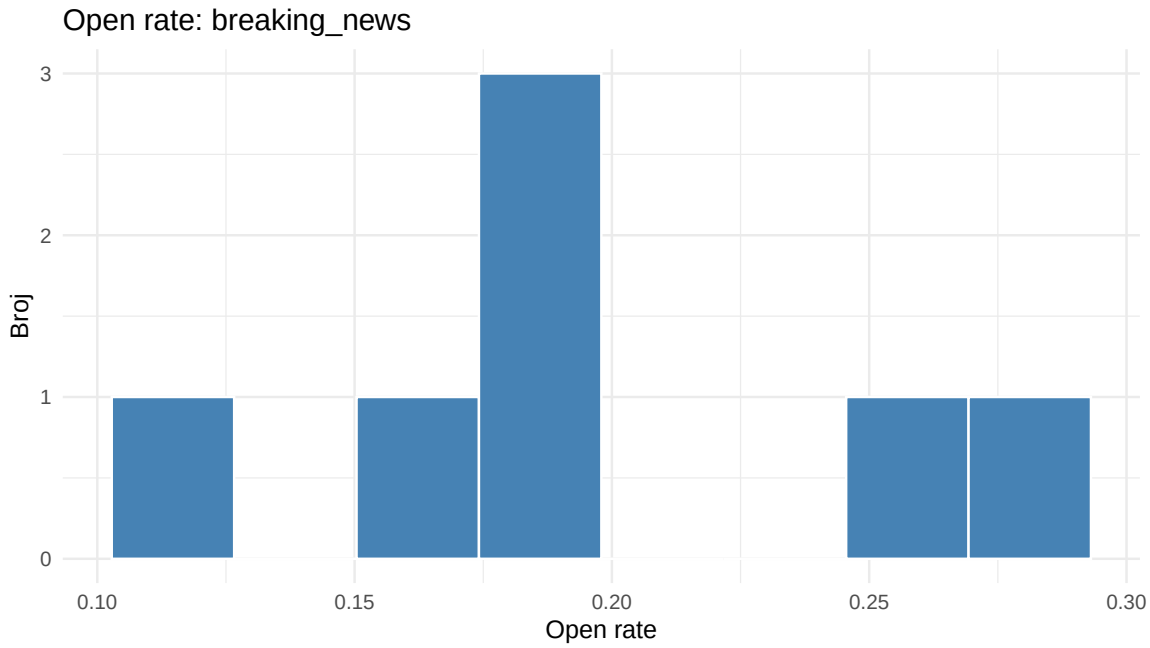
```
2 weekly_digest     12       0.289       0.029
3 breaking_news      7       0.198       0.0382
4 sponsored          8       0.248       0.0223
5 event_promo        6       0.25        0.0294
```

Kreiramo praznu listu `rezultati`, u svakoj iteraciji računamo sažetak i pohranjujemo ga u listu pod imenom tipa kampanje, a na kraju sve spajamo u jedan tibble s `bind_rows()`.

5.6.3 For petlja za generiranje grafova

```
# Generiranje grafa za svaki tip kampanje
for (tip in c("weekly_digest", "breaking_news")) {
  p <- nl |>
    filter(campaign_type == tip) |>
    ggplot(aes(x = open_rate)) +
    geom_histogram(fill = "steelblue", color = "white", bins = 8) +
    labs(
      title = paste("Open rate:", tip),
      x = "Open rate",
      y = "Broj"
    ) +
    theme_minimal()

  print(p)
}
```





Unutar for petlje, ggplot grafove morate eksplicitno ispisati s `print()`. Izvan petlje, R automatski ispisuje zadnji objekt, ali unutar petlje to ne radi. Ovo je čest izvor frustracije za početnike.

> **!** Važna napomena
>
> For petlje u R-u nisu pogrešne ni zastarjele, ali za većinu zadataka u tidyverse ekosustavu postoje elegantnije alternative. `group_by() |> summarise()` zamjenjuje petlje za grupirane sažetke. `across()` zamjenjuje petlje za primjenu iste operacije na više stupaca. `map()` iz paketa purrr zamjenjuje petlje za primjenu funkcije na svaki element liste ili vektora. Petlje koristite kad alternative ne postoje ili kad je petlja jasnija (što se ponekad događa).

5.7 map(): moderna alternativa petljama

Paket purrr (dio tidyverse) pruža obitelj `map()` funkcija koje primjenjuju funkciju na svaki element vektora ili liste. Rezultat ovisi o varijanti map-a koju koristite.

5.7.1 Osnovni map()

```r
tipovi <- unique(nl$campaign_type)

# map() vraća listu
rezultati <- map(tipovi, function(tip) {
  nl |>
    filter(campaign_type == tip) |>
    summarise(
      tip = tip,
      n = n(),
      prosjek_or = round(mean(open_rate), 3)
    )
})

bind_rows(rezultati)
```

```
# A tibble: 5 x 3
  tip                n prosjek_or
  <chr>          <int>      <dbl>
1 special_report    17      0.259
2 weekly_digest     12      0.289
3 breaking_news      7      0.198
4 sponsored          8      0.248
5 event_promo        6      0.25
```

`map()` prima vektor (ili listu) i funkciju, primjenjuje funkciju na svaki element i vraća listu rezultata. Ovo je funkcionalni ekvivalent for petlje ali u kompaktnijem obliku.

5.7.2 Skraćena lambda sintaksa

Umjesto `function(tip) { ... }` možete koristiti skraćenu lambda sintaksu s tildom.

```r
# Skraćena lambda sintaksa: \(x) umjesto function(x)
rezultati <- map(tipovi, \(tip) {
  nl |>
    filter(campaign_type == tip) |>
    summarise(tip = tip, n = n(), prosjek_or = round(mean(open_rate), 3))
})

bind_rows(rezultati)
```

```
# A tibble: 5 x 3
  tip                n prosjek_or
```

```
  <chr>          <int>      <dbl>
1 special_report    17      0.259
2 weekly_digest     12      0.289
3 breaking_news      7      0.198
4 sponsored          8      0.248
5 event_promo        6      0.25
```

Notacija `\(tip)` je R-ova nova (od verzije 4.1) skraćenica za `function(tip)`. Obje verzije rade identično, ali `\(x)` je kraća za pisanje.

5.7.3 Varijante map-a

`map()` uvijek vraća listu. Kad znate kakav tip rezultata očekujete, koristite specifičniju varijantu.

```
# map_dbl() vraća numerički vektor
prosjeci <- map_dbl(tipovi, \(tip) {
  nl |>
    filter(campaign_type == tip) |>
    pull(open_rate) |>
    mean()
})

tibble(tip = tipovi, prosjek_or = round(prosjeci, 3))
```

```
# A tibble: 5 x 2
  tip            prosjek_or
  <chr>               <dbl>
1 special_report      0.259
2 weekly_digest       0.289
3 breaking_news       0.198
4 sponsored           0.248
5 event_promo         0.25
```

`map_dbl()` vraća numerički (double) vektor umjesto liste. `map_chr()` vraća tekstualni vektor. `map_lgl()` vraća logički. `map_df()` vraća tibble (spaja sve rezultate). Koristite specifičnu varijantu kad god možete jer je rezultat predvidljiviji i jednostavniji za daljnji rad.

5.7.4 map() unutar tibble radnog toka

Najelegantnija primjena `map()` je unutar tibble radnog toka, kombinirano s `nest()` i `unnest()`.

```
nl |>
  group_by(campaign_type) |>
  nest() |>
  mutate(
    n = map_dbl(data, nrow),
    prosjek_or = map_dbl(data, \(df) mean(df$open_rate)),
    prosjek_ctr = map_dbl(data, \(df) mean(df$click_rate))
  ) |>
  select(-data) |>
  mutate(across(c(prosjek_or, prosjek_ctr), \(x) round(x, 4)))
```

```
# A tibble: 5 x 4
# Groups:   campaign_type [5]
  campaign_type      n prosjek_or prosjek_ctr
  <chr>          <dbl>      <dbl>       <dbl>
1 special_report    17      0.259      0.0519
2 weekly_digest     12      0.289      0.029
3 breaking_news      7      0.198      0.0382
4 sponsored          8      0.248      0.0223
5 event_promo        6      0.250      0.0294
```

Funkcija `nest()` pakira podatke svake grupe u zasebni tibble unutar liste-stupca `data`. Zatim `map_dbl()` primjenjuje funkciju na svaki od tih ugniježđenih tibbleova. Rezultat je jedan redak po grupi s izračunatim metrikama.

Ovo je napredni obrazac koji ćete cijeniti kad budete radili složenije analize (na primjer, fitanje zasebnog regresijskog modela za svaku grupu u tjednu 14).

5.8 DRY princip i organizacija skripte

DRY (Don't Repeat Yourself) je programerski princip koji kaže da svaka informacija u kodu treba postojati na jednom mjestu. Kad se ponavljate, stvarate više točaka koje trebate ažurirati kad nešto promijenite, a to je recept za greške.

5.8.1 Primjer: parametri na jednom mjestu

```r
# PARAMETRI (mijenjajte ovdje, promjena se propagira svugdje)
min_kampanja_za_analizu <- 5
decimale <- 3
boja_grafova <- "steelblue"
kategorije_interesa <- c("weekly_digest", "breaking_news", "special_report")

# ANALIZA (koristi parametre odozgo)
nl_filtered <- nl |>
  filter(campaign_type %in% kategorije_interesa)

nl_filtered |>
  group_by(campaign_type) |>
  summarise(
    n = n(),
    prosjek_or = round(mean(open_rate), decimale),
    prosjek_ctr = round(mean(click_rate), decimale),
    .groups = "drop"
  ) |>
  filter(n >= min_kampanja_za_analizu)
```

```
# A tibble: 3 x 4
  campaign_type      n prosjek_or prosjek_ctr
  <chr>          <int>      <dbl>       <dbl>
1 breaking_news      7      0.198       0.038
2 special_report    17      0.259       0.052
3 weekly_digest     12      0.289       0.029
```

Svi ključni parametri su definirani na jednom mjestu na vrhu. Kad klijent kaže "pokaži mi analizu samo za weekly digest i special report", mijenjate jednu liniju i cijela analiza se ažurira. Ovo je fundamentalno drugačije od traženja i zamjenjivanja vrijednosti razbacanih po cijelom kodu.

5.8.2 Struktura analitičke skripte

Dobro organizirana skripta ima jasne sekcije. Svaka sekcija radi jednu stvar i jasno je označena.

```r
# ============================================
# Analiza newsletter kampanja
# Autor: Ime Prezime
# Datum: 2025-03-29
# Opis: Sažetak performansi email kampanja
# ============================================
```

```r
# 1. PAKETI ----
library(tidyverse)

# 2. PARAMETRI ----
input_file <- "../resources/datasets/newsletter_campaigns.csv"
output_dir <- "../outputs/"
min_n <- 5

# 3. UČITAVANJE ----
raw <- read_csv(input_file)

# 4. ČIŠĆENJE ----
clean <- raw |>
  filter(!is.na(open_rate)) |>
  mutate(
    campaign_type = factor(campaign_type),
    ocjena = case_when(
      open_rate > 0.30 ~ "izvrsna",
      open_rate > 0.20 ~ "dobra",
      open_rate > 0.10 ~ "prosjecna",
      .default = "losa"
    )
  )

# 5. ANALIZA ----
sazetak <- clean |>
  group_by(campaign_type) |>
  summarise(
    n = n(),
    M_or = mean(open_rate),
    SD_or = sd(open_rate),
    M_ctr = mean(click_rate),
    .groups = "drop"
  )

# 6. VIZUALIZACIJA ----
graf <- ggplot(clean, aes(x = campaign_type, y = open_rate)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  theme_minimal() +
  labs(title = "Open rate po tipu kampanje")

# 7. IZVOZ ----
write_csv(sazetak, paste0(output_dir, "sazetak_kampanja.csv"))
ggsave(paste0(output_dir, "boxplot_open_rate.png"), graf, width = 8, height = 5)
```

Komentari s četiri crtice (`# 1. PAKETI ----`) stvaraju navigacijske oznake u Positronu (ili RStudiju) koje omogućuju brzo skakanje između sekcija. Ovo je konvencija, ne sintaktičko pravilo, ali je široko prihvaćena u R zajednici.

> Vaša skripta je vaš laboratorijski dnevnik. Svaki korak je dokumentiran, svaka odluka komentirana, svaki rezultat ponovljiv. Netko (uključujući vas za šest mjeseci) mora moći pokrenuti skriptu od početka do kraja i dobiti identične rezultate.

5.8.3 Pomoćne funkcije na vrhu skripte

Kad imate funkcije koje koristite na više mjesta u analizi, definirajte ih odmah nakon učitavanja paketa. Ovo ih čini vidljivima kroz cijelu skriptu.

```
# Pomoćne funkcije za newsletter analizu
sazetak_metrike <- function(data, metrika, decimale = 3) {
  data |>
    summarise(
      M = round(mean(.data[[metrika]], na.rm = TRUE), decimale),
      Med = round(median(.data[[metrika]], na.rm = TRUE), decimale),
      SD = round(sd(.data[[metrika]], na.rm = TRUE), decimale),
      Min = round(min(.data[[metrika]], na.rm = TRUE), decimale),
      Max = round(max(.data[[metrika]], na.rm = TRUE), decimale)
    )
}

ocjena_kampanje <- function(open_rate) {
  case_when(
    open_rate > 0.30 ~ "izvrsna",
    open_rate > 0.20 ~ "dobra",
    open_rate > 0.10 ~ "prosjecna",
    .default = "losa"
  )
}

# Korištenje pomoćnih funkcija
nl |>
  group_by(campaign_type) |>
  sazetak_metrike("open_rate")
```

```
# A tibble: 5 x 6
  campaign_type      M   Med    SD   Min   Max
  <chr>          <dbl> <dbl> <dbl> <dbl> <dbl>
1 breaking_news  0.198 0.195 0.055 0.115 0.281
2 event_promo    0.25  0.23  0.086 0.161 0.408
```

```
3 special_report 0.259 0.255 0.067 0.155 0.393
4 sponsored      0.248 0.246 0.043 0.191 0.309
5 weekly_digest  0.289 0.29  0.066 0.178 0.388
```

```r
nl |>
  mutate(ocjena = ocjena_kampanje(open_rate)) |>
  count(ocjena, sort = TRUE)
```

```
# A tibble: 3 x 2
  ocjena        n
  <chr>     <int>
1 dobra        25
2 prosjecna    14
3 izvrsna      11
```

Definirajući `ocjena_kampanje()` kao funkciju, logiku rekodiranja pišete jednom. Ako se kriteriji promijene (recimo, prag za "izvrsno" padne na 0.28), mijenjate na jednom mjestu.

5.9 Praktični primjer: automatizirana analiza po kampanjama

Spojimo sve naučene koncepte u jednom praktičnom primjeru. Cilj je napisati kod koji za svaki tip kampanje generira sažetak tablica i graf, koristeći funkcije, map i DRY principe.

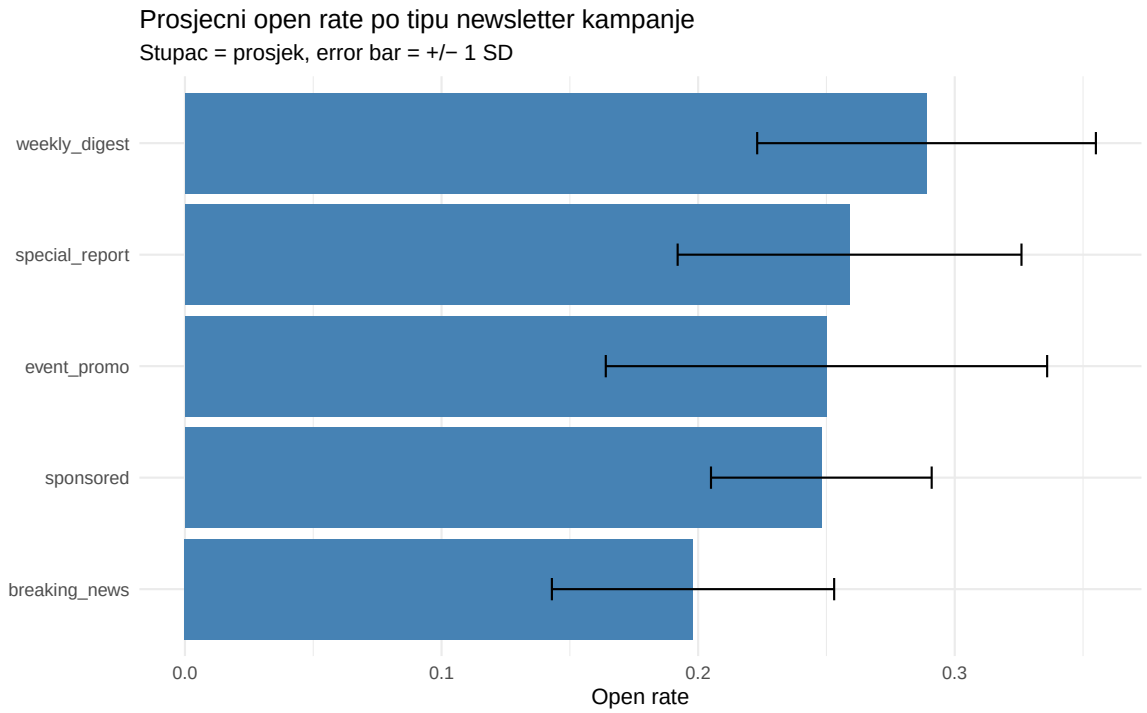
```r
# Funkcija za kompletnu analizu jednog tipa kampanje
analiziraj_tip <- function(data, tip) {
  podaci <- data |> filter(campaign_type == tip)

  if (nrow(podaci) < 3) {
    return(NULL)  # Preskoči tipove s premalo podataka
  }

  saz <- podaci |>
    summarise(
      tip = tip,
      n = n(),
      or_M = round(mean(open_rate), 3),
      or_SD = round(sd(open_rate), 3),
      ctr_M = round(mean(click_rate), 4),
      prosj_pretplatnika = round(mean(subscribers), 0),
      prosj_rijeci = round(mean(word_count), 0)
    )
```

```
  saz
}

# Primjena na sve tipove
svi_tipovi <- unique(nl$campaign_type)

rezultati <- map(svi_tipovi, \(tip) analiziraj_tip(nl, tip)) |>
  bind_rows()

rezultati |>
  arrange(desc(or_M))
```

```
# A tibble: 5 x 7
  tip                n  or_M or_SD  ctr_M prosj_pretplatnika prosj_rijeci
  <chr>          <int> <dbl> <dbl>  <dbl>              <dbl>        <dbl>
1 weekly_digest     12 0.289 0.066 0.029                18109          380
2 special_report    17 0.259 0.067 0.0519               17581          486
3 event_promo        6 0.25  0.086 0.0294               12938          210
4 sponsored          8 0.248 0.043 0.0223               13609          204
5 breaking_news      7 0.198 0.055 0.0382               17160          140
```

Ovaj pristup ima nekoliko prednosti. Logika analize je enkapsulirana u jednu funkciju. Validacija (`if (nrow(podaci) < 3)`) osigurava da ne radimo besmislene izračune na premalo podataka. `map()` elegantno primjenjuje funkciju na sve tipove. Rezultat je čist tibble sortiran po open rateu.

```
# Vizualizacija rezultata
rezultati |>
  mutate(tip = fct_reorder(tip, or_M)) |>
  ggplot(aes(x = tip, y = or_M)) +
  geom_col(fill = "steelblue") +
  geom_errorbar(
    aes(ymin = or_M - or_SD, ymax = or_M + or_SD),
    width = 0.2
  ) +
  coord_flip() +
  labs(
    title = "Prosječni open rate po tipu newsletter kampanje",
    subtitle = "Stupac = prosjek, error bar = +/- 1 SD",
    x = NULL,
    y = "Open rate"
  ) +
  theme_minimal()
```



Error barovi (crte pogreške) prikazuju jednu standardnu devijaciju iznad i ispod prosjeka, dajući vizualni uvid u varijabilnost unutar svake kategorije. Breaking news kampanje imaju viši prosječni open rate ali i veću varijabilnost, dok su sponsored kampanje konzistentno niže.

Podsjetnik

U prvom dijelu naučili smo pisati vlastite funkcije, koristiti uvjetne naredbe, for petlje i map() iz paketa purrr. U ovom dijelu primjenjujemo te vještine na realne radne tokove, kao što su rad s više datoteka, debugging, Quarto izvještaji i kompletna automatizirana analiza.

5.10 Rad s više datoteka

U praksi, podaci rijetko dolaze u jednoj datoteci. Možda imate zasebne CSV datoteke za svaki mjesec, za svaku kampanju ili za svaki izvor podataka. Umjesto ručnog učitavanja svake datoteke, možete automatizirati taj proces koristeći koncepte koje smo upravo naučili.

5.10.1 Pronalaženje datoteka

Funkcija `list.files()` pronalazi datoteke u direktoriju po zadanom uzorku.

```
# Popis svih CSV datoteka u datasets direktoriju
csv_datoteke <- list.files(
  path = "../resources/datasets/",
  pattern = "\\.csv$",
  full.names = TRUE
)

csv_datoteke
```

```
 [1] "../resources/datasets/ab_test_headlines.csv"
 [2] "../resources/datasets/article_engagement.csv"
 [3] "../resources/datasets/article_visuals.csv"
 [4] "../resources/datasets/instagram_ab_test.csv"
 [5] "../resources/datasets/media_habits_raw.csv"
 [6] "../resources/datasets/media_population.csv"
 [7] "../resources/datasets/media_survey_chi2.csv"
 [8] "../resources/datasets/media_trust.csv"
 [9] "../resources/datasets/news_credibility.csv"
[10] "../resources/datasets/newsletter_campaign.csv"
[11] "../resources/datasets/newsletter_campaigns.csv"
[12] "../resources/datasets/social_engagement.csv"
[13] "../resources/datasets/social_media_survey.csv"
[14] "../resources/datasets/social_posts.csv"
[15] "../resources/datasets/tiktok_usage.csv"
```

Argument `pattern = "\\.csv$"` koristi regularni izraz za pronalaženje datoteka koje završavaju s .csv. `full.names = TRUE` vraća kompletne putanje (ne samo imena datoteka), što je bitno jer ih trebamo za učitavanje.

5.10.2 Učitavanje više datoteka odjednom

Kombinirajmo `list.files()`, `map()` i `bind_rows()` za učitavanje i spajanje svih CSV datoteka u jednom koraku.

```
# Učitaj sve CSV datoteke i spoji ih
svi_podaci <- csv_datoteke |>
  map(\(f) read_csv(f, show_col_types = FALSE)) |>
  bind_rows()
```

Ovo je moćan obrazac. `map()` primjenjuje `read_csv()` na svaku putanju, vraćajući listu tibbleova. `bind_rows()` ih vertikalno spaja u jedan veliki tibble. Ako datoteke imaju iste stupce, rezultat je jednostavna konkatenacija. Ako se stupci razlikuju, `bind_rows()` popunjava nedostajuće s NA.

5.10.3 Dodavanje informacije o izvoru

Često želite znati iz koje datoteke dolazi koji redak. Funkcija `set_names()` pomaže.

```
# Učitaj sve datoteke i dodaj stupac s imenom datoteke
svi_podaci <- csv_datoteke |>
  set_names() |>
  map(\(f) read_csv(f, show_col_types = FALSE)) |>
  bind_rows(.id = "izvor")
```

Argument `.id = "izvor"` u `bind_rows()` kreira novi stupac `izvor` koji sadrži ime elementa liste (u ovom slučaju putanju datoteke). Ovo je korisno za praćenje porijekla podataka.

> **Praktični savjet**
>
> Obrazac `list.files() |> map(read_csv) |> bind_rows()` je jedan od najkorisnijih obrazaca u cijelom R radnom toku. Naučite ga napamet. Koristit ćete ga svaki put kad dobijete podatke razdijeljene u više datoteka (mjesečni izvještaji, odvojene ankete, logovi po danima).

5.11 Debugging: pronalaženje i ispravljanje grešaka

Greške su neizbježan dio programiranja. Pitanje nije hoćete li naletjeti na grešku, nego koliko ćete brzo identificirati i ispraviti problem. R daje poruke o greškama koje su ponekad jasne, a ponekad kriptične. Evo strategija za sustavno traženje problema.

5.11.1 Čitanje poruka o greškama

```
# Tipična greška: objekt ne postoji
nl |>
  filter(kampanja_tip == "weekly_digest")
# Error: object 'kampanja_tip' not found

# Čitamo: R ne može naći objekt 'kampanja_tip'
```

```
# Rješenje: provjerimo imena stupaca
names(nl)
# Ah, stupac se zove 'campaign_type', ne 'kampanja_tip'
```

Poruka "object not found" gotovo uvijek znači jednu od tri stvari. Ili ste napravili tipfeler u imenu, ili objekt još nije kreiran (izvršili ste kod izvan redoslijeda), ili je objekt u drugom okruženju (na primjer, kreiran unutar funkcije ali ne i izvan nje).

5.11.2 Strategija: izoliraj problem

Kad imate dugački pipeline koji ne radi, razbijte ga na dijelove i pokrenite svaki zasebno.

```
# Umjesto pokretanja cijelog pipelinea odjednom:
# nl |> filter(...) |> mutate(...) |> group_by(...) |> summarise(...)

# Pokrenite korak po korak:
korak1 <- nl |> filter(campaign_type == "weekly_digest")
korak1  # Provjerite: izgleda li ovo kako očekujete?
```

```
# A tibble: 12 x 13
   campaign_id campaign_type subject_style   day_sent    send_hour subscribers
   <chr>       <chr>         <chr>           <chr>           <dbl>       <dbl>
 1 NL-002      weekly_digest hitno           petak              11       14266
 2 NL-007      weekly_digest upitni          srijeda            13       23450
 3 NL-009      weekly_digest personalizirani subota             11       12444
 4 NL-011      weekly_digest personalizirani utorak             11       23941
 5 NL-016      weekly_digest informativni    cetvrtak           15       12188
 6 NL-022      weekly_digest hitno           utorak             18       12768
 7 NL-026      weekly_digest brojke          utorak              7       12035
 8 NL-032      weekly_digest personalizirani utorak             18       18310
 9 NL-034      weekly_digest brojke          cetvrtak           20       24014
10 NL-035      weekly_digest hitno           ponedjeljak         7       19975
11 NL-037      weekly_digest personalizirani nedjelja           18       20070
12 NL-038      weekly_digest informativni    nedjelja           14       23842
# i 7 more variables: open_rate <dbl>, click_rate <dbl>,
#   unsubscribe_rate <dbl>, word_count <dbl>, n_links <dbl>, has_image <lgl>,
#   revenue <dbl>
```

```
korak2 <- korak1 |> mutate(or_pct = open_rate * 100)
korak2 |> select(campaign_id, open_rate, or_pct) |> head(3)
```

```
# A tibble: 3 x 3
  campaign_id open_rate or_pct
```

```
  <chr>           <dbl>  <dbl>
1 NL-002          0.270   27.0
2 NL-007          0.288   28.8
3 NL-009          0.283   28.3
```

```r
# OK, ovo radi. Idemo dalje...

korak3 <- korak2 |>
  summarise(
    n = n(),
    prosjek = round(mean(or_pct), 1)
  )
korak3
```

```
# A tibble: 1 x 2
      n prosjek
  <int>   <dbl>
1    12    28.9
```

Pohranjivanjem svakog koraka u zasebni objekt, možete točno identificirati na kojem koraku nastaje problem. Kad pronađete i ispravite grešku, spojite korake natrag u pipeline.

5.11.3 print() i glimpse() kao dijagnostika

Unutar funkcija i petlji, dodajte privremene `print()` naredbe da vidite što se događa.

```r
# Debugging s print naredbama
analiziraj_debug <- function(data, tip) {
  podaci <- data |> filter(campaign_type == tip)
  cat("Tip:", tip, "| Redova:", nrow(podaci), "\n")  # Debug ispis

  if (nrow(podaci) == 0) {
    cat("UPOZORENJE: nema podataka za tip", tip, "\n")
    return(NULL)
  }

  podaci |>
    summarise(
      tip = tip,
      n = n(),
      or_M = round(mean(open_rate), 3)
    )
}
```

```
# Testirajte s poznatim i nepoznatim tipom
analiziraj_debug(nl, "weekly_digest")
```

```
Tip: weekly_digest | Redova: 12

# A tibble: 1 x 3
  tip                n  or_M
  <chr>          <int> <dbl>
1 weekly_digest    12 0.289
```

```
analiziraj_debug(nl, "nepostojeci_tip")
```

```
Tip: nepostojeci_tip | Redova: 0
UPOZORENJE: nema podataka za tip nepostojeci_tip

NULL
```

Kad ste riješili problem, uklonite debug ispise. Ostavljanje privremenih `cat()` i `print()` naredbi u gotovom kodu je loša praksa jer zatrpava konzolu nepotrebnim ispisom.

5.11.4 Česte greške i rješenja

Pogledajmo najčešće greške koje ćete susresti i kako ih riješiti.

```
# 1. "could not find function" -> paket nije učitan
summarise(nl, n = n())
# Rješenje: library(tidyverse) na početku

# 2. "unexpected symbol" -> nedostaje zarez, operator ili zagrada
nl |>
  mutate(x = open_rate y = click_rate)  # Nedostaje zarez
# Rješenje: mutate(x = open_rate, y = click_rate)

# 3. "+ ggplot" umjesto "|> ggplot"
nl |>
  filter(open_rate > 0.2) +  # Krivo: + umjesto |>
  ggplot(aes(x = open_rate))
# Rješenje: koristiti |> do ggplot(), pa + za slojeve

# 4. "object of type 'closure' is not subsettable"
mean[1]  # mean je funkcija, ne vektor
# Rješenje: provjerite jeste li slučajno prepisali ime varijable imenom funkcije
```

> Svaki iskusni programer bio je početnik koji je satima tražio zarez koji nedostaje. Debugging nije znak neznanja, nego sastavni dio posla. Razlika između početnika i iskusnog korisnika nije u tome što iskusni ne griješe, nego u tome da imaju sustavan pristup traženju grešaka.

5.12 Quarto: integracija koda, teksta i rezultata

Do sada ste pisali R kod u skriptama (.R datoteke) koje proizvode tablice i grafove u konzoli. Quarto dokumenti (.qmd datoteke) omogućuju nešto moćnije—integraciju teksta, koda i rezultata u jedan dokument koji se renderira u HTML, PDF ili Word.

Zapravo, svako predavanje na ovom kolegiju je Quarto dokument. Tekst koji čitate, kod koji vidite i grafovi koji se prikazuju nastaju iz jedne .qmd datoteke.

5.12.1 Struktura Quarto dokumenta

```
# Quarto dokument ima tri dijela:

# 1. YAML zaglavlje (između --- oznaka)
# ---
# title: "Analiza newsletter kampanja"
# author: "Ime Prezime"
# date: today
# format: html
# ---

# 2. Tekst u Markdown formatu
# ## Uvod
# Ova analiza ispituje performanse naših newsletter kampanja...

# 3. R code chunkovi (između ``` oznaka)
# ```{r}
# library(tidyverse)
# nl <- read_csv("newsletter_campaigns.csv")
# ```
```

Kad pokrenete `quarto render`, Quarto izvršava R kod, hvata rezultate (tablice, grafove, ispis) i umeće ih u dokument zajedno s tekstom. Rezultat je profesionalan izvještaj u kojem su analiza i prezentacija neodvojivi.

5.12.2 Chunk opcije za kontrolu ispisa

Opcije unutar code chunkova kontroliraju što se prikazuje u dokumentu.

```
# echo: true   -> prikaži kod u dokumentu
# echo: false  -> sakrij kod, prikaži samo rezultat
# eval: true   -> izvrši kod
# eval: false  -> ne izvršavaj (samo prikaži kod)
# message: false -> sakrij poruke paketa
# warning: false -> sakrij upozorenja
# fig-width: 8  -> širina grafa u inčima
# fig-height: 5 -> visina grafa u inčima

# Za izvještaj klijentu: echo: false (ne želi vidjeti kod)
# Za kolegicu analitičarku: echo: true (želi vidjeti kako ste to napravili)
```

Ova fleksibilnost je ključna. Isti Quarto dokument možete renderirati s `echo: true` za interni tim (koji želi vidjeti kod) i s `echo: false` za klijenta (koji želi samo rezultate). Mijenjate jednu opciju u YAML zaglavlju i dobivate potpuno drugačiji dokument.

5.12.3 Inline R kod

Osim code chunkova, R vrijednosti možete umetnuti direktno u tekst.

```
n_kampanja <- nrow(nl)
prosjek_or <- round(mean(nl$open_rate) * 100, 1)
najbolji_tip <- nl |>
  group_by(campaign_type) |>
  summarise(or = mean(open_rate), .groups = "drop") |>
  slice_max(or) |>
  pull(campaign_type)
```

U Quarto dokumentu biste napisali tekst poput: “Analizirali smo 50 kampanja. Prosječni open rate je 25.5%. Najbolji rezultat ima tip weekly_digest.”

Kad se dokument renderira, R vrijednosti se automatski umeću u tekst. Ako se podaci promijene, tekst se automatski ažurira. Nikad više ne morate ručno ažurirati brojke u izvještaju.

5.12.4 Quarto vs R skripta: kad koristiti što

R skripta (.R) je pravi izbor kad je cilj izračun, transformacija ili generiranje outputa (tablice, grafovi, datoteke). Skripta je brza za izvršavanje i laka za debugging.

Quarto dokument (.qmd) je pravi izbor kad je cilj komunikacija rezultata. Izvještaj za klijenta, akademski rad, interna prezentacija, kolegijalni materijal. Quarto integrira narativ i rezultate u jedinstven dokument.

U praksi, mnogi analitičari koriste oboje. Skriptu koriste za teški posao (čišćenje, modeliranje), a Quarto za prezentaciju rezultata. Skripta generira čiste podatke i grafove, Quarto ih ugrađuje u priču.

5.13 Funkcionali za složenije radne tokove

Vratimo se purrr paketu i pogledajmo naprednije obrasce koji su korisni u praksi.

5.13.1 walk(): map() bez povratne vrijednosti

Ponekad želite izvršiti nešto za svaki element (na primjer, spremiti graf) ali ne trebate povratnu vrijednost. `walk()` je varijanta `map()` koja izvršava funkciju ali tiho odbacuje rezultat.

```
# Spremi zaseban graf za svaki tip kampanje
tipovi <- unique(nl$campaign_type)

walk(tipovi, \(tip) {
  p <- nl |>
    filter(campaign_type == tip) |>
    ggplot(aes(x = open_rate)) +
    geom_histogram(fill = "steelblue", color = "white", bins = 8) +
    labs(title = paste("Open rate:", tip)) +
    theme_minimal()

  ggsave(paste0("graf_", tip, ".png"), p, width = 7, height = 4)
})
```

`walk()` je idiomatski R način za petlje koje proizvode popratne efekte (side effects) poput spremanja datoteka, ispisa na konzolu ili slanja emailova. Za razliku od `map()`, ne zatrpava konzolu listom NULL vrijednosti.

5.13.2 map2(): paralelna iteracija preko dva vektora

```
# Dva vektora: metrike i njihovi naslovi
metrike <- c("open_rate", "click_rate")
naslovi <- c("Open rate kampanja", "Click rate kampanja")

# map2 iterira paralelno: prvi element s prvim, drugi s drugim
rezultati <- map2(metrike, naslovi, \(metrika, naslov) {
  nl |>
    sazetak_metrike(metrika) |>
    mutate(metrika = naslov)
})

bind_rows(rezultati)
```

```
# A tibble: 2 x 6
      M   Med    SD   Min   Max metrika
  <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 0.255 0.252 0.068 0.115 0.408 Open rate kampanja
2 0.037 0.037 0.019 0.005 0.086 Click rate kampanja
```

`map2()` prima dva vektora i iterira paralelno. Na prvoj iteraciji koristi `metrike[1]` i `naslovi[1]`, na drugoj `metrike[2]` i `naslovi[2]`, i tako dalje. Ovo je korisno kad imate parove ulaznih podataka.

5.13.3 imap(): iteracija s indeksom

```
# imap daje i element i njegovo ime/indeks
nl |>
  group_by(campaign_type) |>
  group_split() |>
  set_names(unique(nl$campaign_type) |> sort()) |>
  imap(\(podaci, ime) {
    tibble(
      tip = ime,
      n = nrow(podaci),
      or_M = round(mean(podaci$open_rate), 3)
    )
  }) |>
  bind_rows()
```

```
# A tibble: 5 x 3
  tip                n  or_M
  <chr>          <int> <dbl>
```

```
1 breaking_news      7 0.198
2 event_promo        6 0.25
3 special_report    17 0.259
4 sponsored          8 0.248
5 weekly_digest     12 0.289
```

`imap()` je varijanta `map()` koja automatski proslijeđuje i element i njegovo ime (ili indeks). Korisna je kad trebate znati koji element trenutno obrađujete, na primjer za imenovanje rezultata ili za dijagnostiku.

5.13.4 possibly(): zaštita od grešaka

Kad primjenjujete funkciju na mnogo elemenata, jedna greška može srušiti cijeli pipeline. `possibly()` omotava funkciju u zaštitni sloj koji hvata greške i vraća default vrijednost umjesto da prekida izvršavanje.

```
# Funkcija koja ponekad pada
opasna_funkcija <- function(tip) {
  podaci <- nl |> filter(campaign_type == tip)
  if (nrow(podaci) < 3) stop("Premalo podataka!")
  mean(podaci$open_rate)
}

# Bez zaštite: jedna greška ruši sve
# map_dbl(c("weekly_digest", "nepostojeci"), opasna_funkcija)  # Error!

# S zaštitom: greška vraća NA, ostali rezultati ostaju
sigurna_funkcija <- possibly(opasna_funkcija, otherwise = NA_real_)

map_dbl(c("weekly_digest", "nepostojeci", "breaking_news"), sigurna_funkcija)
```

```
[1] 0.2886833        NA 0.1978286
```

`possibly(f, otherwise = NA)` kreira novu funkciju koja radi isto kao `f`, ali umjesto da baci grešku, vraća `otherwise` vrijednost. Ovo je neprocjenjivo kad učitavate 50 datoteka i jedna je korumpirana, ili kad analizirate 20 grupa i jedna ima nedovoljno podataka.

5.14 Kompletna analiza: automatizirani izvještaj o kampanjama

Spojimo sve iz ovog predavanja u jednu koherentnu analizu. Cilj je napisati kod koji bi mogao biti tijelo Quarto izvještaja o performansama newsletter kampanja.

```
library(patchwork)

# PARAMETRI
min_kampanja <- 3
decimale <- 3
fokus_metrike <- c("open_rate", "click_rate", "unsubscribe_rate")
```

```
# POMOĆNE FUNKCIJE
sazetak_tipa <- function(data, tip, dec = 3) {
  d <- data |> filter(campaign_type == tip)

  if (nrow(d) < min_kampanja) return(NULL)

  tibble(
    tip = tip,
    n = nrow(d),
    or_M = round(mean(d$open_rate), dec),
    or_SD = round(sd(d$open_rate), dec),
    ctr_M = round(mean(d$click_rate), dec + 1),
    unsub_M = round(mean(d$unsubscribe_rate), dec + 2),
    prosj_rijeci = round(mean(d$word_count), 0),
    udio_sa_slikom = round(mean(d$has_image), 2)
  )
}

graf_usporedba <- function(data, metrika, naslov, boja = "steelblue") {
  data |>
    ggplot(aes(x = fct_reorder(campaign_type, .data[[metrika]]),
               y = .data[[metrika]])) +
    geom_boxplot(fill = boja, alpha = 0.6) +
    coord_flip() +
    labs(title = naslov, x = NULL, y = metrika) +
    theme_minimal()
}
```

```
# ANALIZA
tipovi <- unique(nl$campaign_type)

# Sažetak za sve tipove (s automatskim preskakanjem malih grupa)
tablica_sazetka <- map(tipovi, \(t) sazetak_tipa(nl, t, decimale)) |>
```

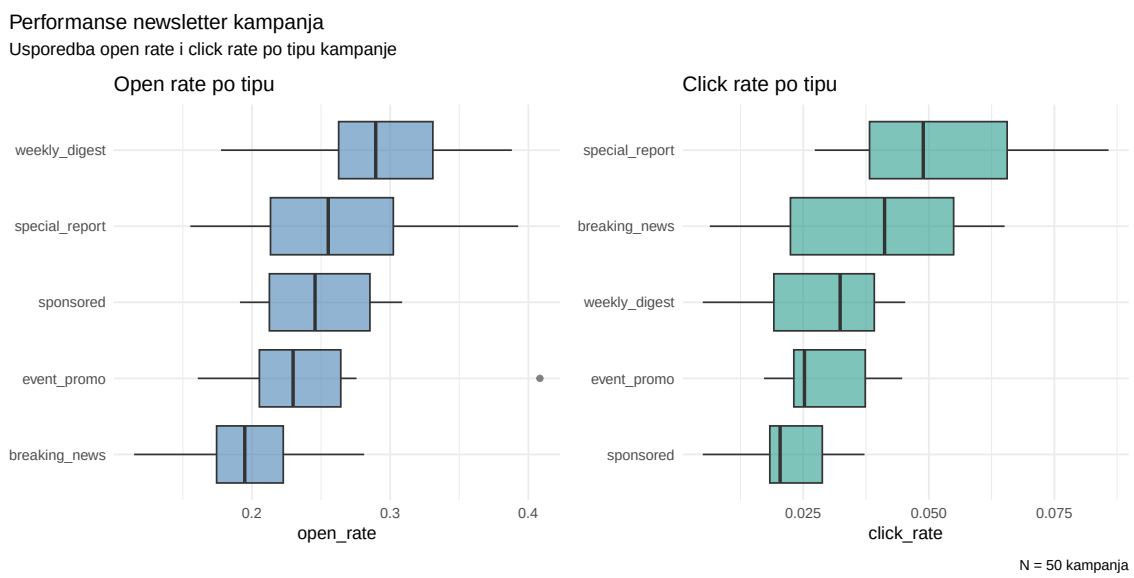
```
  bind_rows() |>
  arrange(desc(or_M))

tablica_sazetka
```

```
# A tibble: 5 x 8
  tip                n  or_M or_SD  ctr_M unsub_M prosj_rijeci udio_sa_slikom
  <chr>          <int> <dbl> <dbl>  <dbl>   <dbl>        <dbl>          <dbl>
1 weekly_digest     12 0.289 0.066 0.029  0.00331          380           0.67
2 special_report    17 0.259 0.067 0.0519 0.00279          486           0.76
3 event_promo        6 0.25  0.086 0.0294 0.00207          210           1
4 sponsored          8 0.248 0.043 0.0223 0.00645          204           0.75
5 breaking_news      7 0.198 0.055 0.0382 0.00405          140           0.71
```

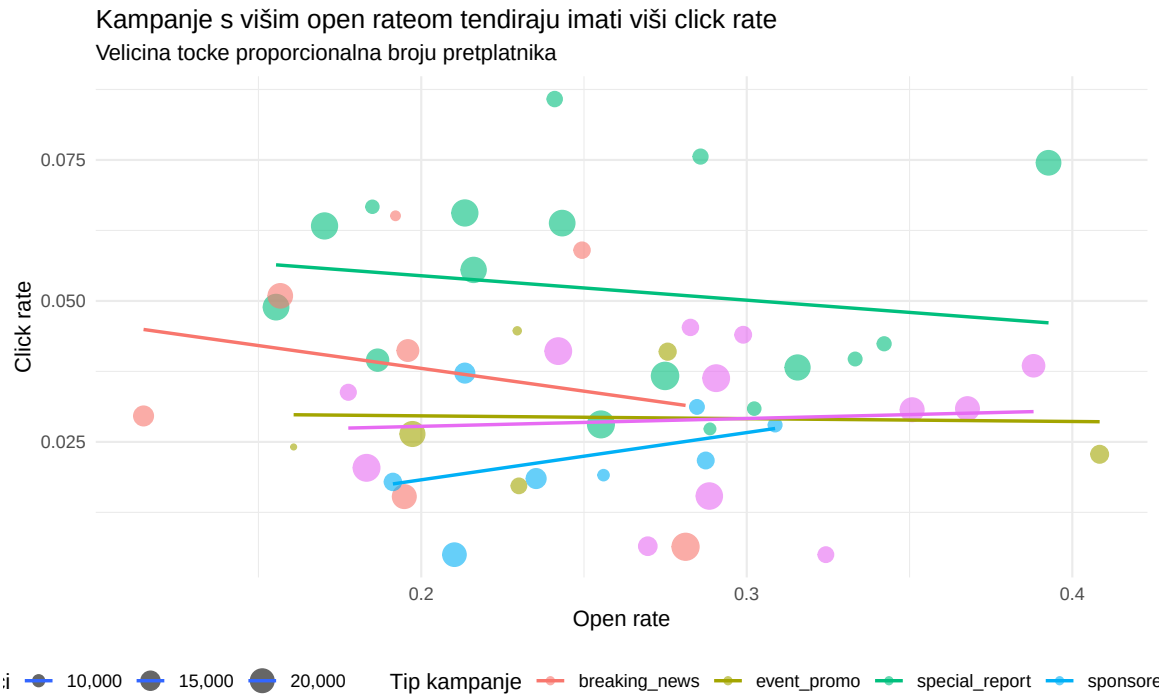
```
# Analiza po stilu naslova (unutar svake kampanje)
nl |>
  group_by(campaign_type, subject_style) |>
  summarise(
    n = n(),
    or_M = round(mean(open_rate), 3),
    .groups = "drop"
  ) |>
  filter(n >= 2) |>
  pivot_wider(
    names_from = subject_style,
    values_from = or_M
  )
```

```
# A tibble: 12 x 7
   campaign_type      n informativni personalizirani upitni brojke  hitno
   <chr>          <int>        <dbl>           <dbl>  <dbl>  <dbl>  <dbl>
 1 breaking_news      5        0.182          NA     NA     NA     NA
 2 event_promo        2       NA               0.285 NA     NA     NA
 3 event_promo        3       NA              NA      0.245 NA     NA
 4 special_report     2       NA               0.242 NA      0.237 NA
 5 special_report     4       NA              NA     NA     NA      0.329
 6 special_report     6        0.2            NA     NA     NA     NA
 7 special_report     3       NA              NA      0.31  NA     NA
 8 sponsored          3       NA              NA     NA      0.252 NA
 9 sponsored          2       NA              NA     NA     NA      0.224
10 weekly_digest      2        0.21           NA     NA      0.254 NA
11 weekly_digest      3       NA              NA     NA     NA      0.312
12 weekly_digest      4       NA               0.328 NA     NA     NA
```

```
# VIZUALIZACIJA
g1 <- graf_usporedba(nl, "open_rate", "Open rate po tipu")
g2 <- graf_usporedba(nl, "click_rate", "Click rate po tipu", boja = "#2a9d8f")

g1 + g2 +
  plot_annotation(
    title = "Performanse newsletter kampanja",
    subtitle = "Usporedba open rate i click rate po tipu kampanje",
    caption = paste("N =", nrow(nl), "kampanja")
  )
```




```
# Odnos open rate i click rate
nl |>
  ggplot(aes(x = open_rate, y = click_rate, color = campaign_type, size = subscribers)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, linewidth = 0.8) +
  scale_size_continuous(labels = scales::label_comma()) +
  labs(
    title = "Kampanje s višim open rateom tendiraju imati viši click rate",
    subtitle = "Veličina točke proporcionalna broju pretplatnika",
    x = "Open rate",
    y = "Click rate",
    color = "Tip kampanje",
    size = "Pretplatnici"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
# Kada slati newsletter?
nl |>
  mutate(
    dio_dana = case_when(
      send_hour < 10 ~ "jutro (6-9)",
      send_hour < 14 ~ "prijepodne (10-13)",
      send_hour < 18 ~ "poslijepodne (14-17)",
      .default = "navečer (18+)"
    )
  ) |>
  group_by(dio_dana) |>
  summarise(
    n = n(),
    or_M = round(mean(open_rate), 3),
    ctr_M = round(mean(click_rate), 4),
    .groups = "drop"
  ) |>
  arrange(desc(or_M))
```

```
# A tibble: 4 x 4
  dio_dana                n  or_M  ctr_M
  <chr>               <int> <dbl>  <dbl>
1 navečer (18+)           9 0.277 0.0388
2 poslijepodne (14-17)    15 0.26  0.04
```

```
3 prijepodne (10-13)      10 0.245 0.0307
4 jutro (6-9)             16 0.243 0.0373
```

```
# KLJUČNI NALAZ: koji faktori predviđaju open rate?
nl |>
  summarise(
    kor_rijeci_or = round(cor(word_count, open_rate), 3),
    kor_linkovi_ctr = round(cor(n_links, click_rate), 3),
    kor_pretplatnici_or = round(cor(subscribers, open_rate), 3)
  )
```

```
# A tibble: 1 x 3
  kor_rijeci_or kor_linkovi_ctr kor_pretplatnici_or
          <dbl>           <dbl>               <dbl>
1         0.127          -0.239              -0.075
```

Analiza otkriva nekoliko nalaza. Breaking news i kampanje s hitnim stilom naslova imaju najviši open rate, ali uz veću varijabilnost. Click rate ne prati savršeno open rate, što sugerira da su čimbenici koji navode ljude da otvore email (naslov, hitnost) različiti od onih koji ih navode da kliknu na sadržaj (relevantnost, format). Korelacija između broja riječi i open ratea govori o preferiranom formatu, dok veličina baze pretplatnika sama po sebi ne predviđa bolji angažman.

Cijela ova analiza, od učitavanja podataka do gotovih grafova i nalaza, koristi principe koje smo naučili ovaj tjedan. Parametri su na jednom mjestu. Pomoćne funkcije enkapsuliraju ponovljivu logiku. map() automatizira iteraciju. Vizualizacija prati principe iz prošlog tjedna. I sve je napisano tako da se može pokrenuti ponovno s novim podacima bez ikakvih promjena u kodu (osim, eventualno, putanje do datoteke).

> Kad pišete analizu, zamislite da ju netko pokreće prvi put, bez ikakvog konteksta. Može li taj netko razumjeti što kod radi, zašto, i kako interpretirati rezultate? Ako da, napisali ste dobru analizu.

! Ključni zaključci

1. Funkcije su alat za izbjegavanje ponavljanja koda. Prema Pravilu tri, ako ste kopirali isti kod tri puta, pretvorite ga u funkciju. Default vrijednosti argumenata čine funkcije fleksibilnima.

2. Klasični `if/else` radi s jednom vrijednošću (za skripte i funkcije). `if_else()` i `case_when()` su vektorizirani (za `mutate()`). Ne miješajte ih.

3. Validacija ulaza u funkcijama sprečava tihe greške. Koristite `warning()` za

upozorenja i `return()` za rano izlaženje.

4. `for` petlje ponavljaju kod za svaki element. Unutar petlje, grafove morate ispisati s `print()`. Za većinu zadataka postoje elegantnije alternative.
5. `map()` iz paketa purrr je moderna alternativa petljama. `map_dbl()`, `map_chr()` i `map_lgl()` vraćaju specifične tipove. `walk()` je za popratne efekte (spremanje datoteka).
6. `map2()` iterira paralelno preko dva vektora. `imap()` daje i element i njegovo ime. `possibly()` štiti od grešaka unutar iteracije.
7. Obrazac `list.files() |> map(read_csv) |> bind_rows()` učitava i spaja više datoteka u jednom koraku.
8. Debugging zahtijeva sustavan pristup, uključujući čitanje poruka, izolaciju problema (korak po korak) i privremene `cat()`/`print()` ispise.
9. DRY princip se primjenjuje na nekoliko načina. Parametri trebaju biti na jednom mjestu, logika u funkcijama, a struktura skripte u jasnim sekcijama.
10. Quarto dokumenti integriraju tekst, kod i rezultate. Koristite ih za izvještaje, radove i prezentacije. R skripte su za teški izračun, Quarto za komunikaciju.
11. Chunk opcije, kao što su `echo`, `eval`, `message`, `warning` i `fig-width`, kontroliraju što se prikazuje u renderiranom dokumentu. Na primjer, `echo: false` sakriva kod za klijente.
12. Cilj ponovljive analize jest taj da netko može pokrenuti vaš kod od početka do kraja s novim podacima i dobiti ažurirane rezultate bez ručnih promjena.

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan ulazimo u **uvod u vjerojatnost**, gdje ćemo naučiti što je vjerojatnost, kako ju računamo, te kakvi su binomna i normalna distribucija. Ovo je konceptualni temelj za sve statističke testove koje ćemo raditi u drugom dijelu kolegija.
Za pripremu:

1. Napišite vlastitu funkciju koja prima tibble i ime kategoričke varijable te vraća tibble s brojem i udjelom (%) svake kategorije. Testirajte je na datasetu `newsletter_campaigns.csv`.
2. Koristeći `map()`, generirajte sažetak open ratea za svaki dan u tjednu (stupac `day_sent`). Spojite rezultate u jedan tibble.
3. Napišite kratki Quarto dokument (.qmd) koji učitava podatke, prikazuje jedan graf i jednu tablicu, s popratnim tekstom. Renderirajte ga u HTML.

4. Pročitajte poglavlje 9 iz Navarro (Learning Statistics with R) o vjerojatnosti. Fokusirajte se na intuiciju, ne na formule.

5.15 Dodatno čitanje

Obavezno

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 26, 27 i 29. Besplatno dostupno na r4ds.hadley.nz. Poglavlje 26 pokriva funkcije, poglavlje 27 iteraciju s purrr, poglavlje 29 Quarto dokumente.

Navarro, D. (2018). *Learning Statistics with R*, Chapter 8. Besplatno dostupno na learningstatisticswithr.com. Osnove programiranja u R-u.

Preporučeno

Wickham, H. (2019). *Advanced R* (2nd edition), Chapters 6 i 9. Besplatno dostupno na adv-r.hadley.nz. Poglavlje 6 detaljno pokriva funkcije, poglavlje 9 funkcionalno programiranje (map i prijatelji).

Quarto dokumentacija. Besplatno dostupno na quarto.org. Kompletna dokumentacija za Quarto sustav sa tutorijalima za HTML, PDF i Word dokumente.

Bryan, J. & Hester, J. *What They Forgot to Teach You About R*. Besplatno dostupno na rstats.wtf. Praktični savjeti o organizaciji projekata, debugging-u i radnim tokovima koji se ne uče u udžbenicima statistike.

5.16 Pojmovnik

Pojam	Objašnjenje
Funkcija	Objekt koji prima argumente, izvršava operacije i vraća rezultat. Definira se s `function()`.
Argument	Ulazni podatak funkcije. Navodi se unutar zagrada pri definiciji i pozivu.
Default vrijednost	Podrazumijevana vrijednost argumenta. Definira se s `=` u listi argumenata.
Povratna vrijednost	Rezultat funkcije. Zadnji izraz u tijelu, ili eksplicitno s `return()`.

Pojam	Objašnjenje
`return()`	Eksplicitno vraća vrijednost i izlazi iz funkcije. Korisno za ranu validaciju.
`if/else`	Uvjetna naredba za kontrolu toka. Radi s jednom vrijednošću (nije vektorizirana).
`if_else()`	Vektorizirana uvjetna funkcija za `mutate()`. Radi na cijelom stupcu.
`case_when()`	Vektorizirana funkcija za složeno rekodiranje s više uvjeta.
`for` petlja	Ponavlja blok koda za svaki element u skupu. Sintaksa: `for (x in skup) { ... }`.
`map()`	purrr funkcija koja primjenjuje funkciju na svaki element i vraća listu.
`map_dbl()`	Varijanta `map()` koja vraća numerički vektor.
`map_chr()`	Varijanta `map()` koja vraća tekstualni vektor.
`map_lgl()`	Varijanta `map()` koja vraća logički vektor.
`map2()`	purrr funkcija za paralelnu iteraciju preko dva vektora.
`imap()`	purrr funkcija koja proslijeđuje i element i njegovo ime/indeks.
`walk()`	Varijanta `map()` za popratne efekte (spremanje datoteka). Ne vraća rezultat.
`possibly()`	purrr funkcija koja omotava funkciju u zaštitni sloj. Greška vraća default vrijednost umjesto prekida.
purrr	Paket iz tidyverse za funkcijsko programiranje.
`nest()`	tidyr funkcija koja pakira podatke grupe u ugniježđeni tibble.
`bind_rows()`	Vertikalno spaja listu tibbleova u jedan.
`list.files()`	Base R funkcija za pronalaženje datoteka u direktoriju po uzorku.
`set_names()`	Dodjeljuje imena elementima vektora ili liste.
DRY	Don't Repeat Yourself. Princip da informacija postoji na jednom mjestu u kodu.
Lambda funkcija	Anonimna funkcija. Piše se kao `\(x) x + 1` ili `function(x) x + 1`.
`.data[[var]]`	Pristup stupcu po imenu pohranjenom u varijabli. Za tidyverse funkcije.
`cat()`	Ispis teksta u konzolu. Bez navodnih oznaka i indeksa.

Pojam	Objašnjenje
`warning()`	Ispis upozorenja. Ne zaustavlja program.
`stop()`	Ispis greške i zaustavljanje programa. Za kritične probleme.
Validacija ulaza	Provjera ispravnosti argumenata prije izvršavanja. Sprečava tihe greške.
Skripta (.R)	R datoteka s nizom naredbi. Za izračune i transformacije.
Quarto dokument (.qmd)	Datoteka koja integrira tekst, kod i rezultate. Za izvještaje i komunikaciju.
Chunk opcije	Postavke R code chunka u Quarto dokumentu (echo, eval, message, warning, fig-width).
Inline R kod	R izraz umetnut u tekst Quarto dokumenta. Automatski se evaluira pri renderiranju.
Debugging	Proces pronalaženja i ispravljanja grešaka u kodu.
Side effect	Popratni efekt funkcije (ispis, spremanje datoteke) koji nije povratna vrijednost.
Ponovljiva analiza	Analiza napisana tako da se može pokrenuti od početka do kraja s novim podacima bez ručnih promjena.

Dio II

Deskriptivna statistika i vizualizacija

6 Tjedan 5: Deskriptivna statistika

Kako brojkama opisati ono što podaci govore

```
library(tidyverse)
library(scales)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti razliku između mjera centralne tendencije (aritmetička sredina, skraćena sredina, medijan, mod) i odabrati odgovarajuću mjeru za različite tipove podataka.
2. Izračunati i interpretirati mjere varijabilnosti (raspon, prosječno apsolutno odstupanje, varijanca, standardna devijacija, interkvartilni raspon) te objasniti zašto varijanca dijeli s $N - 1$, a ne s N.
3. Prepoznati asimetriju i zaobljenost distribucije te objasniti zašto su te karakteristike važne za izbor statističkih metoda.
4. Generirati cjeloviti sažetak varijable koristeći `summarise()` i `across()`.
5. Koristiti `group_by()` i `summarise()` za izračunavanje deskriptivnih statistika po grupama.
6. Izračunati i interpretirati standardne rezultate (z-scores) te objasniti zašto su korisni za usporedbu varijabli na različitim skalama.
7. Izračunati i interpretirati Pearsonov i Spearmanov koeficijent korelacije te razumjeti njihova ograničenja.
8. Prepoznati različite tipove nedostajućih vrijednosti i primijeniti odgovarajuće strategije za rad s njima.

6.1 Zašto su brojke same po sebi beskorisne

Zamislite da ste upravo završili veliko istraživanje o korištenju TikToka u Hrvatskoj. Proveli ste anketu na 300 ispitanika, prikupili podatke o tome koliko minuta dnevno svaka osoba provodi na platformi i sada sjedite pred ogromnom tablicom punom brojki. Vaš urednik ili klijent vas pita jednostavno pitanje. Koliko ljudi zapravo koriste TikTok i koliko vremena tamo provode?

Mogli biste im poslati cijelu tablicu. Svih 300 redova. Ali to nitko neće čitati i, što je još važnije, nitko iz toga neće izvući nikakav zaključak. Ljudski mozak jednostavno nije dizajniran da iz stotina pojedinačnih brojki spontano prepozna obrasce. Upravo zato postoji deskriptivna statistika. Njezin posao je uzeti gomilu podataka i pretvoriti je u nekoliko smislenih brojki koje opisuju što se u podacima zapravo događa.

To zvuči jednostavno, i donekle jest, ali postoji jedna zamka o kojoj treba voditi računa od samog početka. **Svaki put kad sažmete podatke u jednu ili dvije brojke, nešto izgubite.** Aritmetička sredina od 65 minuta dnevno na TikToku zvuči informativno, ali skriva činjenicu da neki ljudi provode 140 minuta, a neki samo 7. Ta informacija o raspršenosti podataka jednako je važna kao i ta jedna prosječna vrijednost, a ponekad je i važnija. Upravo zato u deskriptivnoj statistici nikad ne gledamo samo jednu mjeru. Trebamo barem dvije stvari. Trebamo nešto što nam govori gdje se podaci nalaze (mjere centralne tendencije) i nešto što nam govori koliko su raspršeni (mjere varijabilnosti).

Navarro u svojoj knjizi koristi zgodni paralelizam. Zamislite da vam netko opisuje grupu ljudi riječima. Reći će vam nešto o prosječnoj osobi u grupi (to je centralna tendencija), ali će vam reći i koliko su ljudi u grupi slični jedni drugima ili se pak drastično razlikuju (to je varijabilnost). Trebate obje informacije da biste stvorili mentalnu sliku o čemu se radi. Upravo to ćemo naučiti danas.

Krenimo od podataka.

6.2 Naši podaci: anketa o korištenju TikToka

Tijekom ovog predavanja koristit ćemo simulirani dataset koji sadrži podatke iz ankete o korištenju TikToka. Anketa je provedena na 300 ispitanika različitih dobnih skupina, a prikupljene su informacije o dnevnom vremenu korištenja, broju pogledanih videozapisa tjedno, aktivnosti na platformi i povjerenju u sadržaj koji tamo pronalaze.

Učitajmo podatke i pogledajmo s čime radimo.

```
tiktok <- read_csv("../resources/datasets/tiktok_usage.csv")
glimpse(tiktok)
```

```
Rows: 300
Columns: 11
$ respondent_id         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1~
$ age                   <dbl> 19, 22, 20, 35, 28, 41, 19, 24, 31, 45, 21, 23, ~
$ age_group             <chr> "18-24", "18-24", "18-24", "25-34", "25-
34", "35~
$ gender                <chr> "female", "male", "female", "male", "female", "m~
$ daily_minutes         <dbl> 95, 78, 112, 45, 62, 22, 130, 88, 55, 18, 105, 7~
```

```
$ weekly_videos_watched <dbl> 320, 250, 410, 140, 200, 70, 480, 290, 175, 55, ~
$ likes_given           <dbl> 45, 30, 60, 15, 25, 8, 70, 35, 20, 5, 50, 28, 10~
$ comments_posted       <dbl> 3, 1, 5, 2, 3, 0, 8, 2, 1, 0, 4, 1, 1, 2, 0, 10,~
$ follows_creators      <dbl> 12, 8, 15, 5, 9, 3, 20, 10, 7, 2, 14, 7, 4, 8, 1~
$ trust_score           <dbl> 6, 5, 7, 4, 5, 3, 8, 6, 4, 3, 7, 5, 3, 5, 2, 8, ~
$ education             <chr> "student", "student", "student", "employed", "em~
```

Imamo 300 redova i 11 stupaca. Svaki red predstavlja jednog ispitanika. Varijabla `daily_minutes` bilježi koliko minuta dnevno osoba koristi TikTok, `age_group` svrstava ispitanike u dobne skupine, `trust_score` mjeri povjerenje u TikTok sadržaj na skali od 1 do 10, a `weekly_videos_watched` bilježi otprilike koliko videozapisa tjedno pogledaju.

Pogledajmo prvih desetak redova da stvorimo osjećaj za podatke.

```
tiktok |>
  select(respondent_id, age, age_group, daily_minutes, trust_score) |>
  head(10)
```

```
# A tibble: 10 x 5
   respondent_id   age age_group daily_minutes trust_score
           <dbl> <dbl> <chr>             <dbl>       <dbl>
 1             1    19 18-24                95           6
 2             2    22 18-24                78           5
 3             3    20 18-24               112           7
 4             4    35 25-34                45           4
 5             5    28 25-34                62           5
 6             6    41 35-44                22           3
 7             7    19 18-24               130           8
 8             8    24 18-24                88           6
 9             9    31 25-34                55           4
10            10    45 35-44                18           3
```

Na prvi pogled vidimo da mlađi ispitanici provode više vremena na TikToku od starijih. Ali koliko više? I koliko se ispitanici unutar iste dobne skupine razlikuju međusobno? Na ta pitanja odgovaraju deskriptivne statistike.

6.3 Mjere centralne tendencije

Kad netko kaže da želi znati koliko ljudi koriste TikTok, zapravo pita za neku vrstu tipične ili prosječne vrijednosti. U statistici to zovemo **mjerom centralne tendencije** jer tražimo središte oko kojeg se podaci grupiraju. Ideja je intuitvna, ali čim pokušate biti precizni, stvari

postaju kompliciranije nego što biste očekivali. Postoji više načina da definirate središte skupa podataka, i ne daju svi iste rezultate. U ovom poglavlju obradit ćemo četiri mjere. To su aritmetička sredina, skraćena sredina, medijan i mod.

6.3.1 Aritmetička sredina

Aritmetička sredina je ono što većina ljudi misli kad kaže prosjek. Zbrojite sve vrijednosti i podijelite s brojem opažanja. Formula je jednostavna.

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

gdje je N broj opažanja, a X_i svaka pojedinačna vrijednost. Matematička notacija može izgledati zastrašujuće ako je vidite prvi put, ali ideja je stvarno banalna. Zbrojite sve, podijelite s ukupnim brojem. To je to.

U R u to izračunavamo funkcijom `mean()`. Izračunajmo prosječno dnevno korištenje TikToka u cijelom uzorku.

```
tiktok |>
  summarise(
    prosjek_minuta = mean(daily_minutes),
    n = n()
  )
```

```
# A tibble: 1 x 2
  prosjek_minuta     n
           <dbl> <int>
1           56.9   300
```

Prosječni ispitanik u našem uzorku provodi otprilike 55 minuta dnevno na TikToku. Ali koliko je ta informacija korisna sama za sebe? Zamislite da pišete članak i u njemu navedete samo taj broj. Čitatelj bi mogao pomisliti da većina ljudi provodi oko sat vremena dnevno na TikToku, što je potpuno pogrešan zaključak jer ta sredina skriva enormnu razliku između dobnih skupina.

Aritmetička sredina ima jednu veliku prednost i jednu veliku manu. Prednost je da koristi svaku vrijednost u podacima, pa je u tom smislu najinformativnija mjera. U statistici se kaže da je sredina **dovoljni statistik** (sufficient statistic) za normalnu distribuciju, što znači da sadrži svu informaciju o centralnoj tendenciji koju podaci nude. To zvuči apstraktno, ali praktična implikacija je važna. Ako su vaši podaci normalno distribuirani, aritmetička sredina je definitivno pravi izbor.

Mana je da je **osjetljiva na ekstremne vrijednosti** (outliers). Ako u vašem uzorku postoji jedna osoba koja koristi TikTok 8 sati dnevno (480 minuta), ta jedna osoba će pomaknuti

prosjek za cijeli uzorak prema gore. To je razlog zašto prosječna plaća u nekoj zemlji može biti znatno viša od plaće koju prima većina zaposlenih. Nekolicina ljudi s ekstremno visokim primanjima vuče prosjek prema gore.

Evo konkretnog primjera koji pokazuje koliko jedna ekstremna vrijednost može utjecati na sredinu. Zamislite da imate pet korisnika koji TikTok koriste 20, 25, 30, 35 i 40 minuta dnevno. Prosjek je 30 minuta. Sada zamislite da šesti korisnik provodi 480 minuta (8 sati!) dnevno. Prosjek skače na 105 minuta, što nikako ne opisuje tipičnog korisnika u toj grupi.

```
bez_outliera <- c(20, 25, 30, 35, 40)
s_outlierom <- c(20, 25, 30, 35, 40, 480)

tibble(
  skup = c("Bez ekstremne vrijednosti", "S ekstremnom vrijednošću"),
  prosjek = c(mean(bez_outliera), mean(s_outlierom)),
  medijan = c(median(bez_outliera), median(s_outlierom))
)
```

```
# A tibble: 2 x 3
  skup                        prosjek medijan
  <chr>                         <dbl>   <dbl>
1 Bez ekstremne vrijednosti        30    30
2 S ekstremnom vrijednošću        105    32.5
```

Primijetite kako prosjek skoči sa 30 na 105, dok medijan ostaje stabilan. Na ovo ćemo se vratiti za trenutak.

> **Praktični savjet**
>
> Kad u medijskim izvještajima vidite izraz prosječna vrijednost bez dodatnog konteksta, uvijek se zapitajte postoje li u tim podacima ekstremne vrijednosti. Ako postoje, aritmetička sredina može biti zavaravajuća. Upravo zato odgovorni novinari uz prosjek uvijek navode i medijan, ili barem napomenu o rasponu podataka. Kad čitate da je prosječna plaća u Hrvatskoj, recimo, 1400 eura, zapitajte se koliki je medijan. Razlika vam govori koliko su plaće neravnomjerno raspodijeljene.

6.3.2 Skraćena sredina (trimmed mean)

Postoji kompromis između aritmetičke sredine (koja koristi sve podatke, ali je osjetljiva na outliere) i medijana (koji je robustan, ali ignorira većinu podataka). Taj kompromis zove se **skraćena sredina** (trimmed mean). Ideja je jednostavna. Prije nego izračunamo prosjek, izbacimo određeni postotak najmanjih i najvećih vrijednosti. Najčešće se koristi 5% skraćivanje, što znači da izbacimo 5% najmanjih i 5% najvećih vrijednosti, pa izračunamo prosjek preostalih 90%.

U R u je to trivijalno jer funkcija `mean()` već ima argument `trim` koji prima proporciju (ne postotak!) koja se skraćuje sa svake strane.

```
tiktok |>
  summarise(
    prosjek = mean(daily_minutes),
    skracena_5 = mean(daily_minutes, trim = 0.05),
    skracena_10 = mean(daily_minutes, trim = 0.10),
    medijan = median(daily_minutes)
  )
```

```
# A tibble: 1 x 4
  prosjek skracena_5 skracena_10 medijan
    <dbl>      <dbl>       <dbl>   <dbl>
1    56.9       55.4        54.2      50
```

Vidimo da se skraćena sredina nalazi negdje između pune aritmetičke sredine i medijana. Što više skraćujemo, to se rezultat više približava medijanu. Zapravo, ako postavimo `trim = 0.5`, dobili bismo upravo medijan, jer bismo izbacili sve osim srednje vrijednosti.

Skraćena sredina je osobito korisna kad znate da vaši podaci imaju neke ekstremne vrijednosti, ali ne želite ih potpuno ignorirati. U praksi, 5% ili 10% skraćivanje obično dobro funkcionira. Navarro u knjizi napominje da se skraćena sredina pojavljuje iznenađujuće rijetko u objavljenim istraživanjima, što je šteta, jer je u mnogim situacijama bolji izbor od obične sredine.

6.3.3 Medijan

Medijan je vrijednost koja dijeli podatke na pola. Kad sve vrijednosti poredamo od najmanje do najveće, medijan je ona koja se nalazi točno na sredini. Ako imamo neparan broj opažanja, medijan je srednje opažanje. Ako imamo paran broj, uzimamo prosjek dvaju srednjih opažanja.

Ova definicija zvuči jednostavno, ali skriva nešto dublje. Medijan je zapravo odgovor na pitanje koje se razlikuje od pitanja na koje odgovara sredina. Aritmetička sredina minimizira zbroj kvadriranih odstupanja od sebe same. To zvuči apstraktno, ali praktično znači da sredina daje veliku težinu velikim odstupanjima. Medijan, s druge strane, minimizira zbroj **apsolutnih** odstupanja. To znači da medijan tretira sva odstupanja jednako, neovisno o tome koliko su velika. Zato je robustan.

Ključna razlika u odnosu na aritmetičku sredinu jest da medijan **ne ovisi o ekstremnim vrijednostima**. Ako jedna osoba koristi TikTok 480 umjesto 140 minuta dnevno, medijan se neće promijeniti ni za minutu jer ta osoba i dalje ostaje na istom kraju poretka. Vidjeli smo to u primjeru iznad. Zato kažemo da je medijan **robusna** mjera centralne tendencije.

Izračunajmo medijan za naše podatke, zajedno sa sredinom, kako bismo ih mogli usporediti.

```
tiktok |>
  summarise(
    prosjek = mean(daily_minutes),
    medijan = median(daily_minutes),
    razlika = mean(daily_minutes) - median(daily_minutes)
  )
```

```
# A tibble: 1 x 3
  prosjek medijan razlika
    <dbl>   <dbl>   <dbl>
1    56.9      50    6.85
```

Vidimo da su prosjek i medijan različiti. Kad je prosjek veći od medijana, to je signal da distribucija ima rep prema desno, odnosno da postoje neke veće vrijednosti koje vuku prosjek prema gore. U našem slučaju ta razlika postoji jer imamo velik broj mladih ispitanika koji koriste TikTok znatno više od ostalih, pa distribucija ukupnog uzorka ima pozitivnu asimetriju.

Odnos između sredine i medijana zapravo je brz dijagnostički alat za oblik distribucije. Ako je sredina otprilike jednaka medijanu, distribucija je vjerojatno prilično simetrična. Ako je sredina znatno veća od medijana, distribucija je pozitivno asimetrična (rep prema desno). Ako je sredina znatno manja od medijana, distribucija je negativno asimetrična (rep prema lijevo). Ovo nije egzaktan test, ali u praksi je korisna brza provjera.

6.3.4 Mod

Mod je najjednostavnija mjera centralne tendencije. To je jednostavno vrijednost koja se najčešće pojavljuje u podacima. Za kontinuirane podatke (poput minuta korištenja) mod nije osobito koristan jer svaka vrijednost može biti jedinstvena. Ali za kategoričke podatke, mod je savršena mjera. Zapravo, mod je **jedina** smislena mjera centralne tendencije za kategoričke podatke jer ne možete izračunati prosjek ili medijan kategorija poput spola ili vrste medija.

Na primjer, koji je najčešći tip korisnika u našem uzorku po dobi?

```
tiktok |>
  count(age_group, sort = TRUE)
```

```
# A tibble: 4 x 2
  age_group     n
  <chr>     <int>
1 18-24       102
2 25-34        86
3 35-44        62
4 45+          50
```

Modalna kategorija je 18 do 24 jer ta dobna skupina ima najviše ispitanika. To je logično jer su mladi ljudi dominantna publika TikToka, pa ih je u anketi bilo najlakše regrutirati.

Pogledajmo i mod za razinu obrazovanja i spol.

```
tiktok |>
  count(education, sort = TRUE)
```

```
# A tibble: 2 x 2
  education     n
  <chr>     <int>
1 employed    198
2 student     102
```

```
tiktok |>
  count(gender, sort = TRUE)
```

```
# A tibble: 2 x 2
  gender     n
  <chr>  <int>
1 female   157
2 male     143
```

Jedna stvar koju vrijedi napomenuti o modu jest da distribucija može imati više modova. Kad distribucija ima dva vrha, kažemo da je **bimodalna**. To se u praksi događa kad su u uzorku pomiješane dvije različite populacije. Na primjer, ako bismo gledali distribuciju dnevnog korištenja TikToka za cijeli uzorak (bez razdvajanja po dobi), mogli bismo vidjeti dva vrha. Jedan je za mlade korisnike (oko 100 minuta) i drugi za starije (oko 15 minuta). To je signal da ukupna distribucija zapravo skriva dvije različite grupe, što je izuzetno korisna informacija.

6.3.5 Kada koristiti koju mjeru?

Ovo je pitanje na koje studenti često žele jednostavan odgovor, ali odgovor zapravo ovisi o kontekstu. Ipak, postoje neka korisna pravila koja se izvode iz matematičkih svojstava svake mjere.

Za **numeričke podatke koji su približno simetrično distribuirani** (nemaju dugačke repove na jednoj strani), aritmetička sredina je sasvim dobra mjera. Ona koristi sve podatke i statistička teorija se u velikoj mjeri oslanja na nju. Ako nemate razloga za sumnju u ekstremne vrijednosti, koristite sredinu.

Za **numeričke podatke s izrazitim ekstremnim vrijednostima** (poput prihoda, cijena nekretnina, broja pratitelja na društvenim mrežama ili broja dijeljenja objave), medijan je

pouzdaniji izbor. Alternativno, možete koristiti skraćenu sredinu koja je kompromis između robusnosti i informativnosti.

Za **kategoričke podatke**, mod je jedina smislena opcija jer ne možete izračunati prosjek spola ili vrste medija. To se čini očitim, ali iznenađujuće je koliko se često u izvještajima pokušavaju interpretirati prosjeci Likertove skale (na primjer, prosječan odgovor 3.7 na skali od 1 do 5) kao da su smisleni. Tehnički, Likertove skale su ordinalne varijable, i prosjek ordinalnih podataka je diskutabilan. U praksi se to ipak često radi, ali vrijedi biti svjestan ograničenja.

> **!** Važna napomena
>
> Nikada nemojte izvijestiti samo jednu mjeru centralne tendencije. Kad pišete izvještaj ili analizu, dobra praksa je navesti i prosjek i medijan za numeričke varijable. Ako su slični, distribucija je vjerojatno prilično simetrična. Ako se razlikuju, to je signal da se u podacima nešto zanimljivo događa i vrijedi istražiti dalje. U akademskim radovima iz komunikologije standardno se navode sredina i standardna devijacija za sve ključne varijable, obično u tablici deskriptivnih statistika.

6.4 Mjere varijabilnosti

Znati gdje se podaci nalaze je tek pola priče. Jednako je važno znati koliko su podaci raspršeni. Navarro u knjizi koristi lijep primjer. Zamislite da vam kažem da je prosječna temperatura u dva grada jednaka, recimo 15°C. Na temelju te informacije mogli biste pomisliti da su ta dva grada klimatski slična. Ali zamislite da u jednom gradu temperatura nikad ne padne ispod 10°C niti naraste iznad 20°C, dok u drugom temperatura varira od minus 20°C zimi do plus 45°C ljeti. Očito, to su potpuno različiti gradovi unatoč istoj prosječnoj temperaturi. Razlika je u varijabilnosti.

Isto vrijedi za medijske podatke. Zamislite dva medijska portala čiji članci u prosjeku dobivaju po 50 komentara. Na prvom portalu svaki članak dobiva između 45 i 55 komentara. Na drugom portalu neki članci dobiju 0, a poneki 200. Prosjek je isti, ali situacija je potpuno drugačija. Upravo to razlikuju mjere varijabilnosti.

6.4.1 Raspon

Najjednostavnija mjera varijabilnosti je raspon. To je razlika između najveće i najmanje vrijednosti u podacima. Izračunajmo raspon dnevnog korištenja TikToka.

```
tiktok |>
  summarise(
    minimum = min(daily_minutes),
    maksimum = max(daily_minutes),
    raspon = max(daily_minutes) - min(daily_minutes)
  )
```

```
# A tibble: 1 x 3
  minimum maksimum raspon
    <dbl>    <dbl>  <dbl>
1       7      140    133
```

Raspon nam govori da se dnevno korištenje kreće od jedva desetak minuta do gotovo dva i pol sata, što je ogromna razlika. Međutim, raspon ima isti problem kao aritmetička sredina, samo s druge strane. Ovisi samo o dvije najekstremnije vrijednosti i potpuno ignorira sve ostale. Ako se u uzorku pojavi jedna osoba koja koristi TikTok 12 sati dnevno, raspon će eksplodirati, a svi ostali podaci ostaju isti. Raspon je koristan kao brzi orijentir, ali za ozbiljnu analizu trebamo nešto bolje.

6.4.2 Prosječno apsolutno odstupanje

Prije nego prijeđemo na varijancu, vrijedi se nakratko zadržati na jednoj mjeri koja je konceptualno jednostavnija, a to je **prosječno apsolutno odstupanje** (average absolute deviation, AAD). Ideja je jednostavna. Za svako opažanje izračunamo koliko se razlikuje od aritmetičke sredine (to je odstupanje ili devijacija), uzmemo apsolutnu vrijednost tog odstupanja (jer nas zanima veličina odstupanja, ne smjer), i izračunamo prosjek svih apsolutnih odstupanja.

$$\text{AAD} = \frac{1}{N} \sum_{i=1}^{N} |X_i - \bar{X}|$$

Zašto ovo uopće spominjemo? Zato što je AAD puno intuitivniji od varijance. Kad kažete da je prosječno apsolutno odstupanje 30 minuta, to doslovno znači da se prosječni ispitanik razlikuje od sredine za oko 30 minuta. To je vrlo jednostavno za interpretirati.

R nema ugrađenu funkciju za AAD, ali ga možemo lako izračunati.

```
tiktok |>
  summarise(
    prosjek = mean(daily_minutes),
    aad = mean(abs(daily_minutes - mean(daily_minutes)))
  )
```

```
# A tibble: 1 x 2
  prosjek   aad
    <dbl> <dbl>
1    56.9  33.0
```

Problem s AAD-om je da apsolutna vrijednost matematički nije ugodna za rad. Nije diferencijabilna u nuli, što otežava izvođenje formula u statistici. Zato su statističari davno odlučili koristiti kvadrate umjesto apsolutnih vrijednosti, i tako smo dobili varijancu. Ta odluka ima duboke posljedice za cijelu statistiku, ali za naše potrebe dovoljno je znati da varijanca postoji zato što su kvadrati matematički elegantniji od apsolutnih vrijednosti, čak i ako su manje intuitivni.

6.4.3 Varijanca

Varijanca je sofisticiranija mjera raspršenosti. Ideja je sljedeća. Uzmemo svaku vrijednost u podacima i izračunamo koliko se razlikuje od aritmetičke sredine. To se zove **odstupanje** (deviation). Zatim ta odstupanja kvadriramo (jer bi se pozitivna i negativna inače poništila) i izračunamo njihov prosjek. Ili, točnije, gotovo prosjek.

Pogledajmo najprije formulu, pa ćemo razjasniti taj gotovo prosjek.

$$s^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2$$

6.4.3.1 Zašto dijelimo s $N-1$? Besselova korekcija

Primijetite da dijelimo s $N-1$, a ne s N. Ovo je jedan od onih detalja koji studente redovito zbunjuju, a vrijedi razumjeti zašto je tako. Dijeljenje s $N-1$ zove se **Besselova korekcija** i postoji iz razloga koji su vezani uz procjenu populacijskih parametara iz uzorka.

Evo intuicije. Kad računamo varijancu uzorka, koristimo sredinu uzorka $\bar{X}$ kao procjenu populacijske sredine μ. Ali sredina uzorka je izračunata iz istih podataka iz kojih računamo odstupanja. To stvara suptilni problem. Odstupanja od sredine uzorka su sustavno manja nego što bi bila odstupanja od prave populacijske sredine, jer je sredina uzorka po definiciji najbliža moguća vrijednost tim konkretnim podacima. Dijeljenje s $N-1$ umjesto N ispravlja tu pristranost i daje **nepristranu procjenu** populacijske varijance.

Ako vam ovo zvuči apstraktno, ne brinite previše. Za velike uzorke (N > 30) razlika između dijeljenja s N i $N-1$ je minimalna. Ali za male uzorke može biti značajna, pa se konvencija $N-1$ koristi uvijek. R-ova funkcija `var()` automatski koristi $N-1$.

Navarro u knjizi posvećuje dosta prostora objašnjavanju ovog koncepta i iskreno kaže da je to jedan od najtežih dijelova uvodnog kolegija statistike. Mi ćemo se na ovu temu detaljno vratiti kad budemo govorili o uzorcima i populacijama u kasnijim tjednima. Za sada je dovoljno zapamtiti da `var()` u R u radi ono što treba.

```
tiktok |>
  summarise(
    varijanca = var(daily_minutes)
  )
```

```
# A tibble: 1 x 1
  varijanca
      <dbl>
1     1487.
```

Problem s varijancom je što je teško interpretirati. Mjerna jedinica varijance je kvadrat izvorne mjerne jedinice, dakle u našem slučaju minute na kvadrat. Što to uopće znači, minute na kvadrat? Ništa intuitivno. Zato postoji standardna devijacija.

6.4.3.2 Ručno izračunavanje varijance korak po korak

Korisno je barem jednom vidjeti kako se varijanca računa ručno, korak po korak, čak i ako to u praksi nikad nećemo raditi. Ovo pomaže izgraditi intuiciju.

```
# Uzmimo mali podskup podataka za demonstraciju
demo <- tiktok |>
  slice(1:6) |>
  select(respondent_id, daily_minutes)

demo |>
  mutate(
    sredina = mean(daily_minutes),
    odstupanje = daily_minutes - mean(daily_minutes),
    kvadrirano_odstupanje = odstupanje^2
  )
```

```
# A tibble: 6 x 5
  respondent_id daily_minutes sredina odstupanje kvadrirano_odstupanje
          <dbl>         <dbl>   <dbl>      <dbl>                 <dbl>
1             1            95      69         26                   676
2             2            78      69          9                    81
3             3           112      69         43                  1849
4             4            45      69        -24                   576
5             5            62      69         -7                    49
6             6            22      69        -47                  2209
```

Varijanca je zbroj svih kvadriranih odstupanja podijeljen s $N-1$. Vidimo da veća odstupanja (pozitivna ili negativna) imaju neproporcionalno velik utjecaj na varijancu jer se kvadriraju. To je upravo razlog zašto je varijanca (i standardna devijacija) osjetljiva na ekstremne vrijednosti.

6.4.4 Standardna devijacija

Standardna devijacija je jednostavno korijen iz varijance.

$$s = \sqrt{s^2} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

Njezina prednost je što je u istim mjernim jedinicama kao i izvorni podaci. Ako je standardna devijacija dnevnog korištenja TikToka 40 minuta, to znači da se ispitanici u prosjeku razlikuju od aritmetičke sredine za otprilike 40 minuta. To nije savršena interpretacija (prisjetite se, standardna devijacija koristi kvadrate, ne apsolutne vrijednosti, pa nije identična prosječnom apsolutnom odstupanju), ali je dovoljno dobra za intuitivno razumijevanje.

```
tiktok |>
  summarise(
    prosjek = mean(daily_minutes),
    sd = sd(daily_minutes),
    medijan = median(daily_minutes),
    aad = mean(abs(daily_minutes - mean(daily_minutes)))
  )
```

```
# A tibble: 1 x 4
  prosjek    sd medijan   aad
    <dbl> <dbl>   <dbl> <dbl>
1    56.9  38.6      50  33.0
```

Primijetite da je standardna devijacija nešto veća od prosječnog apsolutnog odstupanja. To je uvijek tako, jer kvadriranje daje veću težinu velikim odstupanjima.

Standardna devijacija je daleko najčešće korištena mjera varijabilnosti u znanosti, uključujući komunikologiju. Kad u akademskom radu vidite tablicu deskriptivnih statistika, gotovo uvijek će sadržavati sredinu (M) i standardnu devijaciju (SD) za svaku varijablu.

6.4.4.1 Interpretacija standardne devijacije

Za podatke koji su približno normalno distribuirani, vrijedi korisno pravilo palca. Otprilike 68% podataka nalazi se unutar jedne standardne devijacije od sredine. Otprilike 95% podataka nalazi se unutar dvije standardne devijacije. Otprilike 99.7% unutar tri standardne devijacije.

Ovo se ponekad naziva pravilo 68-95-99.7 ili empirijsko pravilo. Ako je sredina 55 i SD 40, onda se otprilike 68% ispitanika nalazi između 15 i 95 minuta. Naravno, ovo pravilo vrijedi samo za normalno distribuirane podatke, a naši podaci o TikToku sigurno nisu savršeno normalno distribuirani. Ali i tada, pravilo daje koristan okvirni uvid.

6.4.5 Interkvartilni raspon

Baš kao što medijan ima prednost nad aritmetičkom sredinom kod ekstremnih vrijednosti, tako i **interkvartilni raspon** (IQR) ima prednost nad standardnom devijacijom. Da bismo razumjeli IQR, moramo najprije razumjeti percentile i kvartile.

6.4.5.1 Percentili i kvartili

Percentil je vrijednost ispod koje se nalazi određeni postotak podataka. 25. percentil (ili prvi kvartil, Q1) je vrijednost ispod koje se nalazi 25% podataka. 50. percentil je medijan. 75. percentil (treći kvartil, Q3) je vrijednost ispod koje se nalazi 75% podataka.

Kvartili dijele podatke na četiri jednaka dijela, baš kao što medijan dijeli na dva.

```
tiktok |>
  summarise(
    Q1 = quantile(daily_minutes, 0.25),
    medijan = quantile(daily_minutes, 0.50),
    Q3 = quantile(daily_minutes, 0.75),
    IQR = IQR(daily_minutes)
  )
```

```
# A tibble: 1 x 4
     Q1 medijan    Q3   IQR
  <dbl>   <dbl> <dbl> <dbl>
1    21      50    90    69
```

IQR je razlika između Q3 i Q1, dakle raspon unutar kojeg se nalaze srednjih 50% podataka. To je robusna mjera koja neće skočiti zbog jedne ekstremne vrijednosti.

Možemo izračunati i detaljnije percentile.

```
percentili <- c(0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95)

tibble(
  percentil = percentili * 100,
  vrijednost = quantile(tiktok$daily_minutes, percentili)
)
```

```
# A tibble: 7 x 2
  percentil vrijednost
      <dbl>      <dbl>
1         5       9.95
2        10      11
```

```
3         25       21
4         50       50
5         75       90
6         90      114.
7         95      124.
```

Ova tablica nam daje bogatu sliku o distribuciji. Vidimo da 5% ispitanika koristi TikTok manje od pedesetak minuta (najniži percentil), a 5% koristi više od gornjeg percentila. Srednja polovica uzorka (od 25. do 75. percentila) pokriva raspon koji nam daje IQR.

> Praktični savjet
>
> Kad opisujete podatke u izvještaju ili radu, kombinirajte mjere centralne tendencije i varijabilnosti. Dobar opis bi glasio otprilike ovako. Ispitanici u prosjeku koriste TikTok M minuta dnevno (SD = X), a medijan je Y minuta. Srednja polovica ispitanika provodi na platformi između Q1 i Q3 minuta dnevno (IQR = Z). Ova kombinacija sredine, SD, medijana i IQR daje čitatelju bogatu sliku podataka u samo dvije rečenice.

6.4.6 Koja mjera varijabilnosti?

Izbor mjere varijabilnosti prati istu logiku kao izbor mjere centralne tendencije. Ako ste izabrali sredinu, standardna devijacija je prirodan par jer obje koriste iste matematičke principe (kvadrate odstupanja). Ako ste izabrali medijan, IQR je prirodan par jer su obje robusne mjere.

U praksi, u akademskim radovima gotovo uvijek vidite sredinu i SD. To je konvencija. Ali to ne znači da je to uvijek najbolji izbor. Za podatke koji su jako asimetrični (što je čest slučaj s medijskim metrikama poput broja dijeljenja, broja pratitelja, ili vremena na stranici), medijan i IQR su informativniji.

6.5 Ukupni sažetak varijable

U praksi, kad dobijete novi dataset, prva stvar koju želite napraviti je brzi pregled svih varijabli. Umjesto da za svaku varijablu posebno računate sredinu, SD, medijan i tako dalje, R nudi načine da to napravite odjednom.

Osnovna funkcija `summary()` daje brzi pregled.

```
tiktok |>
  select(daily_minutes, weekly_videos_watched, trust_score) |>
  summary()
```

```
 daily_minutes    weekly_videos_watched  trust_score
 Min.   :  7.00   Min.   : 20.0         Min.   :2.000
 1st Qu.: 21.00   1st Qu.: 65.0         1st Qu.:3.000
 Median : 50.00   Median :159.0         Median :4.000
 Mean   : 56.85   Mean   :189.6         Mean   :4.497
 3rd Qu.: 90.00   3rd Qu.:300.0         3rd Qu.:6.000
 Max.   :140.00   Max.   :500.0         Max.   :8.000
```

Funkcija `summary()` za numeričke varijable automatski ispisuje minimum, prvi kvartil, medijan, sredinu, treći kvartil i maksimum. To je tzv. **five-number summary** (plus sredina), i daje solidan pregled distribucije.

Još moćniji pristup je koristiti `summarise()` s `across()` da izračunamo točno one statistike koje želimo, za sve numeričke varijable odjednom.

```
tiktok |>
  summarise(
    across(
      where(is.numeric),
      list(
        prosjek = ~mean(.x, na.rm = TRUE),
        sd = ~sd(.x, na.rm = TRUE),
        medijan = ~median(.x, na.rm = TRUE)
      ),
      .names = "{.col}_{.fn}"
    )
  ) |>
  pivot_longer(
    everything(),
    names_to = c("varijabla", "statistika"),
    names_sep = "_(?=[^_]+$)",
    values_to = "vrijednost"
  ) |>
  pivot_wider(
    names_from = statistika,
    values_from = vrijednost
  )
```

```
# A tibble: 8 x 4
  varijabla             prosjek     sd medijan
  <chr>                   <dbl>  <dbl>   <dbl>
1 respondent_id           150.    86.7     150.
2 age                      32.1   10.6      30
3 daily_minutes            56.9   38.6      50
4 weekly_videos_watched   190.   139.      159
```

```
5 likes_given             23.5   20.2      17
6 comments_posted          1.91   2.23      1
7 follows_creators         7.08   5.39      6
8 trust_score              4.50   1.85      4
```

Ovaj kod izgleda komplicirano, ali radi nešto vrlo korisno. Za svaku numeričku varijablu izračunava sredinu, SD i medijan, te rezultate prikazuje u čitljivoj tablici. Funkcija `across()` primjenjuje iste izračune na sve odabrane stupce, a `pivot_longer()` i `pivot_wider()` preoblikuju rezultat u preglednu formu. Ovo je obrazac koji ćete koristiti iznova i iznova, pa ga vrijedi zapamtiti (ili još bolje, spremiti u skriptu za ponovnu upotrebu).

6.6 Deskriptivne statistike po grupama

Ukupne statistike su korisne, ali prava snaga deskriptivne analize dolazi do izražaja kad podatke razbijemo po grupama. U komunikologiji nas gotovo uvijek zanima usporedba. Koriste li žene i muškarci TikTok jednako? Razlikuju li se dobne skupine? Ovisi li povjerenje u sadržaj o intenzitetu korištenja?

Tidyverse čini ovu vrstu analize izuzetno elegantnom. Kombinacija `group_by()` i `summarise()` je jedan od najmoćnijih alata koje ćete naučiti u ovom kolegiju. Logika je jednostavna. `group_by()` podijeli podatke u grupe, a `summarise()` izračuna statistike za svaku grupu zasebno. U pozadini, R ponavlja identičan izračun za svaki podskup podataka i rezultate slaže u jednu tablicu.

Pogledajmo kako se korištenje TikToka razlikuje po dobnim skupinama.

```
tiktok |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek = mean(daily_minutes),
    sd = sd(daily_minutes),
    medijan = median(daily_minutes),
    min = min(daily_minutes),
    max = max(daily_minutes),
    .groups = "drop"
  ) |>
  arrange(desc(prosjek))
```

```
# A tibble: 4 x 7
  age_group     n prosjek    sd medijan   min   max
  <chr>     <int>   <dbl> <dbl>   <dbl> <dbl> <dbl>
```

```
1 18-24       102   104.  17.1    104      72   140
2 25-34        86    52.6  7.78    52.5    40    68
3 35-44        62    22.3  2.70    22      18    30
4 45+          50    10.7  2.17    11       7    15
```

Ovdje se jasno vidi ono što smo slutili iz sirovih podataka. Najmlađa skupina (18 do 24 godine) koristi TikTok u prosjeku preko 100 minuta dnevno, dok najstarija skupina (45+) provodi na platformi tek desetak minuta. Razlika je dramatična.

Primijetite i nešto važno. Standardna devijacija unutar svake grupe je znatno manja nego u ukupnom uzorku. To je zato što smo grupiranjem uklonili najveći izvor varijabilnosti, a to je upravo dob. Ova tehnika grupiranja ključna je za razumijevanje podataka. Kad god vidite veliku standardnu devijaciju u ukupnom uzorku, prvo što biste trebali učiniti jest pogledati postoji li neka grupna varijabla koja objašnjava tu varijabilnost. U našem slučaju, dob objašnjava najveći dio razlika u korištenju TikToka.

Možemo ići i korak dalje te kombinirati dva kriterija grupiranja. Pogledajmo korištenje po dobnoj skupini i spolu.

```
tiktok |>
  group_by(age_group, gender) |>
  summarise(
    n = n(),
    prosjek = round(mean(daily_minutes), 1),
    sd = round(sd(daily_minutes), 1),
    .groups = "drop"
  ) |>
  arrange(age_group, gender)
```

```
# A tibble: 8 x 5
  age_group gender     n prosjek    sd
  <chr>     <chr>  <int>   <dbl> <dbl>
1 18-24     female    58   117.   11.1
2 18-24     male      44    87.6   6
3 25-34     female    45    54.8   8.8
4 25-34     male      41    50.1   5.7
5 35-44     female    30    21.8   2.4
6 35-44     male      32    22.8   2.9
7 45+       female    24    10.7   2.2
8 45+       male      26    10.8   2.2
```

Vidimo da unutar svake dobne skupine žene u prosjeku koriste TikTok nešto više nego muškarci. Ta razlika je konzistentna kroz sve dobne skupine, ali je relativno mala u usporedbi s razlikom između samih dobnih skupina. To je upravo ona vrsta uvida koju dobivate kad pravilno razbijete podatke po relevantnim kategorijama.

> ❗ Važna napomena
>
> Argument `.groups = "drop"` na kraju `summarise()` poziva služi tome da R ukloni grupiranje nakon izračuna. Bez njega, rezultirajući tibble bi ostao grupiran po `age_group` (jer je `summarise()` automatski uklonio samo zadnju razinu grupiranja, a `gender` je zadnja). To može uzrokovati neočekivano ponašanje u kasnijim operacijama. Dobra praksa je uvijek eksplicitno navesti ovaj argument kad koristite više od jedne grupirajuće varijable.

Pogledajmo i deskriptivne statistike za varijablu povjerenja u sadržaj (trust_score) po dobnim skupinama.

```
tiktok |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_trust = round(mean(trust_score), 1),
    sd_trust = round(sd(trust_score), 1),
    medijan_trust = median(trust_score),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 5
  age_group     n prosjek_trust sd_trust medijan_trust
  <chr>     <int>         <dbl>    <dbl>         <dbl>
1 18-24       102           6.7      0.8             7
2 25-34        86           4.5      0.5             4
3 35-44        62           3        0               3
4 45+          50           2        0               2
```

Zanimljivo je da povjerenje u sadržaj prati sličan obrazac kao i korištenje. Mladi korisnici imaju veće povjerenje u TikTok sadržaj. Čini se da što više vremena netko provodi na platformi, to više vjeruje sadržaju koji tamo pronalazi. Ovo bi moglo biti zanimljivo za daljnje istraživanje, ali budite oprezni s uzročno-posljedičnim zaključcima. Korelacija nije uzročnost, a do tog pojma stižemo uskoro.

6.7 Oblik distribucije: asimetrija i zaobljenost

Osim centralne tendencije i varijabilnosti, treća važna karakteristika podataka je **oblik distribucije**. Dva najvažnija aspekta oblika su asimetrija (skewness) i zaobljenost (kurtosis).

6.7.1 Asimetrija (skewness)

Distribucija je **simetrična** kad lijeva i desna strana izgledaju kao zrcalna slika. Normalna distribucija (ona poznata zvonolika krivulja) je savršeno simetrična. U praksi, savršena simetrija je rijetka.

Kad distribucija ima dugačak rep prema desno (više ekstremno visokih vrijednosti), kažemo da je **pozitivno asimetrična** (right-skewed ili positively skewed). Kad ima dugačak rep prema lijevo, kažemo da je **negativno asimetrična** (left-skewed ili negatively skewed).

Matematička definicija asimetrije koristi treći standardizirani moment

$$\text{skewness} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{X_i - \bar{X}}{s}\right)^3$$

Primijetite da je eksponent 3, a ne 2 kao kod varijance. Budući da kubiranje čuva predznak (negativan broj na treću ostaje negativan), ovaj izraz je pozitivan kad distribucija ima rep prema desno (jer veliki pozitivni residuali dominiraju) i negativan kad ima rep prema lijevo.

U komunikologiji se pozitivna asimetrija pojavljuje vrlo često. Broj pratitelja na društvenim mrežama, broj dijeljenja objave, prihod od oglašavanja, vrijeme provedeno na web stranici, broj komentara na članku, sve su to varijable koje tipično imaju pozitivnu asimetriju. Većina vrijednosti je relativno mala, ali postoji dugačak rep velikih vrijednosti. To je gotovo univerzalna karakteristika metrika angažmana u digitalnim medijima i vrijedi je zapamtiti kao opće pravilo.

Jednostavan način da provjerite asimetriju jest usporedba aritmetičke sredine i medijana. Ako je sredina veća od medijana, distribucija je vjerojatno pozitivno asimetrična. Ako je medijan veći, negativno asimetrična.

```
tiktok |>
  summarise(
    across(
      c(daily_minutes, weekly_videos_watched, likes_given, comments_posted),
      list(
        prosjek = ~mean(.x),
        medijan = ~median(.x),
        razlika = ~mean(.x) - median(.x)
      ),
      .names = "{.col}_{.fn}"
    )
  ) |>
  pivot_longer(
    everything(),
    names_to = c("varijabla", "statistika"),
    names_sep = "_(?=[^_]+$)",
```

```
    values_to = "vrijednost"
  ) |>
  pivot_wider(names_from = statistika, values_from = vrijednost)
```

```
# A tibble: 4 x 4
  varijabla             prosjek medijan razlika
  <chr>                   <dbl>   <dbl>   <dbl>
1 daily_minutes           56.9       50   6.85
2 weekly_videos_watched  190.       159  30.6
3 likes_given             23.5       17   6.54
4 comments_posted          1.91       1   0.913
```

Pozitivna razlika između prosjeka i medijana za sve varijable potvrđuje da su distribucije pozitivno asimetrične. To je očekivano jer u uzorku postoji skupina mladih korisnika koji imaju izrazito visoke vrijednosti na svim metrikama korištenja.

6.7.2 Zaobljenost (kurtosis)

Zaobljenost opisuje koliko su repovi distribucije teški u usporedbi s normalnom distribucijom. Koristi četvrti standardizirani moment

$$\text{kurtosis} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{X_i - \bar{X}}{s}\right)^4$$

Za normalnu distribuciju, kurtosis iznosi 3. Zato se često koristi **višak zaobljenosti** (excess kurtosis) koji oduzima 3, pa normalna distribucija ima excess kurtosis jednak 0.

Distribucija s velikom zaobljenošću (leptokurtic, excess kurtosis > 0) ima teže repove i oštriji vrh od normalne. To znači da ima više ekstremnih vrijednosti nego što bismo očekivali. Distribucija s malom zaobljenošću (platykurtic, excess kurtosis < 0) ima lakše repove i zaobljeniji vrh.

Za svakodnevni rad u komunikologiji, zaobljenost je manje važna od asimetrije. Ipak, vrijedi ju poznavati jer se pojavljuje u izvještajima statističkog softvera i u akademskim radovima. Najvažnija praktična implikacija je da distribucija s velikom zaobljenošću ima više ekstremnih vrijednosti nego što bismo očekivali od normalne distribucije, što može utjecati na rezultate statističkih testova koji pretpostavljaju normalnost.

> **Praktični savjet**
>
> Asimetrija i zaobljenost postaju osobito važne kad počnemo raditi inferencijsku statistiku (t-testove, ANOVA-u, regresiju) jer mnogi od tih testova pretpostavljaju normalnost distribucije. U tom kontekstu, asimetrija i zaobljenost služe kao dijagnostički alati za provjeru pretpostavki. O tome ćemo detaljno govoriti u kasnijim tjednima.

6.8 Standardni rezultati (z-scores)

Ponekad trebamo usporediti vrijednosti na potpuno različitim skalama. Na primjer, kako usporediti nekoga tko provodi 120 minuta dnevno na TikToku s nekim tko ima trust_score 8? To su različite varijable s različitim mjernim jedinicama i različitim rasponima. Standardni rezultati (z-scores) rješavaju taj problem.

Standardni rezultat nam govori koliko se standardnih devijacija neka vrijednost nalazi iznad ili ispod aritmetičke sredine. Formula je

$$z_i = \frac{X_i - \bar{X}}{s}$$

Ako je z-score jednak 0, ta vrijednost je jednaka prosjeku. Ako je 1, ta vrijednost je jednu standardnu devijaciju iznad prosjeka. Ako je minus 2, ta je vrijednost dvije standardne devijacije ispod prosjeka.

Z-scores su korisni iz još jednog razloga. Kad pretvorite varijablu u z-scores, rezultirajuća varijabla uvijek ima sredinu 0 i standardnu devijaciju 1. To se zove **standardizacija** i često se koristi u naprednim statističkim metodama.

Izračunajmo z-score za dnevno korištenje TikToka. Možemo to napraviti ručno (koristeći formulu) ili pomoću ugrađene funkcije `scale()`.

```
tiktok |>
  mutate(
    z_rucno = (daily_minutes - mean(daily_minutes)) / sd(daily_minutes),
    z_scale = as.numeric(scale(daily_minutes))
  ) |>
  select(respondent_id, age_group, daily_minutes, z_rucno, z_scale) |>
  head(10)
```

```
# A tibble: 10 x 5
   respondent_id age_group daily_minutes z_rucno z_scale
           <dbl> <chr>             <dbl>   <dbl>   <dbl>
 1             1 18-24                95  0.989   0.989
 2             2 18-24                78  0.548   0.548
 3             3 18-24               112  1.43    1.43
 4             4 25-34                45 -0.307  -0.307
 5             5 25-34                62  0.133   0.133
 6             6 35-44                22 -0.904  -0.904
 7             7 18-24               130  1.90    1.90
 8             8 18-24                88  0.808   0.808
```

```
 9             9 25-34                55 -0.0481 -0.0481
10            10 35-44                18 -1.01   -1.01
```

Obje metode daju identične rezultate. Funkcija `scale()` vraća matricu, pa koristimo `as.numeric()` da pretvorimo rezultat u obični numerički vektor.

Sada vidimo da osoba s 95 minuta dnevno ima pozitivan z-score (iznad prosjeka), dok osoba s 22 minute ima negativan z-score (ispod prosjeka). Osoba čiji je z-score oko 2 nalazi se dvije standardne devijacije iznad prosjeka, što je prilično ekstremna vrijednost.

> Z-scores su poput zajedničkog jezika za različite varijable. Svaki put kad pretvorite podatke u z-scores, omogućujete usporedbu jabuka i naranči.

Pogledajmo kako z-scores izgledaju kad ih izračunamo unutar svake dobne skupine (što je ponekad smislenije nego ukupni z-score).

```
tiktok |>
  group_by(age_group) |>
  mutate(
    z_unutar_grupe = as.numeric(scale(daily_minutes))
  ) |>
  ungroup() |>
  select(respondent_id, age_group, daily_minutes, z_unutar_grupe) |>
  head(12)
```

```
# A tibble: 12 x 4
   respondent_id age_group daily_minutes z_unutar_grupe
           <dbl> <chr>              <dbl>          <dbl>
 1             1 18-24                 95        -0.531
 2             2 18-24                 78        -1.52
 3             3 18-24                112         0.463
 4             4 25-34                 45        -0.972
 5             5 25-34                 62         1.21
 6             6 35-44                 22        -0.114
 7             7 18-24                130         1.52
 8             8 18-24                 88        -0.940
 9             9 25-34                 55         0.314
10            10 35-44                 18        -1.60
11            11 18-24                105         0.0539
12            12 18-24                 72        -1.88
```

Sada z-score govori koliko se osoba razlikuje od prosjeka **svoje vlastite dobne skupine**, što je ponekad informativnije od ukupnog z-scorea. Na primjer, osoba od 19 godina koja koristi TikTok 95 minuta dnevno možda ima negativan z-score unutar skupine 18 do 24 (jer je ispod prosjeka te skupine), ali bi imala pozitivan z-score u ukupnom uzorku. Kontekst je važan.

6.9 Korelacije

Do sada smo opisivali jednu varijablu po jednu. Ali u istraživanjima nas često zanima **veza između dviju varijabli**. Postoji li povezanost između dobi ispitanika i vremena koje provode na TikToku? Jesu li ljudi koji više koriste platformu ujedno i oni koji joj više vjeruju?

6.9.1 Kovarijanca: temelj korelacije

Prije nego uđemo u korelaciju, vrijedi razumjeti koncept koji stoji iza nje. To je **kovarijanca**. Kovarijanca mjeri u kojoj mjeri dvije varijable variraju zajedno. Ako su obje varijable iznadprosječne za istog ispitanika i ispodprosječne za istog ispitanika, one pozitivno kovariraju. Ako jedna tendira biti iznadprosječna kad je druga ispodprosječna, negativno kovariraju.

Formula za kovarijancu je

$$\mathrm{Cov}(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})$$

Primijetite sličnost s varijancom. Varijanca je zapravo kovarijanca varijable same sa sobom. Jedina razlika je da umjesto kvadriranja odstupanja jedne varijable, množimo odstupanja dviju različitih varijabli.

Problem s kovarijancom je isti kao s varijancom. Rezultat ovisi o mjernim jedinicama varijabli. Kovarijanca između dobi (u godinama) i korištenja TikToka (u minutama) bit će potpuno drugačija od kovarijance između dobi (u mjesecima) i korištenja TikToka (u satima), čak i ako je veza identična. Zato trebamo standardiziranu mjeru, a to je Pearsonov koeficijent korelacije.

6.9.2 Pearsonov koeficijent korelacije

Najčešća mjera linearne povezanosti dviju numeričkih varijabli je **Pearsonov koeficijent korelacije**, označen s r. To je zapravo standardizirana kovarijanca, što znači da kovarijancu podijelimo s produktom standardnih devijacija obiju varijabli.

$$r_{XY} = \frac{\mathrm{Cov}(X,Y)}{s_X \cdot s_Y} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{X_i - \bar{X}}{s_X}\right)\left(\frac{Y_i - \bar{Y}}{s_Y}\right)$$

Ako prepoznajete z-scores u ovoj formuli, u pravu ste. Korelacija je zapravo prosjek umnožaka z-scores dviju varijabli. To je elegantna definicija jer pokazuje da korelacija zapravo mjeri u kojoj mjeri dvije varijable variraju zajedno, nakon što smo obje stavili na istu skalu.

Vrijednost korelacije kreće se od minus 1 do plus 1. Korelacija blizu plus 1 znači snažnu pozitivnu linearnu vezu (kad jedna varijabla raste, raste i druga). Korelacija blizu minus 1 znači snažnu negativnu linearnu vezu (kad jedna raste, druga pada). Korelacija blizu 0 znači da linearna veza ne postoji ili je vrlo slaba.

Izračunajmo korelaciju između dobi i dnevnog korištenja.

```
tiktok |>
  summarise(
    kovarijanca = cov(age, daily_minutes),
    korelacija = cor(age, daily_minutes)
  )
```

```
# A tibble: 1 x 2
  kovarijanca korelacija
        <dbl>      <dbl>
1       -380.     -0.925
```

Korelacija je snažno negativna, što znači da stariji ispitanici koriste TikTok manje. Kovarijanca je negativna i velika, ali njezina apsolutna vrijednost nam ne govori ništa korisno bez konteksta jer ovisi o mjernim jedinicama. Korelacija od minus 0.9 (ili koliko god iznosi) odmah nam govori da je veza snažna.

6.9.3 Interpretacija korelacija

Jedna od najčešćih pitanja je koliko velika mora biti korelacija da bismo je smatrali značajnom ili važnom. Cohen (1988) je predložio sljedeće smjernice koje se još uvijek široko koriste.

Za korelacije oko $|r| = 0.10$ kažemo da je veza **slaba** (small). Za korelacije oko $|r| = 0.30$ kažemo da je **umjerena** (medium). Za korelacije oko $|r| = 0.50$ ili više kažemo da je **snažna** (large).

No, Navarro s pravom upozorava da su ove smjernice samo grubi orijentiri i da ovise o kontekstu. U nekim područjima (na primjer, u predviđanju ponašanja) korelacija od 0.30 je zapravo prilično impresivna. U drugim (na primjer, u procjeni pouzdanosti testa) korelacija od 0.50 može biti nedovoljna. Kontekst je uvijek ključan.

Pogledajmo više korelacija odjednom.

```
tiktok |>
  summarise(
    r_dob_minute = cor(age, daily_minutes),
    r_minute_trust = cor(daily_minutes, trust_score),
    r_minute_videos = cor(daily_minutes, weekly_videos_watched),
    r_dob_trust = cor(age, trust_score),
    r_likes_comments = cor(likes_given, comments_posted)
  )
```

```
# A tibble: 1 x 5
  r_dob_minute r_minute_trust r_minute_videos r_dob_trust r_likes_comments
         <dbl>          <dbl>           <dbl>       <dbl>            <dbl>
1       -0.925          0.988           0.998      -0.935            0.965
```

Korelacija između dnevnog korištenja i povjerenja u sadržaj je pozitivna i prilično snažna. Ljudi koji više koriste TikTok ujedno iskazuju veće povjerenje u sadržaj koji tamo pronalaze. Korelacija između minuta i broja pogledanih videozapisa je gotovo savršena, što je logično jer su to dvije strane iste medalje. Korelacija između lajkova i komentara je umjereno pozitivna, što sugerira da aktivniji korisnici tendiraju biti aktivni na više načina.

6.9.4 Matrica korelacija

Kad imamo više numeričkih varijabli, korisno je izračunati korelacije između svih parova odjednom. To daje **matricu korelacija**.

```
tiktok |>
  select(age, daily_minutes, weekly_videos_watched, likes_given,
         comments_posted, follows_creators, trust_score) |>
  cor() |>
  round(2)
```

```
                      age daily_minutes weekly_videos_watched likes_given
age                  1.00         -0.92                 -0.90       -0.86
daily_minutes        -0.92          1.00                  1.00        0.98
weekly_videos_watched -0.90         1.00                  1.00        0.99
likes_given          -0.86          0.98                  0.99        1.00
comments_posted      -0.77          0.93                  0.94        0.97
follows_creators     -0.87          0.98                  0.99        0.99
trust_score          -0.93          0.99                  0.98        0.96
                     comments_posted follows_creators trust_score
age                            -0.77            -0.87       -0.93
daily_minutes                   0.93             0.98        0.99
weekly_videos_watched           0.94             0.99        0.98
likes_given                     0.97             0.99        0.96
comments_posted                 1.00             0.98        0.91
follows_creators                0.98             1.00        0.97
trust_score                     0.91             0.97        1.00
```

Matrica korelacija je simetrična (korelacija između X i Y je jednaka korelaciji između Y i X) i na dijagonali su uvijek jedinice (svaka varijabla je savršeno korelirana sama sa sobom). Ovo su svojstva koja slijede direktno iz matematičke definicije korelacije.

6.9.5 Spearmanov koeficijent korelacije

Pearsonov koeficijent mjeri linearnu vezu. Ali što ako veza između varijabli postoji, ali nije linearna? Na primjer, možda korištenje TikToka i povjerenje u sadržaj rastu zajedno, ali ne linearno nego u obliku krivulje. U tom slučaju Pearsonov r može podcijeniti snagu veze.

Za takve situacije postoji **Spearmanov koeficijent korelacije** (ρ ili r_s). Spearmanova korelacija funkcionira tako da najprije pretvori podatke u rangove, a zatim izračuna Pearsonov koeficijent na rangovima. Budući da rangovi čuvaju redoslijed ali ne i udaljenosti, Spearmanova korelacija mjeri **monotonost** veze, to jest koliko dosljedno jedna varijabla raste kad druga raste (ili pada), neovisno o tome je li veza linearna.

U R u, Spearmanova korelacija se računa jednostavno dodavanjem argumenta `method = "spearman"` u funkciju `cor()`.

```
tiktok |>
  summarise(
    pearson = cor(age, daily_minutes, method = "pearson"),
    spearman = cor(age, daily_minutes, method = "spearman")
  )
```

```
# A tibble: 1 x 2
  pearson spearman
    <dbl>    <dbl>
1  -0.925   -0.972
```

Kad su Pearsonov i Spearmanov koeficijent slični, to sugerira da je veza približno linearna. Kad se razlikuju (osobito kad je Spearmanov veći od Pearsonovog), to sugerira da postoji monotona, ali nelinearna veza.

Spearmanova korelacija ima i dodatnu prednost. Manje je osjetljiva na ekstremne vrijednosti nego Pearsonova, jer radi s rangovima, a ne izvornim vrijednostima. Zato se ponekad koristi kao robusna alternativa Pearsonovom koeficijentu.

> Praktični savjet
>
> Kad radite s podacima za koje sumnjate da imaju nelinearnu vezu ili ekstremne vrijednosti, izračunajte i Pearsonov i Spearmanov koeficijent. Ako su slični, veza je vjerojatno linearna i bez problematičnih outliera. Ako se razlikuju, istražite dalje (najčešće pomoću scatterplota, o čemu ćemo govoriti sljedećeg tjedna).

6.9.6 Ograničenja korelacije

Korelacija je izuzetno korisna mjera, ali ima nekoliko važnih ograničenja koja morate poznavati. Navarro im u knjizi posvećuje značajan prostor, i to s dobrim razlogom.

Korelacija mjeri samo linearnu vezu. Ako je veza između dviju varijabli zakrivljena (na primjer, performanse rastu s vježbom ali se onda stabiliziraju), Pearsonov r može biti nizak čak i kad je veza vrlo snažna. Čak i Spearmanov koeficijent zahtijeva monotonost. Ako je veza U-oblik (na primjer, zadovoljstvo je nisko i pri vrlo niskom i pri vrlo visokom radnom opterećenju), ni jedna korelacija neće to uhvatiti.

Korelacija nije uzročnost. Ovo je toliko važno da zaslužuje poseban odlomak. Činjenica da je korištenje TikToka korelirano s povjerenjem u sadržaj ne znači da korištenje TikToka uzrokuje veće povjerenje (niti obratno). Moguće je da treća varijabla, poput dobi, objašnjava obje pojave. Mladi ljudi i više koriste TikTok i općenito imaju drugačiji odnos prema digitalnim medijima. Ovo je toliko čest problem da ima i ime, **confounding** (zbunjivanje) varijabli.

Navarro koristi sjajan primjer. Broj utopljavanja koreliran je s prodajom sladoleda. To ne znači da sladoled uzrokuje utapljanje. Treća varijabla (vrućina) objašnjava oboje, jer kad je vruće, ljudi i kupuju sladoled i idu plivati, a plivanje povećava rizik od utapljanja.

Ekstremne vrijednosti mogu drastično utjecati na korelaciju. Jedna ili dvije ekstremne točke mogu stvoriti iluziju korelacije tamo gdje je zapravo nema, ili maskirati korelaciju koja zapravo postoji.

Ograničeni raspon smanjuje korelaciju. Ako vaš uzorak pokriva samo uzak raspon jedne varijable, korelacija će biti niža nego u populaciji. Na primjer, ako istražujete vezu između IQ-a i akademskog uspjeha, ali vaš uzorak uključuje samo studente na elitnom sveučilištu (gdje svi imaju visok IQ), korelacija će biti niska jer nema dovoljno varijabilnosti u IQ-u.

> **!** Važna napomena
>
> Kad god izračunate korelaciju, obavezno napravite i scatterplot. Postoje poznati primjeri (poput Anscombeovog kvarteta, a u novije vrijeme i Datasaurus Dozen) u kojima potpuno različiti skupovi podataka imaju identičnu korelaciju, a vizualno su potpuno različiti. Grafiku ćemo detaljno raditi sljedeći tjedan, ali zapamtite ovo pravilo od sada. Brojke bez grafike mogu lako zavarati.

6.10 Rad s nedostajućim vrijednostima

U stvarnom svijetu podaci gotovo nikad nisu potpuni. Ispitanici preskoče pitanje u anketi, senzor prestane raditi, sustav ne zabilježi klik. R koristi oznaku `NA` (not available) za nedostajuće vrijednosti, i ove vrijednosti zahtijevaju posebnu pažnju.

6.10.1 Kako R tretira nedostajuće vrijednosti

Problem je u tome što većina R funkcija za deskriptivnu statistiku vraća `NA` ako u podacima postoji ijedna nedostajuća vrijednost.

```
primjer <- c(10, 20, NA, 40, 50)

# Ovo vraća NA
mean(primjer)
```

```
[1] NA
```

```
# Ovo ignorira NA i računa prosjek od preostalih vrijednosti
mean(primjer, na.rm = TRUE)
```

```
[1] 30
```

To je zapravo dobro ponašanje jer vas prisiljava da svjesno odlučite što ćete učiniti s nedostajućim podacima. Ali u praksi, najčešće rješenje je dodati argument `na.rm = TRUE` koji govori R u da ignorira nedostajuće vrijednosti.

Unutar tidyverse pipeline, `na.rm = TRUE` se stavlja unutar svake funkcije u `summarise()`.

```
tiktok |>
  summarise(
    prosjek = mean(daily_minutes, na.rm = TRUE),
    sd = sd(daily_minutes, na.rm = TRUE),
    medijan = median(daily_minutes, na.rm = TRUE)
  )
```

```
# A tibble: 1 x 3
  prosjek    sd medijan
    <dbl> <dbl>   <dbl>
1    56.9  38.6      50
```

6.10.2 Tipovi nedostajućih vrijednosti

Navarro u knjizi ne ulazi duboko u ovu temu, ali za komunikologe je korisno poznavati osnove. Postoje tri tipa nedostajućih vrijednosti.

MCAR (Missing Completely at Random) znači da je nedostajanje potpuno nasumično, nepovezano ni s jednom varijablom u podacima. Na primjer, ispitanik je slučajno preskočio pitanje jer mu je kliznuo prst na touchscreenu. U tom slučaju, ignoriranje nedostajućih vrijednosti (`na.rm = TRUE`) ne uvodi nikakvu pristranost.

MAR (Missing at Random) znači da je nedostajanje povezano s drugim varijablama u podacima, ali ne s nedostajućom vrijednošću samom. Na primjer, stariji ispitanici češće preskaču pitanja o TikToku jer smatraju da se to na njih ne odnosi. Nedostajanje je povezano s dobi, ali ne izravno s korištenjem TikToka. U tom slučaju, ignoriranje nedostajućih vrijednosti može uvesti pristranost, ali postoje statističke metode za korekciju.

MNAR (Missing Not at Random) znači da je nedostajanje izravno povezano s vrijednošću koja nedostaje. Na primjer, ljudi koji provode izrazito mnogo vremena na TikToku možda preskaču to pitanje jer ih je sram priznati koliko vremena tamo provode. Ovo je najproblematičniji slučaj jer ga je teško detektirati i ispraviti.

Za uvodni kolegij, dovoljno je biti svjestan da nedostajuće vrijednosti nisu uvijek nasumične. Kad god imate nedostajuće podatke, razmislite zašto nedostaju i je li sigurno jednostavno ih ignorirati.

6.10.3 Provjera nedostajućih vrijednosti

Prije bilo kakve analize uvijek provjerite koliko nedostajućih vrijednosti imate.

```
tiktok |>
  summarise(across(everything(), ~sum(is.na(.x)))) |>
  pivot_longer(everything(), names_to = "varijabla", values_to = "broj_NA")
```

```
# A tibble: 11 x 2
   varijabla             broj_NA
   <chr>                   <int>
 1 respondent_id               0
 2 age                         0
 3 age_group                   0
 4 gender                      0
 5 daily_minutes               0
 6 weekly_videos_watched       0
 7 likes_given                 0
 8 comments_posted             0
 9 follows_creators            0
10 trust_score                 0
11 education                   0
```

U našem datasetu nema nedostajućih vrijednosti jer su podaci simulirani. Ali u stvarnom radu ih gotovo sigurno hoćete imati. Navikavanje na `na.rm = TRUE` od početka će vam uštedjeti mnogo frustracije.

Praktični savjet

Ako neka varijabla ima velik postotak nedostajućih vrijednosti (recimo više od 20%), razmislite treba li uopće koristiti tu varijablu u analizi. Također, umjesto `na.rm = TRUE`, ponekad je bolje koristiti `tidyr::drop_na()` na početku analize kako biste radili s kompletnim opažanjima. Razlika je u tome što `na.rm = TRUE` radi po varijabli (svaka statistika koristi sva dostupna opažanja), dok `drop_na()` eliminira cijele retke koji imaju bilo koju nedostajuću vrijednost. Koja opcija je bolja ovisi o konkretnoj situaciji.

6.11 Sve zajedno: kompletna deskriptivna analiza

Da bismo zaokružili ovo predavanje, napravimo kompletnu deskriptivnu analizu našeg dataseta o korištenju TikToka. Ovo je obrazac koji ćete koristiti na početku gotovo svake analize. Učitajte podatke, pogledajte strukturu, izračunajte deskriptivne statistike ukupno i po grupama, provjerite korelacije.

```
tiktok |>
  group_by(age_group) |>
  summarise(
    n = n(),
    prosjek_min = round(mean(daily_minutes), 1),
    sd_min = round(sd(daily_minutes), 1),
    medijan_min = median(daily_minutes),
    prosjek_trust = round(mean(trust_score), 1),
    sd_trust = round(sd(trust_score), 1),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 7
  age_group     n prosjek_min sd_min medijan_min prosjek_trust sd_trust
  <chr>     <int>       <dbl>  <dbl>       <dbl>         <dbl>    <dbl>
1 18-24       102       104.    17.1       104             6.7      0.8
2 25-34        86        52.6    7.8        52.5           4.5      0.5
3 35-44        62        22.3    2.7        22             3        0
4 45+          50        10.7    2.2        11             2        0
```

Ova tablica u šest redova sažima informacije koje bi vam inače trebale stranice i stranice sirovih podataka. Vidimo jasne obrasce. Mladi (18 do 24) koriste TikTok oko 100 minuta dnevno s umjerenom varijabilnošću. Srednja skupina (25 do 34) koristi ga upola manje. Starije skupine jedva ga koriste, ali su unutar tih skupina ispitanici prilično ujednačeni (mala

standardna devijacija). Povjerenje u sadržaj prati isti obrazac jer je snažno korelirano s intenzitetom korištenja.

Dopunimo ovu tablicu korelacijama unutar svake skupine.

```
tiktok |>
  group_by(age_group) |>
  summarise(
    n = n(),
    r_minute_trust = round(cor(daily_minutes, trust_score), 2),
    r_minute_videos = round(cor(daily_minutes, weekly_videos_watched), 2),
    .groups = "drop"
  )
```

```
# A tibble: 4 x 4
  age_group     n r_minute_trust r_minute_videos
  <chr>     <int>          <dbl>           <dbl>
1 18-24       102           0.94            1
2 25-34        86           0.86            1
3 35-44        62          NA               0.99
4 45+          50          NA               0.99
```

Ovaj korak je važan jer korelacije mogu biti različite u podskupovima nego u ukupnom uzorku. Na primjer, ukupna korelacija između korištenja i povjerenja može biti visoka dijelom zato što obje varijable koreliraju s dobi (mlađi koriste više i imaju veće povjerenje). Korelacija unutar svake dobne skupine govori nam postoji li veza i nakon što smo kontrolirali za dob. Ovo je uvod u koncept kontrole varijabli koji ćemo detaljno obraditi kad budemo radili regresiju.

! Ključni zaključci

1. Deskriptivna statistika sažima podatke u manji broj smislenih brojki, ali svako sažimanje znači i gubitak informacija.
2. Mjere centralne tendencije (sredina, skraćena sredina, medijan, mod) odgovaraju na pitanje gdje se podaci nalaze. Svaka ima svoja svojstva, uključujući sredinu koja koristi sve podatke ali je osjetljiva na outliere, medijan koji je robustan ali ignorira većinu podataka, i skraćenu sredinu kao kompromis.
3. Mjere varijabilnosti (raspon, AAD, varijanca, SD, IQR) odgovaraju na pitanje koliko su podaci raspršeni. Varijanca dijeli s $N-1$ umjesto N (Besselova korekcija) kako bi bila nepristrana procjena populacijske varijance.
4. Asimetrija i zaobljenost opisuju oblik distribucije. Medijske metrike gotovo uvijek

imaju pozitivnu asimetriju (dugačak rep prema desno).

5. Z-scores standardiziraju varijable na zajedničku skalu (sredina 0, SD 1) i omogućuju usporedbu varijabli s različitim mjernim jedinicama.
6. Kombinacija `group_by()` i `summarise()` je temeljni alat za izračunavanje deskriptivnih statistika po grupama u R u.
7. Pearsonov koeficijent korelacije mjeri linearnu povezanost, a Spearmanov mjeri monotonu povezanost. Korelacija nije uzročnost i mjeri samo specifičan tip veze.
8. Nedostajuće vrijednosti (`NA`) zahtijevaju svjesnu odluku o tome kako ih tretirati. Različiti tipovi nedostajanja (MCAR, MAR, MNAR) impliciraju različite posljedice za analizu.
9. Uvijek kombinirajte numeričke statistike s vizualizacijom. Brojke bez grafike mogu zavarati (Anscombeov kvartet).

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan bavimo se **vizualizacijom podataka s ggplot2**. To je prirodni nastavak ovog predavanja jer ćemo naučiti kako sve statistike koje smo danas izračunali prikazati grafički. Histogrami, boxplotovi, scatterplotovi i bar chartovi su alati koji daju život brojkama.
Za pripremu napravite sljedeće:

1. Ponovite današnje R primjere i eksperimentirajte s njima. Promijenite varijable u `summarise()` pozivu i pogledajte što se događa.
2. Razmislite o tome koje bi grafičke prikaze htjeli vidjeti za naše TikTok podatke. Histogram dnevnog korištenja? Boxplot po dobnim skupinama? Scatterplot dobi i korištenja?
3. Pročitajte poglavlje 3 iz knjige Kieran Healy, *Data Visualization* (besplatno dostupno online).
4. Instalirajte paket `patchwork` ako ga nemate sa `install.packages("patchwork")`. Koristit ćemo ga za kombiniranje grafika.

6.12 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 5: Descriptive Statistics. Besplatno dostupno na learningstatisticswithr.com. Poglavlje pokriva iste teme kao ovo predavanje, uključujući trimmed mean, kovarijancu i Spearmanovu korelaciju, ali s primjerima iz psihologije i u base R sintaksi.

Preporučeno

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 3 i 4. Besplatno dostupno na r4ds.hadley.nz. Odlično pokrivanje tidyverse pristupa manipulaciji i sažimanju podataka.

Healy, K. (2018). *Data Visualization: A Practical Introduction*. Besplatno dostupno na socviz.co. Poglavlje 1 daje izvrsnu motivaciju zašto je vizualizacija neodvojiva od deskriptivne statistike.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd edition). Klasična referenca za interpretaciju veličine efekata, uključujući smjernice za korelacije.

6.13 Pojmovnik

Pojam	Objašnjenje
Aritmetička sredina (mean)	Zbroj svih vrijednosti podijeljen s brojem opažanja. Koristi sve podatke ali je osjetljiva na ekstremne vrijednosti.
Skraćena sredina (trimmed mean)	Aritmetička sredina izračunata nakon uklanjanja određenog postotka najmanjih i najvećih vrijednosti. Kompromis između sredine i medijana.
Medijan (median)	Srednja vrijednost kad se podaci poredaju po veličini. Robusna mjera centralne tendencije jer ne ovisi o ekstremnim vrijednostima.
Mod (mode)	Vrijednost koja se najčešće pojavljuje. Jedina smislena mjera centralne tendencije za kategoričke podatke.
Raspon (range)	Razlika između najveće i najmanje vrijednosti u podacima. Jednostavna ali nerobusna mjera varijabilnosti.
Prosječno apsolutno odstupanje (AAD)	Prosjek apsolutnih razlika svake vrijednosti od sredine. Intuitivnija ali matematički manje pogodna od varijance.

Pojam	Objašnjenje
Varijanca (variance)	Prosječno kvadrirano odstupanje od aritmetičke sredine (s $N-1$ u nazivniku). Mjeri raspršenost podataka.
Standardna devijacija (SD)	Korijen iz varijance. Izražena u istim mjernim jedinicama kao izvorni podaci.
Besselova korekcija	Dijeljenje s $N-1$ umjesto N pri izračunu varijance uzorka, kako bi procjena populacijske varijance bila nepristrana.
Interkvartilni raspon (IQR)	Razlika između 75. i 25. percentila. Raspon unutar kojeg se nalaze srednjih 50% podataka. Robusna mjera varijabilnosti.
Percentil	Vrijednost ispod koje se nalazi određeni postotak podataka. 50. percentil je medijan.
Asimetrija (skewness)	Mjera simetrije distribucije. Pozitivna asimetrija znači dugačak rep prema desno, negativna prema lijevo.
Zaobljenost (kurtosis)	Mjera težine repova distribucije u usporedbi s normalnom distribucijom.
Standardni rezultat (z-score)	Broj standardnih devijacija za koji se neka vrijednost razlikuje od aritmetičke sredine. Omogućuje usporedbu varijabli na različitim skalama.
Kovarijanca (covariance)	Mjera zajedničkog variranja dviju varijabli. Ovisi o mjernim jedinicama, pa se koristi korelacija kao standardizirana verzija.
Pearsonov koeficijent korelacije (r)	Standardizirana kovarijanca. Mjera linearne povezanosti dviju numeričkih varijabli. Kreće se od minus 1 do plus 1.
Spearmanov koeficijent korelacije (r_s)	Pearsonov koeficijent izračunat na rangovima. Mjeri monotonost veze i robusniji je od Pearsonovog koeficijenta.
MCAR, MAR, MNAR	Tri tipa nedostajućih vrijednosti: potpuno nasumično, nasumično uvjetovano drugim varijablama, te sustavno povezano s nedostajućom vrijednošću.

7 Tjedan 6: Vizualizacija podataka s ggplot2

Od brojeva do priča koje se vide

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti logiku gramatike grafike (grammar of graphics) i zašto je ggplot2 organiziran oko slojeva.
2. Identificirati tri obavezne komponente svakog ggplot2 grafa — podatke, estetike (`aes()`) i geometriju (`geom_*()`) — te razumjeti njihovu ulogu.
3. Kreirati histograme i grafove gustoće za vizualizaciju distribucija jedne kontinuirane varijable.
4. Kreirati stupčaste grafove (`geom_bar()` i `geom_col()`) za prikaz kategoričkih varijabli i njihovih frekvencija ili sažetaka.
5. Kreirati boxplotove i violin grafove za usporedbu distribucija jedne varijable između grupa.
6. Kreirati točkaste grafove (scatterplots) za vizualizaciju odnosa između dviju kontinuiranih varijabli.
7. Koristiti estetike boje, ispune, oblika i veličine za kodiranje dodatnih varijabli u grafu.
8. Prilagoditi oznake osi, naslove i podnaslove pomoću `labs()`.

7.1 Zašto je vizualizacija važna

Prošli tjedan naučili smo izračunati prosjek, medijan, standardnu devijaciju i korelaciju. To su korisni brojevi, ali sami po sebi rijetko govore cijelu priču. Anscombe je 1973. konstruirao četiri dataseta koji imaju identičan prosjek (7.5 za x, identičan za y), identičnu standardnu devijaciju, identičnu korelaciju (0.816) i identičnu regresijsku liniju. A kad ih nacrtate, vidite četiri potpuno različita uzorka. Jedan je linearan, drugi je zakrivljen, treći ima jedan outlier koji povlači liniju, četvrti ima grupirane točke s jednim ekstremom. Bez vizualizacije, sva četiri izgledaju jednako.

Ista logika vrijedi za svaku analizu koju ćete raditi kao komunikolozi. Recimo da vam netko kaže "prosječno vrijeme čitanja članaka na našem portalu je 83 sekunde". Zvuči informativno. Ali to vam ne govori je li distribucija simetrična ili iskrivljena. Možda većina čitatelja provede 30 sekundi, a nekolicina koja čita detaljno diže prosjek. Ili možda postoje dva jasna klastera — oni koji odmah odu (bounce) i oni koji čitaju do kraja. Histogram bi to pokazao u sekundi. Broj sam po sebi ne može.

U ovom tjednu učimo ggplot2, paket za vizualizaciju koji je dio tidyverse ekosustava. ggplot2 nije samo alat za crtanje grafova. On implementira konzistentnu logiku (gramatiku grafike) koja vam omogućuje da razmišljate o vizualizaciji na strukturiran način. Kad jednom shvatite tu logiku, moći ćete kreirati bilo koji graf od istih temeljnih komponenti.

7.2 Naši podaci: angažman čitatelja na portalima

Koristit ćemo simulirani dataset koji sadrži podatke o 1000 članaka objavljenih na hrvatskim informativnim portalima. Za svaki članak imamo informacije o izvoru, kategoriji, stilu naslova, formatu, broju riječi, vremenu provedenom na stranici, broju dijeljenja, komentara, dubini scrollanja i drugim metrikama angažmana.

```
clanci <- read_csv("../resources/datasets/article_engagement.csv")
glimpse(clanci)
```

```
Rows: 1,000
Columns: 16
$ article_id     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ~
$ source         <chr> "Večernji.hr", "Index.hr", "Index.hr", "Index.hr", "24s~
$ category       <chr> "Politika", "Tehnologija", "Politika", "Sport", "Politi~
$ headline_style <chr> "informativni", "senzacionalistički", "narativni", "inf~
$ format         <chr> "tekst+slika", "tekst+slika", "tekst+video", "tekst+sli~
$ word_count     <dbl> 624, 191, 763, 249, 1117, 766, 661, 795, 394, 177, 466,~
$ has_image      <lgl> TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, FALSE,~
$ has_video      <lgl> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
$ publish_hour   <dbl> 20, 0, 14, 7, 11, 19, 8, 22, 22, 20, 18, 16, 4, 16, 7, ~
$ day_of_week    <chr> "ponedjeljak", "subota", "nedjelja", "utorak", "nedjelj~
$ time_on_page   <dbl> 88, 22, 191, 30, 113, 100, 86, 208, 21, 23, 55, 102, 44~
$ shares         <dbl> 0, 1, 7, 0, 11, 9, 0, 2, 1, 0, 0, 1, 1, 0, 1, 5, 10, 3,~
$ comments       <dbl> 0, 5, 5, 0, 14, 3, 0, 1, 0, 0, 0, 0, 1, 1, 3, 7, 1, 0, ~
$ scroll_depth   <dbl> 57, 33, 88, 48, 30, 34, 54, 75, 21, 100, 5, 86, 53, 81,~
$ bounce         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE,~
$ return_visit   <lgl> TRUE, TRUE, TRUE, FALSE, FALSE, TRUE, FALSE, FALSE, FAL~
```

Dataset ima 1000 redova i 16 stupaca. Već na prvi pogled vidimo mješavinu kontinuiranih (word_count, time_on_page, scroll_depth), kategoričkih (source, category, headline_style) i logičkih (has_image, bounce, return_visit) varijabli. Ova raznolikost je idealna za učenje vizualizacije jer svaki tip varijable traži drugačiji tip grafa.

Pogledajmo osnovne karakteristike.

```
clanci |>
  count(source, sort = TRUE)
```

```
# A tibble: 7 x 2
  source          n
  <chr>       <int>
1 Index.hr      254
2 24sata.hr     203
3 Jutarnji.hr   175
4 Večernji.hr   157
5 N1info.hr      92
6 Telegram.hr    69
7 tportal.hr     50
```

```
clanci |>
  count(category, sort = TRUE)
```

```
# A tibble: 7 x 2
  category         n
  <chr>        <int>
1 Politika       224
2 Lifestyle      198
3 Sport          186
4 Tehnologija    138
5 Kultura        118
6 Znanost         82
7 Crna kronika    54
```

```
clanci |>
  summarise(
    prosjek_vrijeme = round(mean(time_on_page), 1),
    prosjek_rijeci = round(mean(word_count), 0),
    prosjek_dijeljenja = round(mean(shares), 1)
  )
```

```
# A tibble: 1 x 3
  prosjek_vrijeme prosjek_rijeci prosjek_dijeljenja
```

```
            <dbl>          <dbl>              <dbl>
1            83.4            511                4.5
```

Sad kad znamo s čime radimo, krenimo graditi grafove.

7.3 Gramatika grafike: kako ggplot2 razmišlja

Paket ggplot2 temelji se na knjizi *The Grammar of Graphics* (Wilkinson, 2005), koja opisuje vizualizaciju podataka kao sustav komponenti koje se slažu u slojeve. Ideja je da svaki graf, koliko god bio složen, nastaje kombinacijom istih temeljnih elemenata.

Tri elementa su obavezna za svaki ggplot2 graf.

Podaci (data) — tibble koji sadrži varijable koje želite prikazati.

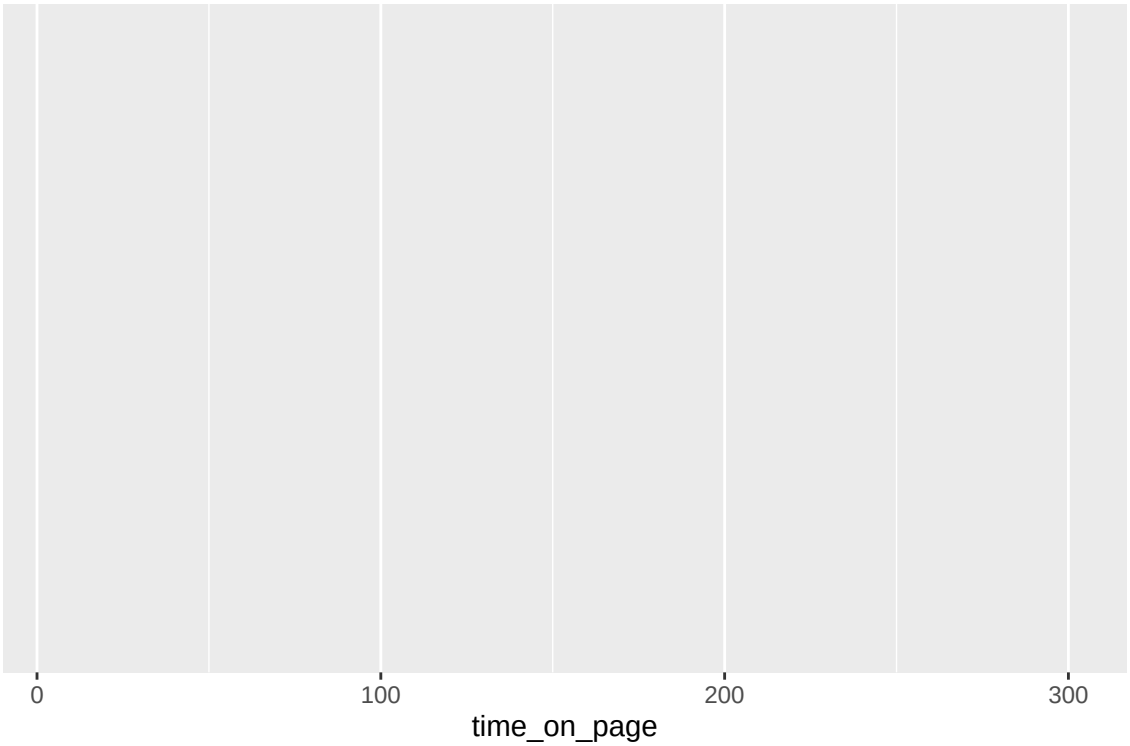
Estetike (aesthetics, `aes()`) — mapiranja varijabli na vizualne dimenzije grafa. Na primjer, varijabla `time_on_page` ide na x os, `shares` na y os, `category` određuje boju.

Geometrija (geometry, `geom_*()`) — oblik kojim se podaci prikazuju, kao što su točke za scatterplot (`geom_point()`), stupci za bar chart (`geom_bar()`) ili linije za linijski graf (`geom_line()`).

Osim ova tri, graf može imati i dodatne slojeve poput statističkih transformacija (stat), prilagodbi skala (scale), podjele u panele (facet), koordinatnog sustava (coord) i vizualne teme (theme). Svaki od ovih elemenata se dodaje operatorom `+`.

Pogledajmo najjednostavniji mogući ggplot2 kod.

```
ggplot(data = clanci, mapping = aes(x = time_on_page))
```

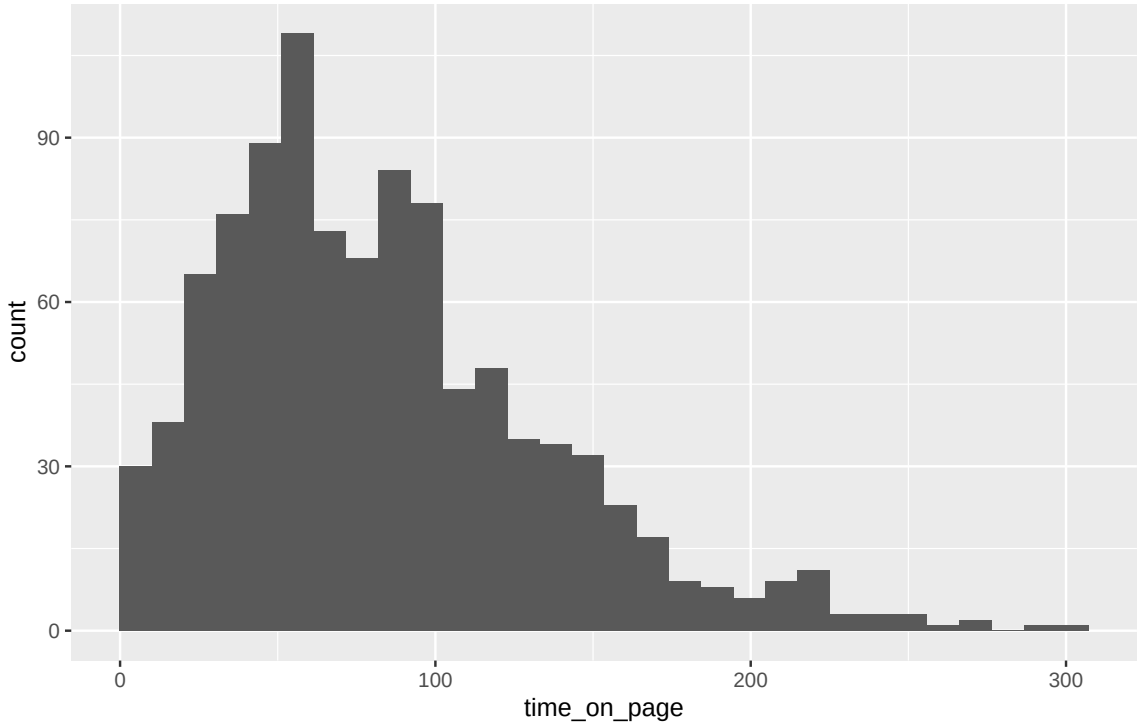


Ovaj kod kreira prazan graf. Definirali smo podatke (`clanci`) i jednu estetiku (varijabla `time_on_page` na x osi), ali nismo rekli ggplotu KAKO da prikaže te podatke (nismo dodali geometriju). Rezultat je prazan koordinatni sustav s ispravno postavljenom x osi. Ggplot zna raspon varijable i pripremio je platno, ali čeka da mu kažemo što da nacrta.

7.4 Histogrami: distribucija jedne varijable

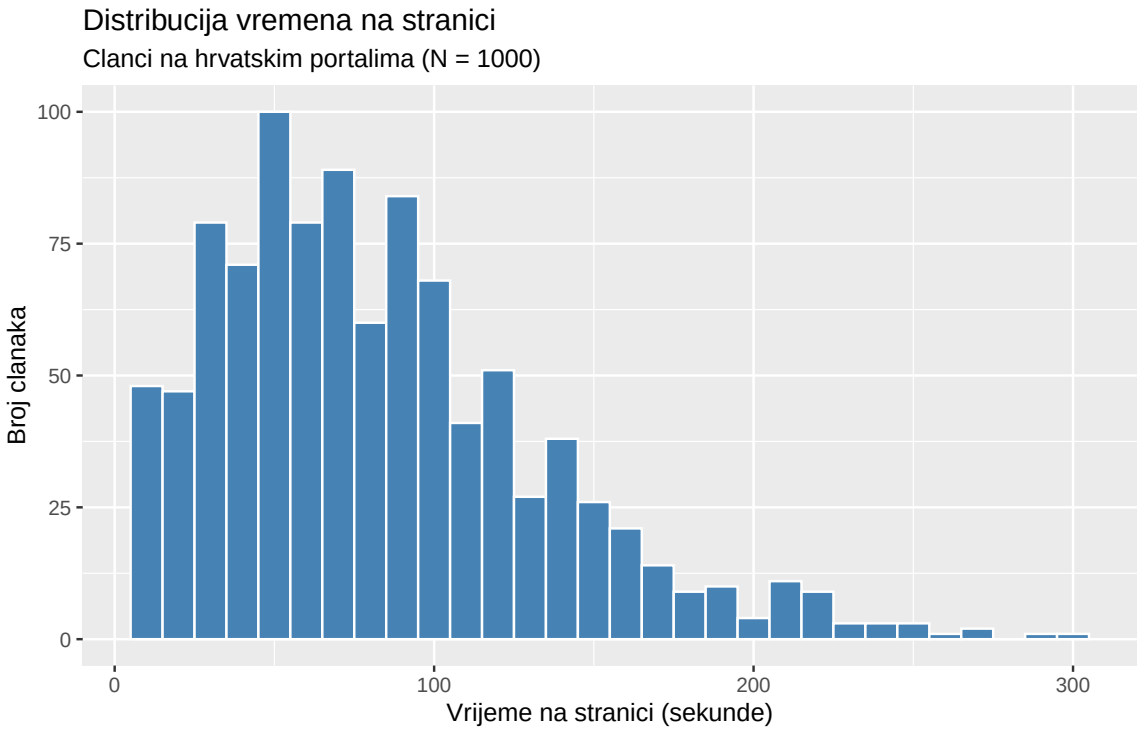
Histogram je najvažniji graf za razumijevanje distribucije jedne kontinuirane varijable. Dijeli raspon vrijednosti u jednake intervale (binove) i prikazuje koliko opažanja pada u svaki interval.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram()
```



Ggplot nas upozorava da koristi 30 binova i sugerira da eksperimentiramo s `bins` ili `binwidth` argumentom. Upozorenje je korisno jer broj binova značajno utječe na to što vidimo. S premalo binova gubimo detalje, s previše binova graf postaje neuredan.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 10, fill = "steelblue", color = "white") +
  labs(
    title = "Distribucija vremena na stranici",
    subtitle = "Članci na hrvatskim portalima (N = 1000)",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



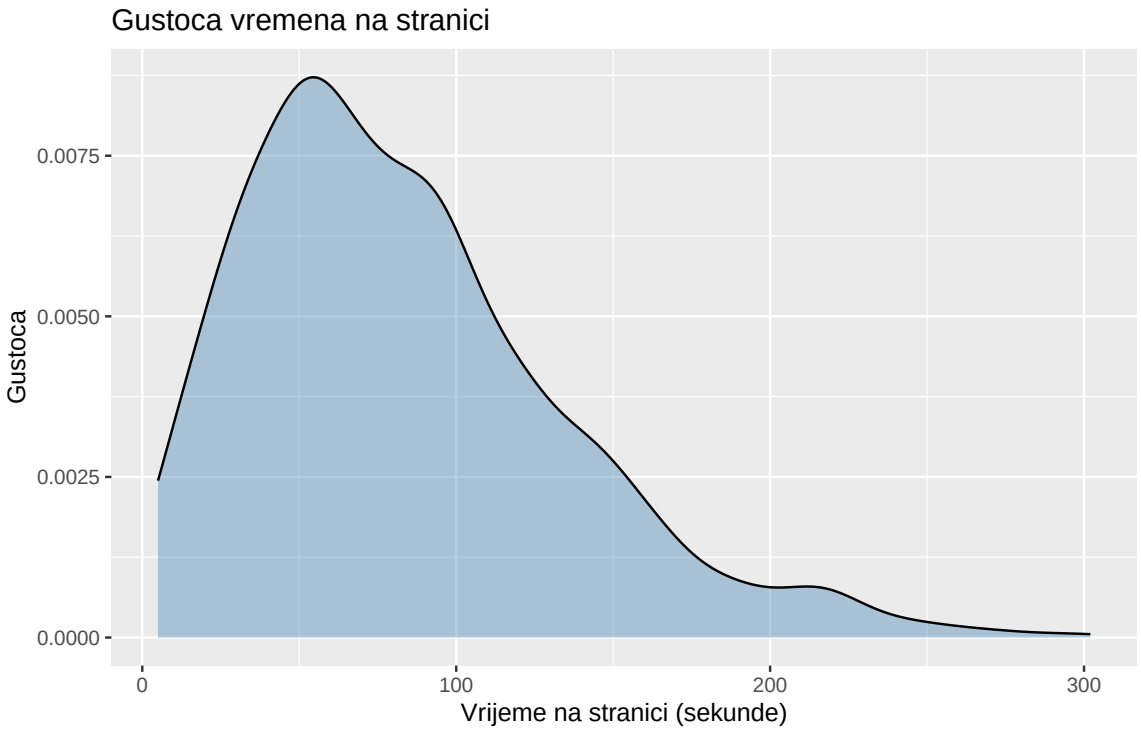
Ovdje smo napravili nekoliko poboljšanja. Argument `binwidth = 10` postavlja širinu svakog bina na 10 sekundi, što daje jasniju sliku od defaultnih 30 binova. `fill = "steelblue"` boja ispunu stupaca, a `color = "white"` crta bijeli rub između stupaca za bolju čitljivost. Funkcija `labs()` dodaje naslov, podnaslov i oznake osi.

Distribucija je desno iskrivljena (pozitivan skew), što je tipično za metriku angažmana. Većina članaka ima relativno kratko vrijeme čitanja, ali postoji dugačak rep članaka s izuzetno dugim vremenom čitanja.

7.4.1 Graf gustoće (density plot)

Alternativa histogramu je graf gustoće koji prikazuje procijenjenu krivulju gustoće vjerojatnosti. Prednost je što ne ovisi o odabiru binova i daje glatku krivulju.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_density(fill = "steelblue", alpha = 0.4) +
  labs(
    title = "Gustoća vremena na stranici",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća"
  )
```

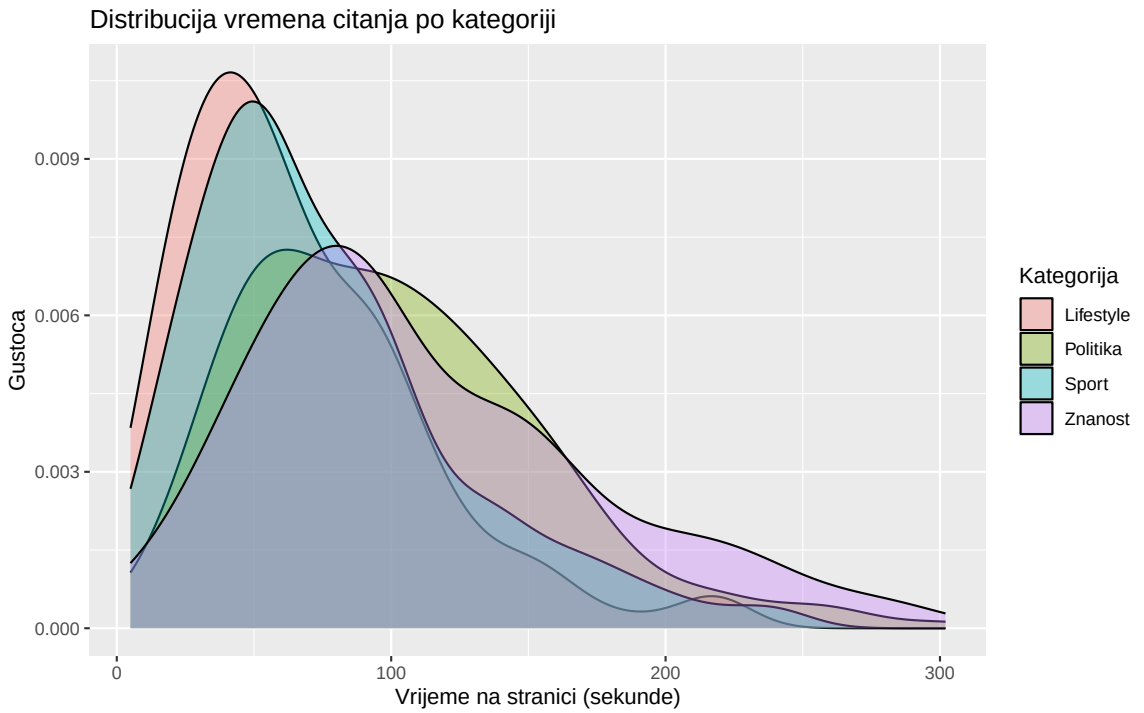


Argument `alpha = 0.4` kontrolira transparentnost ispune (0 je potpuno prozirno, 1 potpuno neprozirno). Transparentnost je osobito korisna kad preklapate više distribucija.

7.4.2 Usporedba distribucija s density plotom

Recimo da želimo usporediti distribuciju vremena čitanja između različitih kategorija članaka. Histogram bi bio nepregledan s pet ili više preklopljenih boja, ali graf gustoće radi dobro.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = time_on_page, fill = category)) +
  geom_density(alpha = 0.35) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća",
    fill = "Kategorija"
  )
```



Primijetite novu estetiku `fill = category` unutar `aes()`, koja govori ggplotu da koristi različitu boju ispune za svaku kategoriju. Kad je estetika mapirana na varijablu (unutar `aes()`), ggplot automatski kreira legendu.

Vidimo da znanstveni članci imaju širu distribuciju pomaknuto udesno (duže čitanje), dok su lifestyle članci koncentrirani na kraćem kraju. Politički članci su negdje između. Ovo ima smisla jer znanstveni članci tendiraju biti duži i zahtijevaju više pozornosti.

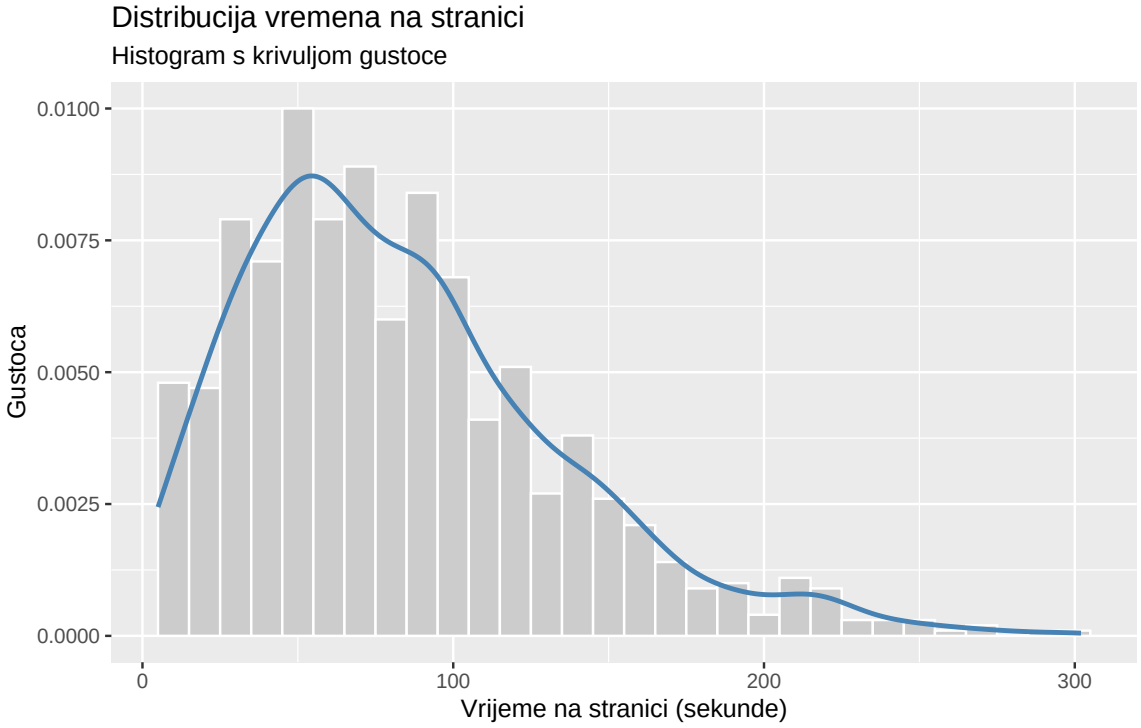
> **Praktični savjet**
>
> Kad prikazujete distribucije više grupa, density plot je gotovo uvijek bolji izbor od preklopljenih histograma. Histogrami se preklapaju i zaklanjaju jedni druge, dok su density krivulje s transparentnošću (`alpha < 0.5`) lako čitljive i za četiri ili pet grupa. Za više od pet grupa, razmislite o facetiranju (koje ćemo naučiti u drugom dijelu).

7.4.3 Histogram i density zajedno

Ponekad je korisno nacrtati oboje na istom grafu. Za to moramo histogramu reći da na y osi prikaže gustoću umjesto broja opažanja, kako bi skale bile usporedive.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(
    aes(y = after_stat(density)),
    binwidth = 10,
```

```
    fill = "grey80",
    color = "white"
  ) +
  geom_density(color = "steelblue", linewidth = 1) +
  labs(
    title = "Distribucija vremena na stranici",
    subtitle = "Histogram s krivuljom gustoće",
    x = "Vrijeme na stranici (sekunde)",
    y = "Gustoća"
  )
```



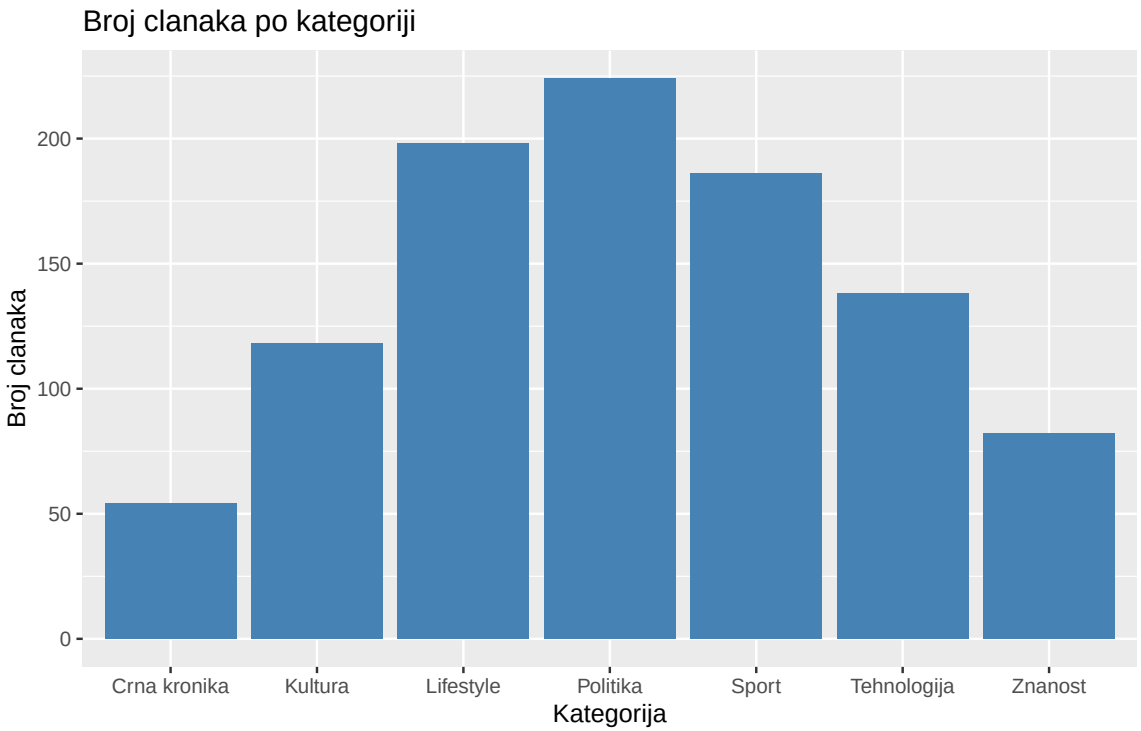
Ključan detalj je `aes(y = after_stat(density))` unutar `geom_histogram()`. Ovo govori ggplotu da na y osi prikaže gustoću (proporciju) umjesto apsolutnog broja, čime histogram i krivulja gustoće postaju usporedivi.

7.5 Stupčasti grafovi: kategoričke varijable

Stupčasti grafovi (bar charts) prikazuju frekvencije ili sažetke za kategoričke varijable. U ggplot2 postoje dvije varijante — `geom_bar()` koja sama broji opažanja i `geom_col()` koja prikazuje unaprijed izračunate vrijednosti.

7.5.1 geom_bar(): automatsko prebrojavanje

```
ggplot(clanci, aes(x = category)) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po kategoriji",
    x = "Kategorija",
    y = "Broj članaka"
  )
```



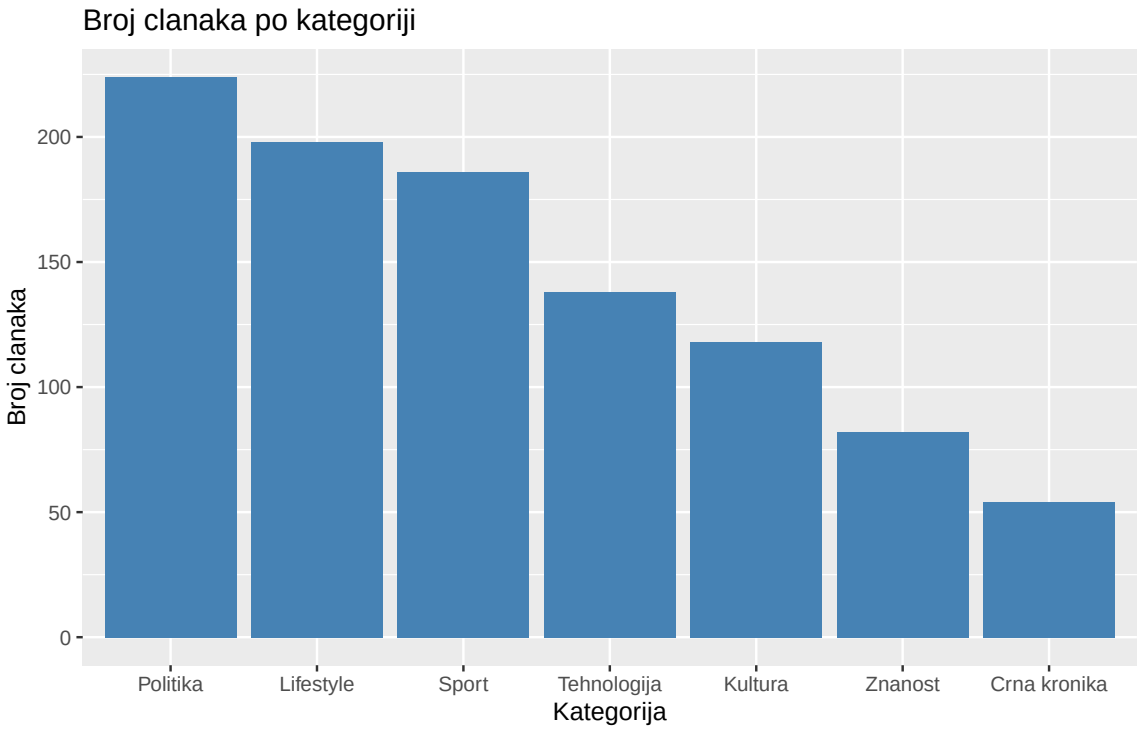
`geom_bar()` automatski broji koliko redova (članaka) pripada svakoj kategoriji i prikazuje rezultat kao stupac. Ovo je ekvivalent pozivanja `count()` na podatke, ali vizualno.

7.5.2 Sortiranje stupaca po veličini

Abecedni redoslijed kategorija rijetko je informativan. Bolje je sortirati stupce po veličini pomoću `fct_infreq()` iz paketa forcats (dio tidyverse).

```
ggplot(clanci, aes(x = fct_infreq(category))) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po kategoriji",
```

```
    x = "Kategorija",
    y = "Broj članaka"
  )
```

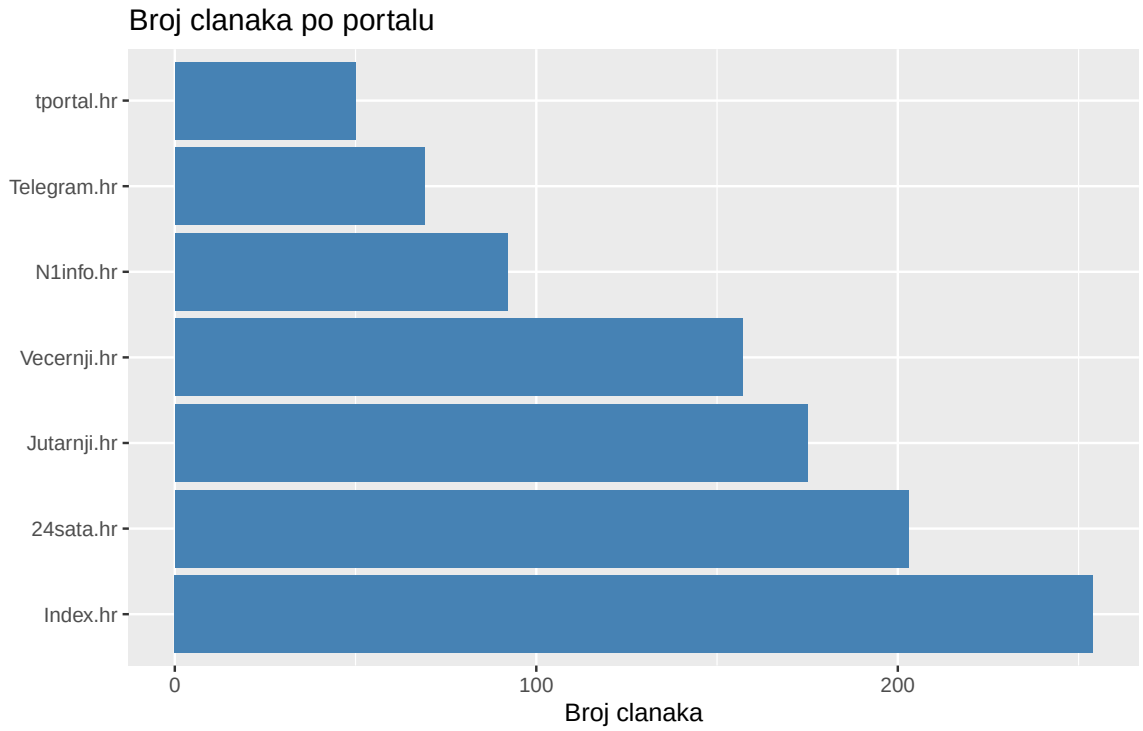


Sad je odmah vidljivo da Politika ima najviše članaka, a Crna kronika najmanje. Funkcija `fct_infreq()` slaže kategorije od najčešće prema najrjeđoj. Za obrnuti redoslijed, omotajte u `fct_rev()` — `fct_rev(fct_infreq(category))`.

7.5.3 Horizontalni stupčasti graf

Za kategorije s dugačkim imenima, horizontalni graf je čitljiviji.

```
ggplot(clanci, aes(y = fct_infreq(source))) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Broj članaka po portalu",
    x = "Broj članaka",
    y = NULL
  )
```

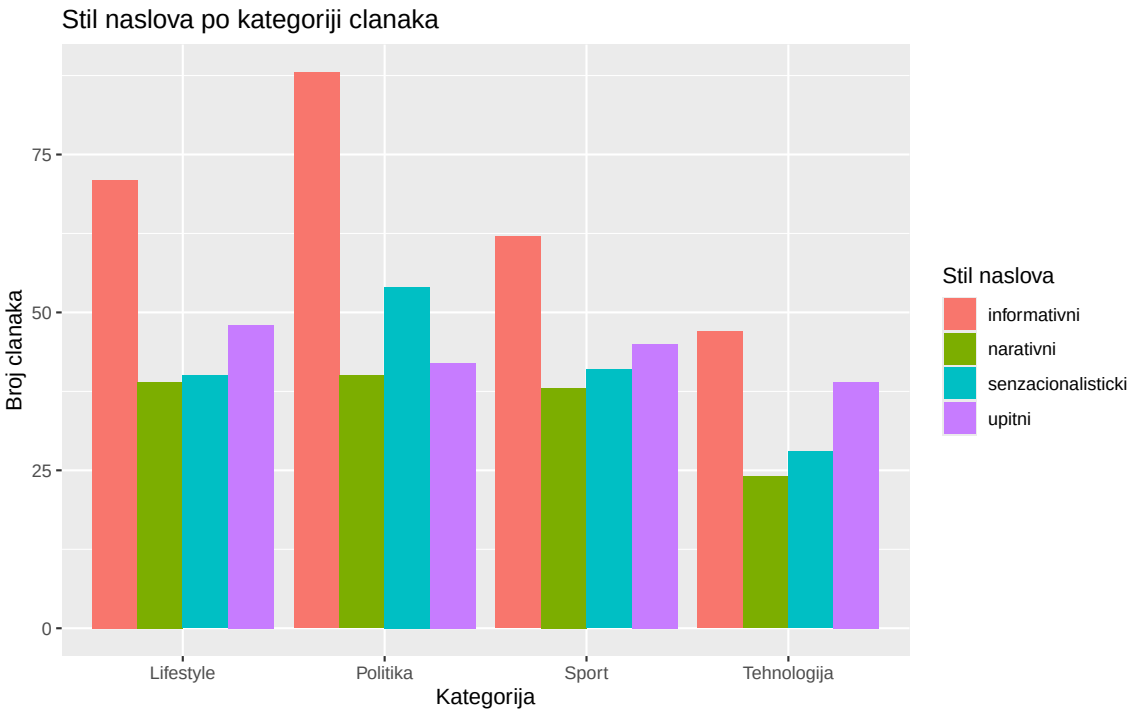


Trik je jednostavan — umjesto `x` koristite `y` u `aes()`, i ggplot automatski crta horizontalne stupce. Postavili smo `y = NULL` u `labs()` da uklonimo nepotrebnu oznaku osi jer su imena portala samorazumljiva.

7.5.4 Grupirani i složeni stupčasti grafovi

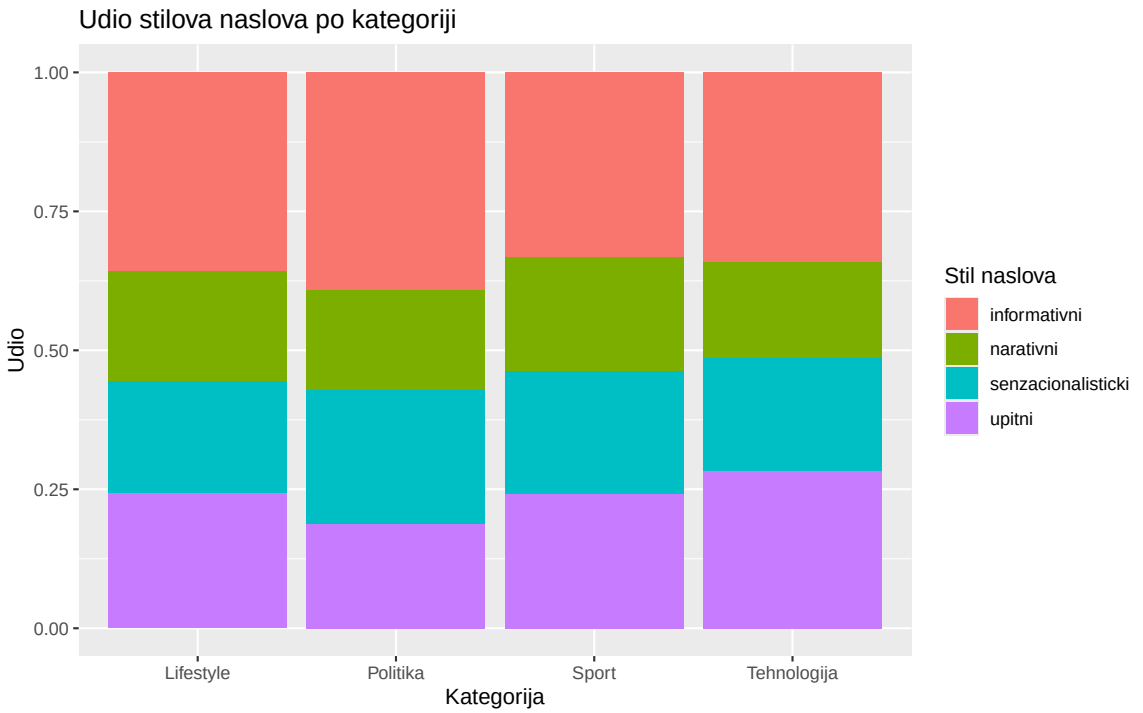
Kad želimo prikazati odnos između dviju kategoričkih varijabli, koristimo boju za drugu varijablu.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Stil naslova po kategoriji članaka",
    x = "Kategorija",
    y = "Broj članaka",
    fill = "Stil naslova"
  )
```



Argument `position = "dodge"` postavlja stupce jedne do drugih umjesto da ih slaže. Alternativa je `position = "fill"` koji prikazuje proporcije umjesto apsolutnih brojeva.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = category, fill = headline_style)) +
  geom_bar(position = "fill") +
  labs(
    title = "Udio stilova naslova po kategoriji",
    x = "Kategorija",
    y = "Udio",
    fill = "Stil naslova"
  )
```

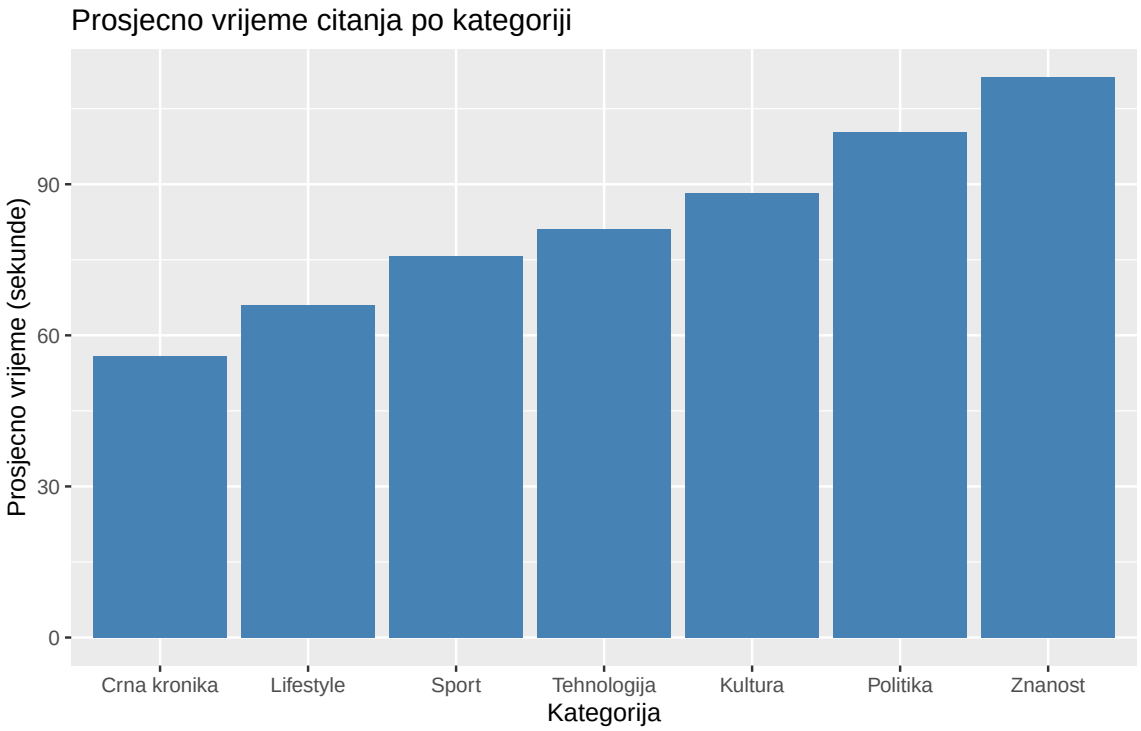


Ovaj graf otkriva zanimljive obrasce. Proporcija senzacionalističkih naslova se razlikuje po kategorijama. Ovo je vizualna verzija tablice unakrsnih frekvencija (contingency table) i koristit ćete ju kad budemo radili hi-kvadrat testove u tjednu 11.

7.5.5 geom_col(): vlastiti sažeci

Kad ste već izračunali sažetke (prosjeke, medijane, postotke) pomoću `summarise()`, koristite `geom_col()` koji očekuje gotove y vrijednosti.

```
clanci |>
  group_by(category) |>
  summarise(prosjek_vrijeme = mean(time_on_page), .groups = "drop") |>
  mutate(category = fct_reorder(category, prosjek_vrijeme)) |>
  ggplot(aes(x = category, y = prosjek_vrijeme)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Prosječno vrijeme čitanja po kategoriji",
    x = "Kategorija",
    y = "Prosječno vrijeme (sekunde)"
  )
```



Ovdje smo najprije izračunali prosjeke, zatim koristili `fct_reorder()` da sortiramo kategorije po prosječnom vremenu (ne po frekvenciji kao `fct_infreq()`), i onda prikazali te prosjeke s `geom_col()`.

Razlika između `geom_bar()` i `geom_col()` je ključna. `geom_bar()` sam broji retke i ne treba y estetiku. `geom_col()` prikazuje y vrijednosti koje ste vi pripremili. Koristite `geom_bar()` za frekvencije, `geom_col()` za sve ostalo (prosjeke, medijane, postotke, bilo kakve prethodno izračunate sažetke).

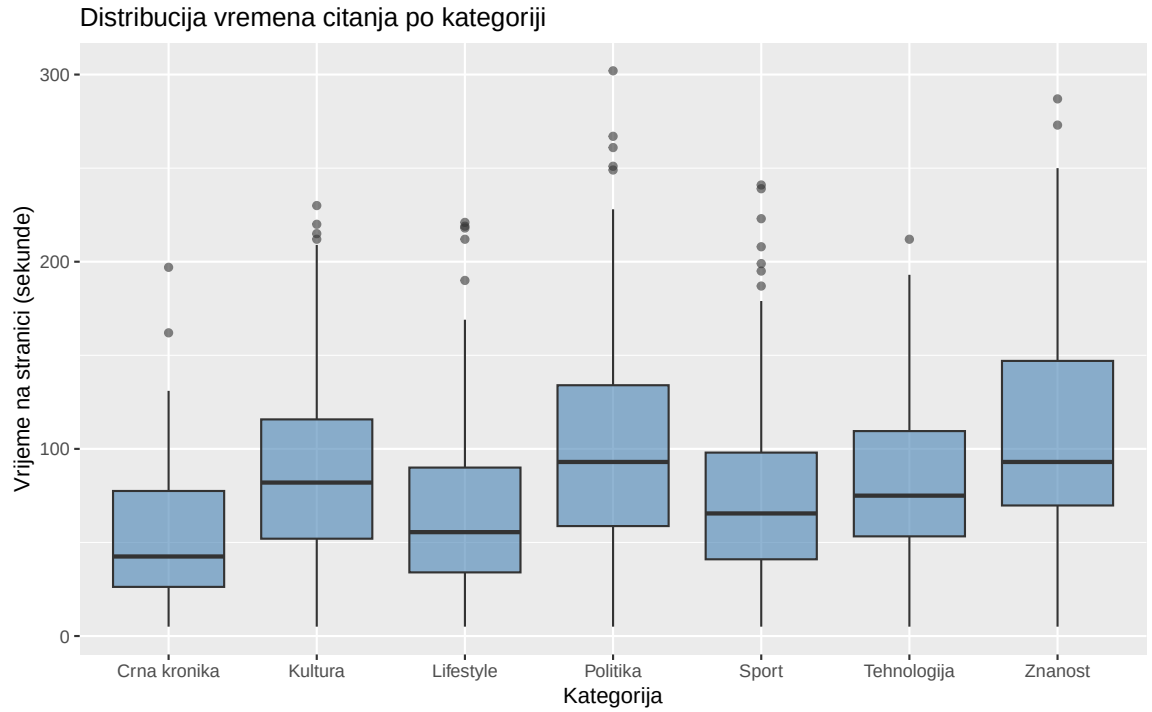
> **!** Važna napomena
>
> Stupčasti graf prosjeka skriva distribuciju podataka. Kad vidite samo stupac visine 83, ne znate je li to zato što su svi oko 83 ili zato što su pola ljudi na 10 i pola na 156. Za usporedbu distribucija između grupa, boxplot ili violin plot su gotovo uvijek bolji izbor. Stupčaste grafove prosjeka koristite samo kad je publici dovoljna informacija o prosjecima (na primjer, u izvještaju za klijenta koji ne želi vidjeti boxplotove).

7.6 Boxplot: usporedba distribucija između grupa

Boxplot (dijagram pravokutnika) prikazuje pet ključnih brojeva distribucije — minimum, prvi kvartil (Q1), medijan, treći kvartil (Q3) i maksimum. Također identificira potencijalne

outliere. Za usporedbu distribucija između grupa, boxplot je jedan od najkorisnijih grafova.

```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```



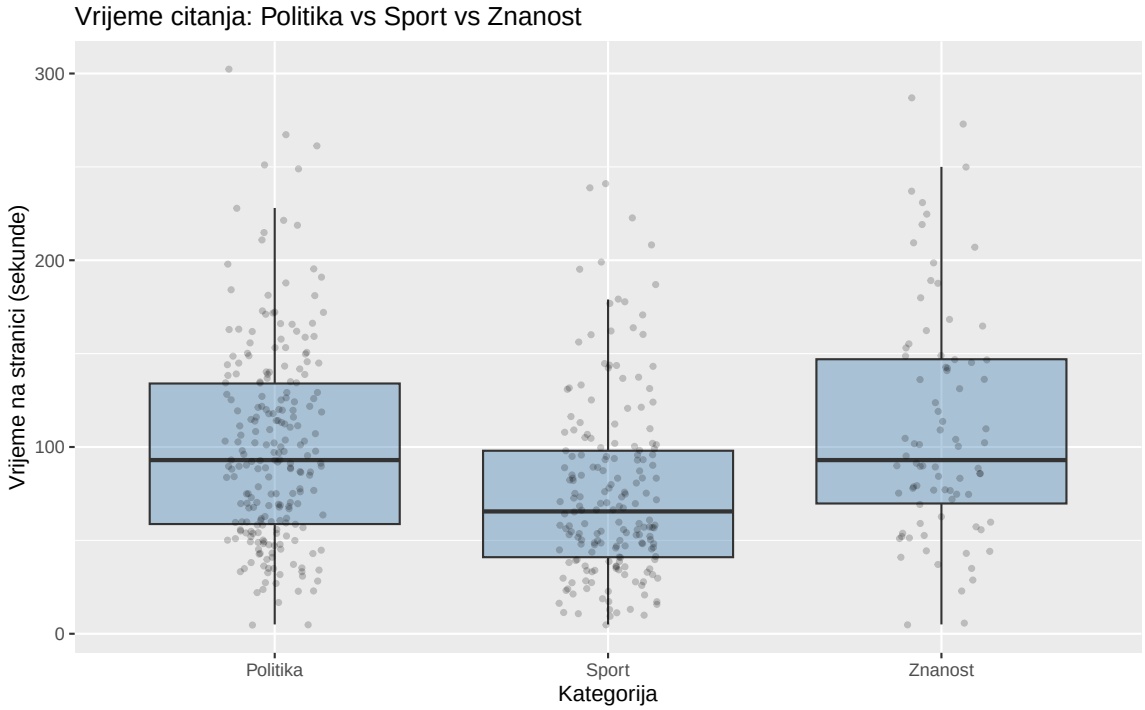
Čitanje boxplota ide na sljedeći način. Deblja crta unutar pravokutnika je medijan, donji rub pravokutnika je Q1 (25. percentil), a gornji rub je Q3 (75. percentil). Visina pravokutnika je interkvartilni raspon (IQR = Q3 − Q1), koji obuhvaća srednjih 50% podataka. Linije (whiskers) se protežu do najudaljenije točke koja je unutar 1.5 × IQR od ruba pravokutnika, dok su točke izvan toga potencijalni outlieri.

Iz ovog grafa jasno vidimo da znanstveni članci imaju ne samo viši medijan vremena čitanja nego i veću varijabilnost. Lifestyle članci su koncentrirani na nižim vrijednostima. Svaka kategorija ima outliere na desnoj strani, što je očekivano za metriku angažmana.

7.6.1 Boxplot s točkama

Boxplot sažima distribuciju u pet brojeva, pa neke informacije gubi. Dodavanje pojedinačnih točaka vraća taj kontekst.

```r
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost")) |>
  ggplot(aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.4, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.2, size = 1) +
  labs(
    title = "Vrijeme čitanja: Politika vs Sport vs Znanost",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```



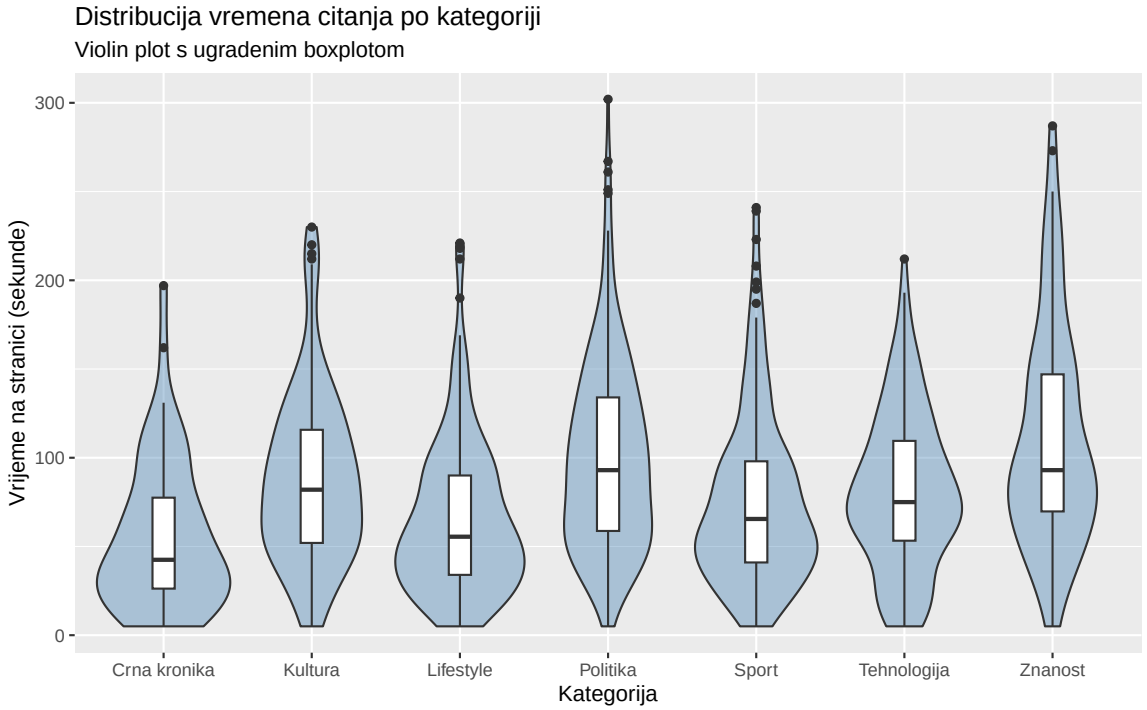
`geom_jitter()` dodaje točke s malim nasumičnim pomakom po horizontali (`width = 0.15`) da se ne preklapaju, a `alpha = 0.2` čini točke poluprozirnim kako bismo vidjeli gustoću. `outlier.shape = NA` u boxplotu isključuje prikaz outliera jer bi se inače udvostručili s jitter točkama.

Ovaj kombinirani prikaz daje kompletnu sliku: boxplot za sažetak distribucije i točke za uvid u stvarne podatke.

7.6.2 Violin plot: oblik distribucije

Violin plot je varijanta boxplota koja prikazuje oblik distribucije pomoću zrcaljene krivulje gustoće. Tamo gdje je graf širi, ima više podataka.

```r
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_violin(fill = "steelblue", alpha = 0.4) +
  geom_boxplot(width = 0.15, fill = "white") +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    subtitle = "Violin plot s ugrađenim boxplotom",
    x = "Kategorija",
    y = "Vrijeme na stranici (sekunde)"
  )
```

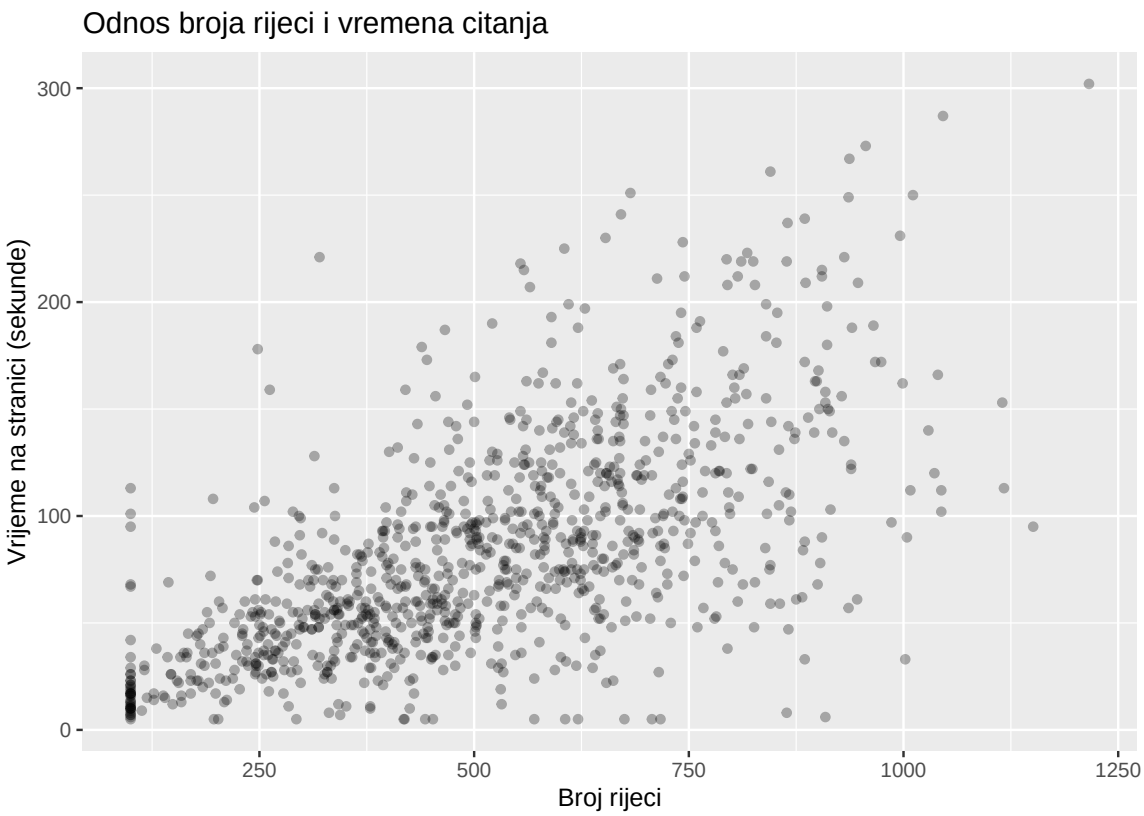


Kombinacija violin plota (za oblik distribucije) i uskog boxplota (za medijan i kvartile) daje bogat prikaz koji je ujedno i informativan i vizualno privlačan.

7.7 Točkasti grafovi (scatterplots): odnos dviju varijabli

Scatterplot je temeljni graf za vizualizaciju odnosa (korelacije) između dviju kontinuiranih varijabli. Svaka točka predstavlja jedno opažanje, s jednom varijablom na x osi i drugom na y osi.

```r
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.3) +
  labs(
    title = "Odnos broja riječi i vremena čitanja",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



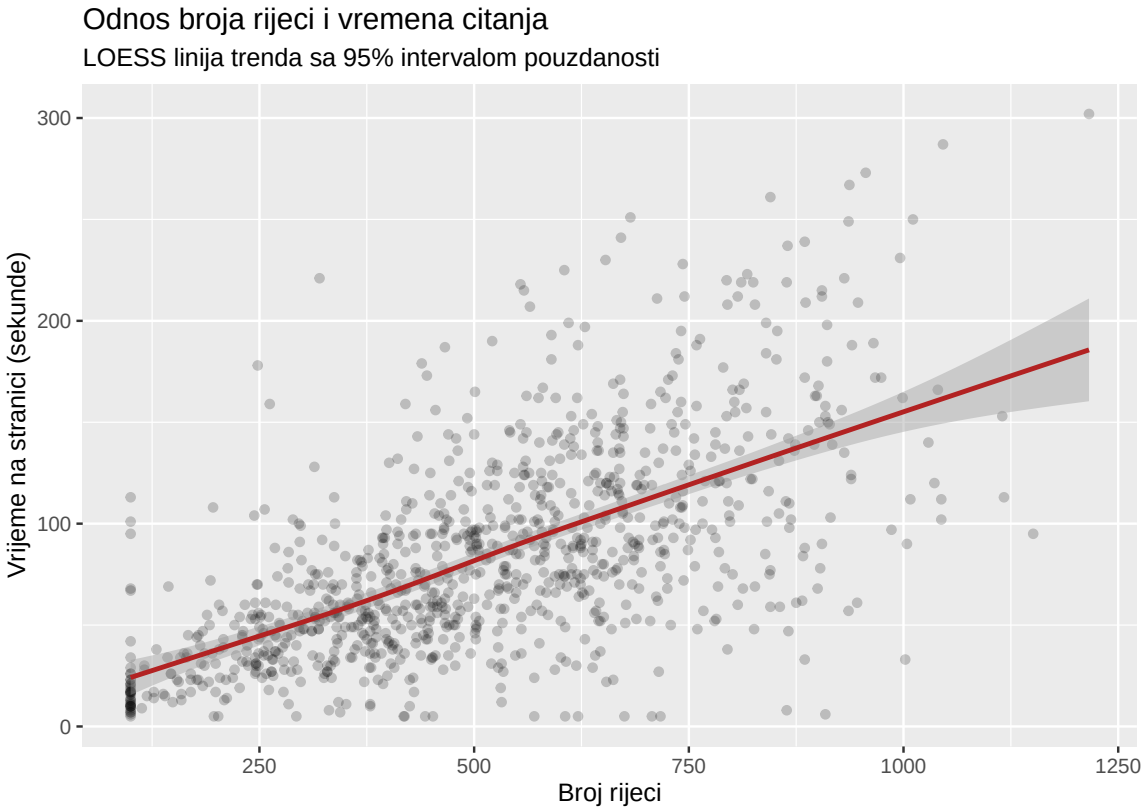
Vidimo pozitivan trend — članci s više riječi tendiraju imati duže vrijeme čitanja. Ali odnos nije savršen i postoji značajna varijabilnost. `alpha = 0.3` je bitan jer s 1000 točaka bi se bez transparentnosti mnoge preklapale i graf bi bio nečitljiv.

7.7.1 Dodavanje linije trenda

Funkcija `geom_smooth()` dodaje statističku liniju trenda na scatterplot. Po defaultu koristi LOESS (lokalno ponderiranu regresiju) koja je fleksibilna i prati oblik podataka.

```r
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(color = "firebrick", linewidth = 1) +
  labs(
```

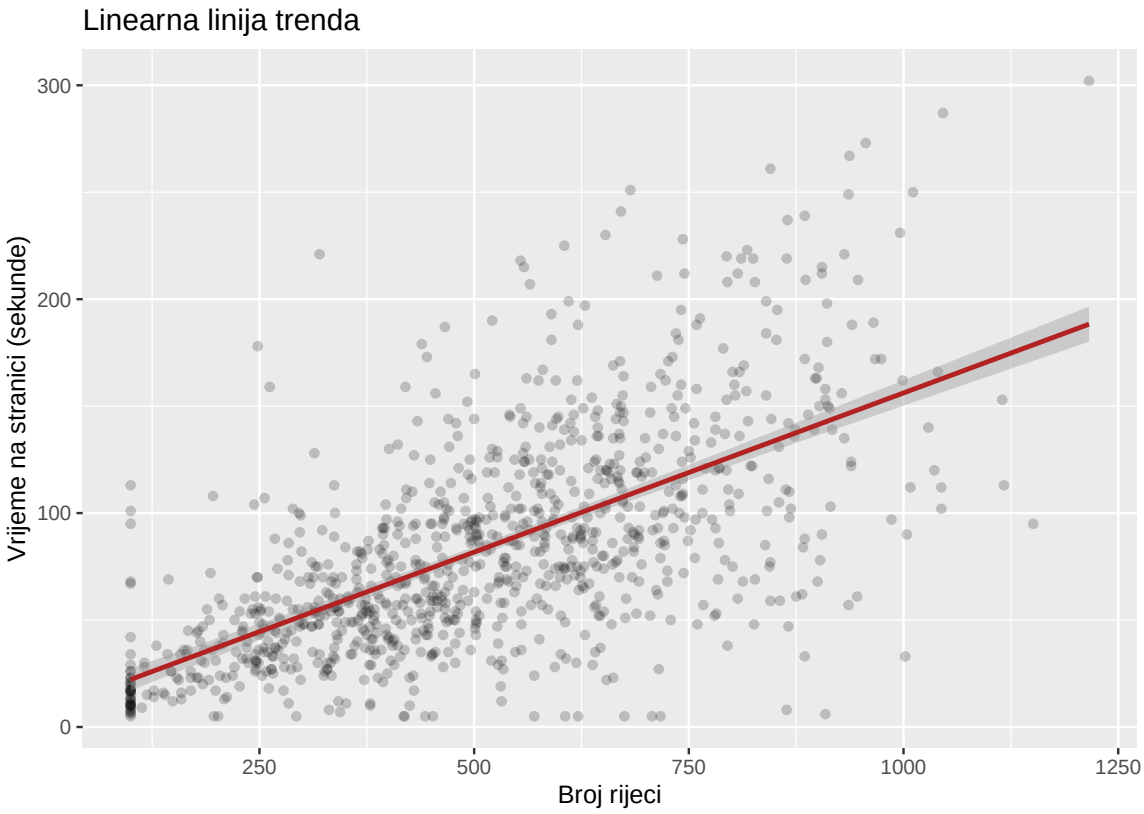
```
    title = "Odnos broja riječi i vremena čitanja",
    subtitle = "LOESS linija trenda sa 95% intervalom pouzdanosti",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



Sivi pojas oko linije je 95% interval pouzdanosti za procjenu trenda. Što je pojas uži, to smo sigurniji u procjenu. Na krajevima distribucije (malo i mnogo riječi) pojas je širi jer imamo manje podataka.

Za linearnu liniju trenda (onu koju smo računali pri radu na korelaciji u tjednu 4), koristite `method = "lm"`.

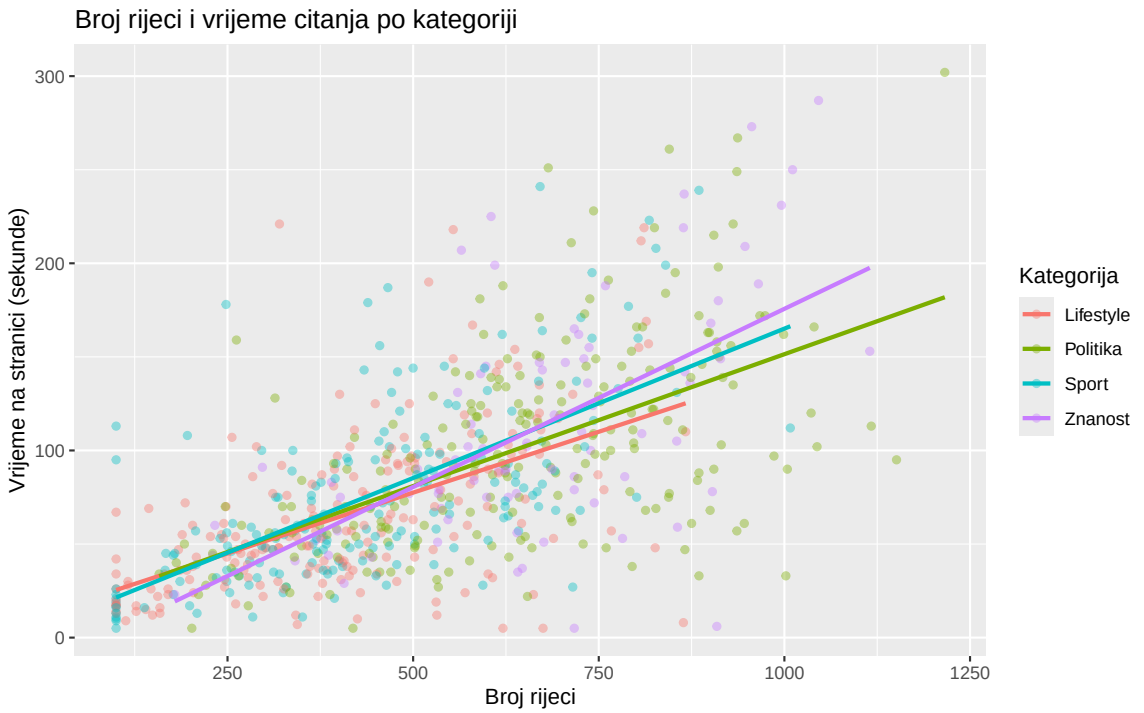
```
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick", linewidth = 1) +
  labs(
    title = "Linearna linija trenda",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



7.7.2 Kodiranje treće varijable bojom

Dodavanjem treće varijable kao estetike boje, scatterplot može prikazati tri dimenzije podataka na dvodimenzionalnom grafu.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Broj riječi i vrijeme čitanja po kategoriji",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)",
    color = "Kategorija"
  )
```



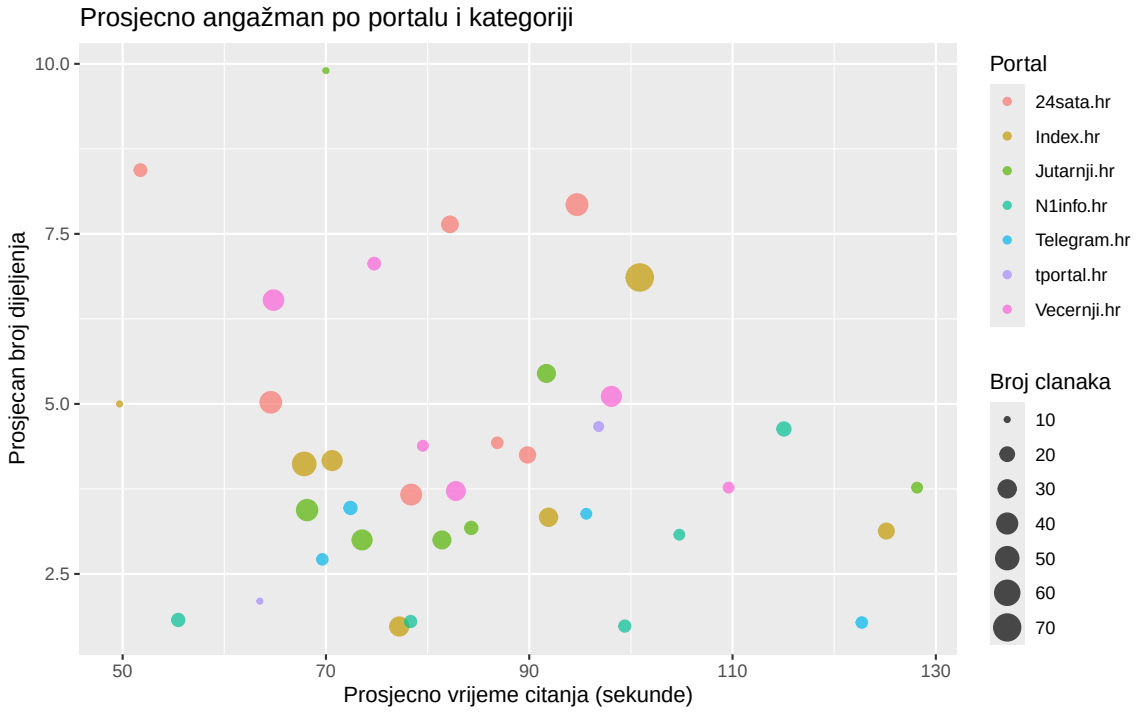
Argument `se = FALSE` uklanja interval pouzdanosti da graf ne bude pretrpan. Sada vidimo da je pozitivan odnos između broja riječi i vremena čitanja prisutan u svim kategorijama. Međutim, kategorije se razlikuju po razini (intercept) — za isti broj riječi, znanstveni članci imaju duže prosječno čitanje od lifestyle članaka.

7.7.3 Kodiranje veličine i oblika

Osim boje, ggplot2 nudi i druge estetike za kodiranje varijabli.

```
clanci |>
  group_by(source, category) |>
  summarise(
    prosjek_vrijeme = mean(time_on_page),
    prosjek_dijeljenja = mean(shares),
    n = n(),
    .groups = "drop"
  ) |>
  filter(n >= 10) |>
  ggplot(aes(x = prosjek_vrijeme, y = prosjek_dijeljenja,
             color = source, size = n)) +
  geom_point(alpha = 0.7) +
  labs(
    title = "Prosječno angažman po portalu i kategoriji",
    x = "Prosječno vrijeme čitanja (sekunde)",
```

```
    y = "Prosječan broj dijeljenja",
    color = "Portal",
    size = "Broj članaka"
  )
```



Ovdje smo najprije izračunali sažetke po kombinaciji portala i kategorije, a onda veličinu točke mapirali na broj članaka. Veće točke predstavljaju kombinacije s više članaka (i stoga pouzdanijim prosjekom). Ovaj tip grafa se naziva bubble chart i koristan je za prikaz tri ili četiri dimenzije podataka istovremeno.

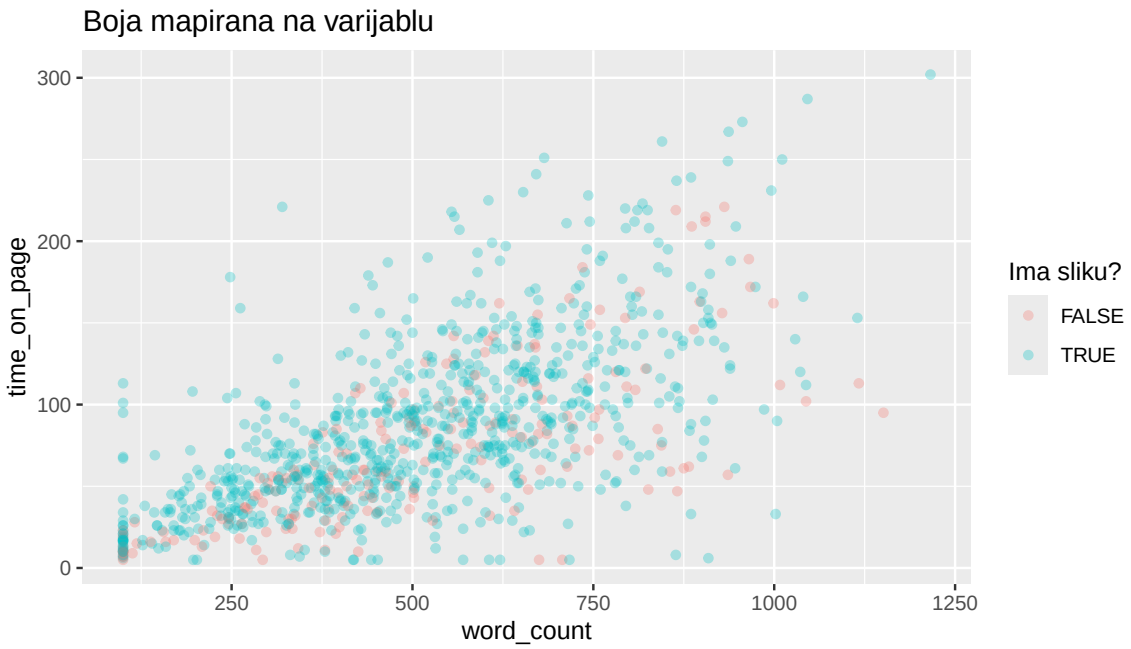
7.8 Estetike unutar i izvan aes()

Česta zbunjenica za početnike je razlika između estetika unutar i izvan `aes()`. Ovo je konceptualno važno razumjeti.

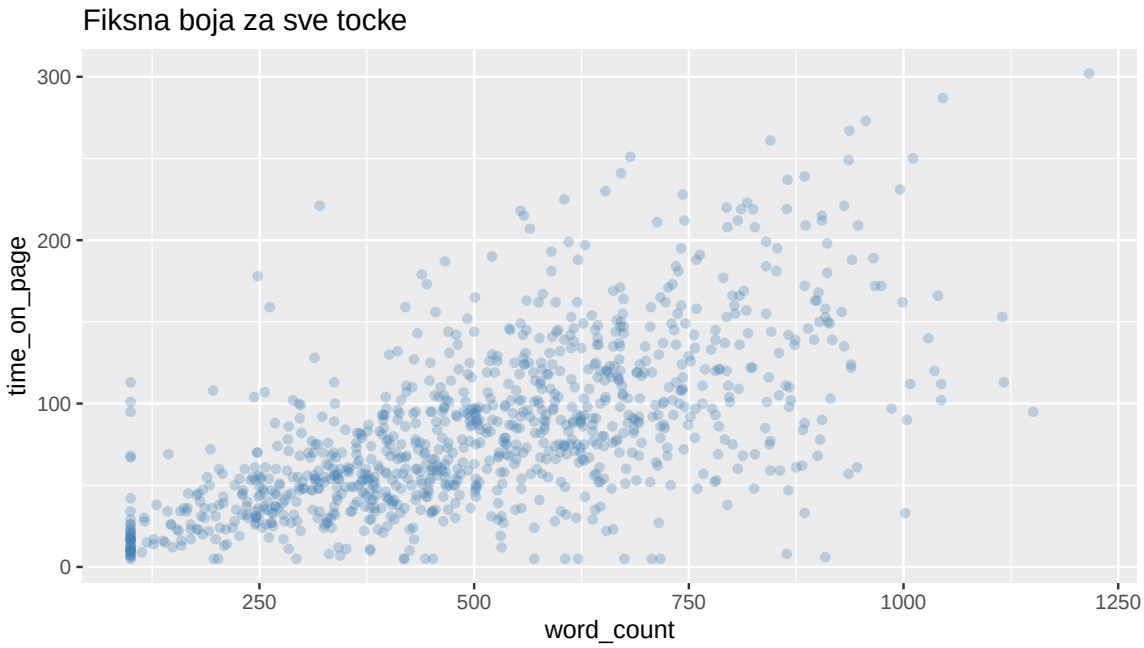
Kad stavite estetiku **unutar** `aes()`, mapirate varijablu na vizualno svojstvo. Boja ovisi o podatcima i ggplot automatski kreira legendu.

Kad stavite estetiku **izvan** `aes()`, postavljate fiksnu vrijednost za sve točke. Nema legende jer boja ne ovisi o podatcima.

```r
# UNUTAR aes(): boja ovisi o varijabli
p1 <- ggplot(clanci, aes(x = word_count, y = time_on_page, color = has_image)) +
  geom_point(alpha = 0.3) +
  labs(title = "Boja mapirana na varijablu", color = "Ima sliku?")

# IZVAN aes(): boja je fiksna za sve točke
p2 <- ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  labs(title = "Fiksna boja za sve točke")

p1
```



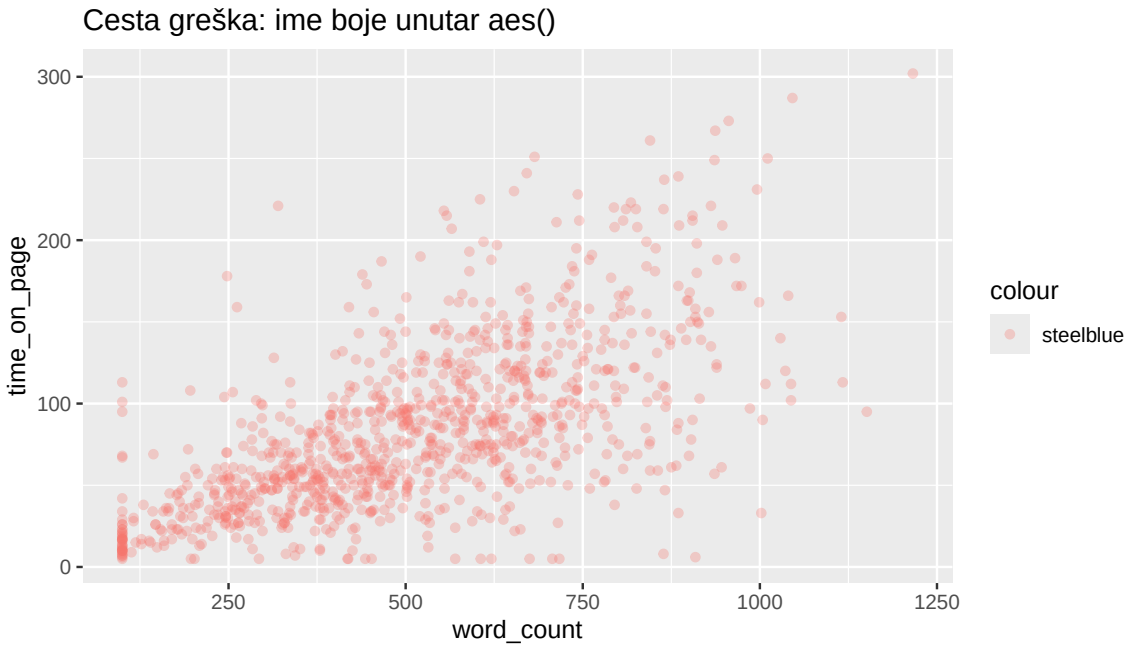

```r
p2
```



Ova razlika se proteže na sve estetike — `fill`, `color`, `size`, `shape`, `alpha`, `linewidth` — gdje varijabilne estetike idu unutar `aes()`, a fiksne vrijednosti idu izvan.

Česta greška je staviti ime boje unutar `aes()`:

```
# KRIVO: "steelblue" se tretira kao kategorija, ne kao boja
ggplot(clanci, aes(x = word_count, y = time_on_page, color = "steelblue")) +
  geom_point(alpha = 0.3) +
  labs(title = "Česta greška: ime boje unutar aes()")
```

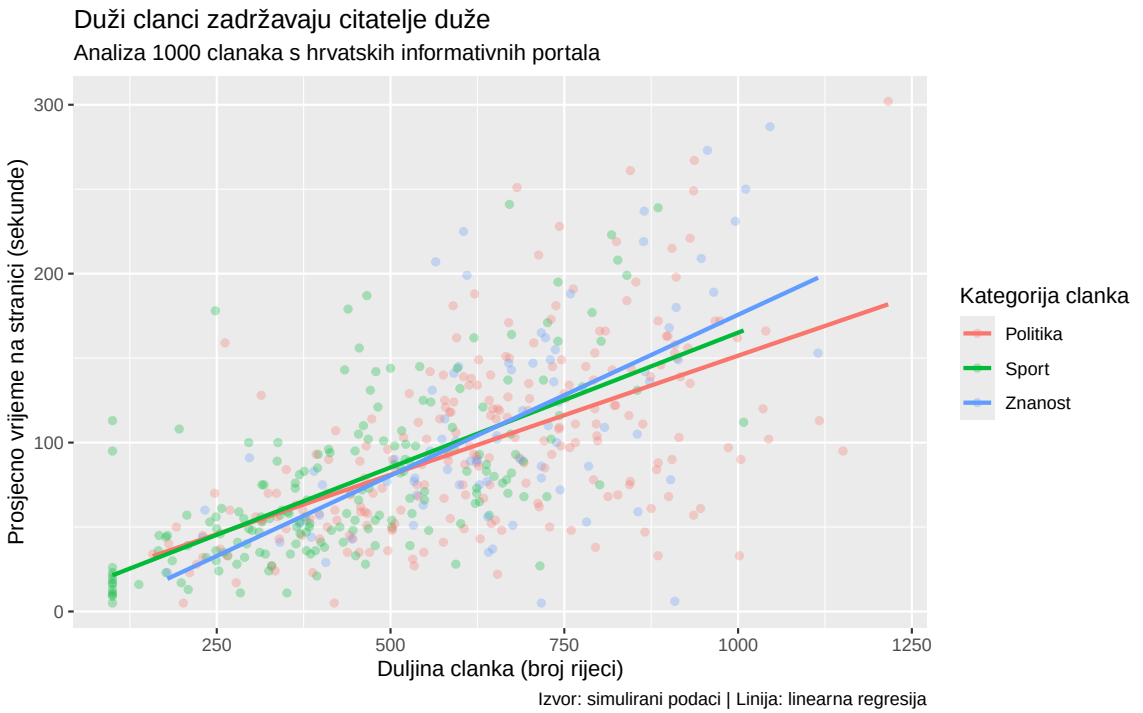


Ggplot interpretira "steelblue" kao tekstualnu varijablu s jednom kategorijom i dodjeljuje joj svoju paletu (obično crvenu) — rezultat je legenda s jednom stavkom "steelblue" obojana u boju koju ggplot sam odabere. To je jedan od najčešćih bugova kod početnaka.

7.9 labs(): naslovi, oznake i natpisi

Svaki graf koji dijete s drugima mora imati jasne oznake. Funkcija `labs()` kontrolira naslove, podnaslove, oznake osi i legende.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Duži članci zadržavaju čitatelje duže",
    subtitle = "Analiza 1000 članaka s hrvatskih informativnih portala",
    x = "Duljina članka (broj riječi)",
    y = "Prosječno vrijeme na stranici (sekunde)",
    color = "Kategorija članka",
    caption = "Izvor: simulirani podaci | Linija: linearna regresija"
  )
```



Primijetite da je naslov formuliran kao nalaz ("Duži članci zadržavaju čitatelje duže"), ne kao opis ("Odnos duljine članka i vremena čitanja"). Ovo je najbolja praksa za vizualizaciju u novinarstvu i izvještajima jer čitatelju odmah komunicira ključnu poruku. Za akademske radove, opisni naslovi su prihvatljiviji.

Argument `caption` dodaje tekst u donji desni kut grafa — koristite ga za izvor podataka ili metodološke napomene.

7.10 Brzi pregled: koji graf za koji podatak?

Do sada smo naučili četiri tipa grafova. Evo sažetka kada koristiti koji.

Histogram / density plot prikazuje distribuciju jedne kontinuirane varijable — koristite ga kad želite vidjeti oblik distribucije, identificirati outliere, provjeriti normalnost ili usporediti distribucije između grupa (npr. distribucija vremena čitanja članaka).

Stupčasti graf (bar chart) prikazuje frekvencije ili sažetke kategoričkih varijabli — koristite `geom_bar()` za automatsko prebrojavanje i `geom_col()` za prethodno izračunate sažetke (npr. broj članaka po kategoriji ili portalu).

Boxplot / violin plot uspoređuje distribucije jedne kontinuirane varijable između grupa — posebno je koristan za identifikaciju razlika u medijanima, varijabilnosti i outlierima (npr. usporedba vremena čitanja između kategorija članaka).

Scatterplot prikazuje odnos (korelaciju) između dviju kontinuiranih varijabli — dodajte `geom_smooth()` za liniju trenda i koristite boju/veličinu za kodiranje dodatnih varijabli (npr. odnos broja riječi i vremena čitanja).

```
# Praktična provjera: koje varijable imamo i što s njima prikazati?
tribble(
  ~varijable, ~tip_grafa, ~geom,
  "1 kontinuirana", "Histogram / density", "geom_histogram() / geom_density()",
  "1 kategorička", "Bar chart", "geom_bar()",
  "1 kontinuirana + 1 kategorička", "Boxplot / violin", "geom_boxplot() / geom_violin()",
  "2 kontinuirane", "Scatterplot", "geom_point() + geom_smooth()",
  "2 kategoričke", "Grupirani bar chart", "geom_bar(position = 'dodge'/'fill')"
)
```

```
# A tibble: 5 x 3
  varijable                     tip_grafa           geom
  <chr>                         <chr>               <chr>
1 1 kontinuirana                Histogram / density geom_histogram() / geom_de~
2 1 kategorička                 Bar chart           geom_bar()
3 1 kontinuirana + 1 kategorička Boxplot / violin   geom_boxplot() / geom_viol~
4 2 kontinuirane                Scatterplot         geom_point() + geom_smooth~
5 2 kategoričke                 Grupirani bar chart geom_bar(position = 'dodge~
```

Ova tablica je vaš vodič za odabir grafa. Prije nego nacrtate bilo što, postavite si pitanje — koje varijable imam i kakve su (kontinuirane ili kategoričke)? Odgovor vas automatski vodi do pravog tipa grafa.

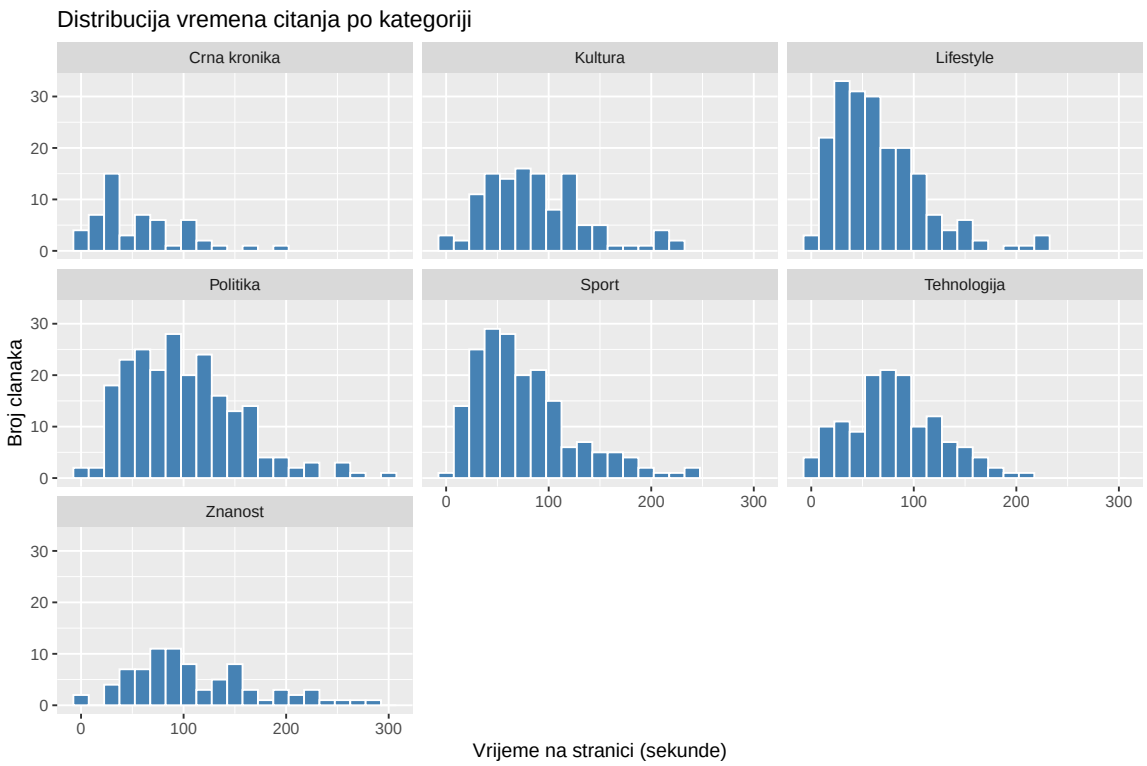
> **Podsjetnik**
>
> U prvom dijelu naučili smo četiri temeljna tipa grafova — histogram/density za distribucije, bar chart za kategorije, boxplot/violin za usporedbu grupa i scatterplot za odnos dviju varijabli. U ovom dijelu učimo kako grafove učiniti profesionalnima i prezentabilnima.

7.11 Facetiranje: mali višestruki grafovi

Facetiranje je jedna od najmoćnijih značajki ggplot2. Umjesto da sve grupe trpate u jedan graf s više boja, facetiranje dijeli graf na zasebne panele, po jedan za svaku grupu. Rezultat je čitljiviji jer svaki panel ima vlastite osi i nije zatrpan preklapajućim elementima.

7.11.1 facet_wrap(): paneli u jednom retku ili mreži

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```

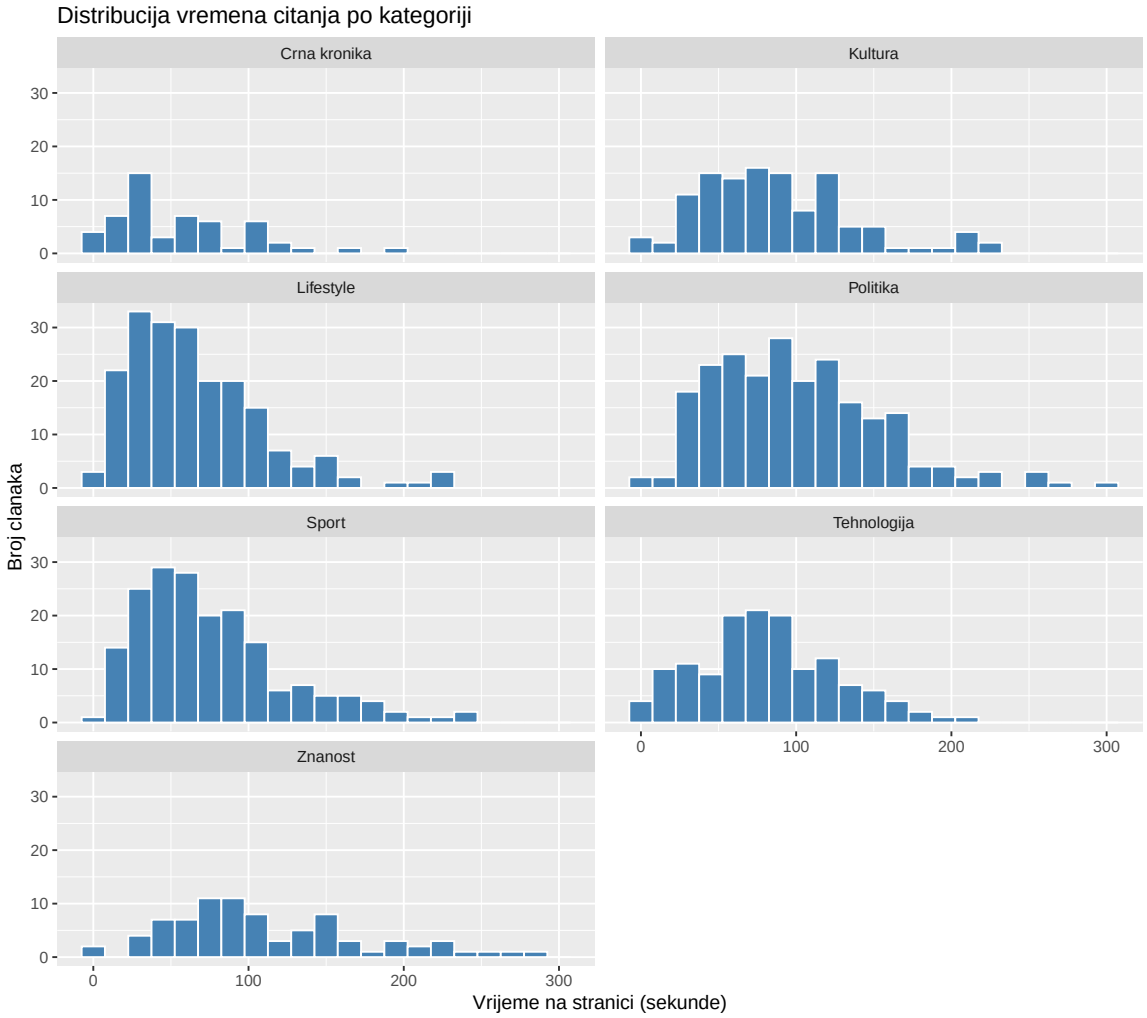


Sintaksa `facet_wrap(~category)` govori ggplotu da napravi zaseban panel za svaku razinu varijable `category`. Tilda (`~`) je obavezna i čita se kao "po", što znači podijeli po kategoriji. Paneli se automatski slažu u mrežu.

Argument `ncol` kontrolira broj stupaca u mreži.

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category, ncol = 2) +
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    x = "Vrijeme na stranici (sekunde)",
```

```
    y = "Broj članaka"
  )
```



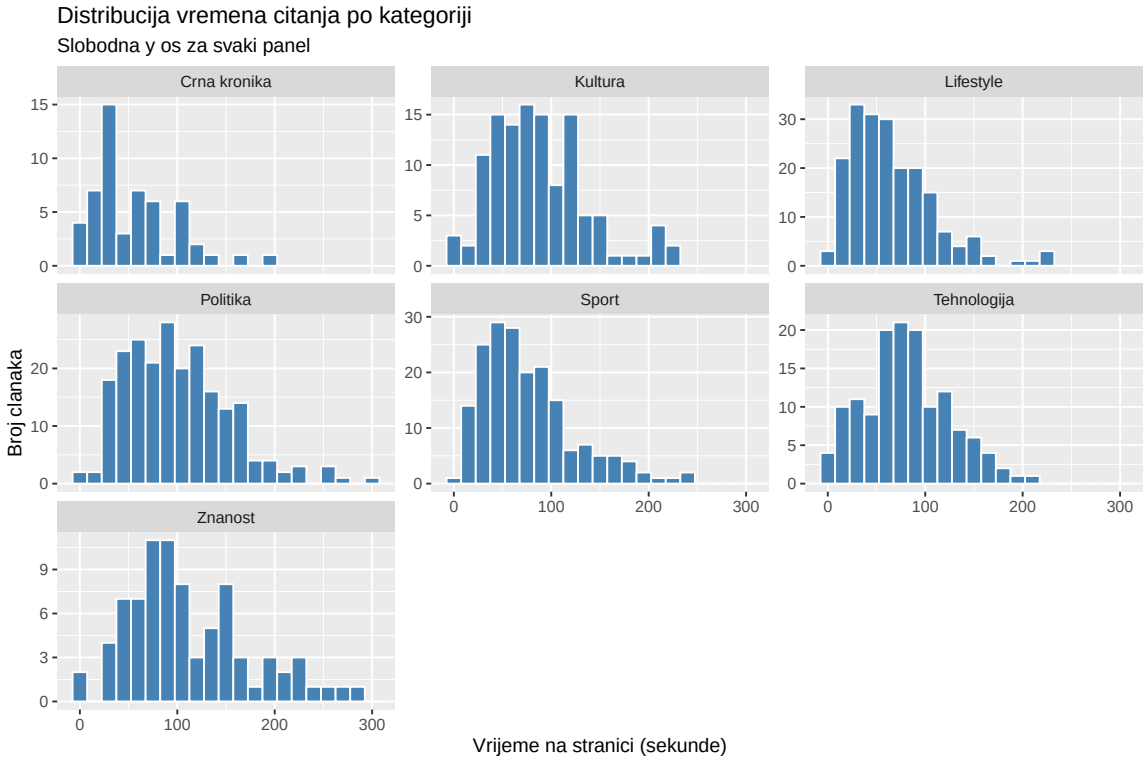
S `ncol = 2` dobivamo dva stupca panela, što je čitljivije kad imate mnogo kategorija jer su paneli širi i histogram je jasniji.

7.11.2 Slobodne osi u facetima

Po defaultu, svi paneli dijele iste osi. Ovo je dobro za usporedbu apsolutnih vrijednosti, ali ponekad želite da svaki panel ima vlastitu skalu (na primjer, kad se grupe drastično razlikuju po broju opažanja).

```
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_wrap(~category, scales = "free_y") +
```

```
  labs(
    title = "Distribucija vremena čitanja po kategoriji",
    subtitle = "Slobodna y os za svaki panel",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



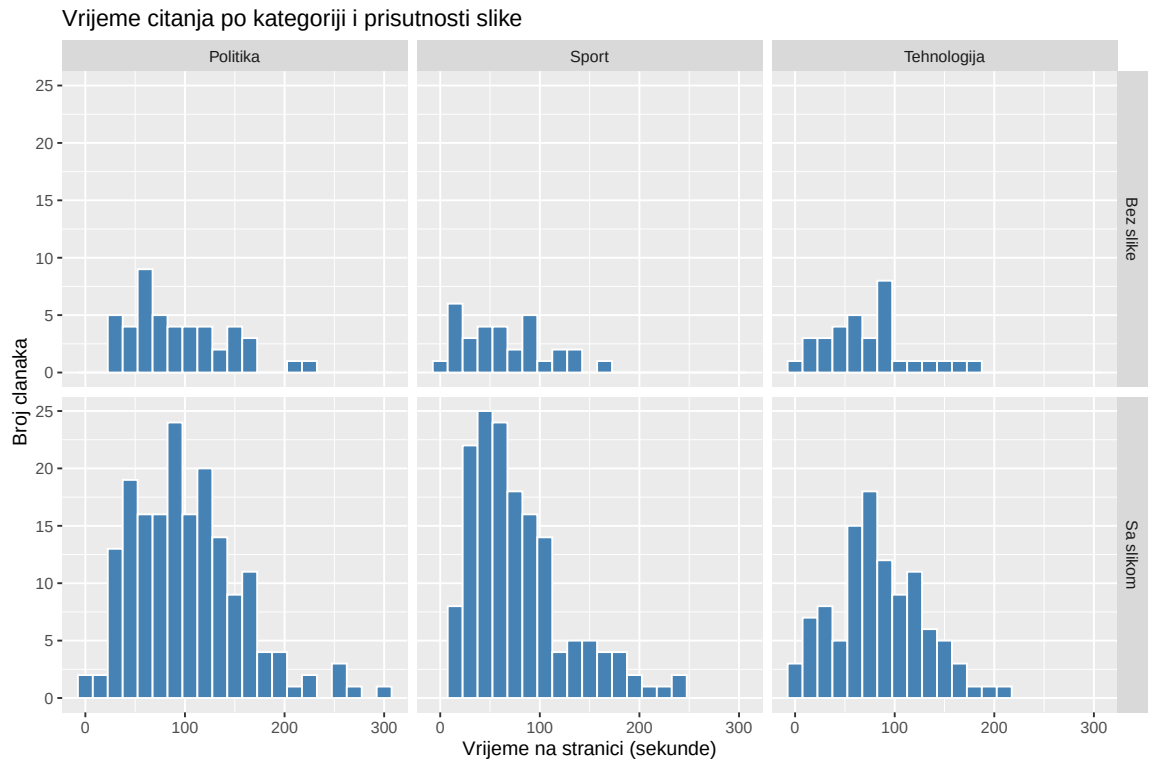
Opcije za `scales` uključuju `"fixed"` (default, iste osi), `"free_x"` (slobodna x os), `"free_y"` (slobodna y os) i `"free"` (obje slobodne). Koristite slobodne osi samo kad imate dobar razlog jer otežavaju izravnu usporedbu između panela.

7.11.3 facet_grid(): paneli u matrici dviju varijabli

Dok `facet_wrap()` dijeli po jednoj varijabli i slaže panele u mrežu, `facet_grid()` kreira matricu panela po dvjema varijablama — jedna definira retke, druga stupce.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija")) |>
  mutate(ima_sliku = if_else(has_image, "Sa slikom", "Bez slike")) |>
  ggplot(aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_grid(ima_sliku ~ category) +
```

```
  labs(
    title = "Vrijeme čitanja po kategoriji i prisutnosti slike",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj članaka"
  )
```



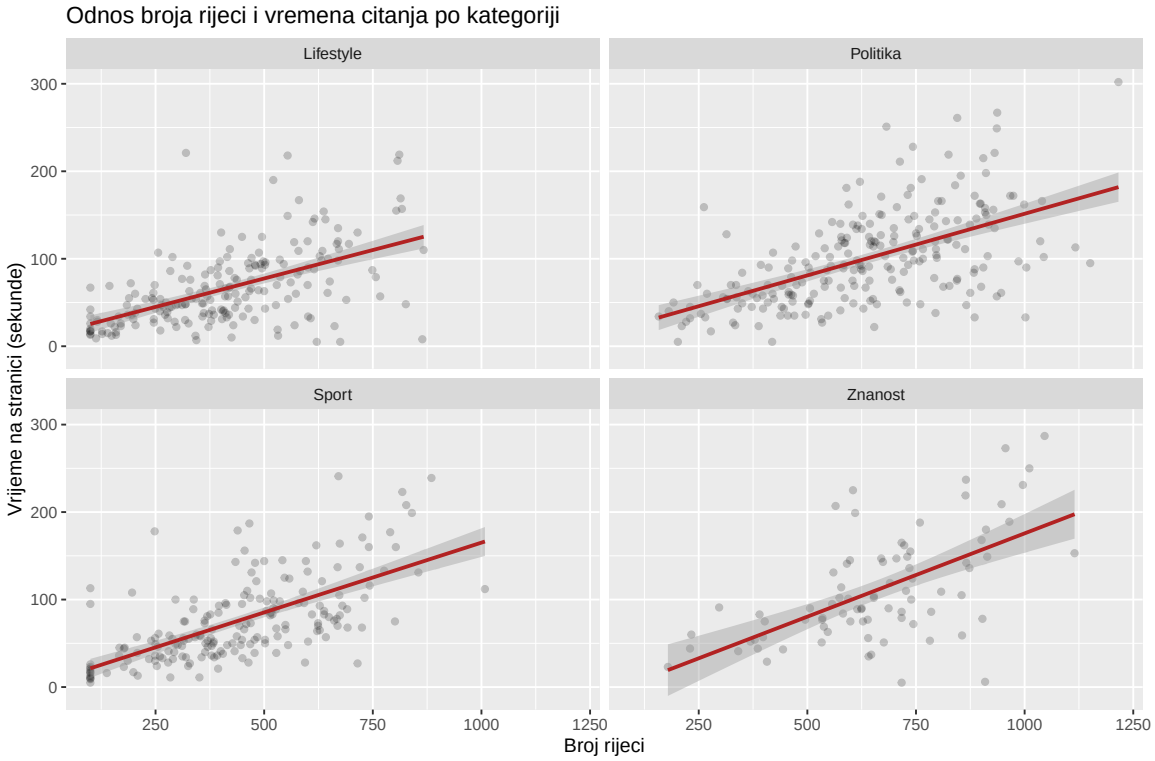
Sintaksa `facet_grid(retci ~ stupci)` postavlja varijable u redove i stupce matrice. Ovo je idealno za prikaz interakcije dviju kategoričkih varijabli jer možete uspoređivati kako vertikalno (unutar iste kategorije, sa i bez slike) tako i horizontalno (između kategorija, za isti format).

7.11.4 Facetiranje scatterplota

Facetiranje radi sa svakim tipom grafa, ne samo s histogramima.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick") +
  facet_wrap(~category) +
  labs(
```

```
    title = "Odnos broja riječi i vremena čitanja po kategoriji",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)"
  )
```



Svaki panel ima vlastitu regresijsku liniju, pa možemo vidjeti je li odnos sličan u svim kategorijama ili se razlikuje. Ovo je vizualna prethodnica interakcije u regresijskoj analizi (tjedan 14).

> **Praktični savjet**
>
> Facetiranje je gotovo uvijek bolje od preklapanja mnogo grupa u jednom grafu. Kad imate više od tri ili četiri grupe, graf s jednim panelom postaje nečitljiv bez obzira koliko pažljivo birate boje i transparentnost. Facet_wrap s 6 ili 8 panela je čitljiviji od jednog pretrpanog grafa.

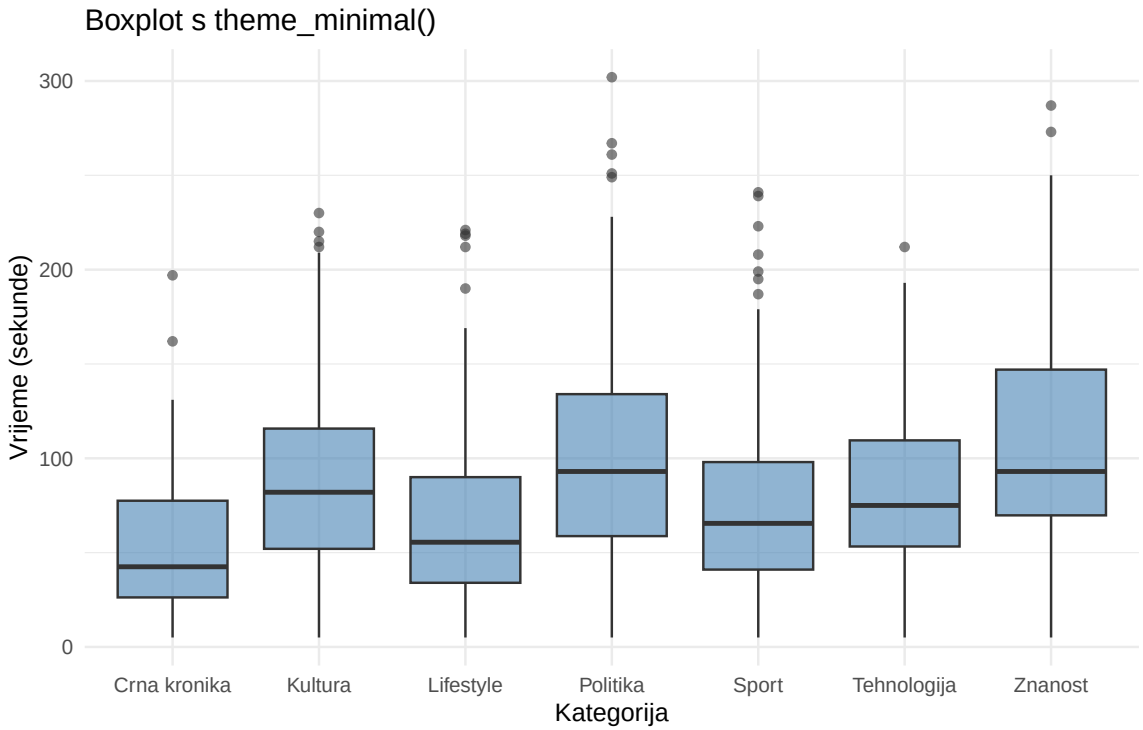
7.12 Teme: vizualni izgled grafa

Svaki ggplot2 graf ima temu koja kontrolira sve vizualne elemente koji nisu podaci — pozadinu, mrežu (grid lines), fontove, margine, poziciju legende i slično. Defaultna tema

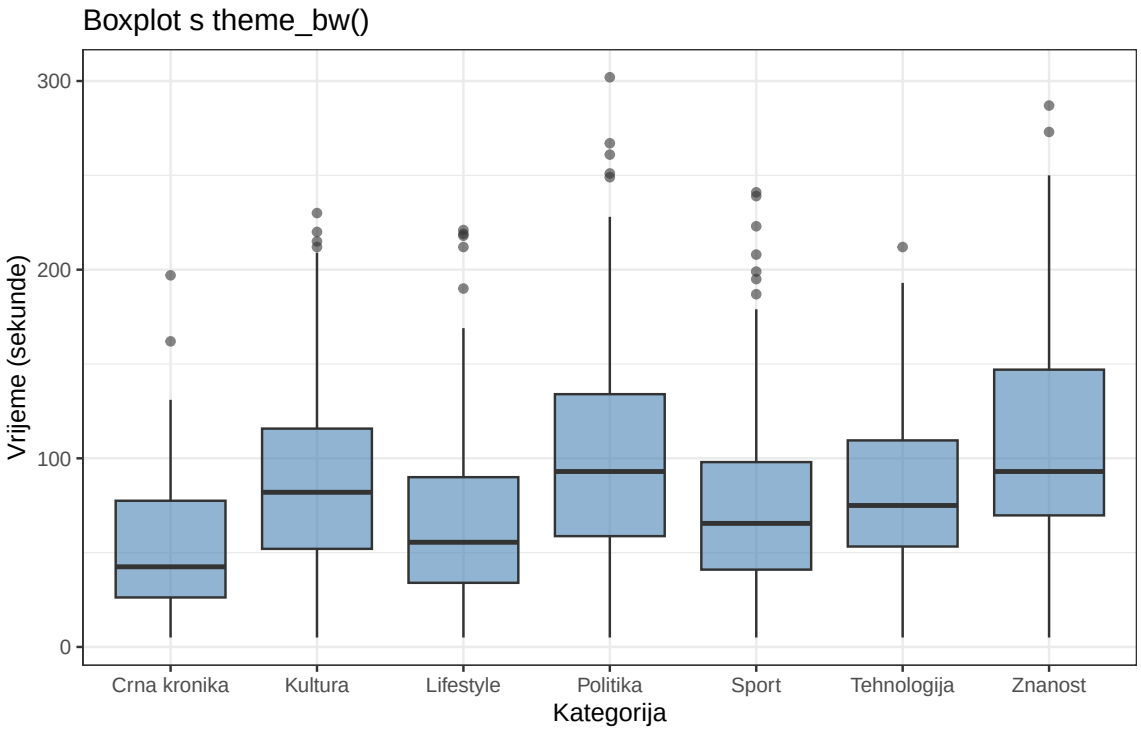
(`theme_gray()`) ima sivu pozadinu s bijelom mrežom. Za profesionalni rad, gotovo uvijek ćete koristiti neku drugu temu.

7.12.1 Ugrađene teme

```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Boxplot s theme_minimal()",
    x = "Kategorija",
    y = "Vrijeme (sekunde)"
  )
```




```
ggplot(clanci, aes(x = category, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  theme_bw() +
  labs(
    title = "Boxplot s theme_bw()",
    x = "Kategorija",
    y = "Vrijeme (sekunde)"
  )
```



`theme_minimal()` je čista tema bez okvira i s minimalnom mrežom. Odlična za prezentacije i izvještaje. `theme_bw()` je slična ali s crnim okvirom oko grafa. Obje su popularnije od defaultne sive teme.

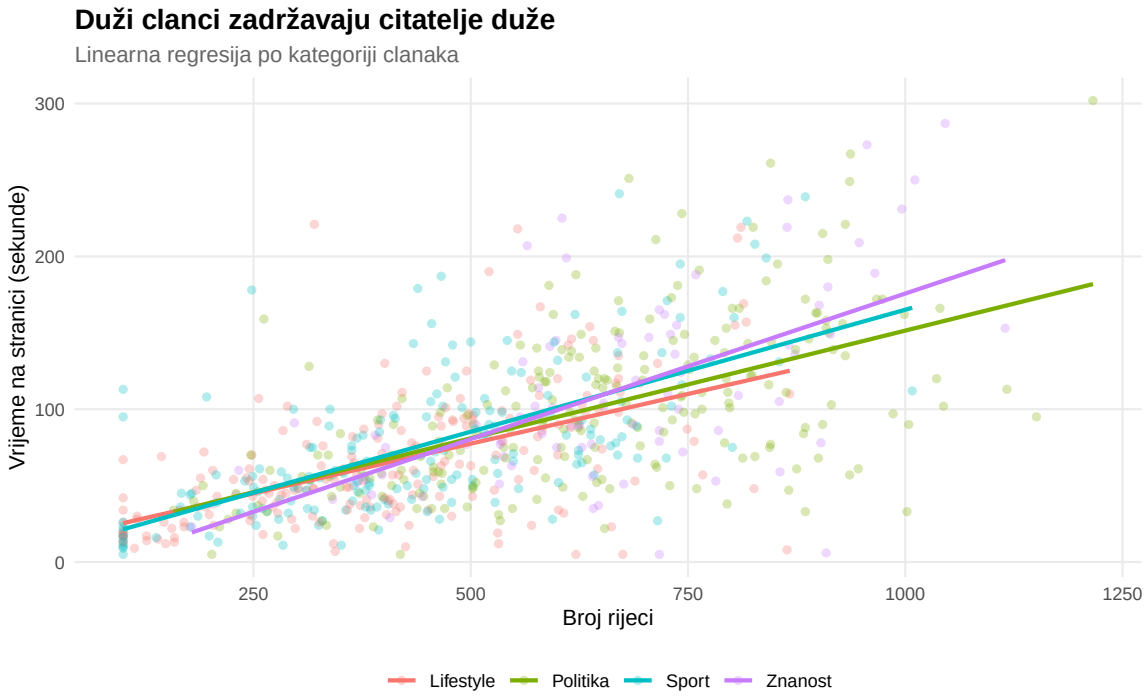
Ostale ugrađene teme uključuju `theme_classic()` (samo osi, bez mreže, tradicionalan izgled), `theme_light()` (svijetla pozadina s tankom mrežom) i `theme_void()` (prazan prostor, korisno za karte i dijagrame).

7.12.2 Prilagodba s theme()

Funkcija `theme()` omogućuje prilagodbu pojedinačnih vizualnih elemenata. Ovo je detaljni alat za fino podešavanje izgleda.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    plot.subtitle = element_text(size = 11, color = "grey40"),
    axis.title = element_text(size = 11),
    legend.position = "bottom",
```

```
    panel.grid.minor = element_blank()
  ) +
  labs(
    title = "Duži članci zadržavaju čitatelje duže",
    subtitle = "Linearna regresija po kategoriji članaka",
    x = "Broj riječi",
    y = "Vrijeme na stranici (sekunde)",
    color = NULL
  )
```



Raščlanimo `theme()` argumente — `element_text()` kontrolira fontove (veličinu, bold/italic, boju), dok `element_blank()` potpuno uklanja element (u ovom slučaju minor grid linije). `legend.position = "bottom"` premješta legendu ispod grafa. Postavljanje `color = NULL` u `labs()` uklanja naslov legende kad je očit iz konteksta.

Redoslijed je bitan: prvo dodajte ugrađenu temu (`theme_minimal()`), pa onda vlastite prilagodbe s `theme()`. Obrnuti redoslijed ne bi radio jer bi ugrađena tema pregazila vaše prilagodbe.

7.12.3 Postavljanje globalne teme

Ako želite da svi grafovi u dokumentu koriste istu temu, postavite je globalno na početku.

```
# Postavljanje globalne teme za sve grafove
theme_set(
  theme_minimal() +
    theme(
      plot.title = element_text(size = 13, face = "bold"),
      plot.subtitle = element_text(size = 10, color = "grey40"),
      panel.grid.minor = element_blank()
    )
)
```

Od ovog trenutka, svaki graf u dokumentu automatski koristi ovu temu. Ne morate je dodavati svakom grafu posebno. Ovo osigurava vizualnu konzistentnost kroz cijeli izvještaj ili prezentaciju.

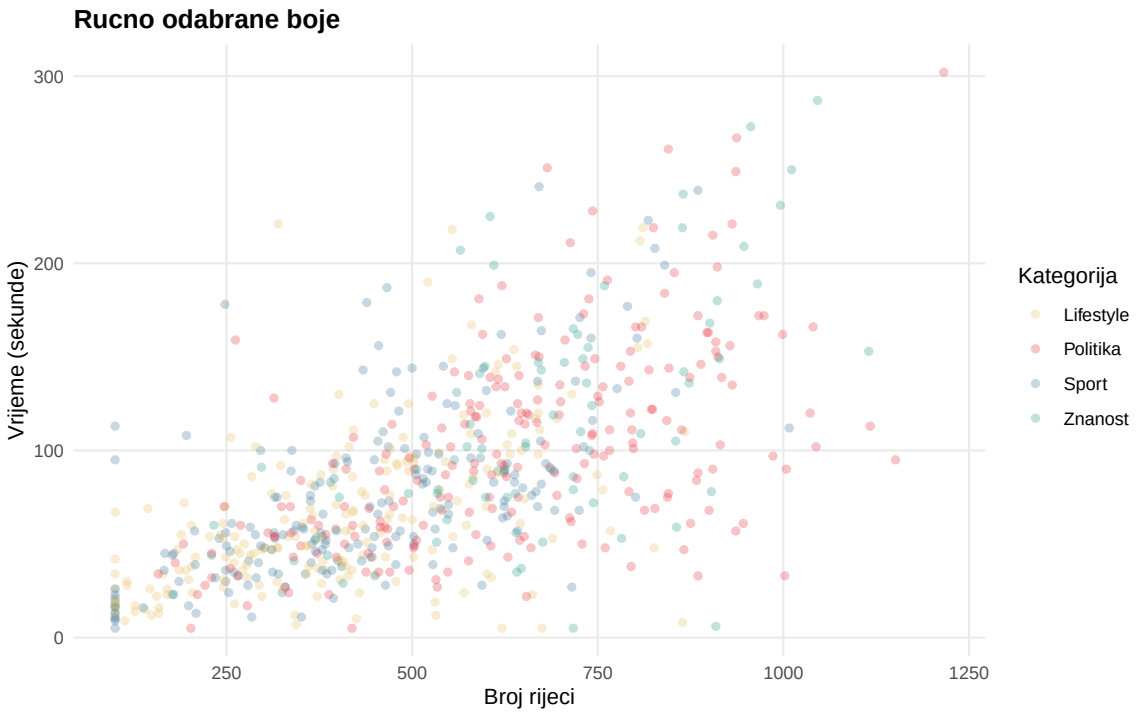
7.13 Skale boja

Defaultne boje u ggplot2 su funkcionalne ali ne uvijek idealne. Paket nudi više sustava boja prilagođenih različitim potrebama.

7.13.1 Ručni odabir boja

Za kategoričke varijable s nekoliko razina, ponekad je najbolje ručno odrediti boje.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page, color = category)) +
  geom_point(alpha = 0.3) +
  scale_color_manual(values = c(
    "Politika" = "#e63946",
    "Sport" = "#457b9d",
    "Znanost" = "#2a9d8f",
    "Lifestyle" = "#e9c46a"
  )) +
  labs(
    title = "Ručno odabrane boje",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)",
    color = "Kategorija"
  )
```

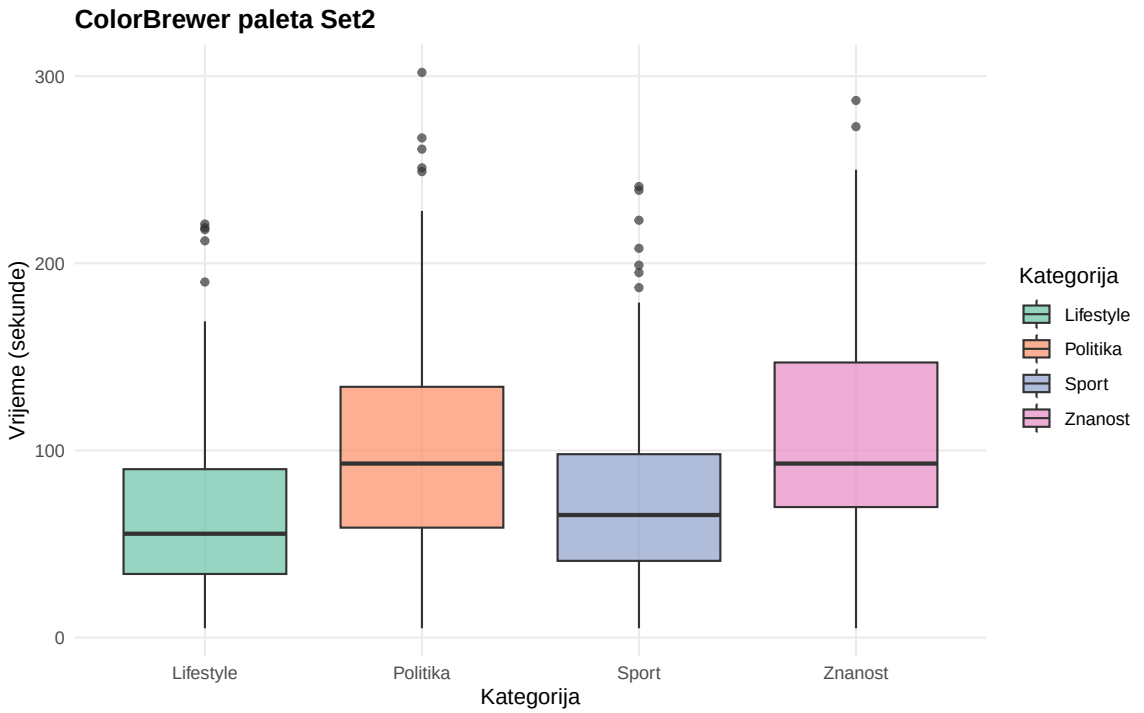


`scale_color_manual()` (za boju linija i točaka) i `scale_fill_manual()` (za boju ispune) primaju imenovani vektor boja. Prednost ručnog odabira je potpuna kontrola, ali zahtijeva poznavanje hex kodova boja ili korištenje alata poput coolors.co za odabir usklađenih paleta.

7.13.2 ColorBrewer palete

Paket RColorBrewer nudi provjerene palete dizajnirane za kartografiju i vizualizaciju podataka. Dostupne su kroz `scale_color_brewer()` i `scale_fill_brewer()`.

```
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = category, y = time_on_page, fill = category)) +
  geom_boxplot(alpha = 0.7) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "ColorBrewer paleta Set2",
    x = "Kategorija",
    y = "Vrijeme (sekunde)",
    fill = "Kategorija"
  )
```

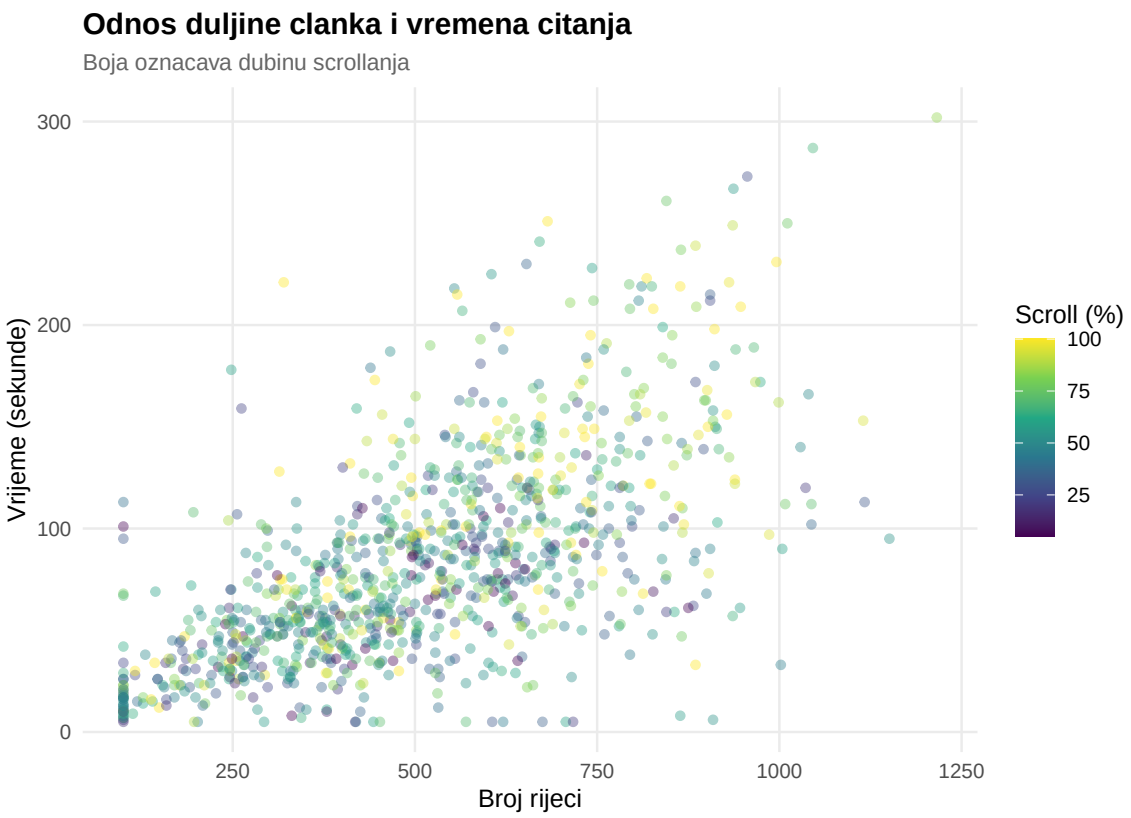


Brewer palete dolaze u tri tipa — **kvalitativne** za kategorije (npr. “Set1”, “Set2”, “Dark2”, “Pastel1”), **sekvencijalne** za gradijent od niske do visoke vrijednosti (npr. “Blues”, “Reds”, “YlOrRd”) i **divergentne** za vrijednosti koje se razilaze od sredine (npr. “RdBu”, “PRGn”).

7.13.3 Viridis palete

Viridis palete su dizajnirane da budu perceptualno uniformne (jednaki koraci u boji odgovaraju jednakim koracima u podatcima), čitljive u crno-bijelom ispisu i pristupačne osobama s poremećajem vida boja.

```
ggplot(clanci, aes(x = word_count, y = time_on_page, color = scroll_depth)) +
  geom_point(alpha = 0.4) +
  scale_color_viridis_c() +
  labs(
    title = "Odnos duljine članka i vremena čitanja",
    subtitle = "Boja označava dubinu scrollanja",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)",
    color = "Scroll (%)"
  )
```



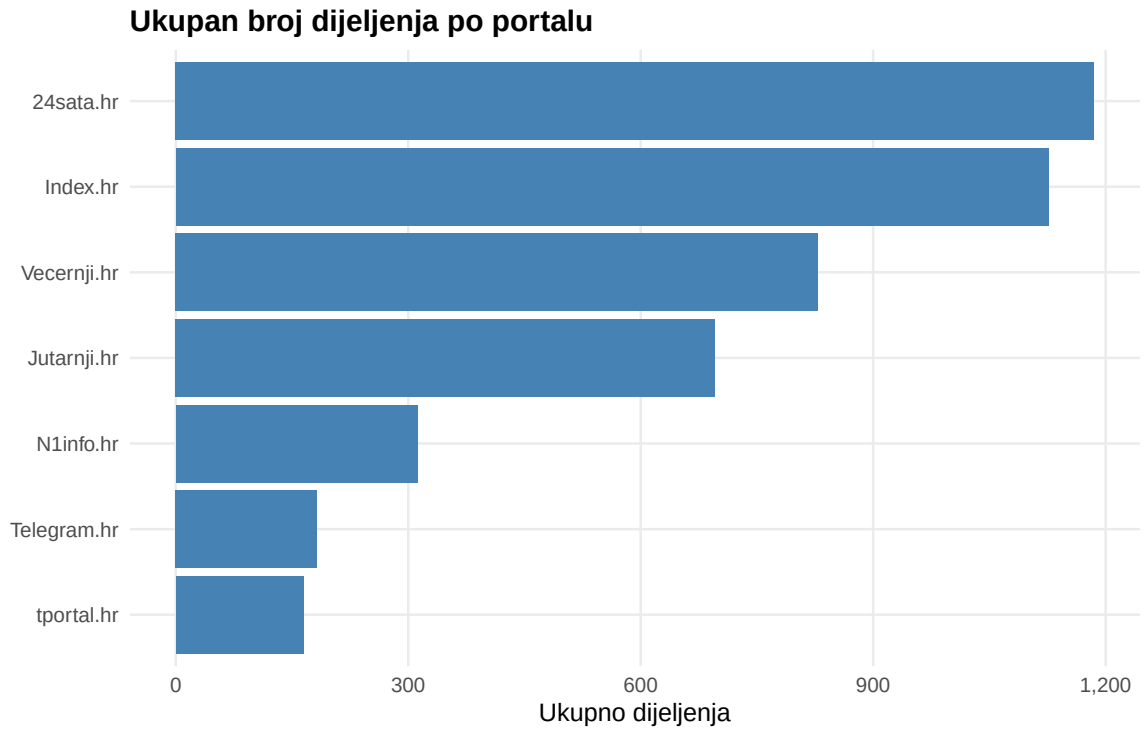
`scale_color_viridis_c()` koristi kontinuiranu viridis paletu za numeričke varijable, dok `scale_color_viridis_d()` koristi diskretnu verziju za kategoričke varijable. Opcija `option` bira između varijanti: "viridis" (default, plavo-zeleno-žuta), "magma" (crno-crveno-žuta), "plasma", "inferno" i "turbo".

> **!** Važna napomena
>
> Boja ima dva kanala u ggplot2 — `color` za rubove i linije i `fill` za ispunu. Svaki ima vlastite scale funkcije. Za boxplot koristite `scale_fill_*()`, za scatterplot `scale_color_*()`, za bar chart `scale_fill_*()`. Ako koristite krivi kanal, boja se neće promijeniti i nećete dobiti grešku, samo prazan vizualni rezultat.

7.14 Formatiranje osi

Ponekad defaultne oznake na osima nisu optimalne. Paket scales (automatski učitan s tidyverse) pruža pomoćne funkcije za formatiranje.

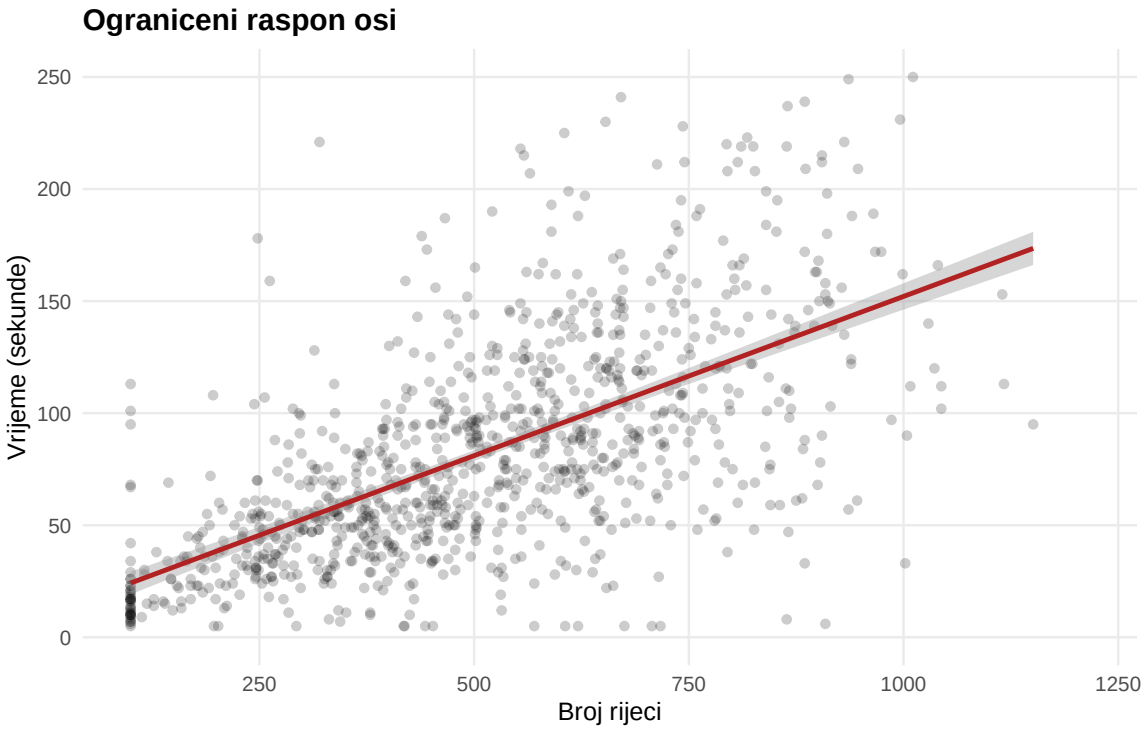
```r
library(scales)

clanci |>
  group_by(source) |>
  summarise(ukupno_dijeljenja = sum(shares), .groups = "drop") |>
  mutate(source = fct_reorder(source, ukupno_dijeljenja)) |>
  ggplot(aes(x = source, y = ukupno_dijeljenja)) +
  geom_col(fill = "steelblue") +
  scale_y_continuous(labels = label_comma()) +
  coord_flip() +
  labs(
    title = "Ukupan broj dijeljenja po portalu",
    x = NULL,
    y = "Ukupno dijeljenja"
  )
```



Funkcija `label_comma()` formatira brojeve s tisućicama (1,000 umjesto 1000). Druge korisne funkcije iz paketa scales uključuju `label_percent()` za postotke, `label_dollar()` za valute i `label_number(suffix = " min")` za dodavanje mjernih jedinica.

7.14.1 Kontrola raspona osi

```
ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick") +
  scale_x_continuous(breaks = seq(0, 1500, by = 250)) +
  scale_y_continuous(limits = c(0, 250)) +
  labs(
    title = "Ograničeni raspon osi",
    x = "Broj riječi",
    y = "Vrijeme (sekunde)"
  )
```

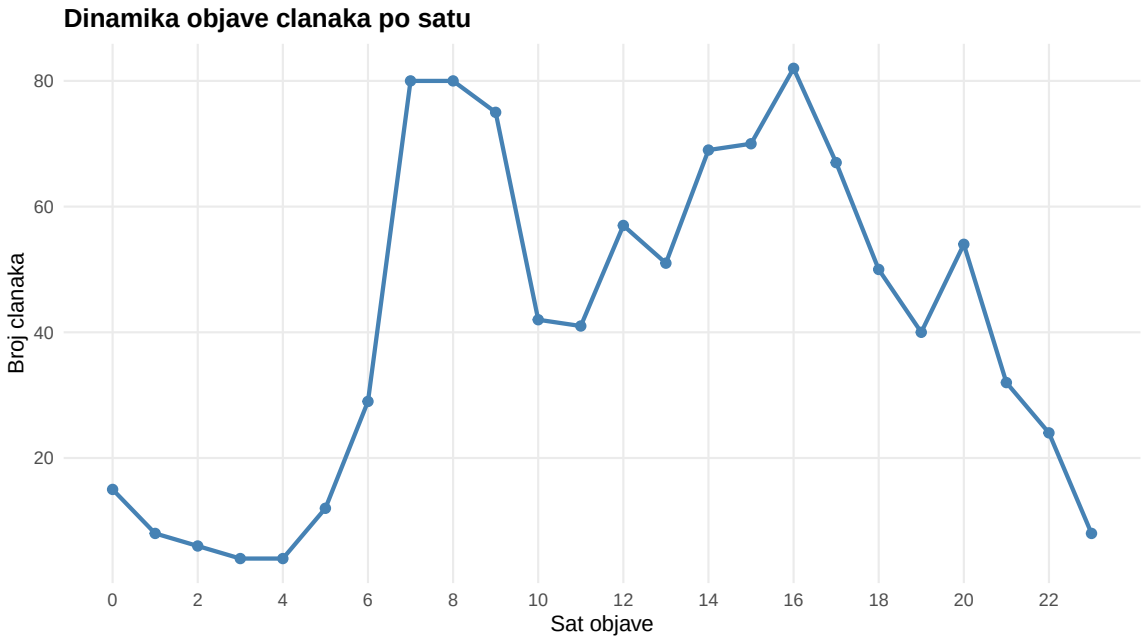


`breaks` kontrolira gdje se pojavljuju oznake na osi, dok `limits` ograničava raspon osi (točke izvan raspona se uklanjaju iz grafa). Koristite `coord_cartesian(ylim = c(0, 250))` umjesto `limits` ako želite "zumirati" bez uklanjanja podataka, jer `coord_cartesian()` samo sužava prikaz dok `limits` zaista filtrira podatke prije nego ih ggplot obradi (što može utjecati na linije trenda).

7.15 Linijski grafovi: trendovi i serije

Linijski grafovi su prirodan izbor za podatke koji imaju redoslijed, posebno vremenski. Pogledajmo distribuciju objava po satu.

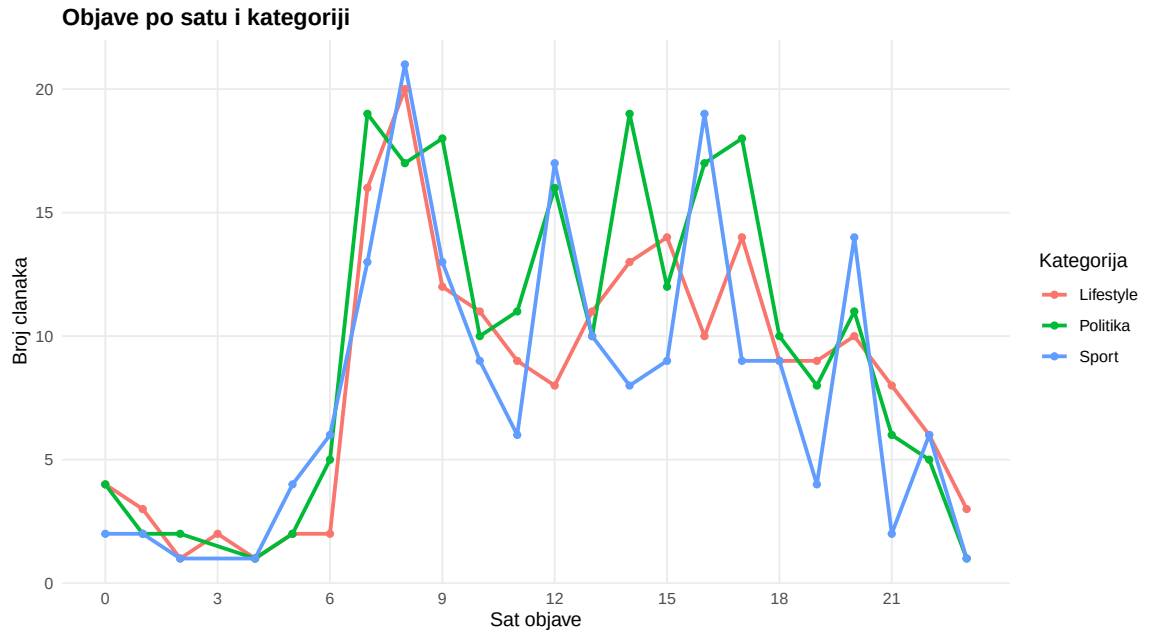
```
clanci |>
  count(publish_hour) |>
  ggplot(aes(x = publish_hour, y = n)) +
  geom_line(color = "steelblue", linewidth = 1) +
  geom_point(color = "steelblue", size = 2) +
  scale_x_continuous(breaks = seq(0, 23, by = 2)) +
  labs(
    title = "Dinamika objave članaka po satu",
    x = "Sat objave",
    y = "Broj članaka"
  )
```



Kombinacija `geom_line()` i `geom_point()` je uobičajena — linije pokazuju trend, dok točke označavaju stvarne podatke. Vidimo jasne vrhunce u jutarnjim satima i kasno popodne, što odgovara redakcijskim ciklusima.

7.15.1 Više linija u jednom grafu

```r
clanci |>
  filter(category %in% c("Politika", "Sport", "Lifestyle")) |>
  count(publish_hour, category) |>
  ggplot(aes(x = publish_hour, y = n, color = category)) +
  geom_line(linewidth = 1) +
  geom_point(size = 1.5) +
  scale_x_continuous(breaks = seq(0, 23, by = 3)) +
  labs(
    title = "Objave po satu i kategoriji",
    x = "Sat objave",
    y = "Broj članaka",
    color = "Kategorija"
  )
```

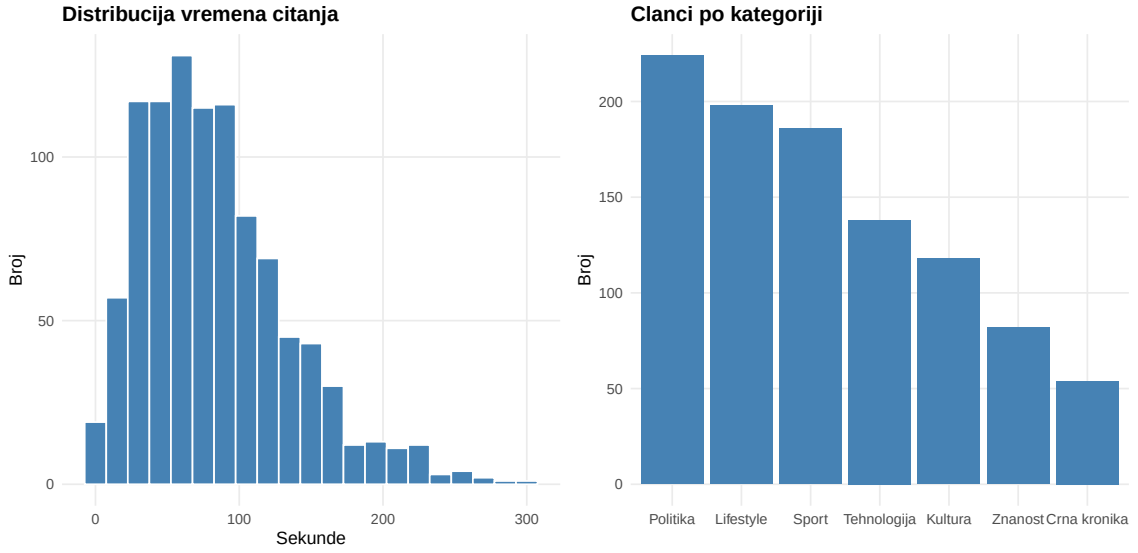


Svaka kategorija ima vlastitu liniju jer je `color = category` mapirana unutar `aes()`. Politika i sport imaju različite dnevne ritmove, što ima smisla. Sportski sadržaj se više objavljuje popodne i navečer (kad su rezultati utakmica), dok je politika koncentrirana u jutarnjim satima.

7.16 Kombiniranje grafova s patchwork

U izvještajima i radovima često trebate više grafova na jednoj stranici. Paket patchwork omogućuje elegantno slaganje ggplot2 grafova.

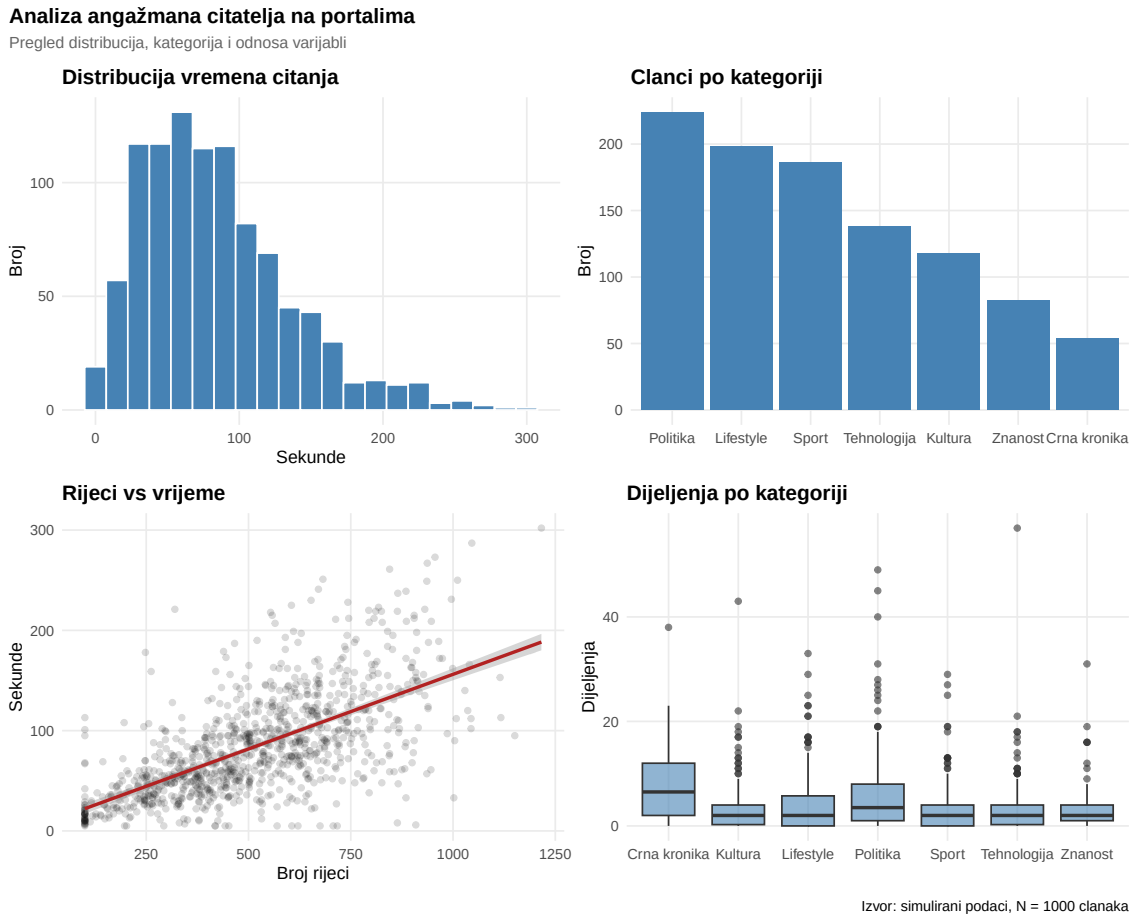
```r
library(patchwork)

p1 <- ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  labs(title = "Distribucija vremena čitanja", x = "Sekunde", y = "Broj")

p2 <- ggplot(clanci, aes(x = fct_infreq(category))) +
  geom_bar(fill = "steelblue") +
  labs(title = "Članci po kategoriji", x = NULL, y = "Broj")

p1 + p2
```



Operator + slaže grafove jedan do drugoga. Alternativno, / slaže vertikalno (jedan iznad drugoga), a | eksplicitno horizontalno.

```r
p3 <- ggplot(clanci, aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "lm", color = "firebrick") +
  labs(title = "Riječi vs vrijeme", x = "Broj riječi", y = "Sekunde")

p4 <- ggplot(clanci, aes(x = category, y = shares)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(title = "Dijeljenja po kategoriji", x = NULL, y = "Dijeljenja")

(p1 | p2) / (p3 | p4) +
  plot_annotation(
    title = "Analiza angažmana čitatelja na portalima",
    subtitle = "Pregled distribucija, kategorija i odnosa varijabli",
```

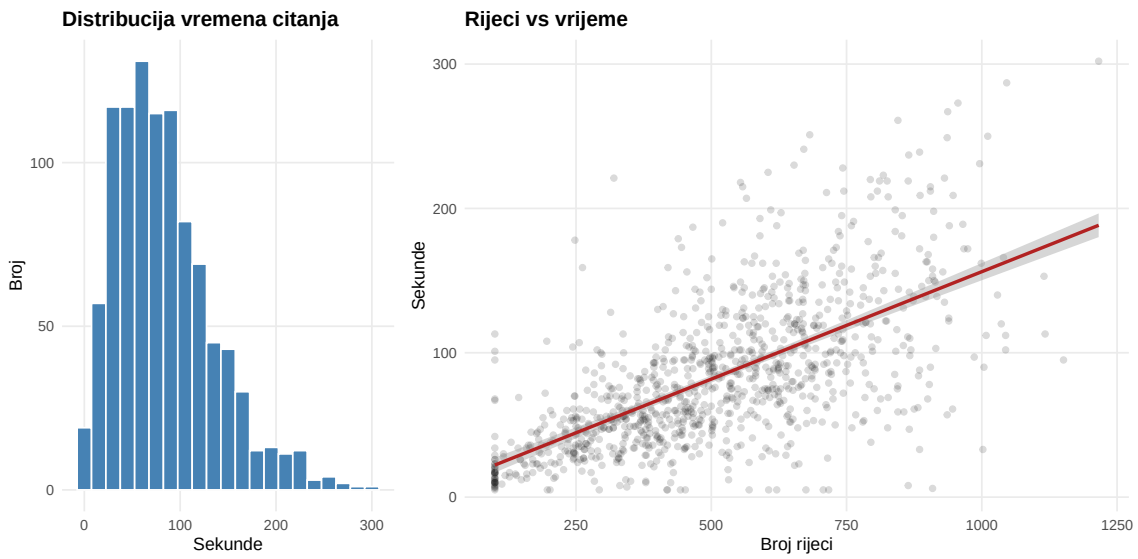
```
    caption = "Izvor: simulirani podaci, N = 1000 članaka"
  )
```



Zagrade i operatori kontroliraju raspored — `(p1 | p2) / (p3 | p4)` kreira matricu 2x2. `plot_annotation()` dodaje zajednički naslov, podnaslov i caption cijeloj kompoziciji. Ovo je profesionalan način za prezentiranje više analiza na jednom mjestu.

Patchwork podržava i `plot_layout()` za finiju kontrolu.

```r
p1 + p3 + plot_layout(widths = c(1, 2))
```



Argument `widths = c(1, 2)` daje drugom grafu dvostruku širinu — slično, `heights` kontrolira relativne visine za vertikalni raspored.

7.17 Spremanje grafova: ggsave()

Funkcija `ggsave()` sprema zadnji ggplot2 graf u datoteku. Podržava sve uobičajene formate poput PNG, PDF, SVG, JPEG i TIFF.

```r
# Spremi zadnji graf kao PNG
ggsave("angažman_portali.png", width = 10, height = 6, dpi = 300)

# Spremi specifični graf kao PDF (vektorski format, idealan za tisak)
ggsave("scatterplot.pdf", plot = p3, width = 8, height = 5)

# Spremi za prezentaciju (veće dimenzije)
ggsave("prezentacija.png", width = 12, height = 7, dpi = 150)
```

Tri ključna argumenta uključuju `width` i `height` (dimenzije u inčima) te `dpi` (rezolucija za rasterske formate). Za tisak koristite `dpi = 300`, za prezentacije `dpi = 150`, za web `dpi = 96`.

PDF format je vektorski, što znači da se skalira bez gubitka kvalitete — idealan je za akademske radove i tisak. PNG je rasterski i bolji je za web i prezentacije.

Praktični savjet

Definirajte standardne dimenzije za svoj projekt i koristite ih konzistentno. Na primjer, za Quarto dokument koji se renderira u HTML, `fig-width: 8` i `fig-height: 5` u chunk opcijama rade dobro za većinu grafova. Za prezentacije, koristite šire dimenzije (10x6). Za akademske radove, uže (6x4). Konzistentne dimenzije daju profesionalan izgled cijelom dokumentu.

7.18 Česte greške i kako ih izbjeći

Učenje ggplot2 dolazi s karakterističnim setom grešaka. Prepoznavanje najčešćih štedi sate frustracije.

7.18.1 Greška 1: + umjesto |> (i obrnuto)

```
# KRIVO: pipe unutar ggplot lanca
ggplot(clanci, aes(x = time_on_page)) |>
  geom_histogram()

# ISPRAVNO: + za dodavanje slojeva
ggplot(clanci, aes(x = time_on_page)) +
  geom_histogram()
```

Unutar ggplot2 lanca koristite `+` za dodavanje slojeva. Pipe (`|>`) koristite za dplyr operacije PRIJE `ggplot()`. Tipičan obrazac je `data |> filter(...) |> ggplot(...) + geom_*()` — prelazak s pipe na plus događa se na poziv `ggplot()`.

7.18.2 Greška 2: kontinuirana varijabla u fill/color za bar chart

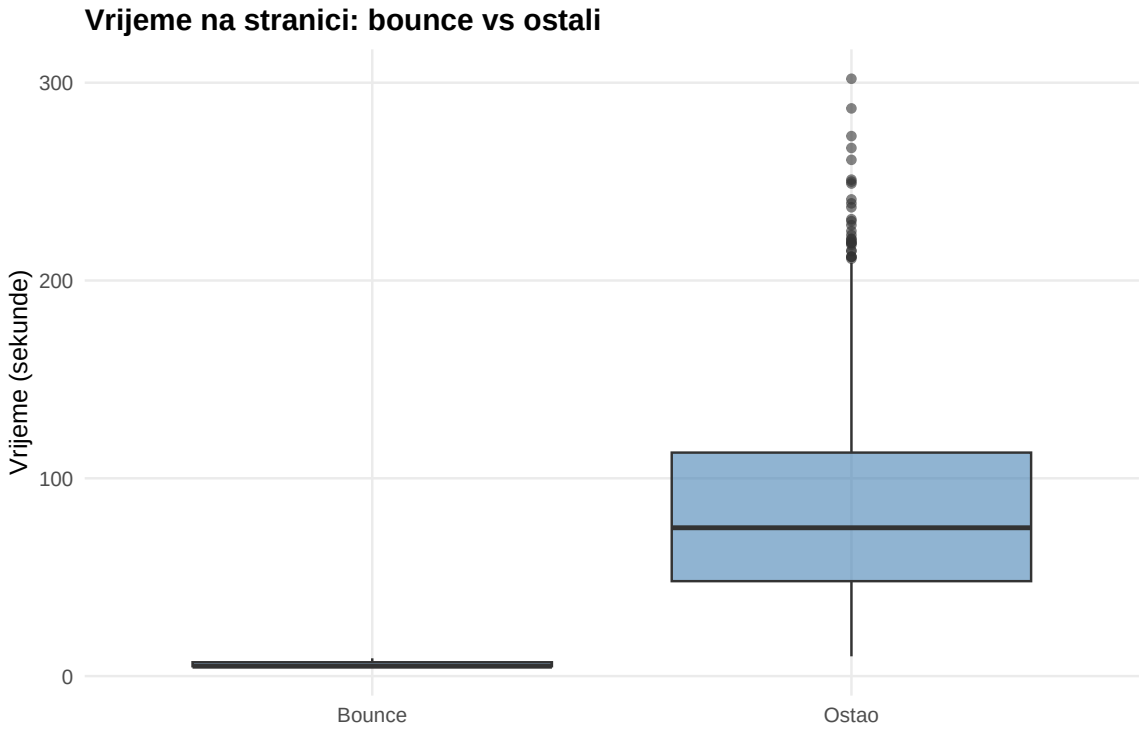
```
# ZBUNJUJUĆE: numerička varijabla kao boja u bar chartu
ggplot(clanci, aes(x = category, fill = word_count)) +
  geom_bar()

# BOLJE: kategorička varijabla za fill
ggplot(clanci, aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge")
```

7.18.3 Greška 3: previše informacija u jednom grafu

Ako imate sedam kategorija, četiri boje, liniju trenda, legendu i facetiranje, rezultat je vizualni kaos. Dobra vizualizacija komunicira jednu poruku jasno. Ako trebate reći više, napravite više grafova.

7.18.4 Greška 4: zaboravljanje na NA

```
# Logičke varijable TRUE/FALSE se ponekad pretvaraju u NA
clanci |>
  mutate(bounce_label = if_else(bounce, "Bounce", "Ostao")) |>
  ggplot(aes(x = bounce_label, y = time_on_page)) +
  geom_boxplot(fill = "steelblue", alpha = 0.6) +
  labs(
    title = "Vrijeme na stranici: bounce vs ostali",
    x = NULL,
    y = "Vrijeme (sekunde)"
  )
```



Ako u podatcima postoji NA u varijabli koja definira grupu, ggplot će napraviti zaseban panel ili stupac za NA. Uvijek provjerite podatke prije vizualizacije i odlučite želite li NA prikazati, filtrirati ili rekodirati.

7.19 Kompletna analiza: od pitanja do gotovog grafa

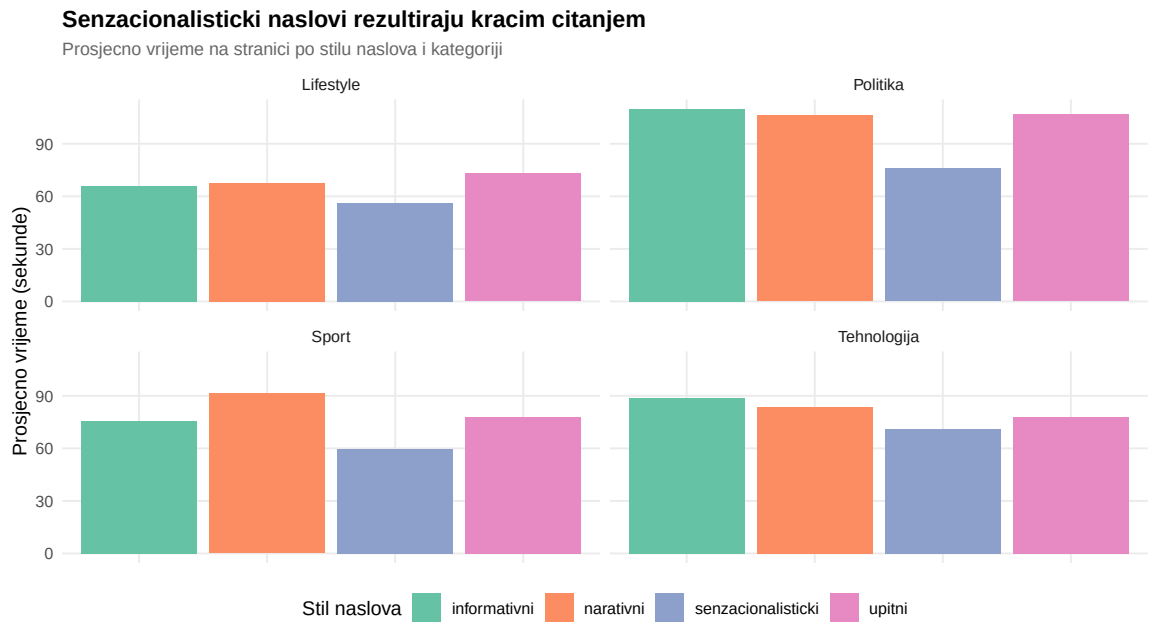
Zaokružimo predavanje kompletnim primjerom koji prolazi sve korake — definiranje pitanja, priprema podataka, odabir grafa, izgradnja i poliranje.

Pitanje — Kako se angažman čitatelja (vrijeme čitanja i dijeljenje) razlikuje ovisno o stilu naslova i kategoriji članka?

```
# Priprema podataka
angazman <- clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  group_by(category, headline_style) |>
  summarise(
    n = n(),
    prosjek_vrijeme = mean(time_on_page),
    prosjek_dijeljenja = mean(shares),
    udio_bounce = mean(bounce),
    .groups = "drop"
  ) |>
  filter(n >= 5)

angazman
```

```
# A tibble: 16 x 6
   category  headline_style     n prosjek_vrijeme prosjek_dijeljenja udio_bounce
   <chr>     <chr>          <int>           <dbl>              <dbl>       <dbl>
 1 Lifestyle informativni      71            66.0               1.87      0.0141
 2 Lifestyle narativni         39            67.5               1.79      0.0513
 3 Lifestyle senzacionalis~    40            56.1               9         0.05
 4 Lifestyle upitni            48            73.0               5.77      0
 5 Politika  informativni      88           110.                3.43      0
 6 Politika  narativni         40           106.                3.62      0.025
 7 Politika  senzacionalis~    54            75.9              11.6       0.0185
 8 Politika  upitni            42           107.                6.36      0
 9 Sport     informativni      62            75.5               1.94      0
10 Sport     narativni         38            91.3               1.68      0
11 Sport     senzacionalis~    41            59.6               6.24      0.0488
12 Sport     upitni            45            77.7               4.13      0
13 Tehnolog~ informativni      47            89.0               1.04      0.0638
14 Tehnolog~ narativni         24            83.5               2.21      0
15 Tehnolog~ senzacionalis~    28            70.9               8.21      0.0357
16 Tehnolog~ upitni            39            77.7               4.59      0.0513
```

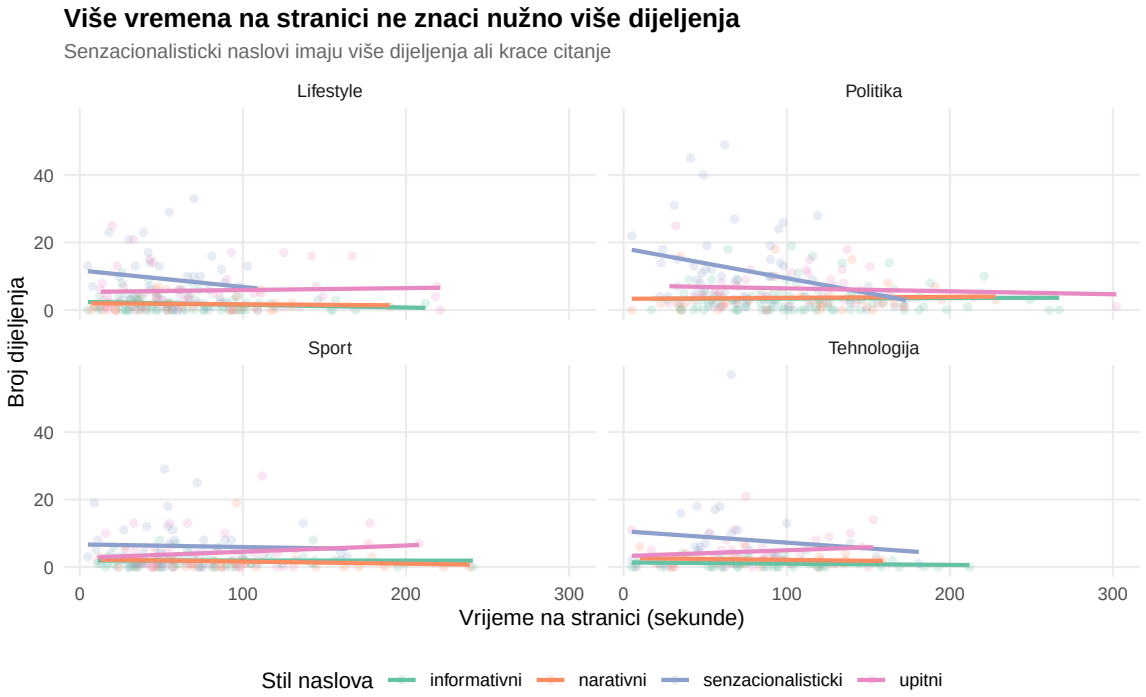
```
# Graf 1: Prosječno vrijeme čitanja po stilu naslova i kategoriji
angazman |>
  ggplot(aes(x = headline_style, y = prosjek_vrijeme, fill = headline_style)) +
  geom_col() +
  facet_wrap(~category) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Senzacionalistički naslovi rezultiraju kraćim čitanjem",
    subtitle = "Prosječno vrijeme na stranici po stilu naslova i kategoriji",
    x = NULL,
    y = "Prosječno vrijeme (sekunde)",
    fill = "Stil naslova"
  ) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  )
```



Uklonili smo oznake na x osi (`element_blank()`) jer legenda na dnu sadrži istu informaciju. Ovo smanjuje vizualni šum i čini graf čitljivijim.

```
# Graf 2: Odnos vremena i dijeljenja, po stilu naslova
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = time_on_page, y = shares, color = headline_style)) +
  geom_point(alpha = 0.2) +
```

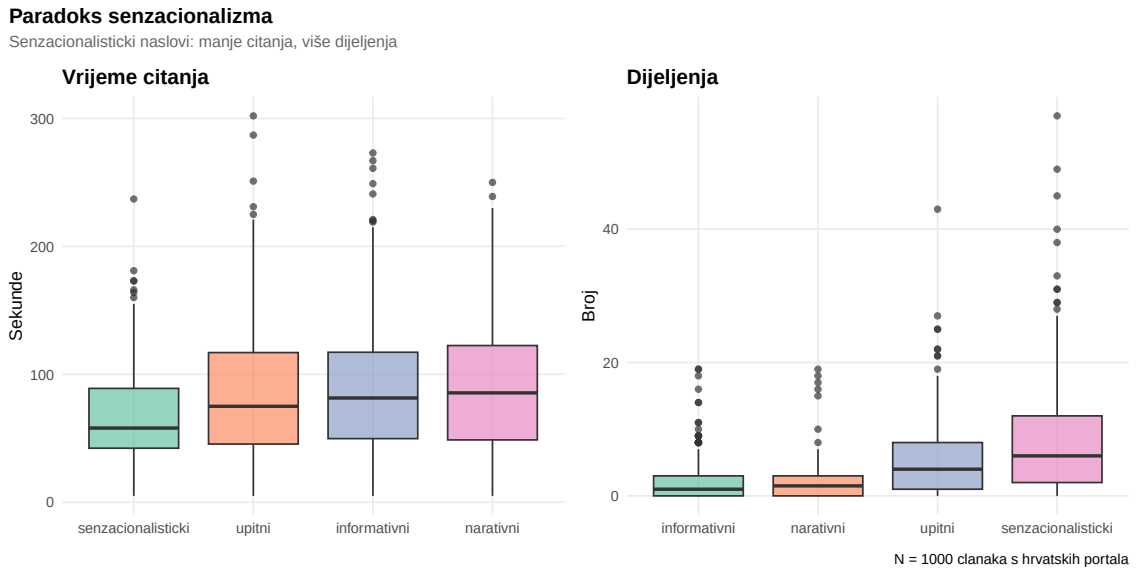
```
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_brewer(palette = "Set2") +
  facet_wrap(~category) +
  labs(
    title = "Više vremena na stranici ne znači nužno više dijeljenja",
    subtitle = "Senzacionalistički naslovi imaju više dijeljenja ali kraće čitanje",
    x = "Vrijeme na stranici (sekunde)",
    y = "Broj dijeljenja",
    color = "Stil naslova"
  ) +
  theme(legend.position = "bottom")
```



Ovaj graf otkriva zanimljiv paradoks. Senzacionalistički naslovi privlače klikove i dijeljenja, ali čitatelji provode manje vremena na članku. Informativni i narativni naslovi imaju manje dijeljenja ali duže čitanje. Ovo je klasična dilema digitalnog novinarstva — optimizirate li za klikove ili za dubinski angažman?

```
# Graf 3: Kompozitni prikaz s patchwork
graf_a <- clanci |>
  mutate(headline_style = fct_reorder(headline_style, time_on_page)) |>
  ggplot(aes(x = headline_style, y = time_on_page, fill = headline_style)) +
  geom_boxplot(alpha = 0.7, show.legend = FALSE) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Vrijeme čitanja", x = NULL, y = "Sekunde")
```

```r
graf_b <- clanci |>
  mutate(headline_style = fct_reorder(headline_style, shares)) |>
  ggplot(aes(x = headline_style, y = shares, fill = headline_style)) +
  geom_boxplot(alpha = 0.7, show.legend = FALSE) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Dijeljenja", x = NULL, y = "Broj")

graf_a + graf_b +
  plot_annotation(
    title = "Paradoks senzacionalizma",
    subtitle = "Senzacionalisticki naslovi: manje čitanja, više dijeljenja",
    caption = "N = 1000 članaka s hrvatskih portala"
  )
```



Tri grafa zajedno ispričali su kompletnu priču. Od sažetka podataka do usmjerenog nalaza, svaki graf ima jasnu poruku. Ovo je razina vizualizacije koja se očekuje u akademskim radovima, poslovnim izvještajima i novinarskim analizama.

! Ključni zaključci

1. ggplot2 graf se gradi od tri obavezne komponente: podaci, estetike (`aes()`) i geometrija (`geom_*()`). Sve ostalo (skale, faceti, teme) je opcionalno ali važno za profesionalan izgled.

2. Estetike unutar `aes()` mapiraju varijable na vizualna svojstva (i kreiraju legendu). Estetike izvan `aes()` postavljaju fiksne vrijednosti. Miješanje ova dva pristupa je

najčešći izvor zbunjenosti.

3. Histogram i density prikazuju distribuciju jedne varijable. Bar chart prikazuje kategorije. Boxplot uspoređuje distribucije između grupa. Scatterplot prikazuje odnos dviju varijabli. Odabir grafa ovisi o tipovima varijabli koje imate.

4. Facetiranje (`facet_wrap()`, `facet_grid()`) dijeli graf na panele po grupama. Gotovo uvijek je čitljivije od preklapanja mnogo grupa u jednom grafu.

5. Teme kontroliraju vizualne elemente koji nisu podaci. `theme_minimal()` i `theme_bw()` su dobri izbori za profesionalan rad. `theme_set()` postavlja globalnu temu za cijeli dokument.

6. Boje se biraju ovisno o tipu podataka: kvalitativne palete (Set2, Dark2) za kategorije, sekvencijalne (Blues, viridis) za kontinuirane varijable. Viridis palete su pristupačne osobama s poremećajem vida boja.

7. `labs()` je obavezna funkcija za svaki graf. Formulirajte naslov kao nalaz, ne kao opis. Dodajte `caption` za izvor podataka.

8. `ggsave()` sprema grafove u datoteku. PDF za tisak (vektorski), PNG za web (rasterski). Koristite `dpi = 300` za tisak.

9. Patchwork kombinira više grafova u jednu kompoziciju operatorima `+`, `/`, `|`. `plot_annotation()` dodaje zajednički naslov.

10. Linijski grafovi (`geom_line()`) su prirodan izbor za podatke s redoslijedom, posebno vremenske serije.

11. Unutar ggplot2 lanca koristite `+` za slojeve. Pipe (`|>`) koristite za dplyr PRIJE `ggplot()`. Prelazak je na poziv `ggplot()`.

12. Dobra vizualizacija komunicira jednu poruku jasno. Ako trebate reći više, napravite više grafova. Vizualni kaos s previše slojeva je gori od praznog platna.

⚠ Priprema za sljedeći tjedan

Sljedeći tjedan bavimo se **programiranjem u R-u**. Naučit ćete pisanje funkcija, uvjetne naredbe, petlje i organizaciju ponovljivih analiza. Fokus nije na tome da postanete programeri, nego na tome da napišete čist, ponovljiv kod koji možete pokrenuti ponovno kad dobijete nove podatke.
Za pripremu:

1. Ponovite sve tipove grafova iz ovog predavanja. Za svaki pokušajte promijeniti barem jedan argument i vidjeti što se događa.
2. Napravite tri grafa iz podataka `article_engagement.csv` koji odgovaraju na

sljedeće pitanje — razlikuju li se portali po angažmanu čitatelja? Koristite barem jedan histogram, jedan boxplot i jedan bar chart.
3. Kombinirajte ta tri grafa pomoću patchwork u jednu kompoziciju s zajedničkim naslovom.
4. Pročitajte poglavlje 8 iz Navarro (Learning Statistics with R) o osnovama programiranja.

7.20 Dodatno čitanje

Obavezno

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Chapters 2, 10, 11 i 12. Besplatno dostupno na r4ds.hadley.nz. Poglavlje 2 daje brzi uvod u vizualizaciju, poglavlje 10 pokriva EDA, poglavlja 11 i 12 detaljno obrađuju komunikaciju putem grafova i slojeve ggplot2.

Navarro, D. (2018). *Learning Statistics with R*, Chapter 6. Besplatno dostupno na learningstatisticswithr.com. Poglavlje koristi base R grafiku, ali koncepti izbora grafa su univerzalni.

Preporučeno

Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press. Besplatno dostupno na socviz.co. Izvrsna knjiga o ggplot2 s naglaskom na principe vizualizacije u društvenim znanostima.

Wilke, C. O. (2019). *Fundamentals of Data Visualization*. O'Reilly. Besplatno dostupno na clauswilke.com/dataviz. Fokus na principima vizualizacije neovisno o alatu.

Scherer, C. (2022). *A ggplot2 Tutorial for Beautiful Plotting in R*. Besplatno dostupno na cedricscherer.netlify.app/2019/08/05/a-ggplot2-tutorial-for-beautiful-plotting-in-r. Detaljan vodič za profesionalno poliranje grafova u ggplot2.

7.21 Pojmovnik

Pojam	Objašnjenje
ggplot2	R paket za vizualizaciju podataka temeljen na gramatici grafike. Dio tidyverse ekosustava.

Pojam	Objašnjenje
Gramatika grafike	Sustav komponenti (podaci, estetike, geometrija, skale, faceti, teme) koje se kombiniraju u slojeve za kreiranje grafova.
`aes()`	Funkcija za mapiranje varijabli na vizualne dimenzije grafa (x os, y os, boja, veličina, oblik).
Geometrija (`geom_*()`)	Vizualni oblik za prikaz podataka. Svaki tip grafa ima svoju geom funkciju.
`geom_histogram()`	Geometrija za histogram. Argument `binwidth` kontrolira širinu bina.
`geom_density()`	Geometrija za graf gustoće distribucije.
`geom_bar()`	Geometrija za bar chart koji automatski broji opažanja po kategorijama.
`geom_col()`	Geometrija za bar chart s prethodno izračunatim y vrijednostima.
`geom_boxplot()`	Geometrija za boxplot koji prikazuje medijan, kvartile i outliere.
`geom_violin()`	Geometrija za violin plot koji prikazuje oblik distribucije.
`geom_point()`	Geometrija za scatterplot.
`geom_jitter()`	Varijanta `geom_point()` s nasumičnim pomakom za izbjegavanje preklapanja.
`geom_smooth()`	Geometrija za liniju trenda. Default LOESS, `method = "lm"` za linearnu.
`geom_line()`	Geometrija za linijski graf. Pogodna za vremenske serije i podatke s redoslijedom.
`facet_wrap()`	Dijeli graf na panele po jednoj varijabli. Argumenti: `ncol`, `scales`.
`facet_grid()`	Dijeli graf na matricu panela po dvjema varijablama. Sintaksa: `retci ~ stupci`.
`labs()`	Funkcija za naslove, podnaslove, oznake osi, legende i caption.
`theme_minimal()`	Ugrađena tema: čista, bez okvira, minimalna mreža. Popularna za profesionalni rad.
`theme_bw()`	Ugrađena tema: bijela pozadina s crnim okvirom.
`theme()`	Funkcija za detaljnu prilagodbu vizualnih elemenata (fontovi, margine, legenda, mreža).
`theme_set()`	Postavlja globalnu temu za sve grafove u dokumentu.
`element_text()`	Unutar `theme()`: kontrolira svojstva teksta (veličina, bold, boja, kut).

Pojam	Objašnjenje
`element_blank()`	Unutar `theme()`: potpuno uklanja element (mreža, oznake, rubovi).
`scale_color_manual()`	Ručni odabir boja za `color` estetiku.
`scale_fill_manual()`	Ručni odabir boja za `fill` estetiku.
`scale_color_brewer()`	ColorBrewer palete za `color`. Tipovi: kvalitativne, sekvencijalne, divergentne.
`scale_fill_brewer()`	ColorBrewer palete za `fill`.
`scale_color_viridis_c()`	Viridis paleta za kontinuirane varijable. Pristupačna za poremećaj vida boja.
`scale_color_viridis_d()`	Viridis paleta za diskretne varijable.
`alpha`	Estetika za transparentnost. Od 0 (prozirno) do 1 (neprozirno).
`fill`	Estetika za boju ispune (stupci, pravokutnici, područja).
`color`	Estetika za boju ruba ili linije (točke, linije, rubovi).
`position = "dodge"`	Stupci jedne do drugih u grupiranom bar chartu.
`position = "fill"`	Normalizira stupce na proporcije.
`fct_infreq()`	Sortira faktor po frekvenciji.
`fct_reorder()`	Sortira faktor po vrijednostima druge varijable.
`coord_flip()`	Zamjenjuje x i y os za horizontalne grafove.
`coord_cartesian()`	Zumira graf bez uklanjanja podataka.
`ggsave()`	Sprema graf u datoteku. Argumenti: širina, visina, dpi, format.
patchwork	Paket za kombiniranje više ggplot2 grafova. Operatori: `+` (horizontalno), `/` (vertikalno), `|` (horizontalno).
`plot_annotation()`	Patchwork funkcija za zajednički naslov kompozicije.
Whisker	Linije iz boxplota do 1.5 x IQR od kvartila.
Outlier	Točka udaljena više od 1.5 x IQR od kvartila.
DPI	Dots per inch. Rezolucija rasterske slike. 300 za tisak, 150 za prezentacije, 96 za web.

Dio III

Statistička teorija

8 Tjedan 7: Uvod u vjerojatnost

Slučajnost, distribucije i zašto ništa nije sigurno

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti što je vjerojatnost i zašto je temelj sve statističke analize.
2. Opisati razliku između frekvencijskog i bayesijanskog pristupa vjerojatnosti.
3. Primijeniti osnovna pravila vjerojatnosti, uključujući komplementarno pravilo, pravilo zbrajanja i pravilo množenja.
4. Izračunati vjerojatnost nezavisnih i zavisnih događaja.
5. Objasniti što je binomna distribucija i prepoznati situacije u kojima se primjenjuje.
6. Koristiti `dbinom()`, `pbinom()` i `rbinom()` za izračun i simulaciju binomnih vjerojatnosti.
7. Vizualizirati distribucije vjerojatnosti u ggplot2.
8. Povezati koncept vjerojatnosti s praktičnim pitanjima iz komunikologije (viralnost sadržaja, stopa otvaranja emaila, konverzija).

8.1 Zašto vjerojatnost?

Do sada smo se bavili opisivanjem podataka koji postoje. Izračunali smo prosjeke, napravili grafove, očistili neuredne datasete. Ali statistika ne služi samo za opisivanje onoga što znamo. Služi i za donošenje zaključaka o onome što ne znamo. A za to nam treba vjerojatnost.

Zamislite da radite A/B test naslova na portalu. Varijanta A ima click-through rate (CTR) od 4.2%, varijanta B ima 4.8%. Je li B zaista bolja ili je razlika samo slučajnost? Odgovor na to pitanje zahtijeva razumijevanje vjerojatnosti. Kolika je vjerojatnost da bismo vidjeli ovakvu ili veću razliku čistom slučajnošću, čak i da su naslovi jednako dobri? Ako je ta vjerojatnost mala, zaključujemo da B vjerojatno zaista jest bolji. Ako je velika, zaključujemo da nemamo dovoljno dokaza.

Ovo je logika koja stoji iza svakog statističkog testa koji ćemo učiti u nastavku kolegija. Govorimo o t-testovima, hi-kvadrat testovima, ANOVA-i i regresiji. Svi oni koriste vjerojatnost kao temelj za donošenje zaključaka. Bez razumijevanja vjerojatnosti, ti testovi su crne kutije u koje ubacujete brojeve i dobivate misterioznu p-vrijednost. S razumijevanjem vjerojatnosti, ti testovi postaju logični alati s jasnom interpretacijom.

Ovo je možda najkonceptualnije predavanje na kolegiju. Nema mnogo koda, nema čišćenja podataka, nema dugačkih pipeline. Umjesto toga, gradimo intuiciju o slučajnosti i distribucijama koja će nam služiti kroz ostatak kolegija.

8.2 Naši podaci: objave na društvenim mrežama

Za ilustraciju vjerojatnosnih koncepata koristimo dataset od 2000 objava na društvenim mrežama. Za svaku objavu imamo platformu, tip sadržaja, broj pratitelja, lajkova, dijeljenja, komentara i oznaku je li objava postala viralna.

```
posts <- read_csv("../resources/datasets/social_posts.csv")
glimpse(posts)
```

```
Rows: 2,000
Columns: 11
$ post_id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
$ platform     <chr> "Instagram", "Instagram", "Instagram", "TikTok", "Twitter~
$ content_type <chr> "slika", "carousel", "story", "reel", "video", "tekst", "~
$ followers    <dbl> 2398, 19468, 1386, 1221, 26918, 9229, 1603, 17998, 4187, ~
$ likes        <dbl> 202, 255, 75, 55, 0, 44, 132, 721, 439, 37, 3018, 37, 217~
$ shares       <dbl> 7, 20, 9, 13, 0, 7, 20, 28, 29, 4, 431, 7, 10, 89, 45, 6,~
$ comments     <dbl> 11, 8, 3, 3, 0, 4, 0, 9, 46, 1, 22, 0, 15, 118, 8, 5, 0, ~
$ hashtags     <dbl> 14, 10, 6, 4, 0, 15, 3, 14, 16, 10, 9, 2, 15, 4, 8, 1, 3,~
$ post_hour    <dbl> 11, 8, 18, 12, 11, 23, 11, 7, 17, 20, 19, 13, 9, 14, 16, ~
$ day_of_week  <chr> "utorak", "subota", "srijeda", "ponedjeljak", "utorak", "~
$ is_viral     <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, F~
```

```
posts |>
  count(platform, sort = TRUE)
```

```
# A tibble: 6 x 2
  platform      n
  <chr>     <int>
1 Instagram   530
2 TikTok      505
```

```
3 Facebook    361
4 Twitter/X   247
5 YouTube     230
6 LinkedIn    127
```

```
# Koliko je objava viralno?
posts |>
  count(is_viral) |>
  mutate(udio = round(n / sum(n), 3))
```

```
# A tibble: 2 x 3
  is_viral     n  udio
  <lgl>    <int> <dbl>
1 FALSE     1972 0.986
2 TRUE        28 0.014
```

Od 2000 objava, samo mali postotak je viralan. Ovo nam daje savršen kontekst za razmišljanje o vjerojatnosti. Kolika je šansa da objava postane viralna? Ovisi li to o platformi? O tipu sadržaja? O broju pratitelja? Ova pitanja ćemo istraživati kroz predavanje.

8.3 Što je vjerojatnost?

Intuitivan odgovor je da je vjerojatnost broj koji izražava koliko je nešto izvjesno. Ako kažemo da je vjerojatnost kiše sutra 70%, to znači da smo prilično sigurni da će padati, ali ne potpuno. Ako kažemo da je vjerojatnost da novčić padne na glavu 50%, to znači da su oba ishoda jednako vjerovatna.

Formalno, vjerojatnost je broj između 0 i 1. Vrijednost 0 znači da se događaj sigurno neće dogoditi. Vrijednost 1 znači da će se sigurno dogoditi. Sve između izražava stupanj neizvjesnosti.

$$P(\text{događaj}) \in [0, 1]$$

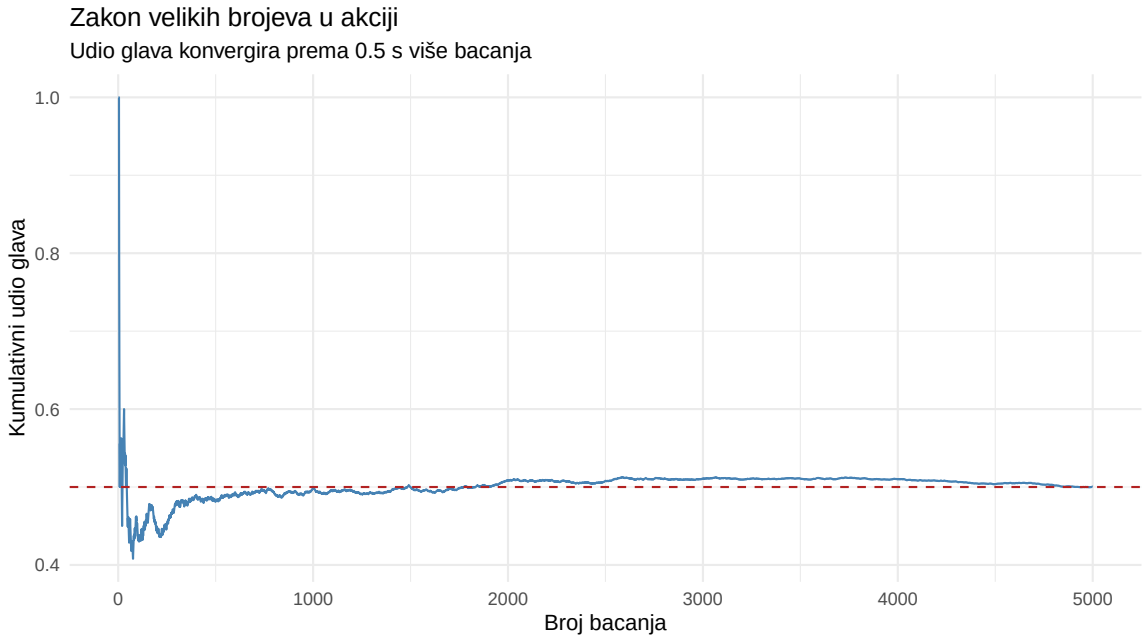
Ponekad se vjerojatnost izražava kao postotak (0% do 100%), ali u statistici i R-u koristimo razlomke (0 do 1).

8.3.1 Frekvencijski pristup

Frekvencijski (ili klasični) pristup definira vjerojatnost kao dugoročnu relativnu frekvenciju. Ako bacite novčić 10 000 puta, otprilike 5 000 puta će pasti na glavu. Omjer 5000/10000 = 0.5 je vjerojatnost.

Pokažimo to simulacijom.

```
set.seed(42)

# Simulacija bacanja novčića
n_bacanja <- 5000
bacanja <- sample(c("glava", "pismo"), size = n_bacanja, replace = TRUE)

# Kumulativni udio glava nakon svakog bacanja
kum_udio <- cumsum(bacanja == "glava") / seq_along(bacanja)

tibble(bacanje = 1:n_bacanja, udio_glava = kum_udio) |>
  ggplot(aes(x = bacanje, y = udio_glava)) +
  geom_line(color = "steelblue") +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "firebrick") +
  labs(
    title = "Zakon velikih brojeva u akciji",
    subtitle = "Udio glava konvergira prema 0.5 s više bacanja",
    x = "Broj bacanja",
    y = "Kumulativni udio glava"
  ) +
  theme_minimal()
```



Na početku, udio skače gore-dolje jer je uzorak mali. Ali s više bacanja, udio se stabilizira oko 0.5. Ovo je **zakon velikih brojeva**. S dovoljno ponavljanja, relativna frekvencija konvergira prema pravoj vjerojatnosti.

Funkcija `set.seed(42)` fiksira generator slučajnih brojeva da bismo svaki put dobili iste rezultate. Bez nje, svako pokretanje koda bi dalo malo drugačiji graf. U ponovljivoj analizi, uvijek postavljamo seed.

8.3.2 Bayesijanski pristup (kratki osvrt)

Bayesijanski pristup definira vjerojatnost kao stupanj uvjerenja. Umjesto da govori o dugoročnoj frekvenciji, bayesijanska statistika kaže da na temelju onoga što znamo, naša uvjerenost da će se X dogoditi je Y.

Ova razlika je filozofska i nema praktične posljedice za većinu analiza koje ćemo raditi. Frekvencijski pristup je dominantan u komunikologiji i društvenim znanostima, pa ćemo ga koristiti. Ali vrijedi znati da postoji alternativni pristup, posebno jer bayesijanska statistika postaje sve popularnija u istraživanjima.

Na ovom kolegiju koristimo frekvencijski pristup. U tjednu 15 ćemo se kratko osvrnuti na bayesijanski kao pogled naprijed.

8.4 Osnovna pravila vjerojatnosti

Postoji nekoliko temeljnih pravila koja vrijede za sve vjerojatnosti. Naučimo ih na primjerima iz našeg dataseta.

8.4.1 Komplementarno pravilo

Ako je P(A) vjerojatnost da se događaj A dogodi, tada je vjerojatnost da se A ne dogodi jednaka 1 - P(A).

$$P(\text{nije A}) = 1 - P(\text{A})$$

```
# Vjerojatnost da je objava viralna
p_viral <- mean(posts$is_viral)
cat("P(viralno) =", round(p_viral, 3), "\n")
```

```
P(viralno) = 0.014
```

```
# Vjerojatnost da NIJE viralna (komplement)
p_ne_viral <- 1 - p_viral
cat("P(nije viralno) =", round(p_ne_viral, 3), "\n")
```

```
P(nije viralno) = 0.986
```

Ovo zvuči trivijalno, ali komplementarno pravilo je izuzetno korisno u praksi. Ponekad je lakše izračunati vjerojatnost da se nešto ne dogodi pa oduzeti od 1. Na primjer, ako želimo znati vjerojatnost da barem jedna od 10 objava postane viralna, lakše je izračunati vjerojatnost da nijedna ne postane viralna i oduzeti od 1.

8.4.2 Pravilo zbrajanja (ILI)

Vjerojatnost da se dogodi A ILI B ovisi o tome jesu li događaji međusobno isključivi.

Ako su **međusobno isključivi** (ne mogu se dogoditi istovremeno), jednostavno zbrajamo.

$$P(\text{A ili B}) = P(\text{A}) + P(\text{B})$$

```
# Vjerojatnost da je objava na Instagramu ILI TikToku
# (svaka objava je samo na jednoj platformi, pa su isključivi)
p_ig <- mean(posts$platform == "Instagram")
p_tt <- mean(posts$platform == "TikTok")

cat("P(Instagram) =", round(p_ig, 3), "\n")
```

```
P(Instagram) = 0.265
```

```
cat("P(TikTok) =", round(p_tt, 3), "\n")
```

```
P(TikTok) = 0.252
```

```
cat("P(Instagram ILI TikTok) =", round(p_ig + p_tt, 3), "\n")
```

```
P(Instagram ILI TikTok) = 0.518
```

```
# Provjera
mean(posts$platform %in% c("Instagram", "TikTok"))
```

```
[1] 0.5175
```

Ako **nisu međusobno isključivi** (mogu se dogoditi istovremeno), moramo oduzeti presjek jer ga inače računamo dvaput.

$$P(\text{A ili B}) = P(\text{A}) + P(\text{B}) - P(\text{A i B})$$

```
# Vjerojatnost da je objava video ILI viralna
# (objava može biti oboje istovremeno)
p_video <- mean(posts$content_type == "video")
p_viral <- mean(posts$is_viral)
p_video_i_viral <- mean(posts$content_type == "video" & posts$is_viral)

cat("P(video) =", round(p_video, 3), "\n")
```

```
P(video) = 0.224
```

```
cat("P(viralno) =", round(p_viral, 3), "\n")
```

```
P(viralno) = 0.014
```

```
cat("P(video I viralno) =", round(p_video_i_viral, 4), "\n")
```

```
P(video I viralno) = 0.009
```

```r
cat("P(video ILI viralno) =", round(p_video + p_viral - p_video_i_viral, 3), "\n")
```

```
P(video ILI viralno) = 0.23
```

```r
# Provjera
mean(posts$content_type == "video" | posts$is_viral)
```

```
[1] 0.2295
```

8.4.3 Pravilo množenja (I)

Vjerojatnost da se dogode i A i B ovisi o tome jesu li događaji nezavisni.

Ako su **nezavisni** (jedan ne utječe na drugi), koristimo formulu

$$P(\text{A i B}) = P(\text{A}) \times P(\text{B})$$

```r
# Jesu li platforma i viralnost nezavisni?
# Ako jesu, P(Instagram I viralno) = P(Instagram) * P(viralno)

p_ig <- mean(posts$platform == "Instagram")
p_viral <- mean(posts$is_viral)

cat("P(Instagram) * P(viralno) =", round(p_ig * p_viral, 4), "\n")
```

```
P(Instagram) * P(viralno) = 0.0037
```

```r
# Stvarna zajednička vjerojatnost
p_ig_viral <- mean(posts$platform == "Instagram" & posts$is_viral)
cat("P(Instagram I viralno) stvarno =", round(p_ig_viral, 4), "\n")
```

```
P(Instagram I viralno) stvarno = 0.002
```

Ako se ove dvije vrijednosti razlikuju, događaji nisu potpuno nezavisni. To znači da platforma utječe na vjerojatnost viralnosti (ili obrnuto). Ovo je važan konceptualni most. Statistički testovi koje ćemo učiti u nastavku kolegija upravo testiraju je li neka razlika rezultat zavisnosti ili čiste slučajnosti.

8.4.4 Uvjetna vjerojatnost

Uvjetna vjerojatnost je vjerojatnost jednog događaja DADO da se drugi već dogodio. Piše se P(A|B) i čita "vjerojatnost A dado B".

$$P(\text{A}|\text{B}) = \frac{P(\text{A i B})}{P(\text{B})}$$

```
# Vjerojatnost viralnosti DADO da je objava na TikToku
p_viral_dado_tt <- posts |>
  filter(platform == "TikTok") |>
  summarise(p = mean(is_viral)) |>
  pull(p)

cat("P(viralno | TikTok) =", round(p_viral_dado_tt, 4), "\n")
```

```
P(viralno | TikTok) = 0.0356
```

```
# Usporedba s ukupnom vjerojatnošću viralnosti
cat("P(viralno) ukupno =", round(mean(posts$is_viral), 4), "\n")
```

```
P(viralno) ukupno = 0.014
```

```
# Uvjetne vjerojatnosti po platformi
posts |>
  group_by(platform) |>
  summarise(
    n = n(),
    n_viral = sum(is_viral),
    p_viral = round(mean(is_viral), 4),
    .groups = "drop"
  ) |>
  arrange(desc(p_viral))
```

```
# A tibble: 6 x 4
  platform      n n_viral p_viral
  <chr>     <int>   <int>   <dbl>
1 TikTok      505      18  0.0356
2 YouTube     230       4  0.0174
3 Instagram   530       4  0.0075
4 Facebook    361       2  0.0055
5 LinkedIn    127       0  0
6 Twitter/X   247       0  0
```

Ako je P(viralno | TikTok) različit od P(viralno), to znači da platforma i viralnost nisu nezavisni. Ovo je temelj za testove koje ćemo raditi u tjednu 11 (hi-kvadrat test) gdje ćemo formalno testirati jesu li kategoričke varijable nezavisne.

Praktični savjet

U komunikologiji, uvjetna vjerojatnost je sveprisutna. Kolika je vjerojatnost konverzije DADO da je korisnik kliknuo na oglas? Kolika je vjerojatnost otvaranja emaila DADO da je poslan utorkom ujutro? Kolika je vjerojatnost dijeljenja DADO da je sadržaj video? Kad god analizirate performanse po segmentima, zapravo računate uvjetne vjerojatnosti.

8.5 Distribucije vjerojatnosti: od podataka do modela

Do sada smo računali vjerojatnosti iz stvarnih podataka (empirijske vjerojatnosti). Ali u statistici koristimo i **teorijske distribucije** koje opisuju kakvi bi podaci trebali izgledati pod određenim pretpostavkama. Dvije najvažnije su binomna i normalna distribucija.

Zašto nam trebaju teorijske distribucije? Zato što nam omogućuju izračun vjerojatnosti za događaje koje nismo opazili. Iz naših podataka možemo izračunati da je 1.4% objava viralno. Ali što ako želimo znati kolika je vjerojatnost da od sljedećih 100 objava točno 5 bude viralno? Ili kolika je vjerojatnost da nijedna ne bude viralna? Za te izračune koristimo distribuciju vjerojatnosti.

8.6 Binomna distribucija

Binomna distribucija opisuje broj uspjeha u fiksnom broju nezavisnih pokušaja, gdje svaki pokušaj ima istu vjerojatnost uspjeha. Ovo je jedna od najvažnijih distribucija u statistici jer modelira mnogo realnih situacija.

Zamislite da imate 20 objava na Instagramu i svaka ima istu vjerojatnost od 2% da postane viralna (nezavisno jedna od druge). Koliko ćete viralnih objava imati? Možda 0. Možda 1. Možda 2. Teorijski čak i svih 20, ali to je izuzetno malo vjerovatno. Binomna distribucija nam daje točnu vjerojatnost za svaki od tih ishoda.

8.6.1 Parametri binomne distribucije

Binomna distribucija ima dva parametra.

n je broj pokušaja (u našem primjeru, 20 objava). **p** je vjerojatnost uspjeha u jednom pokušaju (u našem primjeru, 0.02 ili 2%).

Piše se X ~ Binomial(n, p) i čita "X slijedi binomnu distribuciju s n pokušaja i vjerojatnošću p".

8.6.2 Izračun u R-u: dbinom()

Funkcija `dbinom(x, size, prob)` daje točnu vjerojatnost da dobijemo točno x uspjeha od size pokušaja s vjerojatnošću prob.

```
# Vjerojatnost da NIJEDNA od 20 objava ne postane viralna
dbinom(x = 0, size = 20, prob = 0.02)
```

```
[1] 0.667608
```

```
# Vjerojatnost da TOČNO 1 od 20 postane viralna
dbinom(x = 1, size = 20, prob = 0.02)
```

```
[1] 0.272493
```

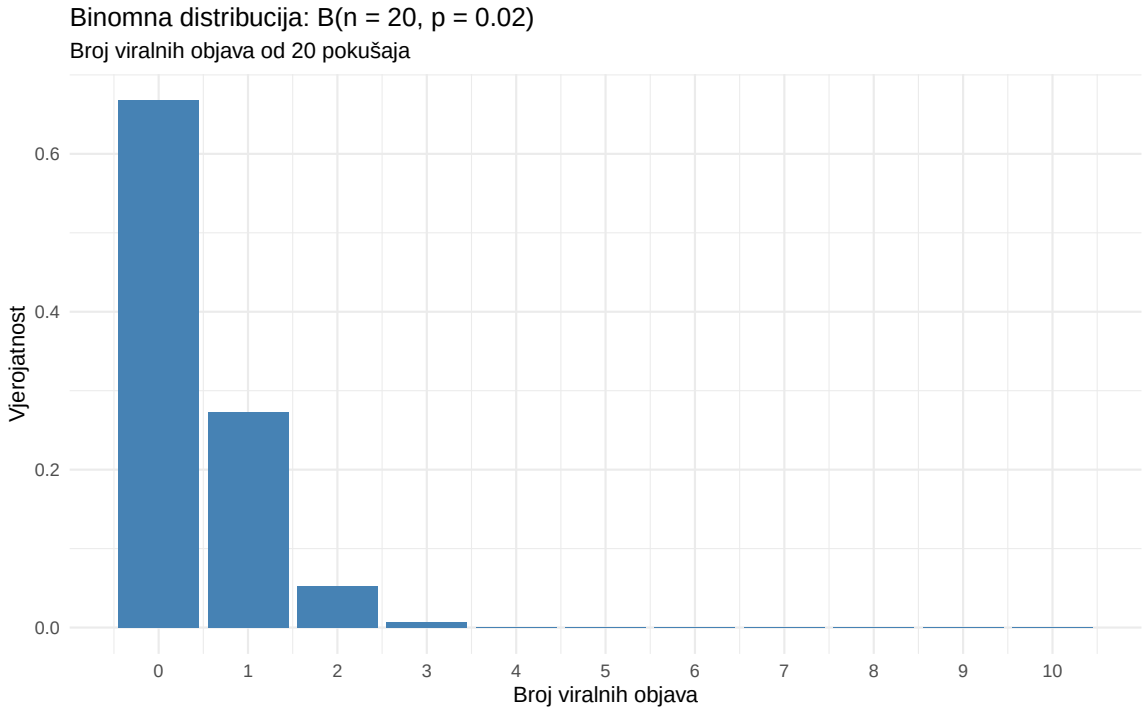
```
# Vjerojatnost da TOČNO 2 od 20 postanu viralne
dbinom(x = 2, size = 20, prob = 0.02)
```

```
[1] 0.05283029
```

Šansa da nijedna objava ne postane viralna je oko 67%. Šansa za točno jednu viralnu je oko 27%. Šansa za točno dvije je oko 5%. Brzo pada. Ovo ima smisla jer kad je vjerojatnost uspjeha samo 2%, većinu vremena nećete imati nijedan uspjeh u 20 pokušaja.

8.6.3 Vizualizacija binomne distribucije

```r
tibble(
  x = 0:10,
  vjerojatnost = dbinom(x, size = 20, prob = 0.02)
) |>
  ggplot(aes(x = x, y = vjerojatnost)) +
  geom_col(fill = "steelblue") +
  scale_x_continuous(breaks = 0:10) +
  labs(
    title = "Binomna distribucija: B(n = 20, p = 0.02)",
    subtitle = "Broj viralnih objava od 20 pokušaja",
    x = "Broj viralnih objava",
    y = "Vjerojatnost"
  ) +
  theme_minimal()
```



Graf jasno pokazuje da je najvjerojatniji ishod 0 viralnih objava, zatim 1, zatim 2. Ishodi s 3 ili više su tako malo vjerovatni da ih jedva vidimo.

8.6.4 Kumulativna vjerojatnost: pbinom()

Funkcija `pbinom(q, size, prob)` daje kumulativnu vjerojatnost, odnosno P(X q). To je vjerojatnost da dobijemo q ili manje uspjeha.

```r
# P(X <= 1): vjerojatnost 0 ili 1 viralne objave od 20
pbinom(q = 1, size = 20, prob = 0.02)
```

```
[1] 0.940101
```

```r
# P(X <= 3): vjerojatnost 3 ili manje
pbinom(q = 3, size = 20, prob = 0.02)
```

```
[1] 0.9994003
```

```r
# P(X >= 2): vjerojatnost 2 ili više (komplement od P(X <= 1))
1 - pbinom(q = 1, size = 20, prob = 0.02)
```

```
[1] 0.05989898
```

Vjerojatnost da ćemo imati jednu ili nijednu viralnu objavu je oko 94%. Vjerojatnost da ćemo imati barem 2 je samo oko 6%. Ovi izračuni su ključni za postavljanje realnih očekivanja u digitalnom marketingu.

8.6.5 Simulacija: rbinom()

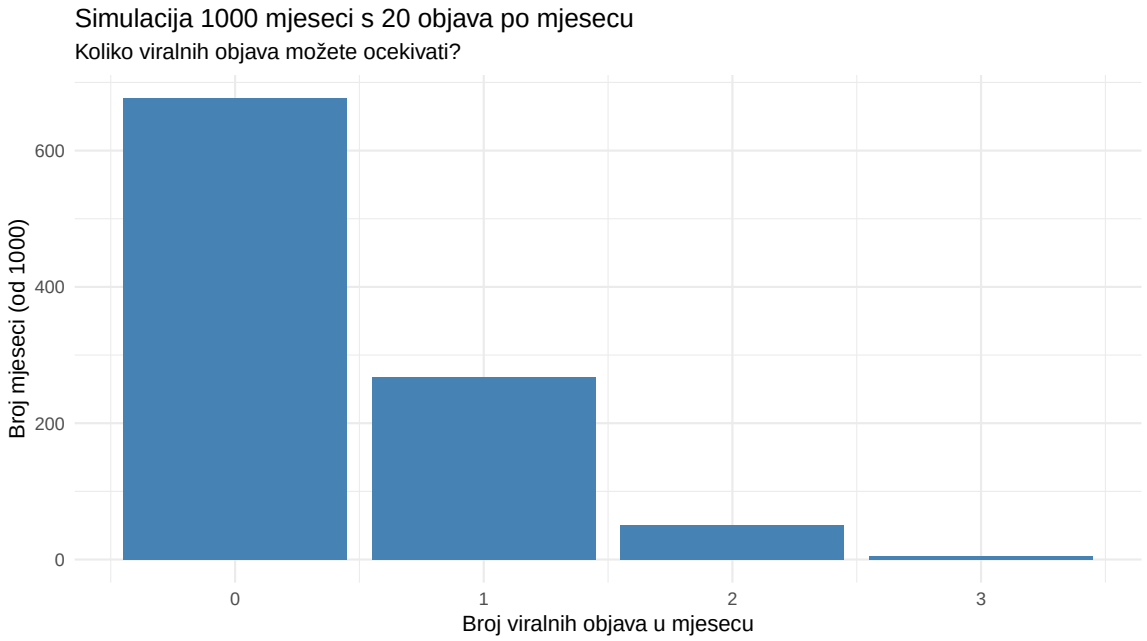
Funkcija `rbinom(n, size, prob)` generira slučajne uzorke iz binomne distribucije. Ovo je korisno za simulaciju scenarija.

```r
set.seed(42)

# Simuliraj 1000 "mjeseci" u kojima imaš po 20 objava
simulacija <- tibble(
  mjesec = 1:1000,
  n_viralnih = rbinom(n = 1000, size = 20, prob = 0.02)
)

# Distribucija rezultata
simulacija |>
  count(n_viralnih) |>
  mutate(udio = round(n / sum(n), 3))
```

```
# A tibble: 4 x 3
  n_viralnih     n  udio
       <int> <int> <dbl>
1          0   677 0.677
```

```
2           1   268 0.268
3           2    50 0.05
4           3     5 0.005
```

```
simulacija |>
  ggplot(aes(x = n_viralnih)) +
  geom_bar(fill = "steelblue") +
  scale_x_continuous(breaks = 0:8) +
  labs(
    title = "Simulacija 1000 mjeseci s 20 objava po mjesecu",
    subtitle = "Koliko viralnih objava možete očekivati?",
    x = "Broj viralnih objava u mjesecu",
    y = "Broj mjeseci (od 1000)"
  ) +
  theme_minimal()
```

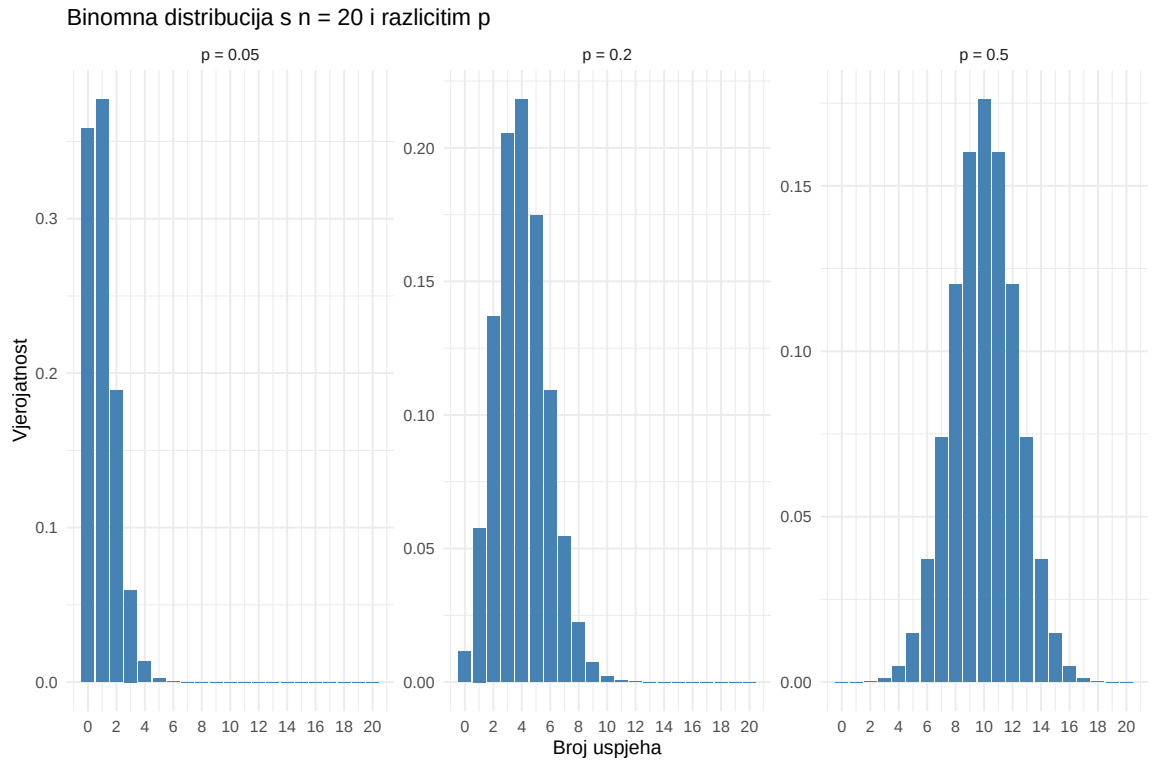


Simulacija potvrđuje teoriju. U većini mjeseci nećete imati nijednu viralnu objavu. Ponekad jednu. Rijetko dvije. I vrlo rijetko tri ili više. Ovo je korisna informacija za postavljanje KPI-jeva (key performance indicators) u komunikacijskim kampanjama.

8.6.6 Kako p mijenja distribuciju

Pogledajmo kako se distribucija mijenja s različitim vjerojatnostima uspjeha.

```r
# Tri različite vjerojatnosti uspjeha
expand_grid(
  p = c(0.05, 0.20, 0.50),
  x = 0:20
) |>
  mutate(
    vjerojatnost = dbinom(x, size = 20, prob = p),
    p_label = paste0("p = ", p)
  ) |>
  ggplot(aes(x = x, y = vjerojatnost)) +
  geom_col(fill = "steelblue") +
  facet_wrap(~p_label, scales = "free_y") +
  scale_x_continuous(breaks = seq(0, 20, by = 2)) +
  labs(
    title = "Binomna distribucija s n = 20 i različitim p",
    x = "Broj uspjeha",
    y = "Vjerojatnost"
  ) +
  theme_minimal()
```



Kad je p malo (0.05), distribucija je jako iskrivljena udesno i koncentrirana oko 0 i 1. Kad je p umjereno (0.20), distribucija se širi i centar se pomiče udesno. Kad je $p = 0.50$, distribucija

je simetrična i izgleda gotovo poput zvona. Ova simetrija kod p = 0.5 nas vodi prema najvažnijoj distribuciji u cijeloj statistici, a to je normalna.

8.6.7 Primjena: A/B test emaila

Zamislite da testirate dva naslova za newsletter. Naslov A ima open rate 22%, naslov B ima 28%. Poslali ste svaki naslov na uzorak od 50 pretplatnika. Naslov B je imao 14 otvaranja od 50, dok je A imao 11. Je li ovo uvjerljiva razlika?

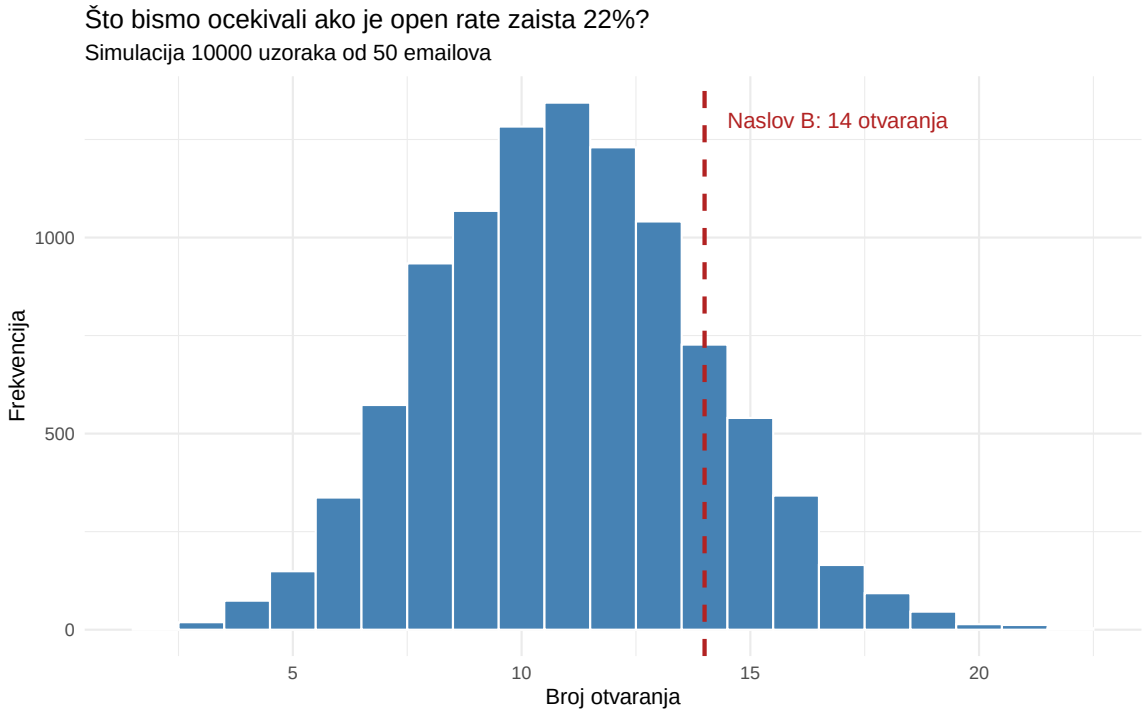
```
# Ako je B zaista isti kao A (p = 0.22), kolika je šansa da vidimo 14 ili više otvaranja?
p_14_ili_vise <- 1 - pbinom(q = 13, size = 50, prob = 0.22)
cat("P(X >= 14 | p = 0.22) =", round(p_14_ili_vise, 3), "\n")
```

```
P(X >= 14 | p = 0.22) = 0.194
```

Šansa da vidimo 14 ili više otvaranja ako je pravi open rate samo 22% iznosi oko 19%. To nije zanemarivo. Ne možemo s velikom sigurnošću tvrditi da je B bolji samo na temelju ovog jednog uzorka.

Ovo je srž statističkog razmišljanja i prethodnica formalnih testova hipoteza koje ćemo učiti u tjednu 10. Pitanje je uvijek isto — koliko je vjerovatno vidjeti ovakav ili ekstremniji rezultat čistom slučajnošću?

```
set.seed(42)

# Simulacija: ako je pravi open rate 22%, koliko otvaranja bismo dobili u 50 emailova?
sim_a <- tibble(
  otvaranja = rbinom(10000, size = 50, prob = 0.22)
)

sim_a |>
  ggplot(aes(x = otvaranja)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "white") +
  geom_vline(xintercept = 14, color = "firebrick", linetype = "dashed", linewidth = 1) +
  annotate("text", x = 14.5, y = 1300, label = "Naslov B: 14 otvaranja",
           hjust = 0, color = "firebrick") +
  labs(
    title = "Što bismo očekivali ako je open rate zaista 22%?",
    subtitle = "Simulacija 10000 uzoraka od 50 emailova",
    x = "Broj otvaranja",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



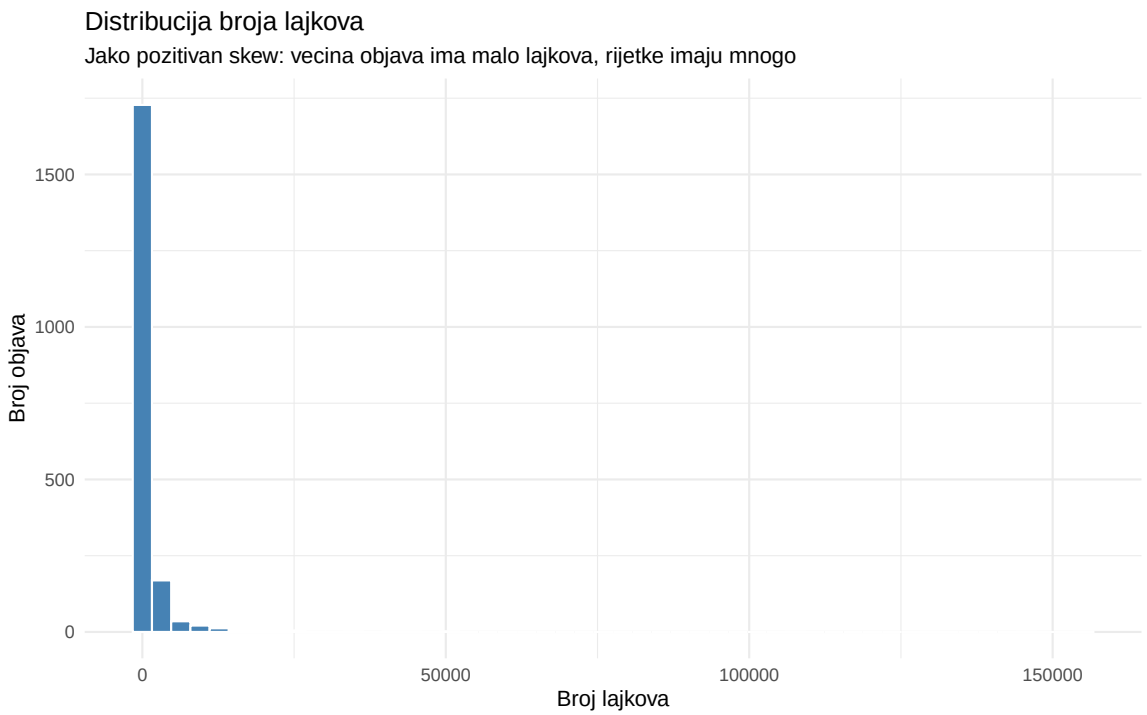
Crvena crta pokazuje rezultat naslova B (14 otvaranja). Vidimo da je to na desnom repu distribucije ali nije ekstreman rezultat. Dobar dio simuliranih uzoraka ima 14 ili više otvaranja čak i kad je pravi open rate samo 22%. Ovo sugerira da razlika možda nije statistički značajna. Trebamo ili veći uzorak ili veću razliku da bismo donijeli sigurnije zaključke.

> **!** Važna napomena
>
> Binomna distribucija se primjenjuje kad imate fiksni broj nezavisnih pokušaja, svaki s istom vjerojatnošću uspjeha. Primjeri iz komunikologije uključuju broj lajkova (svaki pratitelj nezavisno odlučuje), broj otvaranja emaila (svaki pretplatnik nezavisno odlučuje), broj konverzija na landing stranici (svaki posjetitelj nezavisno odlučuje). Pretpostavka nezavisnosti je važna jer ako jedan lajk poveća vidljivost pa uzrokuje sljedeći lajk (viralnost), stroga binomna pretpostavka je narušena.

8.7 Distribucija u stvarnim podacima

Pogledajmo distribucije u našem datasetu i povežimo ih s teorijskim konceptima.

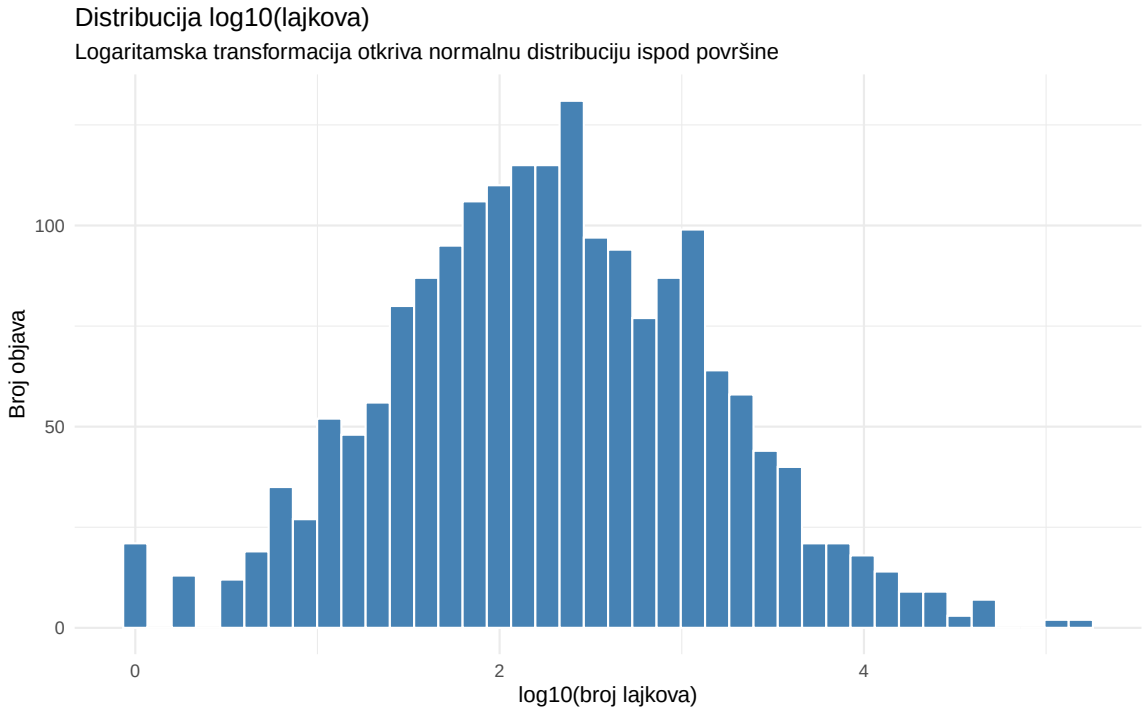
```
posts |>
  ggplot(aes(x = likes)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 50) +
  labs(
    title = "Distribucija broja lajkova",
    subtitle = "Jako pozitivan skew: većina objava ima malo lajkova, rijetke imaju mnogo",
    x = "Broj lajkova",
    y = "Broj objava"
  ) +
  theme_minimal()
```



Ova distribucija je jako iskrivljena udesno. Većina objava ima relativno malo lajkova, ali postoji dugačak rep objava s tisućama ili stotinama tisuća lajkova. Ovo je tipično za metrike angažmana na društvenim mrežama i zove se **power law** ili **log-normalna** distribucija.

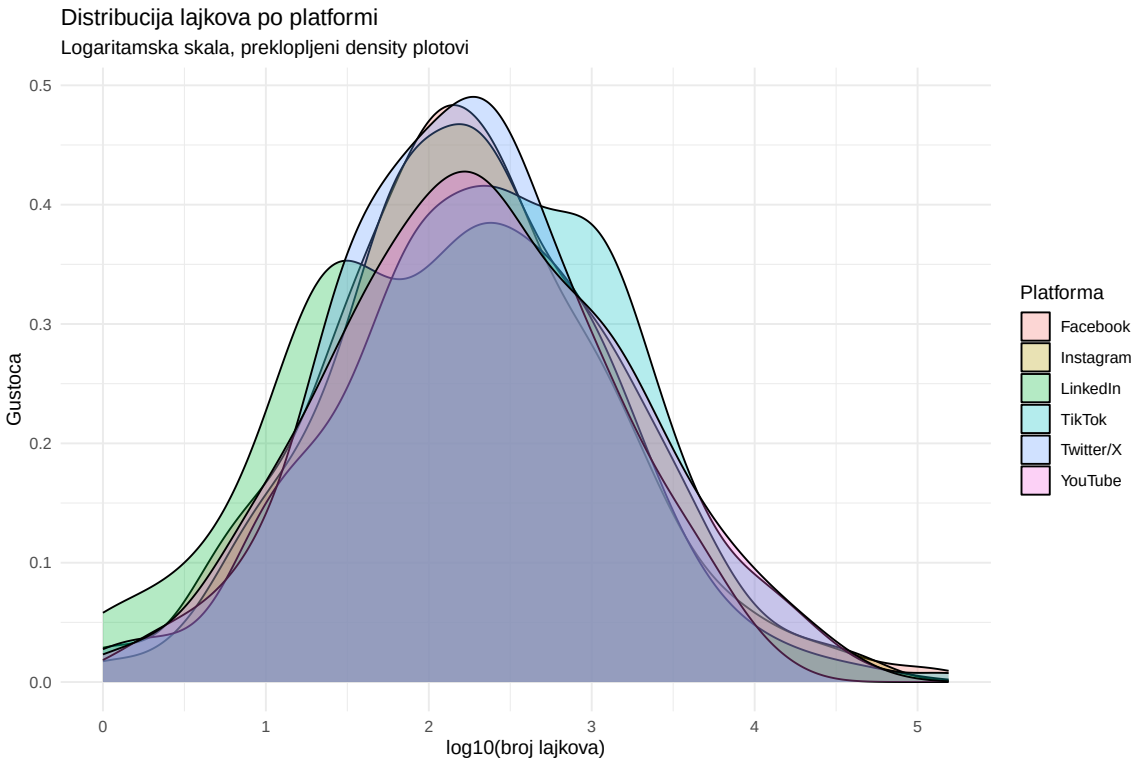
```
# Logaritmirana distribucija izgleda puno "normalnije"
posts |>
  filter(likes > 0) |>
  ggplot(aes(x = log10(likes))) +
  geom_histogram(fill = "steelblue", color = "white", bins = 40) +
  labs(
    title = "Distribucija log10(lajkova)",
    subtitle = "Logaritamska transformacija otkriva normalnu distribuciju ispod površine",
    x = "log10(broj lajkova)",
```

```
    y = "Broj objava"
  ) +
  theme_minimal()
```



Kad primijenimo logaritamsku transformaciju, distribucija počinje nalikovati na zvonoliku krivulju. Ovo je važno zapažanje jer mnoge varijable u komunikologiji koje izgledaju neprirodno iskrivljene zapravo su log-normalno distribuirane. Na logaritamskoj skali, postaju normalne. Normalnu distribuciju ćemo detaljno obraditi u drugom dijelu predavanja.

```
posts |>
  filter(likes > 0) |>
  ggplot(aes(x = log10(likes), fill = platform)) +
  geom_density(alpha = 0.3) +
  labs(
    title = "Distribucija lajkova po platformi",
    subtitle = "Logaritamska skala, preklopljeni density plotovi",
    x = "log10(broj lajkova)",
    y = "Gustoća",
    fill = "Platforma"
  ) +
  theme_minimal()
```



Platforme se razlikuju po distribuciji angažmana. YouTube i TikTok imaju širu distribuciju (veća varijabilnost, češće ekstremne vrijednosti), dok LinkedIn ima užu i pomaknutu ulijevo (manji ali konzistentniji angažman). Ovo odražava fundamentalne razlike u mehanici platformi.

Podsjetnik

U prvom dijelu naučili smo osnovna pravila vjerojatnosti (komplement, zbrajanje, množenje, uvjetna vjerojatnost) i binomnu distribuciju za modeliranje diskretnih ishoda (uspjeh/neuspjeh). U ovom dijelu prelazimo na kontinuirane varijable i upoznajemo najvažniju distribuciju u cijeloj statistici, a to je normalna.

8.8 Normalna distribucija

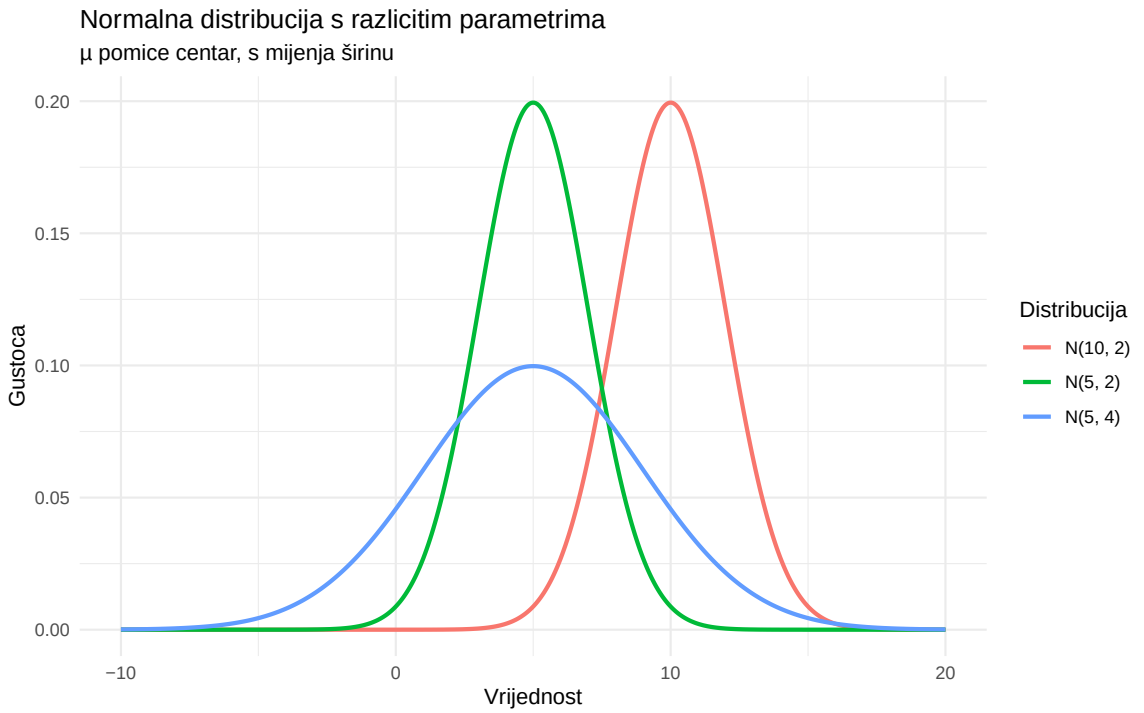
Normalna distribucija (ili Gaussova krivulja, ili zvonolika krivulja) je najvažnija distribucija u statistici. Razlog nije samo u tome što mnoge varijable u prirodi imaju približno normalan oblik. Važniji razlog je **centralni granični teorem** koji kaže da prosjek dovoljno velikog uzorka ima približno normalnu distribuciju, neovisno o obliku izvorne distribucije. Ovo čini normalnu distribuciju temeljom gotovo svih statističkih testova.

8.8.1 Parametri normalne distribucije

Normalna distribucija ima dva parametra. **(mi)** je srednja vrijednost (prosjek), koja određuje centar distribucije. **(sigma)** je standardna devijacija, koja određuje širinu distribucije.

Piše se X ~ N(,) i čita "X slijedi normalnu distribuciju s prosjekom i standardnom devijacijom ."

```
# Vizualizacija normalne distribucije s različitim parametrima
tibble(x = seq(-10, 20, length.out = 500)) |>
  mutate(
    `N(5, 2)` = dnorm(x, mean = 5, sd = 2),
    `N(5, 4)` = dnorm(x, mean = 5, sd = 4),
    `N(10, 2)` = dnorm(x, mean = 10, sd = 2)
  ) |>
  pivot_longer(-x, names_to = "distribucija", values_to = "gustoca") |>
  ggplot(aes(x = x, y = gustoca, color = distribucija)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Normalna distribucija s različitim parametrima",
    subtitle = "  pomiče centar,  mijenja širinu",
    x = "Vrijednost",
    y = "Gustoća",
    color = "Distribucija"
  ) +
  theme_minimal()
```



N(5, 2) i N(10, 2) imaju istu širinu ($= 2$) ali različite centre ($= 5$ vs $= 10$). N(5, 2) i N(5, 4) imaju isti centar ali različite širine. Veća standardna devijacija znači širu, plošniju krivulju. Manja znači užu i višu.

8.8.2 Pravilo 68-95-99.7

Jedno od najkorisnijih svojstava normalne distribucije je da uvijek isti postotak podataka pada unutar istog broja standardnih devijacija od prosjeka.

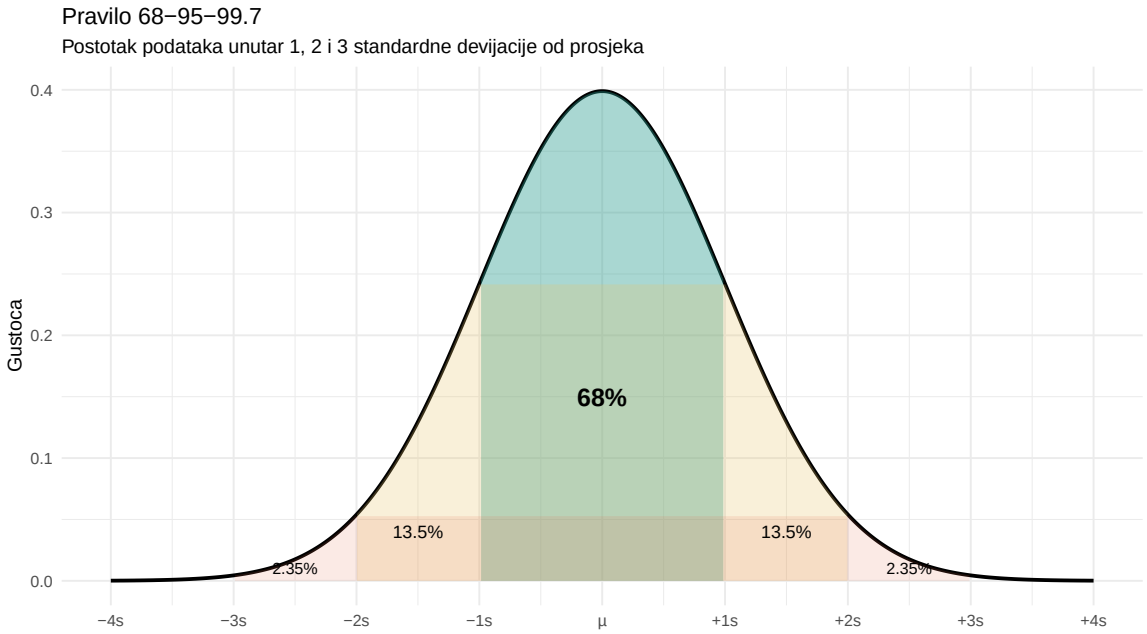
Otprilike **68%** podataka je unutar 1 standardne devijacije od prosjeka (± 1).

Otprilike **95%** podataka je unutar 2 standardne devijacije (± 2).

Otprilike **99.7%** podataka je unutar 3 standardne devijacije (± 3).

```
# Vizualizacija pravila 68-95-99.7
mu <- 0
sigma <- 1

norm_data <- tibble(x = seq(-4, 4, length.out = 500), y = dnorm(x, mu, sigma))

ggplot(norm_data, aes(x = x, y = y)) +
  geom_line(linewidth = 1) +
  geom_area(data = norm_data |> filter(x >= -1, x <= 1), alpha = 0.4, fill = "#2a9d8f") +
  geom_area(data = norm_data |> filter(x >= -2, x <= -1 | x >= 1, x <= 2), alpha = 0.25, f
  geom_area(data = norm_data |> filter(x >= -3, x <= -2 | x >= 2, x <= 3), alpha = 0.15, f
```

```
  annotate("text", x = 0, y = 0.15, label = "68%", size = 5, fontface = "bold") +
  annotate("text", x = 1.5, y = 0.04, label = "13.5%", size = 3.5) +
  annotate("text", x = -1.5, y = 0.04, label = "13.5%", size = 3.5) +
  annotate("text", x = 2.5, y = 0.01, label = "2.35%", size = 3) +
  annotate("text", x = -2.5, y = 0.01, label = "2.35%", size = 3) +
  scale_x_continuous(breaks = -4:4, labels = c("-4 ", "-3 ", "-2 ", "-1 ", " ", "+1 ", "+2 ",
  labs(
    title = "Pravilo 68-95-99.7",
    subtitle = "Postotak podataka unutar 1, 2 i 3 standardne devijacije od prosjeka",
    x = NULL,
    y = "Gustoća"
  ) +
  theme_minimal()
```



Ovo pravilo ima ogromnu praktičnu korist. Ako znate prosjek i standardnu devijaciju, odmah znate raspone u kojima se nalazi većina podataka. Na primjer, ako je prosječno vrijeme čitanja članka 80 sekundi sa SD od 25, tada je 95% čitatelja unutar raspona 80 $\pm$ 50 sekundi, dakle između 30 i 130 sekundi. Netko tko čita 200 sekundi je daleko izvan normalnog raspona.

8.8.3 Provjera pravila na stvarnim podacima

```
# Koristimo log-transformirane lajkove koji su približno normalni
log_likes <- posts |>
  filter(likes > 0) |>
```

```
  pull(likes) |>
  log10()

mu <- mean(log_likes)
sigma <- sd(log_likes)

cat("Prosjek log10(likes):", round(mu, 2), "\n")
```

```
Prosjek log10(likes): 2.29
```

```
cat("SD log10(likes):", round(sigma, 2), "\n\n")
```

```
SD log10(likes): 0.88
```

```
# Koliko podataka pada unutar 1, 2, 3 SD?
unutar_1sd <- mean(log_likes >= mu - sigma & log_likes <= mu + sigma)
unutar_2sd <- mean(log_likes >= mu - 2*sigma & log_likes <= mu + 2*sigma)
unutar_3sd <- mean(log_likes >= mu - 3*sigma & log_likes <= mu + 3*sigma)

cat("Unutar 1 SD:", round(unutar_1sd * 100, 1), "% (teorijski: 68%)\n")
```

```
Unutar 1 SD: 69.1 % (teorijski: 68%)
```

```
cat("Unutar 2 SD:", round(unutar_2sd * 100, 1), "% (teorijski: 95%)\n")
```

```
Unutar 2 SD: 95.1 % (teorijski: 95%)
```

```
cat("Unutar 3 SD:", round(unutar_3sd * 100, 1), "% (teorijski: 99.7%)\n")
```

```
Unutar 3 SD: 99.8 % (teorijski: 99.7%)
```

Rezultati su blizu teorijskih vrijednosti, što potvrđuje da su logaritmirani lajkovi približno normalno distribuirani. Poklapanje nije savršeno jer nijedna stvarna varijabla nije savršeno normalna, ali je dovoljno dobro za praktičnu primjenu.

8.9 Z-score: standardizacija

Z-score (standardizirani rezultat) izražava koliko je neka vrijednost udaljena od prosjeka, mjereno u standardnim devijacijama.

$$z = \frac{x - \mu}{\sigma}$$

Ako je z = 0, vrijednost je na prosjeku. Ako je z = 1, vrijednost je jednu standardnu devijaciju iznad prosjeka. Ako je z = -2, vrijednost je dvije standardne devijacije ispod prosjeka.

```
# Z-score za lajkove (na log skali)
posts_z <- posts |>
  filter(likes > 0) |>
  mutate(
    log_likes = log10(likes),
    z_likes = (log_likes - mean(log_likes)) / sd(log_likes)
  )

# Objave s najvišim z-scoreom (najneobičajniji angažman)
posts_z |>
  select(post_id, platform, content_type, likes, z_likes) |>
  arrange(desc(z_likes)) |>
  head(10)
```

```
# A tibble: 10 x 5
   post_id platform  content_type  likes z_likes
     <dbl> <chr>     <chr>         <dbl>   <dbl>
 1     595 TikTok    video        155132    3.31
 2    1756 TikTok    reel         144871    3.27
 3    1508 Facebook  tekst        129142    3.22
 4    1520 Facebook  slika        108320    3.13
 5     525 Facebook  reel          50834    2.76
 6    1970 Instagram reel          50748    2.75
 7     378 Instagram reel          46499    2.71
 8     254 TikTok    reel          44184    2.69
 9    1569 Instagram story         43805    2.68
10    1984 LinkedIn  tekst         41559    2.66
```

Objave s z-scoreom većim od 2 ili 3 su statistički neobične. U normalnoj distribuciji, samo oko 5% podataka ima z > 2 (ili z < -2), a samo 0.3% ima z > 3 (ili z < -3). Ovo čini z-score korisnim alatom za identifikaciju outliera.

8.9.1 Z-score za usporedbu nepovezanih varijabli

Velika prednost z-scorea je što omogućuje usporedbu varijabli na potpuno različitim skalama. Recimo da želite usporediti koliko je neka objava neobična po broju lajkova i po broju komentara.

```
posts_z2 <- posts |>
  filter(likes > 0, comments > 0) |>
  mutate(
    z_likes = scale(log10(likes))[,1],
    z_comments = scale(log10(comments))[,1]
  )

# Koje objave imaju neproporcionalno više komentara nego lajkova?
posts_z2 |>
  mutate(razlika = z_comments - z_likes) |>
  select(post_id, platform, content_type, likes, comments, z_likes, z_comments, razlika) |
  arrange(desc(razlika)) |>
  head(8)
```

```
# A tibble: 8 x 8
  post_id platform  content_type likes comments z_likes z_comments razlika
    <dbl> <chr>     <chr>        <dbl>    <dbl>   <dbl>      <dbl>   <dbl>
1     659 LinkedIn  tekst           35        5  -1.38      -0.579   0.796
2    1174 LinkedIn  tekst           15        2  -1.89      -1.10    0.787
3    1097 TikTok    reel           144       22  -0.523      0.263   0.785
4    1273 Instagram slika           29        4  -1.49      -0.706   0.783
5     260 Facebook  story          133       20  -0.571      0.209   0.779
6     914 TikTok    reel             8        1  -2.27      -1.49    0.772
7    1212 Instagram slika           57        8  -1.08      -0.312   0.769
8    1706 Facebook  slika          224       34  -0.256      0.510   0.766
```

Funkcija `scale()` u R-u automatski izračunava z-score (oduzima prosjek i dijeli sa SD). `[,1]` na kraju je tehnički detalj koji pretvara matricu u vektor.

Objave s velikom pozitivnom razlikom (z_comments » z_likes) su one koje su generirale neproporcionalno mnogo diskusije s obzirom na ukupni angažman. To su često kontroverzni sadržaji ili sadržaji koji potiču na odgovor. Ovo je primjer kako statistička standardizacija otkriva obrasce koji nisu očiti iz sirovih brojeva.

8.10 R funkcije za normalnu distribuciju

R ima četiri funkcije za normalnu distribuciju, organizirane prema istom obrascu kao binomna (d/p/q/r).

8.10.1 dnorm(): gustoća

```
# Gustoća u točki x za standardnu normalnu N(0,1)
dnorm(0)       # Gustoća na prosjeku (vrh krivulje)
```

```
[1] 0.3989423
```

```
dnorm(1)       # Gustoća na 1 SD iznad prosjeka
```

```
[1] 0.2419707
```

```
dnorm(2)       # Gustoća na 2 SD iznad prosjeka
```

```
[1] 0.05399097
```

```
# Gustoća za nestandardnu normalnu
dnorm(100, mean = 80, sd = 25)  # Koliko je "normalan" rezultat od 100 sekundi?
```

```
[1] 0.01158766
```

Za razliku od binomne, `dnorm()` ne daje vjerojatnost nego gustoću. U kontinuiranoj distribuciji, vjerojatnost jedne specifične vrijednosti je 0 (jer postoji beskonačno mnogo mogućih vrijednosti). Gustoća nam govori koliko je ta vrijednost relativno česta u odnosu na druge.

8.10.2 pnorm(): kumulativna vjerojatnost

`pnorm()` je najkorisnija od četiri funkcije. Daje vjerojatnost P(X x), odnosno postotak distribucije koji je ispod zadane vrijednosti.

```
# Za standardnu normalnu N(0,1):
pnorm(0)    # 50% distribucije je ispod prosjeka
```

```
[1] 0.5
```

```r
pnorm(1)    # ~84% je ispod 1 SD iznad prosjeka
```

```
[1] 0.8413447
```

```r
pnorm(-1)   # ~16% je ispod 1 SD ispod prosjeka
```

```
[1] 0.1586553
```

```r
# Koliki postotak čitatelja provede manje od 60 sekundi?
# (ako je vrijeme ~ N(80, 25))
pnorm(60, mean = 80, sd = 25)
```

```
[1] 0.2118554
```

Oko 21% čitatelja bi provelo manje od 60 sekundi, pod pretpostavkom normalne distribucije s prosjekom 80 i SD 25.

```r
# Postotak čitatelja između 60 i 120 sekundi
pnorm(120, mean = 80, sd = 25) - pnorm(60, mean = 80, sd = 25)
```

```
[1] 0.7333453
```

```r
# Postotak čitatelja iznad 150 sekundi (gornji rep)
1 - pnorm(150, mean = 80, sd = 25)
```

```
[1] 0.00255513
```

```r
# Postotak s z-scoreom između -1 i 1 (provjera pravila 68%)
pnorm(1) - pnorm(-1)
```

```
[1] 0.6826895
```

Razlika `pnorm(120, ...) - pnorm(60, ...)` daje vjerojatnost između dva praga. Komplement `1 - pnorm(...)` daje gornji rep. Ovi izračuni su temelj za p-vrijednosti koje ćemo učiti u tjednu 10.

8.10.3 qnorm(): kvantili (obrnuta funkcija)

`qnorm()` je obrnuta funkcija od `pnorm()`. Daje vrijednost ispod koje se nalazi zadani postotak distribucije.

```r
# Ispod koje vrijednosti je 95% distribucije?
qnorm(0.95, mean = 80, sd = 25)
```

```
[1] 121.1213
```

```r
# Ispod koje je 5%? (donji kvintil)
qnorm(0.05, mean = 80, sd = 25)
```

```
[1] 38.87866
```

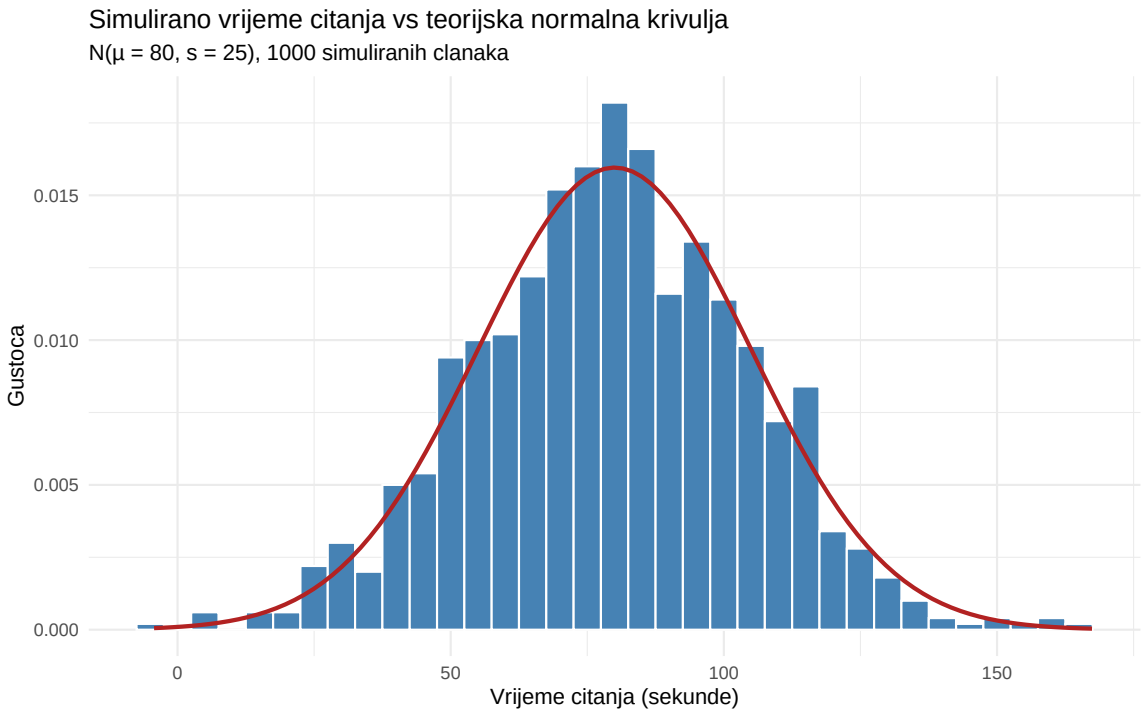
```r
# Top 1% čitatelja (oni koji čitaju najduže)
qnorm(0.99, mean = 80, sd = 25)
```

```
[1] 138.1587
```

Top 1% najdediciranijih čitatelja provodi više od 138 sekundi na članku. Ovo je korisno za postavljanje pragova. Na primjer, možete definirati "super čitatelja" kao onoga koji je u top 5% po vremenu čitanja.

8.10.4 rnorm(): simulacija

```r
set.seed(42)

# Simulacija 1000 članaka s prosječnim vremenom čitanja ~ N(80, 25)
sim_vrijeme <- tibble(
  clanak = 1:1000,
  vrijeme = rnorm(1000, mean = 80, sd = 25)
)

sim_vrijeme |>
  ggplot(aes(x = vrijeme)) +
  geom_histogram(aes(y = after_stat(density)),
                 fill = "steelblue", color = "white", binwidth = 5) +
  stat_function(fun = dnorm, args = list(mean = 80, sd = 25),
                color = "firebrick", linewidth = 1) +
  labs(
    title = "Simulirano vrijeme čitanja vs teorijska normalna krivulja",
    subtitle = "N(  = 80,   = 25), 1000 simuliranih članaka",
    x = "Vrijeme čitanja (sekunde)",
    y = "Gustoća"
  ) +
  theme_minimal()
```



Crvena krivulja je teorijska normalna distribucija. Histogram prikazuje simulirane podatke. Poklapanje je dobro, posebno u sredini distribucije. Na repovima uvijek postoji malo odstupanja jer je uzorak konačan.

Funkcija `stat_function()` je elegantan način za dodavanje teorijske krivulje na ggplot graf. Prima ime distribucijske funkcije i njezine argumente.

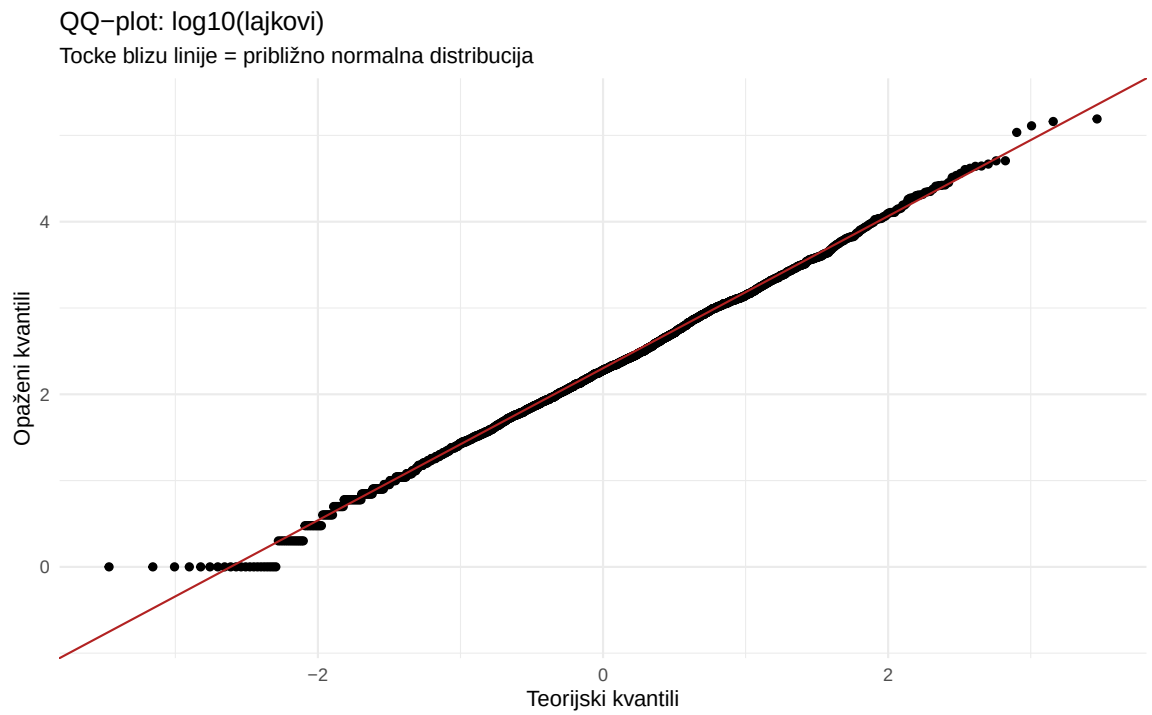
> **Praktični savjet**
>
> Obrazac d/p/q/r vrijedi za sve distribucije u R-u. `d` daje gustoću (ili vjerojatnost za diskretne), `p` daje kumulativnu vjerojatnost, `q` daje kvantile (obrnuto od p), `r` generira slučajne uzorke. Za binomnu: `dbinom`, `pbinom`, `qbinom`, `rbinom`. Za normalnu: `dnorm`, `pnorm`, `qnorm`, `rnorm`. Za t-distribuciju (tjedan 12): `dt`, `pt`, `qt`, `rt`. Naučite obrazac jednom, primijenite svugdje.

8.11 QQ-plot: je li moja varijabla normalno distribuirana?

Vizualna provjera normalnosti je važan korak u mnogim analizama jer mnogi statistički testovi pretpostavljaju (približno) normalnu distribuciju. QQ-plot (quantile-quantile plot) je standardni alat za ovu provjeru.

QQ-plot uspoređuje kvantile vaših podataka s kvantilima teorijske normalne distribucije. Ako su podaci normalno distribuirani, točke padaju na ravnu liniju. Odstupanja od linije ukazuju na nenormalnost.

```
# QQ-plot za log-transformirane lajkove (trebali bi biti približno normalni)
posts |>
  filter(likes > 0) |>
  ggplot(aes(sample = log10(likes))) +
  stat_qq() +
  stat_qq_line(color = "firebrick") +
  labs(
    title = "QQ-plot: log10(lajkovi)",
    subtitle = "Točke blizu linije = približno normalna distribucija",
    x = "Teorijski kvantili",
    y = "Opaženi kvantili"
  ) +
  theme_minimal()
```

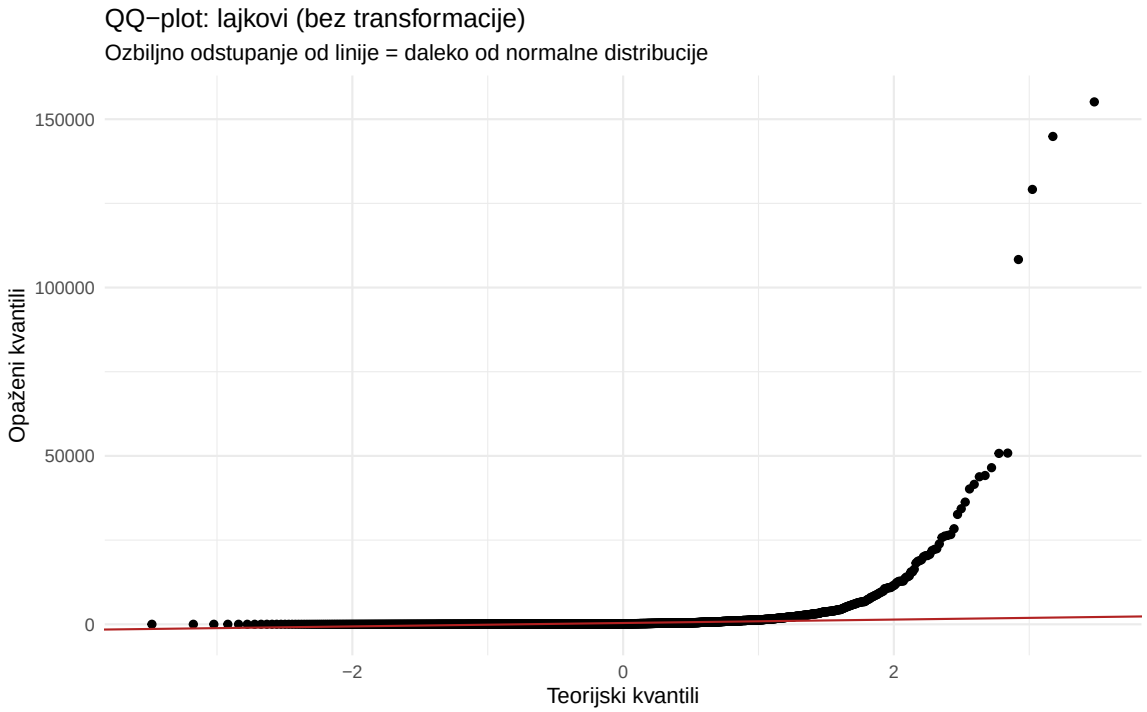


Točke uglavnom prate crvenu liniju, s malim odstupanjima na repovima. Ovo je tipičan rezultat za stvarne podatke i smatra se prihvatljivo normalnim za većinu statističkih testova.

Za usporedbu, pogledajmo QQ-plot za netransformirane lajkove.

```
# QQ-plot za netransformirane lajkove (jako iskrivljeni)
posts |>
```

```
ggplot(aes(sample = likes)) +
stat_qq() +
stat_qq_line(color = "firebrick") +
labs(
  title = "QQ-plot: lajkovi (bez transformacije)",
  subtitle = "Ozbiljno odstupanje od linije = daleko od normalne distribucije",
  x = "Teorijski kvantili",
  y = "Opaženi kvantili"
) +
theme_minimal()
```

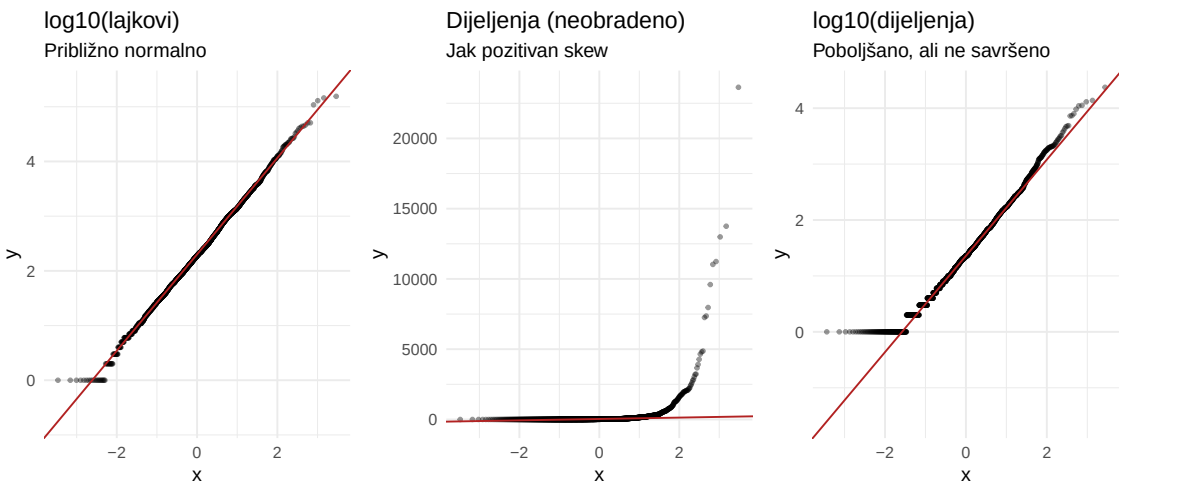


Razlika je dramatična. Netransformirani lajkovi jako odstupaju od normalne distribucije. Desni rep se savija strmo prema gore, što znači da postoje mnogo veće vrijednosti nego što bi normalna distribucija predvidjela. Ovo je vizualni potpis za pozitivno iskrivljenu (right-skewed) distribuciju.

8.11.1 Čitanje QQ-plota

Različiti obrasci na QQ-plotu govore različite priče. Točke na ravnoj liniji znače normalnu distribuciju. Točke koje se savijaju prema gore na desnom kraju znače pozitivan skew (dugačak desni rep). Točke koje se savijaju prema dolje na lijevom kraju znače negativan skew (dugačak lijevi rep). Točke koje se savijaju prema gore na oba kraja (oblik slova S) znače "teže repove" od normalne distribucije (više ekstrema nego što normalna predviđa).

```
library(patchwork)

p_qq1 <- posts |>
  filter(likes > 0) |>
  ggplot(aes(sample = log10(likes))) +
  stat_qq(size = 0.8, alpha = 0.4) +
  stat_qq_line(color = "firebrick") +
  labs(title = "log10(lajkovi)", subtitle = "Približno normalno") +
  theme_minimal()

p_qq2 <- posts |>
  ggplot(aes(sample = shares)) +
  stat_qq(size = 0.8, alpha = 0.4) +
  stat_qq_line(color = "firebrick") +
  labs(title = "Dijeljenja (neobrađeno)", subtitle = "Jak pozitivan skew") +
  theme_minimal()

p_qq3 <- posts |>
  filter(shares > 0) |>
  ggplot(aes(sample = log10(shares))) +
  stat_qq(size = 0.8, alpha = 0.4) +
  stat_qq_line(color = "firebrick") +
  labs(title = "log10(dijeljenja)", subtitle = "Poboljšano, ali ne savršeno") +
  theme_minimal()

p_qq1 + p_qq2 + p_qq3
```



Usporedba tri QQ-plota pokazuje transformacijsku strategiju. Sirovi podaci o dijeljenjima su daleko od normalnih. Log-transformacija ih značajno približava normalnosti, ali ne savršeno.

U praksi, “dovoljno normalno” je uglavnom prihvatljivo za statističke testove, posebno s velikim uzorcima.

8.12 Praktična primjena: postavljanje pragova i identifikacija outliera

Normalna distribucija i z-score daju nam objektivne alate za donošenje odluka koje bi inače bile proizvoljne.

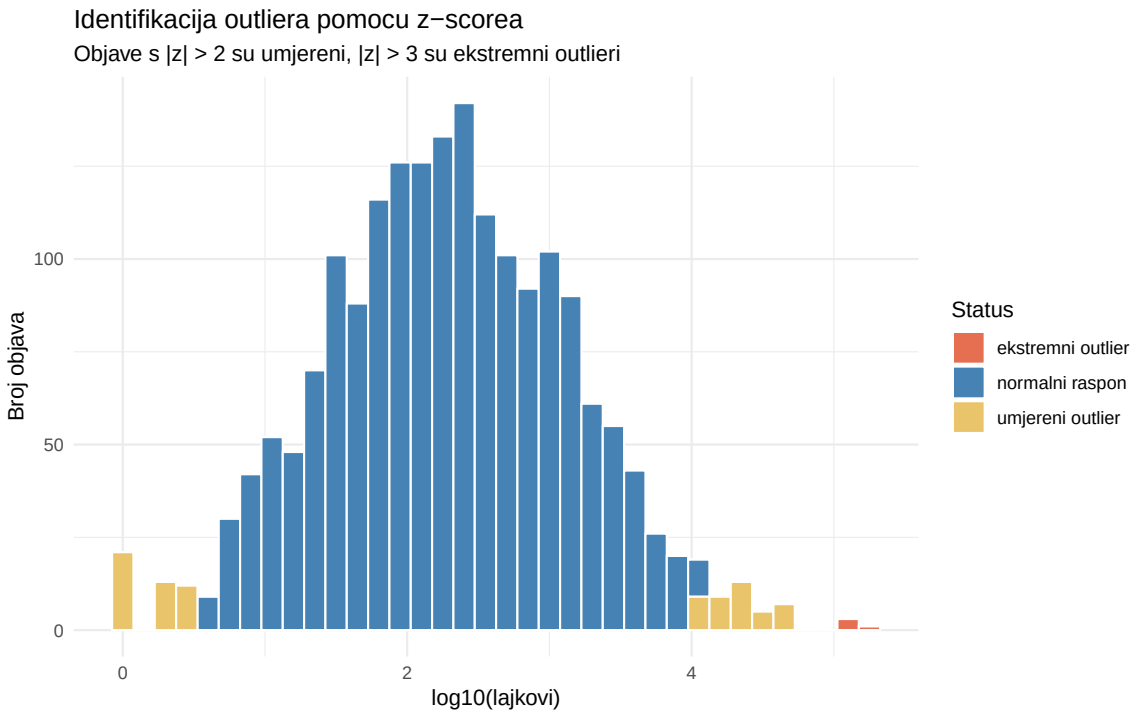
8.12.1 Definiranje “neobičnog” rezultata

```
# Identifikacija outliera u engagement metrikama
posts_analiza <- posts |>
  filter(likes > 0) |>
  mutate(
    log_likes = log10(likes),
    z_likes = scale(log_likes)[,1],
    outlier_status = case_when(
      abs(z_likes) > 3 ~ "ekstremni outlier",
      abs(z_likes) > 2 ~ "umjereni outlier",
      .default = "normalni raspon"
    )
  )

posts_analiza |>
  count(outlier_status) |>
  mutate(udio = round(n / sum(n), 3))
```

```
# A tibble: 3 x 3
  outlier_status        n  udio
  <chr>             <int> <dbl>
1 ekstremni outlier     4 0.002
2 normalni raspon    1795 0.951
3 umjereni outlier     89 0.047
```

```
posts_analiza |>
  ggplot(aes(x = log_likes, fill = outlier_status)) +
  geom_histogram(binwidth = 0.15, color = "white") +
  scale_fill_manual(values = c(
```

```
    "normalni raspon" = "steelblue",
    "umjereni outlier" = "#e9c46a",
    "ekstremni outlier" = "#e76f51"
  )) +
  labs(
    title = "Identifikacija outliera pomoću z-scorea",
    subtitle = "Objave s |z| > 2 su umjereni, |z| > 3 su ekstremni outlieri",
    x = "log10(lajkovi)",
    y = "Broj objava",
    fill = "Status"
  ) +
  theme_minimal()
```



8.12.2 Planiranje: kolika je šansa za uspjeh kampanje?

Normalna distribucija omogućuje izračun vjerojatnosti za buduće kampanje na temelju povijesnih podataka.

```
# Pretpostavimo da je open rate newsletter kampanja ~ N(0.25, 0.07)
# (prosjek 25%, SD 7%)

# Kolika je vjerojatnost da kampanja ima open rate iznad 30%?
```

```r
p_iznad_30 <- 1 - pnorm(0.30, mean = 0.25, sd = 0.07)
cat("P(open rate > 30%) =", round(p_iznad_30, 3), "\n")
```

```
P(open rate > 30%) = 0.238
```

```r
# Kolika je vjerojatnost da padne ispod 15%? (loš rezultat)
p_ispod_15 <- pnorm(0.15, mean = 0.25, sd = 0.07)
cat("P(open rate < 15%) =", round(p_ispod_15, 3), "\n")
```

```
P(open rate < 15%) = 0.077
```

```r
# Koji open rate je na granici top 10% kampanja?
top_10 <- qnorm(0.90, mean = 0.25, sd = 0.07)
cat("Prag za top 10%:", round(top_10 * 100, 1), "%\n")
```

```
Prag za top 10%: 34 %
```

Ovi izračuni omogućuju objektivno postavljanje ciljeva. Umjesto proizvoljnog "cilj nam je 30% open rate", možemo reći "30% open rate je u top 24% naših kampanja, što je ambiciozan ali realan cilj."

8.12.3 Usporedba platformi na zajedničkoj skali

Z-score omogućuje usporedbu angažmana između platformi koje imaju potpuno različite skale.

```r
# Z-score lajkova UNUTAR svake platforme
posts_platform_z <- posts |>
  filter(likes > 0) |>
  mutate(log_likes = log10(likes)) |>
  group_by(platform) |>
  mutate(
    z_likes = (log_likes - mean(log_likes)) / sd(log_likes)
  ) |>
  ungroup()

# Top 5 objava po z-scoreu unutar svake platforme
posts_platform_z |>
  group_by(platform) |>
  slice_max(z_likes, n = 1) |>
  select(platform, content_type, likes, followers, z_likes) |>
  arrange(desc(z_likes))
```

```
# A tibble: 6 x 5
# Groups:   platform [6]
  platform  content_type  likes followers z_likes
  <chr>     <chr>         <dbl>     <dbl>   <dbl>
1 Facebook  tekst        129142   1616076    3.26
2 TikTok    video        155132    903229    3.09
3 Instagram reel          50748    633458    2.81
4 LinkedIn  tekst         41559    553174    2.73
5 YouTube   reel          36288    225790    2.49
6 Twitter/X slika         12775    261180    2.43
```

Objava s 500 lajkova na LinkedInu može biti neobičnija (viši z-score) nego objava s 50 000 lajkova na TikToku, jer su skale potpuno različite. Z-score normalizira tu razliku i omogućuje poštenu usporedbu.

8.13 Od vjerojatnosti do statističkog zaključivanja

Sve što smo naučili danas je temelj za ono što dolazi. Pogledajmo kako se koncepti povezuju.

Binomna distribucija ćemo koristiti u tjednu 11 za hi-kvadrat testove (je li distribucija kategorija onakva kakvu očekujemo?) i u tjednu 10 za razumijevanje logike testiranja hipoteza.

Normalna distribucija je temelj za t-testove (tjedan 12), ANOVA-u (tjedan 13) i regresiju (tjedan 14) jer pretpostavljaju normalnu distribuciju reziduala.

Z-score je temelj za standardizirane veličine učinka i za razumijevanje p-vrijednosti.

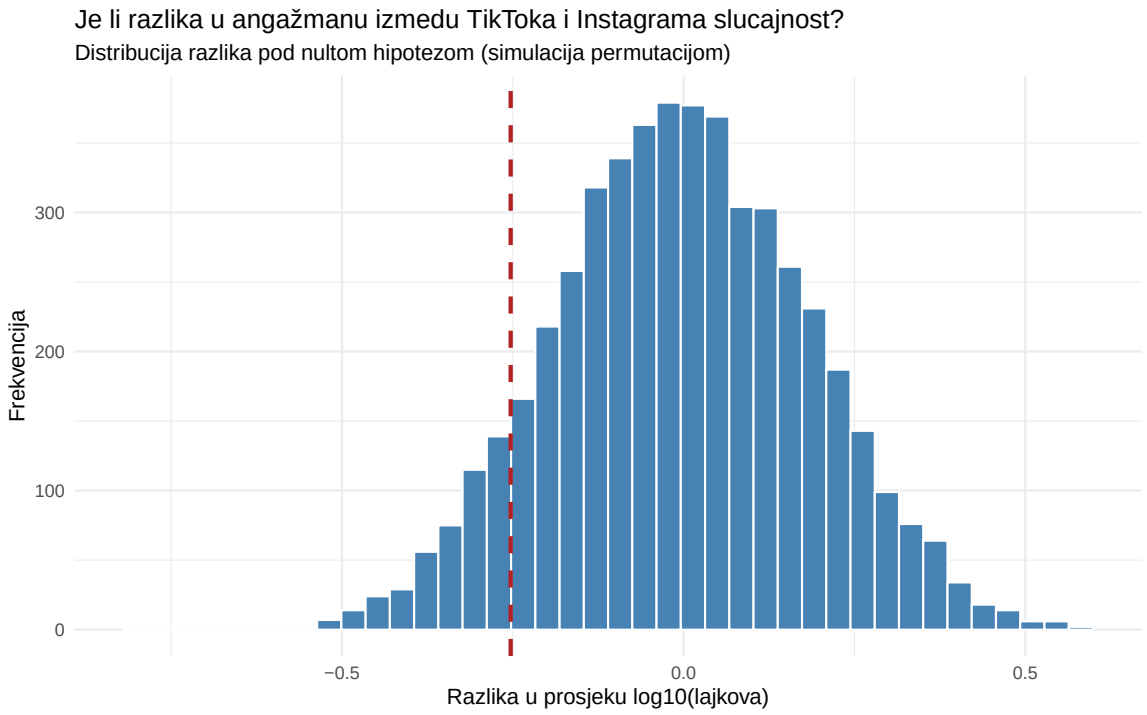
Uvjetna vjerojatnost je logika iza svakog statističkog testa. Kolika je vjerojatnost vidjeti ovakav rezultat DADO da je nulta hipoteza istinita?

```r
# Primjer: je li prosječni angažman na TikToku zaista veći nego na Instagramu?
set.seed(42)

# Uzmimo uzorke od 50 objava s svake platforme
uzorak_tt <- posts |>
  filter(platform == "TikTok", likes > 0) |>
  slice_sample(n = 50) |>
  pull(likes) |>
  log10()

uzorak_ig <- posts |>
  filter(platform == "Instagram", likes > 0) |>
```

```
  slice_sample(n = 50) |>
  pull(likes) |>
  log10()

razlika <- mean(uzorak_tt) - mean(uzorak_ig)
cat("Razlika u prosjeku log10(lajkova):", round(razlika, 3), "\n")
```

```
Razlika u prosjeku log10(lajkova): -0.253
```

```
# Koliko je ova razlika neobična? Simulirajmo!
sim_razlike <- replicate(5000, {
  sve <- c(uzorak_tt, uzorak_ig)
  pomijesano <- sample(sve)
  mean(pomijesano[1:50]) - mean(pomijesano[51:100])
})

tibble(razlika_sim = sim_razlike) |>
  ggplot(aes(x = razlika_sim)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 40) +
  geom_vline(xintercept = razlika, color = "firebrick", linetype = "dashed", linewidth = 1
  labs(
    title = "Je li razlika u angažmanu između TikToka i Instagrama slučajnost?",
    subtitle = "Distribucija razlika pod nultom hipotezom (simulacija permutacijom)",
    x = "Razlika u prosjeku log10(lajkova)",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Crvena crta označava opaženu razliku. Histogram prikazuje distribuciju razlika koje bismo očekivali čistom slučajnošću (ako nema stvarne razlike između platformi). Ako je crvena crta daleko od centra histograma, razlika je neobična i vjerojatno nije slučajna. Ovo je, u suštini, logika t-testa koji ćemo formalno naučiti u tjednu 12, ali ovdje smo ju demonstrirali simulacijom.

> Statistika je u konačnici nauka o donošenju zaključaka u uvjetima neizvjesnosti. Vjerojatnost je jezik kojim tu neizvjesnost izražavamo. Svaki statistički test, svaki interval pouzdanosti, svaka p-vrijednost govori o vjerojatnosti. Razumjeti vjerojatnost znači razumjeti statistiku.

! Ključni zaključci

1. Vjerojatnost je broj između 0 i 1 koji izražava izvjesnost. Frekvencijski pristup definira ju kao dugoročnu relativnu frekvenciju.

2. Zakon velikih brojeva — s više ponavljanja, relativna frekvencija konvergira prema pravoj vjerojatnosti.

3. Osnovna pravila — komplement P(ne A) = 1 − P(A), zbrajanje za isključive P(A ili B) = P(A) + P(B), množenje za nezavisne P(A i B) = P(A) × P(B). Uvjetna vjerojatnost P(A|B) je temelj za segmentnu analizu.

4. Binomna distribucija modelira broj uspjeha u n nezavisnih pokušaja s vjerojat-

nošću p. R funkcije su `dbinom()`, `pbinom()`, `rbinom()`.

5. Normalna distribucija je definirana prosjekom i standardnom devijacijom . Pravilo 68-95-99.7 daje postotak podataka unutar 1, 2 i 3 SD od prosjeka.
6. Z-score z = (x −) / izražava koliko je vrijednost udaljena od prosjeka u jedinicama SD. Omogućuje usporedbu varijabli na različitim skalama.
7. R funkcije za normalnu uključuju `dnorm()` za gustoću, `pnorm()` za kumulativnu vjerojatnost (P(X x)), `qnorm()` za kvantile (obrnuto od pnorm) i `rnorm()` za simulaciju.
8. Obrazac d/p/q/r vrijedi za sve distribucije u R-u.
9. QQ-plot uspoređuje kvantile podataka s teorijskim. Točke na ravnoj liniji znače normalnost. Odstupanja ukazuju na skew ili teške repove.
10. Metrike angažmana na društvenim mrežama su tipično log-normalno distribuirane. Log-transformacija ih često pretvara u (približno) normalne.
11. Z-score služi za identifikaciju outliera ($|z| > 2$ ili 3) i za usporedbu opažanja između grupa s različitim skalama.
12. Svaki statistički test koji ćemo učiti temelji se na pitanju — kolika je vjerojatnost vidjeti ovakav ili ekstremniji rezultat čistom slučajnošću? Ovo predavanje daje konceptualni temelj za to pitanje.

⚠ Priprema za kolokvij (tjedan 8)

Sljedeći tjedan je **kolokvij** koji pokriva gradivo iz tjedana 1 do 7. Kolokvij će uključivati

1. Konceptualna pitanja o istraživačkom dizajnu, mjerenju i vrstama varijabli (tjedan 1).
2. Čitanje i pisanje R koda — tibbleovi, pipe, dplyr glagoli, ggplot2 (tjedni 2 do 5).
3. Interpretaciju deskriptivnih statistika i grafova (tjedan 4 i 5).
4. Razumijevanje funkcija i DRY principa (tjedan 6).
5. Izračun i interpretaciju vjerojatnosti, uključujući binomnu i normalnu distribuciju (tjedan 7).

Za pripremu trebate

1. Ponovite pojmovnike iz svakog tjedna. Svaki pojmovnik sadrži ključne koncepte.
2. Pokrenite sve primjere iz predavanja 2 do 7 i pokušajte ih modificirati.
3. Vježbajte čitanje R koda. Za zadani pipeline, opišite riječima što svaki korak radi.

4. Vježbajte interpretaciju. Za zadani graf ili tablicu, napišite 2 do 3 rečenice o tome što rezultat znači.
5. Vježbajte izračun. Za zadanu situaciju, izračunajte odgovarajuću vjerojatnost koristeći `pbinom()` ili `pnorm()`.

8.14 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 9 (Introduction to Probability). Besplatno dostupno na learningstatisticswithr.com. Pokriva sva pravila vjerojatnosti i distribucije detaljnije nego ovo predavanje.

Wickham, H. & Grolemund, G. (2023). *R for Data Science* (2nd edition), Section 26.4. Besplatno dostupno na r4ds.hadley.nz. Pregled generiranja slučajnih brojeva i simulacije.

Preporučeno

Diez, D., Çetinkaya-Rundel, M., & Barr, C. (2019). *OpenIntro Statistics* (4th edition), Chapters 3 i 4. Besplatno dostupno na openintro.org/book/os. Distribucije s mnogo grafičkih prikaza i primjera.

Spiegelhalter, D. (2019). *The Art of Statistics: Learning from Data*. Pelican Books. Popularnoznanstvena knjiga koja izvrsno objašnjava ulogu vjerojatnosti u donošenju odluka.

Ellenberg, J. (2014). *How Not to Be Wrong: The Power of Mathematical Thinking*. Penguin Press. Poglavlja o vjerojatnosti su izuzetno pristupačna i puna primjera iz svakodnevnog života.

8.15 Pojmovnik

Pojam	Objašnjenje
Vjerojatnost	Broj između 0 i 1 koji izražava izvjesnost događaja.
Frekvencijski pristup	Definira vjerojatnost kao dugoročnu relativnu frekvenciju.
Bayesijanski pristup	Definira vjerojatnost kao stupanj uvjerenja.
Zakon velikih brojeva	S više ponavljanja, relativna frekvencija konvergira prema pravoj vjerojatnosti.

Pojam	Objašnjenje
Komplementarno pravilo	P(ne A) = 1 − P(A).
Pravilo zbrajanja	Za isključive: P(A ili B) = P(A) + P(B). Za neisključive: oduzeti P(A i B).
Pravilo množenja	Za nezavisne: P(A i B) = P(A) × P(B).
Međusobno isključivi	Događaji koji se ne mogu dogoditi istovremeno.
Nezavisni događaji	Jedan ne utječe na vjerojatnost drugoga.
Uvjetna vjerojatnost	P(A|B) = P(A i B) / P(B). Vjerojatnost A dado da se B dogodio.
Distribucija vjerojatnosti	Funkcija koja dodjeljuje vjerojatnost svakom mogućem ishodu.
Binomna distribucija	Distribucija broja uspjeha u n nezavisnih pokušaja s vjerojatnošću p.
Normalna distribucija	Zvonolika, simetrična distribucija definirana prosjekom i standardnom devijacijom . Najvažnija distribucija u statistici.
Standardna normalna	Normalna distribucija s = 0 i = 1. Piše se N(0, 1).
Pravilo 68-95-99.7	U normalnoj distribuciji, 68% podataka je unutar ±1 , 95% unutar ±2 , 99.7% unutar ±3 od prosjeka.
Z-score	Standardizirani rezultat: z = (x −) / . Izražava udaljenost od prosjeka u jedinicama SD.
`scale()`	R funkcija koja izračunava z-score (oduzima prosjek, dijeli s SD).
`dbinom()`	Točna vjerojatnost binomne distribucije.
`pbinom()`	Kumulativna vjerojatnost binomne distribucije P(X q).
`rbinom()`	Generiranje slučajnih uzoraka iz binomne distribucije.
`dnorm()`	Gustoća normalne distribucije u zadanoj točki.
`pnorm()`	Kumulativna vjerojatnost normalne distribucije P(X x).
`qnorm()`	Kvantili normalne distribucije (obrnuta funkcija od pnorm).
`rnorm()`	Generiranje slučajnih uzoraka iz normalne distribucije.
`set.seed()`	Fiksira generator slučajnih brojeva za ponovljivost simulacija.
QQ-plot	Graf koji uspoređuje kvantile podataka s teorijskim kvantilima. Služi za provjeru normalnosti.

Pojam	Objašnjenje
`stat_qq()`	ggplot2 funkcija za QQ-plot. Kombinira se s `stat_qq_line()`.
Power law distribucija	Distribucija s dugačkim desnim repom. Tipična za metrike angažmana.
Log-normalna distribucija	Distribucija čiji logaritam je normalno distribuiran.
Outlier	Opažanje neobično udaljeno od ostatka podataka. Često definirano kao
Centralni granični teorem	Prosjek uzorka ima približno normalnu distribuciju, neovisno o obliku izvorne distribucije, kad je uzorak dovoljno velik.
Permutacijski test	Simulacijski pristup za testiranje razlike između grupa: miješa podatke i uspoređuje opaženu razliku s distribucijom pod nultom hipotezom.

9 Tjedan 8: Uzorkovanje, procjena i intervali pouzdanosti

Kako iz dijela saznati nešto o cjelini

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti razliku između populacije i uzorka te između parametra i statistike.
2. Opisati kako veličina uzorka utječe na preciznost procjene populacijskog parametra.
3. Objasniti što je distribucija uzorkovanja (sampling distribution) i zašto je važna.
4. Opisati centralni granični teorem i demonstrirati ga simulacijom.
5. Izračunati standardnu pogrešku prosjeka i objasniti njezino značenje.
6. Konstruirati i interpretirati interval pouzdanosti za prosjek.
7. Prepoznati uobičajene pristranosti u uzorkovanju i objasniti zašto su online ankete problematične.
8. Kritički ocijeniti marginu pogreške u medijskim izvještajima o anketama.

9.1 Temeljni problem statistike

Statistika rješava jedan temeljni problem. Želimo znati nešto o cijeloj populaciji, ali nemamo pristup cijeloj populaciji. Želimo znati koliki je prosječni dnevni medijski ekran-time svih odraslih Hrvata, ali ne možemo pitati svaku od 3.5 milijuna odraslih osoba. Želimo znati preferiraju li čitatelji kratke ili dugačke članke, ali ne možemo testirati svaki članak na svakom čitatelju. Želimo znati koliki je CTR novog oglasa, ali ne možemo ga pokazati svim korisnicima interneta.

Umjesto toga, uzimamo **uzorak** (manji dio populacije), mjerimo što nas zanima u uzorku i na temelju toga donosimo zaključak o cijeloj populaciji. Ovo zvuči jednostavno, ali otvara niz pitanja. Koliko veliki uzorak trebamo? Koliko možemo vjerovati procjeni iz uzorka? Kako znamo da uzorak nije pristran?

Ovo predavanje daje odgovore na ta pitanja. Koncepti koje ćemo naučiti (distribucija uzorkovanja, centralni granični teorem, standardna pogreška, interval pouzdanosti) su temelj za sve statističke testove koji dolaze u nastavku kolegija.

9.2 Naši podaci: populacija i uzorci

Za ovo predavanje imamo luksuz koji u stvarnom životu nikad nemamo — poznajemo cijelu populaciju. Dataset sadrži 50 000 odraslih osoba iz fiktivnog hrvatskog grada, s podacima o dobi, spolu, obrazovanju, primarnom izvoru vijesti, povjerenju u medije i dnevnoj medijskoj konzumaciji.

```
pop <- read_csv("../resources/datasets/media_population.csv")
glimpse(pop)
```

```
Rows: 50,000
Columns: 8
$ person_id           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
$ age                 <dbl> 42, 24, 28, 68, 22, 42, 58, 19, 69, 35, 22, 69, 56~
$ gender              <chr> "ženski", "muški", "muški", "ženski", "muški", "mu~
$ education           <chr> "viša/prvostupnik", "srednja", "srednja", "osnovna~
$ primary_news_source <chr> "portal", "društvene mreže", "društvene mreže", "T~
$ media_trust         <dbl> 4, 2, 4, 9, 1, 1, 6, 4, 6, 3, 6, 3, 2, 2, 2, 7, 7,~
$ daily_media_min     <dbl> 178, 130, 120, 224, 127, 198, 248, 153, 293, 174, ~
$ willing_to_pay      <dbl> 0, 14, 0, 45, 0, 11, 0, 32, 42, 0, 0, 48, 0, 26, 0~
```

Zato što poznajemo cijelu populaciju, možemo izračunati prave populacijske parametre i onda vidjeti koliko dobro ih procjenjuju uzorci različitih veličina. Ovo je pedagoški trik jer u stvarnom istraživanju nikad ne znate populacijske parametre (da ih znate, ne biste trebali statistiku). Ali ovdje ih znamo pa možemo procijeniti kvalitetu naših procjena.

```
# PRAVI populacijski parametri (u praksi ih NIKAD ne znamo)
pop_params <- pop |>
  summarise(
    mu_trust = round(mean(media_trust), 2),
    sigma_trust = round(sd(media_trust), 2),
    mu_media_min = round(mean(daily_media_min), 1),
    sigma_media_min = round(sd(daily_media_min), 1),
    udio_portal = round(mean(primary_news_source == "portal"), 3)
  )

pop_params
```

```
# A tibble: 1 x 5
  mu_trust sigma_trust mu_media_min sigma_media_min udio_portal
     <dbl>       <dbl>        <dbl>           <dbl>       <dbl>
1     4.87        1.98         174.            65.5       0.304
```

Zapamtite ove brojeve. To su istine o populaciji. Sve što radimo dalje pokušava se približiti ovim vrijednostima iz uzoraka.

9.3 Populacija vs uzorak: terminologija

Statistika strogo razlikuje populaciju i uzorak te njihove mjere.

Populacija je cjelokupni skup jedinica o kojima želimo donijeti zaključak. Mjere populacije zovemo **parametri** i označavamo ih grčkim slovima poput (prosjek), (standardna devijacija), ili p (proporcija).

Uzorak je podskup populacije koji zaista mjerimo. Mjere uzorka zovemo **statistike** i označavamo ih latinskim slovima poput $\bar{x}$ (prosjek uzorka), s (standardna devijacija uzorka), $\hat{p}$ (proporcija uzorka).

Statistike su procjene parametara. Procjena nikad nije savršena jer uzorak nije cijela populacija, ali dobra procjena može biti dovoljno blizu za praktične svrhe.

```
set.seed(42)

# Uzmimo jedan uzorak od 100 osoba
uzorak_100 <- pop |> slice_sample(n = 100)

# Usporedba parametra i statistike
tibble(
  mjera = c("Prosjek povjerenja", "SD povjerenja", "Udio portal"),
  populacija = c(
    round(mean(pop$media_trust), 2),
    round(sd(pop$media_trust), 2),
    round(mean(pop$primary_news_source == "portal"), 3)
  ),
  uzorak_100 = c(
    round(mean(uzorak_100$media_trust), 2),
    round(sd(uzorak_100$media_trust), 2),
    round(mean(uzorak_100$primary_news_source == "portal"), 3)
  )
)
```

```
# A tibble: 3 x 3
  mjera              populacija uzorak_100
  <chr>                   <dbl>      <dbl>
1 Prosjek povjerenja      4.87        5.12
2 SD povjerenja           1.98        2.09
3 Udio portal             0.304       0.31
```

Uzorak od 100 osoba daje procjene koje su blizu populacijskim vrijednostima, ali nisu identične. Razlika između parametra i statistike naziva se **pogreška uzorkovanja** (sampling error). Ovo nije greška u smislu da smo nešto krivo napravili. To je neizbježna posljedica toga što radimo s dijelom umjesto s cjelinom.

9.4 Što se događa kad ponovimo uzorkovanje?

Ključan uvid je da bismo, da smo uzeli drugi uzorak od 100 osoba, dobili malo drugačije rezultate. Svaki uzorak je drugačiji. Statistika varira od uzorka do uzorka.

```
set.seed(1)
u1 <- pop |> slice_sample(n = 100) |> summarise(M = round(mean(media_trust), 2)) |> pull(M

set.seed(2)
u2 <- pop |> slice_sample(n = 100) |> summarise(M = round(mean(media_trust), 2)) |> pull(M

set.seed(3)
u3 <- pop |> slice_sample(n = 100) |> summarise(M = round(mean(media_trust), 2)) |> pull(M

cat("Populacijski prosjek:", round(mean(pop$media_trust), 2), "\n")
```

```
Populacijski prosjek: 4.87
```

```
cat("Uzorak 1 (n=100):", u1, "\n")
```

```
Uzorak 1 (n=100): 4.73
```

```
cat("Uzorak 2 (n=100):", u2, "\n")
```

```
Uzorak 2 (n=100): 4.87
```

```r
cat("Uzorak 3 (n=100):", u3, "\n")
```

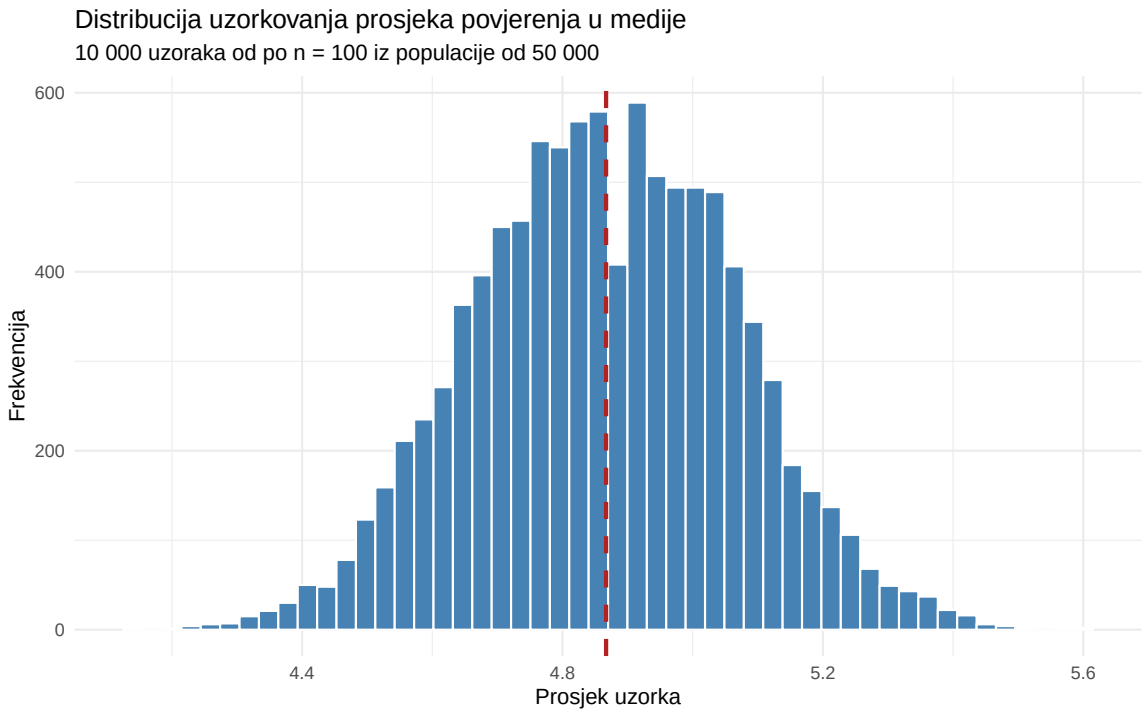
```
Uzorak 3 (n=100): 4.8
```

Svaki uzorak daje malo drugačiji prosjek. Ovo je normalno i neizbježno. Ključna pitanja su sljedeća — koliko ti prosjeci variraju? I kako ta varijacija ovisi o veličini uzorka?

9.5 Distribucija uzorkovanja

Distribucija uzorkovanja je distribucija statistike (npr. prosjeka) kroz mnogo ponovljenih uzoraka. Zamislite da uzimate 10 000 uzoraka od po 100 osoba iz populacije. Svaki uzorak daje jedan prosjek. Distribucija tih 10 000 prosjeka je distribucija uzorkovanja.

```r
set.seed(42)

# 10 000 uzoraka od po 100 osoba
sampling_dist <- tibble(
  uzorak = 1:10000,
  prosjek = map_dbl(1:10000, \(i) {
    pop |> slice_sample(n = 100) |> pull(media_trust) |> mean()
  })
)

# Populacijski prosjek
mu <- mean(pop$media_trust)

sampling_dist |>
  ggplot(aes(x = prosjek)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 50) +
  geom_vline(xintercept = mu, color = "firebrick", linewidth = 1, linetype = "dashed") +
  labs(
    title = "Distribucija uzorkovanja prosjeka povjerenja u medije",
    subtitle = "10 000 uzoraka od po n = 100 iz populacije od 50 000",
    x = "Prosjek uzorka",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Tri stvari su odmah uočljive. Prvo, distribucija je centrirana oko pravog populacijskog prosjeka (crvena crta). Ovo znači da je prosjek uzorka **nepristrana procjena** populacijskog prosjeka, što znači da ne precjenjuje ni podcjenjuje sustavno. Drugo, distribucija je približno normalna (zvonolika), čak i bez pretpostavke o obliku izvorne distribucije. Treće, distribucija je mnogo uža od distribucije pojedinačnih opažanja, gdje prosjeci manje variraju od pojedinačnih vrijednosti.

```
# Usporedba varijabilnosti
cat("SD pojedinačnih opažanja ( ):", round(sd(pop$media_trust), 2), "\n")
```

```
SD pojedinačnih opažanja ( ): 1.98
```

```
cat("SD distribucije uzorkovanja:", round(sd(sampling_dist$prosjek), 3), "\n")
```

```
SD distribucije uzorkovanja: 0.2
```

```
cat("Omjer:", round(sd(pop$media_trust) / sd(sampling_dist$prosjek), 1), "\n")
```

```
Omjer: 9.9
```

SD distribucije uzorkovanja je otprilike 10 puta manji od SD pojedinačnih opažanja. Taj omjer nije slučajan, već je približno jednak $\sqrt{n} = \sqrt{100} = 10$. Ovo nas vodi do ključnog koncepta.

9.6 Standardna pogreška

Standardna pogreška (standard error, SE) je standardna devijacija distribucije uzorkovanja. Ona mjeri koliko tipično prosjeci uzoraka variraju oko populacijskog prosjeka.

Za prosjek, standardna pogreška se računa formulom:

$$SE = \frac{\sigma}{\sqrt{n}}$$

gdje je standardna devijacija populacije, a n veličina uzorka. U praksi ne znamo pa koristimo procjenu iz uzorka (s):

$$SE \approx \frac{s}{\sqrt{n}}$$

```
# Teorijska SE za n = 100
sigma <- sd(pop$media_trust)
se_teorijska <- sigma / sqrt(100)

# Procijenjena SE iz jednog uzorka
set.seed(42)
uzorak <- pop |> slice_sample(n = 100)
se_procijenjena <- sd(uzorak$media_trust) / sqrt(100)

# Empirijska SE (iz 10 000 uzoraka)
se_empirijska <- sd(sampling_dist$prosjek)

cat("Teorijska SE:", round(se_teorijska, 3), "\n")
```

```
Teorijska SE: 0.198
```

```
cat("Procijenjena SE (iz jednog uzorka):", round(se_procijenjena, 3), "\n")
```

```
Procijenjena SE (iz jednog uzorka): 0.209
```

```
cat("Empirijska SE (iz simulacije):", round(se_empirijska, 3), "\n")
```

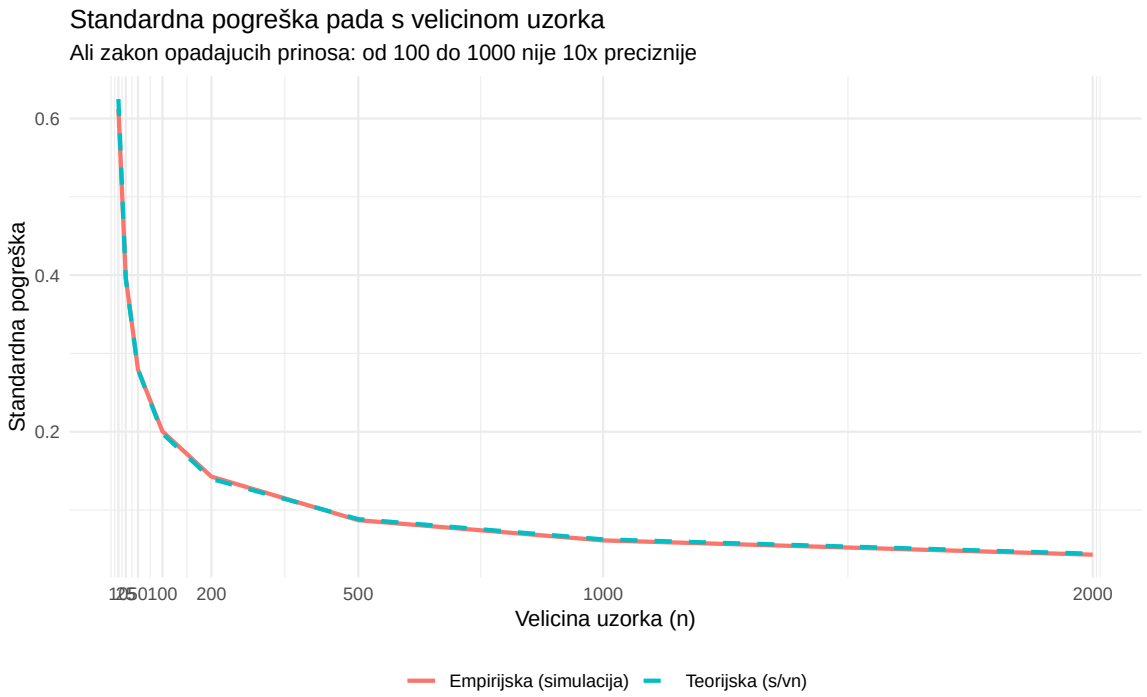
```
Empirijska SE (iz simulacije): 0.2
```

Sve tri vrijednosti su blizu jedna drugoj. Ovo potvrđuje da formula SE = s/√n dobro procjenjuje stvarnu varijabilnost prosjeka uzoraka.

9.6.1 Veličina uzorka i preciznost

Formula SE = /√n odmah otkriva nešto fundamentalno. Preciznost procjene raste s korijenom veličine uzorka, ne linearno.

```
set.seed(42)

velicine <- c(10, 25, 50, 100, 200, 500, 1000, 2000)

sim_se <- map_df(velicine, \(n) {
  prosjeci <- map_dbl(1:2000, \(i) {
    pop |> slice_sample(n = n) |> pull(media_trust) |> mean()
  })
  tibble(n = n, se_empirijska = sd(prosjeci), se_formula = sigma / sqrt(n))
})

sim_se |>
  ggplot(aes(x = n)) +
  geom_line(aes(y = se_empirijska, color = "Empirijska (simulacija)"), linewidth = 1) +
  geom_line(aes(y = se_formula, color = "Teorijska ( /√n)"), linewidth = 1, linetype = "das
  scale_x_continuous(breaks = velicine) +
  labs(
    title = "Standardna pogreška pada s veličinom uzorka",
    subtitle = "Ali zakon opadajućih prinosa: od 100 do 1000 nije 10x preciznije",
    x = "Veličina uzorka (n)",
    y = "Standardna pogreška",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Pad je strm na početku, gdje od n=10 do n=100 je ogromno poboljšanje, ali se postepeno usporava. Da biste prepolovili SE, morate učetverostručiti uzorak. To objašnjava zašto su anketni uzorci obično između 500 i 2000, jer povećanje iznad toga donosi malo dodatne preciznosti u odnosu na trošak.

```
sim_se |>
  mutate(
    se_formula = round(se_formula, 3),
    se_empirijska = round(se_empirijska, 3),
    raspon_95 = paste0("±", round(1.96 * se_formula, 2))
  ) |>
  select(n, se_formula, raspon_95)
```

```
# A tibble: 8 x 3
      n se_formula raspon_95
  <dbl>      <dbl> <chr>
1    10      0.625 ±1.23
2    25      0.395 ±0.77
3    50      0.279 ±0.55
4   100      0.198 ±0.39
5   200      0.14  ±0.27
6   500      0.088 ±0.17
7  1000      0.062 ±0.12
8  2000      0.044 ±0.09
```

Stupac `raspon_95` pokazuje koliko širok je 95% interval oko prosjeka. S uzorkom od 100, prosjek povjerenja je precizan na ±0.39 bodova. S uzorkom od 1000, preciznost je ±0.12 bodova. U praksi, odlučujete kolika je vam preciznost dovoljna i na temelju toga birate veličinu uzorka.

> **Praktični savjet**
>
> Kad čitate medijske izvještaje o anketama, uvijek tražite veličinu uzorka i marginu pogreške. Anketa s n = 500 ima marginu pogreške oko ±4.4 postotna boda za proporcije (na 95% razini). Anketa s n = 1000 ima oko ±3.1. Kad novinar kaže "stranka A ima 32% a stranka B 29%", razlika od 3 postotna boda je unutar margine pogreške za većinu anketa. Zaključak "A vodi" iz takve ankete nije opravdan.

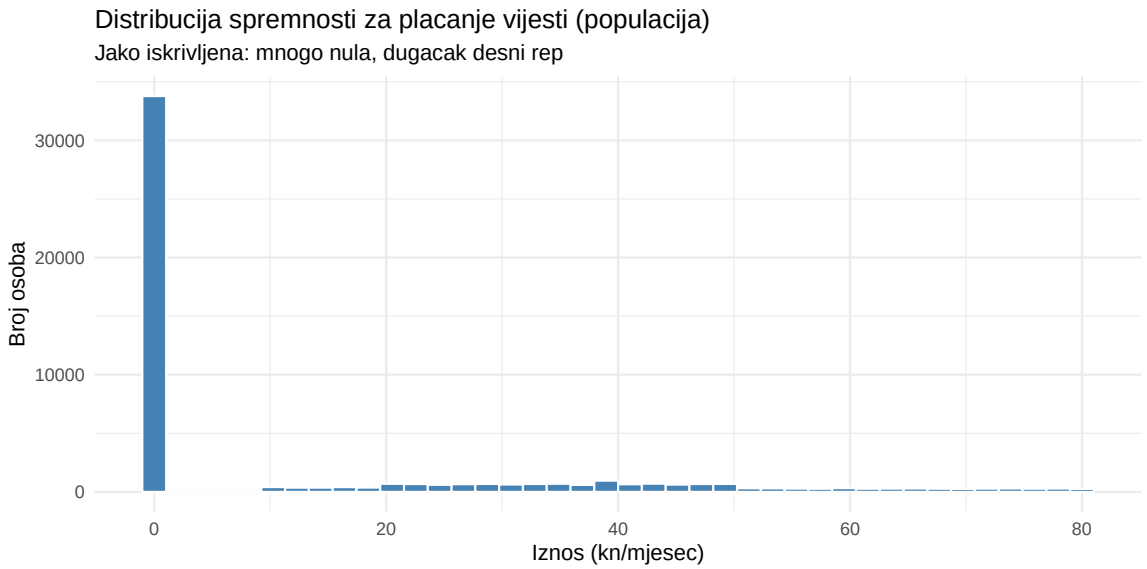
9.7 Centralni granični teorem

Centralni granični teorem (Central Limit Theorem, CLT) je najvažniji teorem u cijeloj statistici. On kaže da je distribucija uzorkovanja prosjeka približno normalna, neovisno o obliku izvorne distribucije, pod uvjetom da je uzorak dovoljno velik.

Ovo je izuzetno moćno jer znači da možemo koristiti normalnu distribuciju za donošenje zaključaka o prosjecima čak i kad izvorna varijabla nije normalna.

Demonstrirajmo to na varijabli `willing_to_pay` koja je jako iskrivljena (mnogo nula i dugačak desni rep).

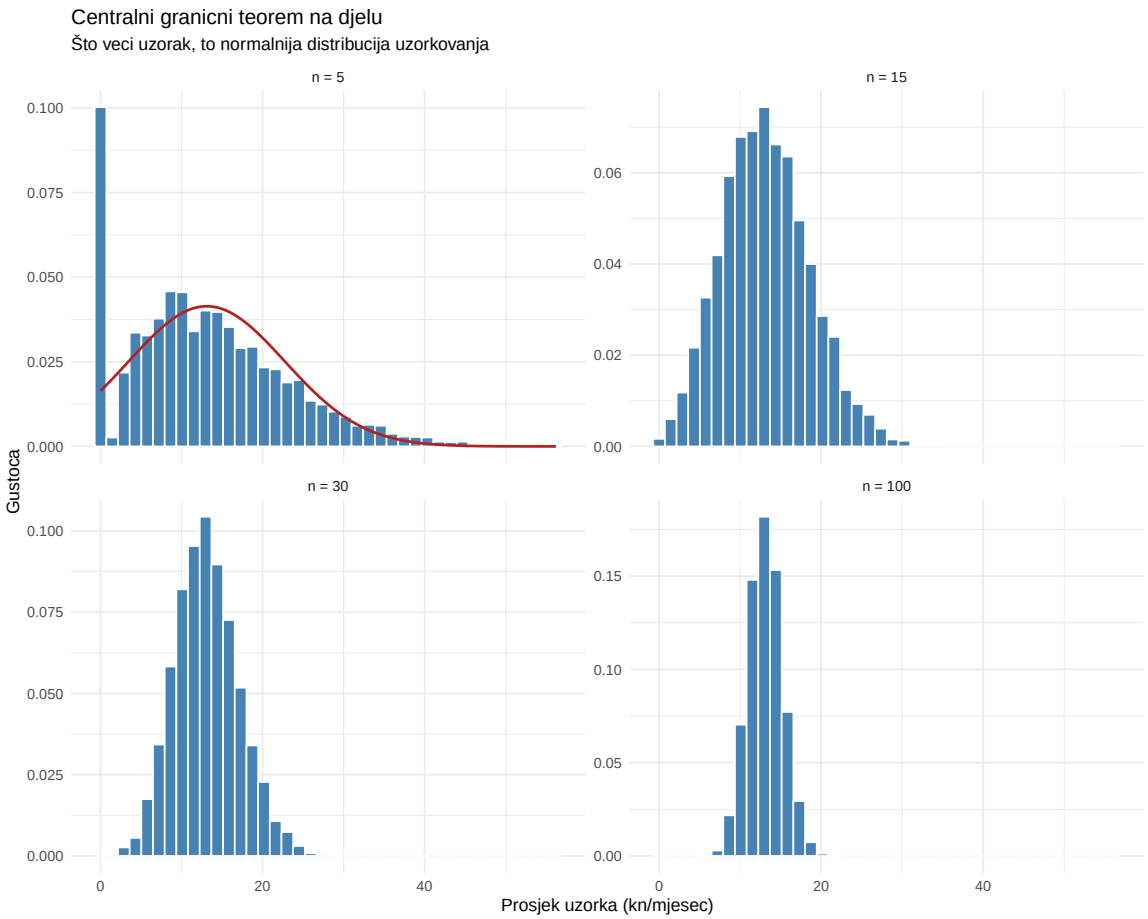
```
# Izvorna distribucija: daleko od normalne
pop |>
  ggplot(aes(x = willing_to_pay)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 40) +
  labs(
    title = "Distribucija spremnosti za plaćanje vijesti (populacija)",
    subtitle = "Jako iskrivljena: mnogo nula, dugačak desni rep",
    x = "Iznos (kn/mjesec)",
    y = "Broj osoba"
  ) +
  theme_minimal()
```



Ovo definitivno nije normalna distribucija. Većina ljudi nije spremna platiti ništa, a oni koji jesu spremni plaćaju različite iznose.

```
set.seed(42)

# Distribucije uzorkovanja za različite veličine uzorka
clt_sim <- map_df(c(5, 15, 30, 100), \(n) {
  prosjeci <- map_dbl(1:5000, \(i) {
    pop |> slice_sample(n = n) |> pull(willing_to_pay) |> mean()
  })
  tibble(n_label = paste("n =", n), n = n, prosjek = prosjeci)
}) |>
  mutate(n_label = fct_reorder(n_label, n))

clt_sim |>
  ggplot(aes(x = prosjek)) +
  geom_histogram(aes(y = after_stat(density)), fill = "steelblue", color = "white", bins =
  stat_function(fun = dnorm,
    args = list(mean = mean(pop$willing_to_pay),
                sd = sd(pop$willing_to_pay) / sqrt(5)),
    data = clt_sim |> filter(n == 5),
    color = "firebrick", linewidth = 0.8) +
  facet_wrap(~n_label, scales = "free_y") +
  labs(
    title = "Centralni granični teorem na djelu",
    subtitle = "Što veći uzorak, to normalnija distribucija uzorkovanja",
    x = "Prosjek uzorka (kn/mjesec)",
    y = "Gustoća"
  ) +
```

```
  theme_minimal()
```



Rezultat je zapanjujuć. S n = 5, distribucija prosjeka je još uvijek iskrivljena (jer je izvorni oblik još dominantan). S n = 15, već se počinje formirati zvonoliki oblik. S n = 30, distribucija je gotovo savršeno normalna. S n = 100, normalna aproksimacija je izvrsna.

Pravilo palca je da je n 30 obično dovoljno za CLT, ali za jako iskrivljene distribucije može trebati i više. Za približno normalne izvorne distribucije, i n = 10 može biti dovoljno.

9.7.1 Zašto je CLT toliko važan?

CLT je razlog zašto većina statističkih testova radi. T-test pretpostavlja da je distribucija prosjeka približno normalna. Ne pretpostavlja da su individualna opažanja normalna. Zahvaljujući CLT-u, distribucija prosjeka je (približno) normalna čak i kad individualna opažanja nisu, pod uvjetom da je uzorak dovoljno velik. Ovo daje statistici ogromnu moć jer možemo koristiti iste alate (normalnu distribuciju) na gotovo svaku vrstu podataka.

9.8 Pristranosti u uzorkovanju

CLT i formula za SE pretpostavljaju da je uzorak **slučajan**, što znači da svaka osoba u populaciji ima jednaku šansu biti odabrana. U praksi je ta pretpostavka često narušena, što uzrokuje veće probleme od male veličine uzorka.

9.8.1 Convenience sampling (prigodan uzorak)

Najčešća pristranost u komunikološkim istraživanjima. Anketirate studente na svom kolegiju jer su dostupni. Ali studenti nisu reprezentativni za opću populaciju ni po dobi, ni po obrazovanju, ni po medijskim navikama.

```
set.seed(42)

# "Populacija" = svi
pop_prosjek_trust <- round(mean(pop$media_trust), 2)

# "Prigodan uzorak" = samo mladi (18-24) s visokim obrazovanjem
pristrani_uzorak <- pop |>
  filter(age <= 24, education %in% c("viša/prvostupnik", "magistar/doktor")) |>
  slice_sample(n = 100)

# "Slučajni uzorak" iste veličine
slucajni_uzorak <- pop |> slice_sample(n = 100)

tibble(
  izvor = c("Populacija", "Slučajni uzorak (n=100)", "Pristrani uzorak (n=100)"),
  prosjek_trust = c(
    round(mean(pop$media_trust), 2),
    round(mean(slucajni_uzorak$media_trust), 2),
    round(mean(pristrani_uzorak$media_trust), 2)
  ),
  udio_portal = c(
    round(mean(pop$primary_news_source == "portal"), 3),
    round(mean(slucajni_uzorak$primary_news_source == "portal"), 3),
    round(mean(pristrani_uzorak$primary_news_source == "portal"), 3)
  )
)
```

```
# A tibble: 3 x 3
  izvor                    prosjek_trust udio_portal
  <chr>                            <dbl>       <dbl>
1 Populacija                        4.87       0.304
2 Slučajni uzorak (n=100)           4.79       0.38
3 Pristrani uzorak (n=100)          4.33       0.27
```

Pristrani uzorak daje drugačije procjene od populacijskih vrijednosti. Mladi visokoobrazovani ljudi imaju drugačije medijske navike od opće populacije. Nijedna količina povećanja uzorka ne može ispraviti ovu pristranost, jer 10 000 studenata i dalje nije reprezentativno za opću populaciju.

9.8.2 Online ankete i self-selection bias

Online ankete, koje su izuzetno popularne u komunikološkim istraživanjima, pate od posebnog oblika pristranosti. Odgovaraju samo ljudi koji su online, koji su na toj platformi, koji su vidjeli poziv na anketu i koji su motivirani odgovoriti. Svaki od ovih koraka filtrira populaciju.

```
# Simulacija: online anketa privlači neproporcijalno mlade korisnike mreža
online_uzorak <- pop |>
  mutate(
    vjerojatnost_odgovora = case_when(
      age < 30 & primary_news_source == "društvene mreže" ~ 0.15,
      age < 30 ~ 0.08,
      age < 50 & primary_news_source %in% c("portal", "društvene mreže") ~ 0.06,
      age < 50 ~ 0.03,
      age >= 50 & primary_news_source %in% c("portal", "društvene mreže") ~ 0.02,
      .default = 0.005
    )
  ) |>
  mutate(odgovorio = runif(n()) < vjerojatnost_odgovora) |>
  filter(odgovorio)

cat("Veličina online uzorka:", nrow(online_uzorak), "\n\n")
```

```
Veličina online uzorka: 2713
```

```
# Usporedba
tribble(
  ~karakteristika, ~populacija, ~online_uzorak,
  "Prosjek dobi", round(mean(pop$age), 1), round(mean(online_uzorak$age), 1),
  "Udio mlađih od 30", round(mean(pop$age < 30) * 100, 1), round(mean(online_uzorak$age <
  "Udio portal kao primarni", round(mean(pop$primary_news_source == "portal") * 100, 1), r
  "Udio društvene mreže", round(mean(pop$primary_news_source == "društvene mreže") * 100,
  "Prosjek povjerenja", round(mean(pop$media_trust), 2), round(mean(online_uzorak$media_tr
)
```

```
# A tibble: 5 x 3
  karakteristika           populacija online_uzorak
  <chr>                         <dbl>         <dbl>
```

```
1 Prosjek dobi                  42.7          31.5
2 Udio mlađih od 30             25.7          56.1
3 Udio portal kao primarni      30.4          32.1
4 Udio društvene mreže          27            48.7
5 Prosjek povjerenja             4.87          4.35
```

Online uzorak je mlađi, koristi više digitalne medije i ima drugačije povjerenje u medije. Čak i s velikim uzorkom, ove procjene su pristrane jer mehanizam uzorkovanja nije slučajan.

> **!** Važna napomena
>
> Veličina uzorka i kvaliteta uzorka su dva različita problema. Velik pristran uzorak je gori od malog slučajnog uzorka. Čuveni primjer je anketa Literary Digesta iz 1936. koja je imala 2.4 milijuna odgovora ali je pogrešno predvidjela američke predsjedničke izbore jer je uzorak bio pristran (bogatiji birači). Gallup je s uzorkom od samo 50 000 pogodio rezultat jer je koristio slučajno uzorkovanje. Veličina bez reprezentativnosti ne vrijedi ništa.

9.9 Procjena proporcija

Do sada smo se fokusirali na procjenu prosjeka. Ali u komunikologiji često procjenjujemo i proporcije (udjele). Koliki je udio ljudi koji portale koriste kao primarni izvor vijesti? Koliki je udio čitatelja koji kliknu na oglas?

Standardna pogreška za proporciju ima drugačiju formulu:

$$SE_p = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

gdje je $\hat{p}$ procijenjena proporcija iz uzorka.

```r
set.seed(42)

# Pravi udio korisnika portala
p_populacija <- mean(pop$primary_news_source == "portal")

# Procjena iz uzorka od 500
uzorak_500 <- pop |> slice_sample(n = 500)
p_hat <- mean(uzorak_500$primary_news_source == "portal")

se_p <- sqrt(p_hat * (1 - p_hat) / 500)
```

```r
cat("Populacijski udio portala:", round(p_populacija, 3), "\n")
```

```
Populacijski udio portala: 0.304
```

```r
cat("Procjena iz uzorka (n=500):", round(p_hat, 3), "\n")
```

```
Procjena iz uzorka (n=500): 0.318
```

```r
cat("SE proporcije:", round(se_p, 3), "\n")
```

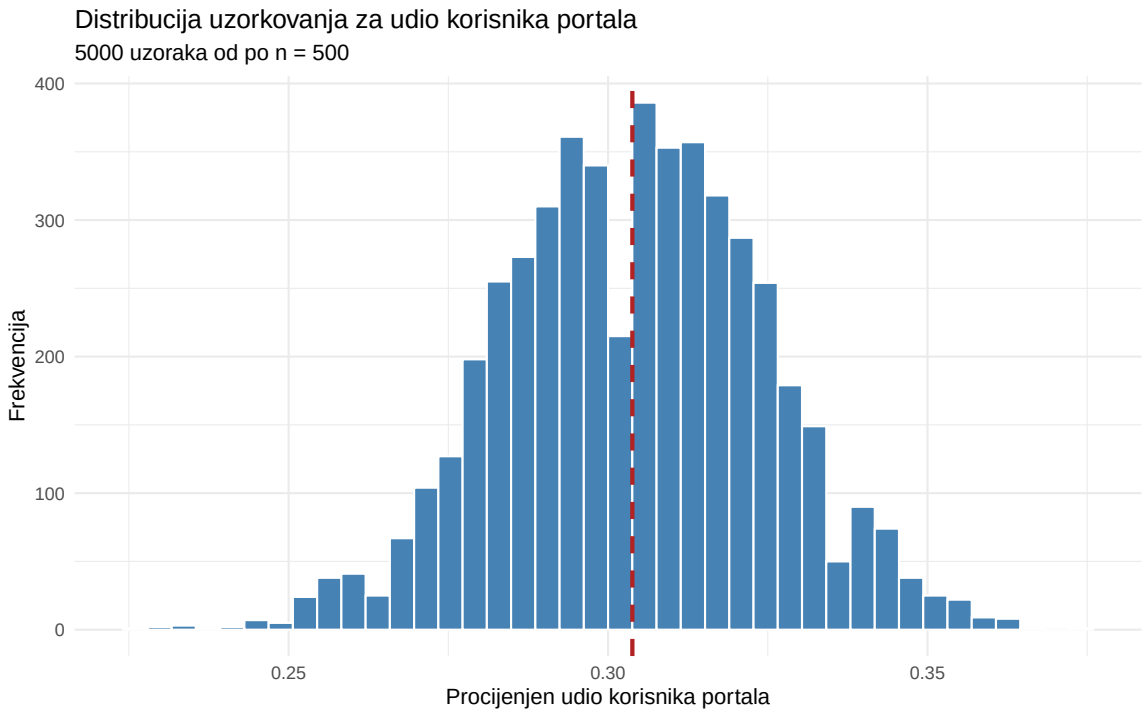
```
SE proporcije: 0.021
```

```r
cat("Margina pogreške (95%):", round(1.96 * se_p, 3), "\n")
```

```
Margina pogreške (95%): 0.041
```

Margina pogreške za proporcije ovisi o samoj proporciji. Najveća je kad je $\hat{p} = 0.5$ (maksimalna neizvjesnost) i smanjuje se kako se $\hat{p}$ približava 0 ili 1 (veća izvjesnost). Zato medijske ankete navode "marginu pogreške $\pm 3\%$" koja zapravo vrijedi samo za proporcije oko 50%.

```r
set.seed(42)

# 5000 uzoraka od po 500 osoba
prop_sim <- tibble(
  uzorak = 1:5000,
  p_hat = map_dbl(1:5000, \(i) {
    pop |> slice_sample(n = 500) |>
      pull(primary_news_source) |>
      (\(x) mean(x == "portal"))()
  })
)

prop_sim |>
  ggplot(aes(x = p_hat)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 40) +
  geom_vline(xintercept = p_populacija, color = "firebrick", linewidth = 1, linetype = "da
  labs(
    title = "Distribucija uzorkovanja za udio korisnika portala",
    subtitle = "5000 uzoraka od po n = 500",
    x = "Procijenjen udio korisnika portala",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Distribucija proporcija uzorka je također normalna (zahvaljujući CLT) i centrirana oko prave populacijske proporcije. Ovo nam omogućuje konstrukciju intervala pouzdanosti za proporcije, što je temelj za interpretaciju anketnih rezultata.

9.10 Interval pouzdanosti: osnovna ideja

Kad kažemo "prosječno povjerenje u medije je 4.87", to je točkasta procjena (point estimate). Problem s točkastom procjenom je da ne govori ništa o tome koliko je precizna. Je li pravi prosjek negdje između 4.5 i 5.2? Ili između 4.85 i 4.89?

Interval pouzdanosti (confidence interval, CI) daje raspon vrijednosti unutar kojeg se, s određenom vjerojatnošću, nalazi pravi populacijski parametar.

Za prosjek, 95% interval pouzdanosti je:

$$CI_{95\%} = \bar{x} \pm 1.96 \times SE$$

```
set.seed(42)
uzorak <- pop |> slice_sample(n = 200)

x_bar <- mean(uzorak$media_trust)
se <- sd(uzorak$media_trust) / sqrt(200)
```

```r
ci_lower <- x_bar - 1.96 * se
ci_upper <- x_bar + 1.96 * se

cat("Prosjek uzorka:", round(x_bar, 2), "\n")
```

```
Prosjek uzorka: 5.07
```

```r
cat("SE:", round(se, 3), "\n")
```

```
SE: 0.145
```

```r
cat("95% CI: [", round(ci_lower, 2), ",", round(ci_upper, 2), "]\n")
```

```
95% CI: [ 4.79 , 5.35 ]
```

```r
cat("Pravi populacijski prosjek:", round(mean(pop$media_trust), 2), "\n")
```

```
Pravi populacijski prosjek: 4.87
```

Interval pouzdanosti pokriva pravi populacijski prosjek u ovom slučaju. Ali ne mora uvijek, jer 5% intervala iz ponovljenih uzoraka neće pokriti pravi parametar. Zato se zove 95% interval, ne 100%.

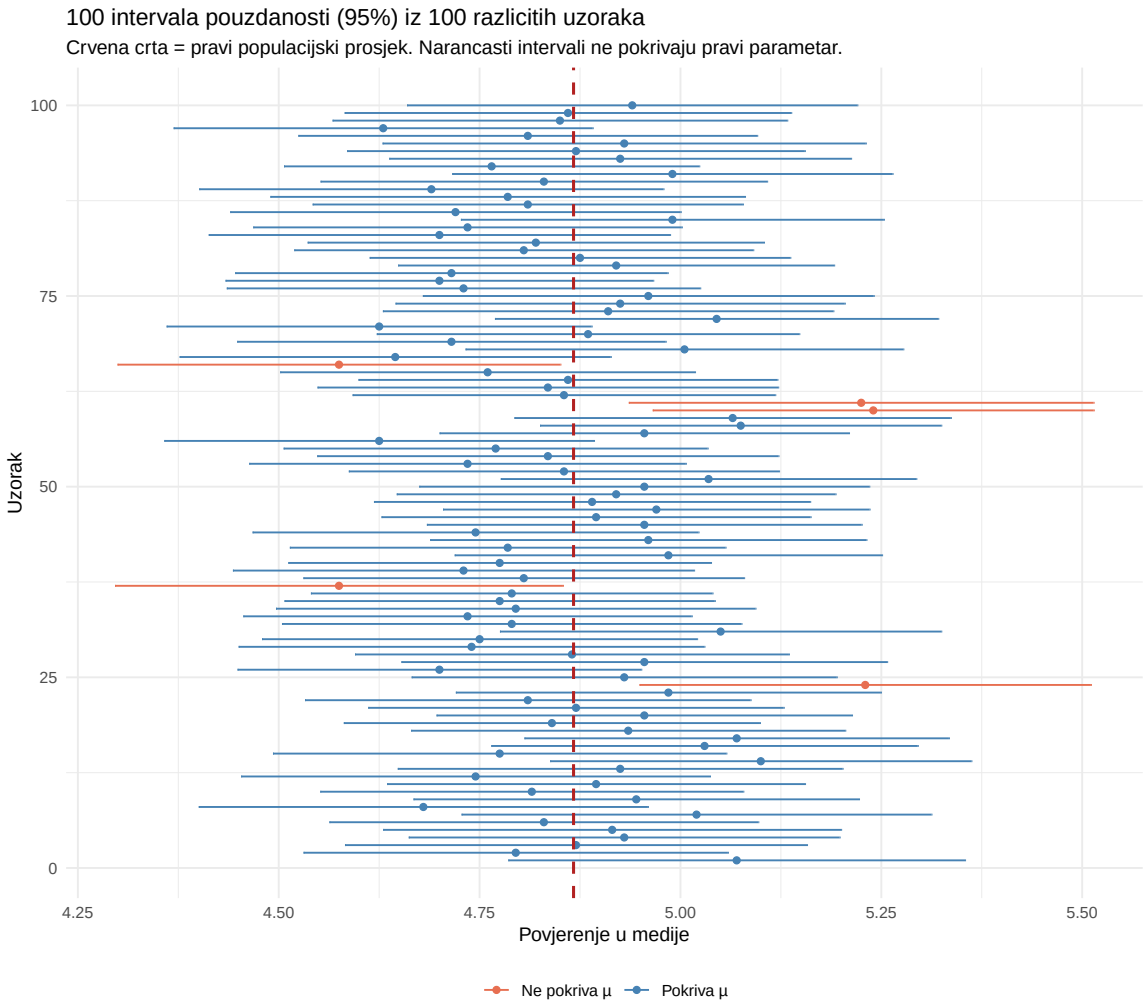
9.10.1 Vizualizacija: 100 intervala pouzdanosti

```r
set.seed(42)

mu_pop <- mean(pop$media_trust)

ci_sim <- map_df(1:100, \(i) {
  u <- pop |> slice_sample(n = 200)
  xbar <- mean(u$media_trust)
  se <- sd(u$media_trust) / sqrt(200)
  tibble(
    uzorak = i,
    xbar = xbar,
    ci_lo = xbar - 1.96 * se,
    ci_hi = xbar + 1.96 * se,
    pokriva_mu = ci_lo <= mu_pop & ci_hi >= mu_pop
  )
```

```
})

cat("Intervala koji pokrivaju pravi prosjek:", sum(ci_sim$pokriva_mu), "od 100\n")
```

```
Intervala koji pokrivaju pravi prosjek: 95 od 100
```

```
ci_sim |>
  ggplot(aes(x = xbar, y = uzorak, color = pokriva_mu)) +
  geom_point(size = 1.5) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0) +
  geom_vline(xintercept = mu_pop, color = "firebrick", linewidth = 0.8, linetype = "dashed
  scale_color_manual(values = c("TRUE" = "steelblue", "FALSE" = "#e76f51"),
                     labels = c("TRUE" = "Pokriva  ", "FALSE" = "Ne pokriva  ")) +
  labs(
    title = "100 intervala pouzdanosti (95%) iz 100 različitih uzoraka",
    subtitle = "Crvena crta = pravi populacijski prosjek. Narančasti intervali ne pokrivaj
    x = "Povjerenje u medije",
    y = "Uzorak",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Ovaj graf je jedna od najvažnijih vizualizacija u cijelom kolegiju. Svaki horizontalni interval je jedan 95% CI iz zasebnog uzorka. Većina, oko 95, pokriva pravi prosjek (crvena crta). Nekolicina, oko 5, ne pokriva. Ovo je precizno značenje 95% intervala pouzdanosti. 95% takvih intervala, konstruiranih iz ponovljenih uzoraka, pokrit će pravi parametar.

! Važna napomena

Česta pogrešna interpretacija je "postoji 95% šansa da je pravi prosjek unutar ovog intervala." Ispravna interpretacija — "ako bismo ponovili uzorkovanje mnogo puta i svaki put konstruirali 95% CI, 95% tih intervala bi pokrilo pravi prosjek." Razlika zvuči suptilno, ali je konceptualno važna. Pravi prosjek je fiksan broj (nije slučajan). Interval je slučajan (jer ovisi o uzorku). Vjerojatnost se odnosi na postupak, ne na parametar.

Podsjetnik

U prvom dijelu predavanja naučili smo da distribucija uzorkovanja prosjeka ima oblik normalne distribucije (CLT), da se njezina širina mjeri standardnom pogreškom SE = s/√n i da 95% interval pouzdanosti pokriva prosjek ± 1.96 × SE. U ovom dijelu prelazimo na alate koji se koriste u praksi, poput t-distribucije, funkcije `t.test()` i planiranja veličine uzorka.

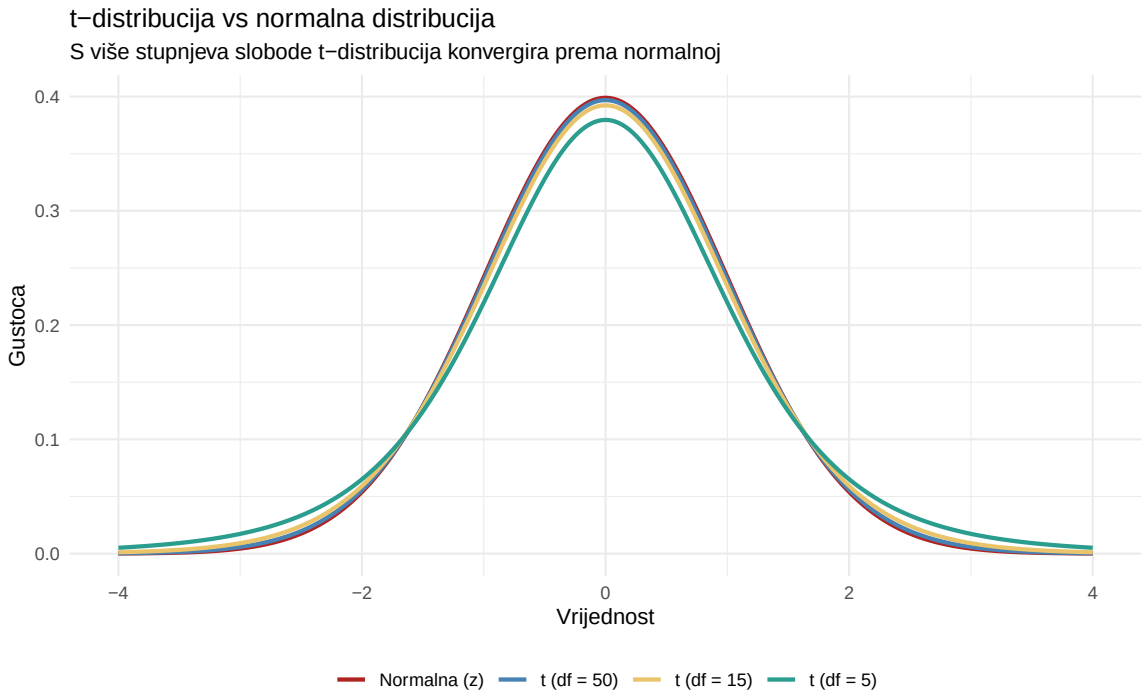
9.11 Od z do t: mali uzorci

Do sada smo koristili z = 1.96 za 95% CI. To je točno kad poznajemo populacijski ili kad je uzorak velik (n > 100). Ali u praksi obično ne poznajemo pa ga procjenjujemo iz uzorka pomoću s. Za male uzorke, ta dodatna nesigurnost znači da trebamo širi interval.

t-distribucija rješava ovaj problem. Izgleda poput normalne distribucije, ali ima deblje repove, gdje je veća vjerojatnost ekstremnijih vrijednosti. Oblik t-distribucije ovisi o **stupnjevima slobode** (degrees of freedom, df), koji su za jedan prosjek df = n − 1.

```
x <- seq(-4, 4, length.out = 300)

t_usporedba <- tibble(x = x) |>
  mutate(
    `Normalna (z)` = dnorm(x),
    `t (df = 5)` = dt(x, df = 5),
    `t (df = 15)` = dt(x, df = 15),
    `t (df = 50)` = dt(x, df = 50)
  ) |>
  pivot_longer(-x, names_to = "distribucija", values_to = "gustoca") |>
  mutate(distribucija = fct_relevel(distribucija,
    "Normalna (z)", "t (df = 50)", "t (df = 15)", "t (df = 5)"))

t_usporedba |>
  ggplot(aes(x = x, y = gustoca, color = distribucija)) +
  geom_line(linewidth = 1) +
  scale_color_manual(values = c(
    "Normalna (z)" = "firebrick",
    "t (df = 5)" = "#2a9d8f",
    "t (df = 15)" = "#e9c46a",
    "t (df = 50)" = "steelblue"
  )) +
  labs(
    title = "t-distribucija vs normalna distribucija",
    subtitle = "S više stupnjeva slobode t-distribucija konvergira prema normalnoj",
    x = "Vrijednost",
```

```
    y = "Gustoća",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



S $\mathrm{df} = 5$ (uzorak od 6), t-distribucija ima znatno deblje repove od normalne. S $\mathrm{df} = 50$, razlika je jedva vidljiva. Praktična posljedica je da za male uzorke koristimo veći multiplikator od 1.96.

```
tibble(
  df = c(5, 10, 15, 25, 50, 100, Inf),
  n = df + 1,
  t_95 = round(qt(0.975, df), 3),
  t_99 = round(qt(0.995, df), 3)
) |>
  mutate(n = if_else(is.infinite(df), "∞ (normalna)", as.character(n)))
```

```
# A tibble: 7 x 4
     df n              t_95  t_99
  <dbl> <chr>         <dbl> <dbl>
1     5 6              2.57  4.03
2    10 11             2.23  3.17
3    15 16             2.13  2.95
```

```
4    25 26            2.06  2.79
5    50 51            2.01  2.68
6   100 101           1.98  2.63
7   Inf ∞ (normalna)  1.96  2.58
```

Za n = 6 (df = 5), kritična vrijednost za 95% CI je 2.571 umjesto 1.960. Interval je značajno širi jer imamo manje podataka pa moramo biti oprezniji. Za n > 100, razlika između t i z je zanemariva i u praksi se često ignorira.

9.12 `t.test()`: sve u jednoj funkciji

R ima ugrađenu funkciju `t.test()` koja automatski računa t-interval pouzdanosti. Za sada je koristimo samo za CI (ne za testiranje hipoteza, to dolazi sljedeći tjedan).

```r
set.seed(42)
uzorak_200 <- pop |> slice_sample(n = 200)

# CI za prosjek povjerenja u medije
rezultat <- t.test(uzorak_200$media_trust, conf.level = 0.95)
rezultat
```

```
    One Sample t-test

data:  uzorak_200$media_trust
t = 34.961, df = 199, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 4.784028 5.355972
sample estimates:
mean of x
     5.07
```

Funkcija `t.test()` vraća mnogo informacija odjednom. Za interval pouzdanosti nas zanima `conf.int` i `estimate`.

```r
# Pristup pojedinim elementima
cat("Prosjek uzorka:", round(rezultat$estimate, 3), "\n")
```

```
Prosjek uzorka: 5.07
```

```r
cat("95% CI: [", round(rezultat$conf.int[1], 3), ",", round(rezultat$conf.int[2], 3), "]\n
```

```
95% CI: [ 4.784 , 5.356 ]
```

```r
cat("Stupnjevi slobode:", rezultat$parameter, "\n")
```

```
Stupnjevi slobode: 199
```

```r
# Usporedba s populacijom
cat("\nProvi populacijski prosjek:", round(mean(pop$media_trust), 3), "\n")
```

```
Provi populacijski prosjek: 4.867
```

```r
cat("Pokriva CI pravi prosjek?",
    mean(pop$media_trust) >= rezultat$conf.int[1] &
    mean(pop$media_trust) <= rezultat$conf.int[2], "\n")
```

```
Pokriva CI pravi prosjek? TRUE
```

9.12.1 Mijenjanje razine pouzdanosti

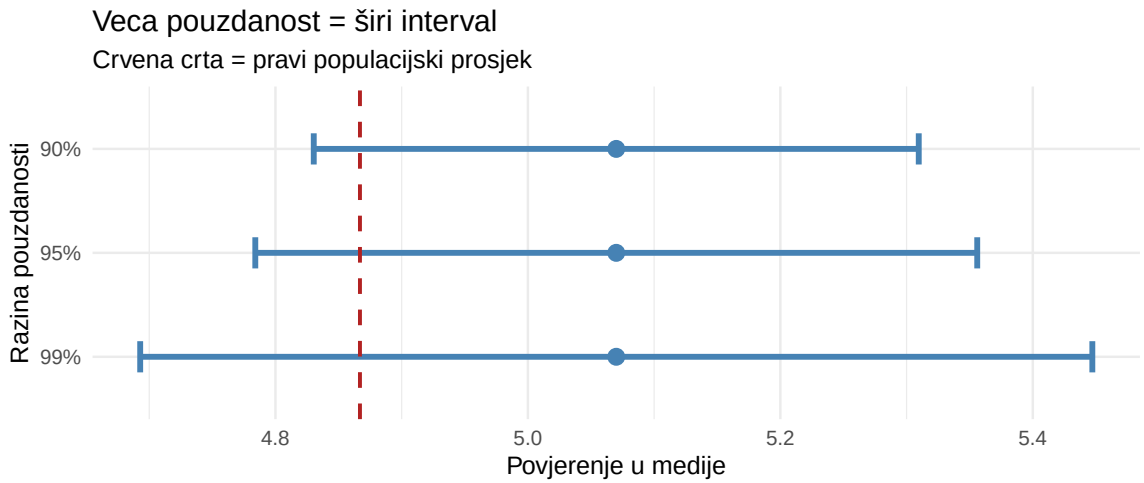
Možemo tražiti i 90% ili 99% interval.

```r
ci_90 <- t.test(uzorak_200$media_trust, conf.level = 0.90)$conf.int
ci_95 <- t.test(uzorak_200$media_trust, conf.level = 0.95)$conf.int
ci_99 <- t.test(uzorak_200$media_trust, conf.level = 0.99)$conf.int

xbar <- mean(uzorak_200$media_trust)

tibble(
  razina = c("90%", "95%", "99%"),
  donja = round(c(ci_90[1], ci_95[1], ci_99[1]), 3),
  gornja = round(c(ci_90[2], ci_95[2], ci_99[2]), 3),
  sirina = round(c(diff(ci_90), diff(ci_95), diff(ci_99)), 3)
)
```

```
# A tibble: 3 x 4
  razina donja gornja sirina
  <chr>  <dbl>  <dbl>  <dbl>
1 90%     4.83   5.31  0.479
2 95%     4.78   5.36  0.572
3 99%     4.69   5.45  0.754
```

Veća pouzdanost znači širi interval. 99% CI je širi od 95% jer morate pokriti više mogućih vrijednosti. Postoji kompromis između pouzdanosti i preciznosti. 100% CI bi bio od $-\infty$ do $+\infty$, što je potpuno beskorisno ali potpuno sigurno. U praksi se najčešće koristi 95% kao konvencija.

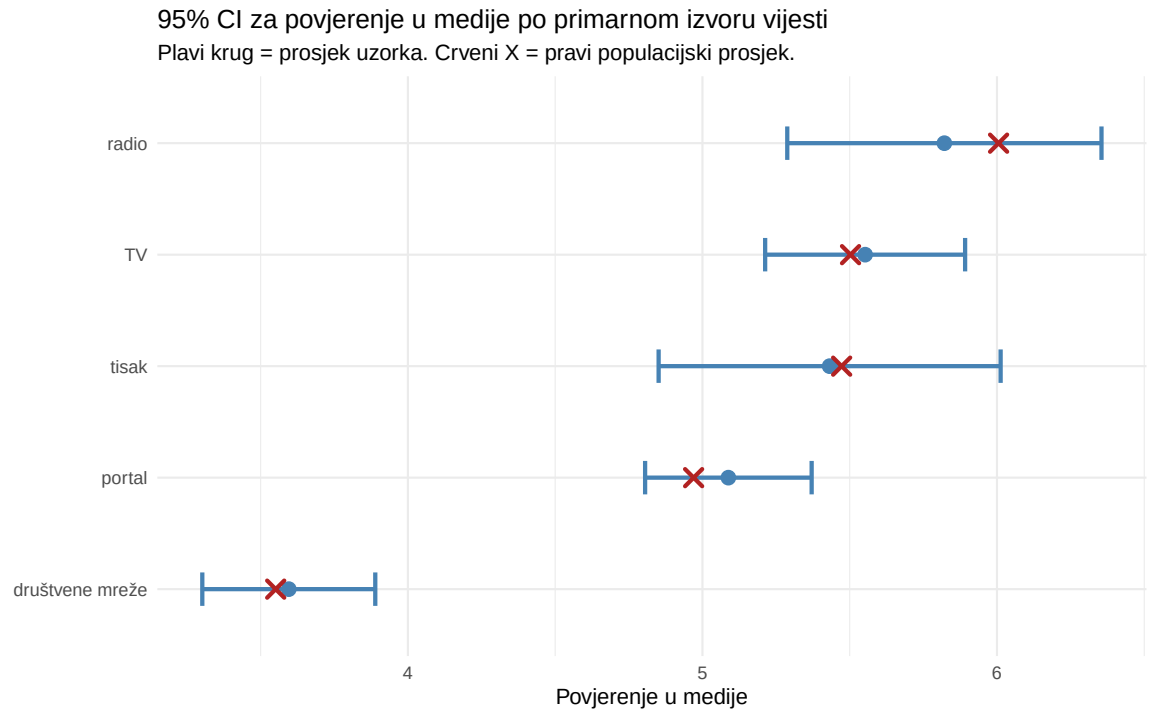
```
mu_pop <- mean(pop$media_trust)

tibble(
  razina = factor(c("90%", "95%", "99%"), levels = c("99%", "95%", "90%")),
  lo = c(ci_90[1], ci_95[1], ci_99[1]),
  hi = c(ci_90[2], ci_95[2], ci_99[2]),
  xbar = xbar
) |>
  ggplot(aes(y = razina)) +
  geom_errorbarh(aes(xmin = lo, xmax = hi), height = 0.3, linewidth = 1.2, color = "steelb
  geom_point(aes(x = xbar), size = 3, color = "steelblue") +
  geom_vline(xintercept = mu_pop, color = "firebrick", linewidth = 0.8, linetype = "dashed
  labs(
    title = "Veća pouzdanost = širi interval",
    subtitle = "Crvena crta = pravi populacijski prosjek",
    x = "Povjerenje u medije",
    y = "Razina pouzdanosti"
  ) +
  theme_minimal()
```



9.12.2 CI za podgrupe

U praksi nas često zanima CI za specifične podgrupe, ne samo za cijeli uzorak.

```r
set.seed(42)
uzorak_500 <- pop |> slice_sample(n = 500)

ci_po_izvoru <- uzorak_500 |>
  group_by(primary_news_source) |>
  filter(n() >= 20) |>
  summarise(
    n = n(),
    prosjek = mean(media_trust),
    se = sd(media_trust) / sqrt(n()),
    ci_lo = prosjek - qt(0.975, n() - 1) * se,
    ci_hi = prosjek + qt(0.975, n() - 1) * se,
    .groups = "drop"
  )

# Pravi populacijski prosjeci za usporedbu
pop_prosjeci <- pop |>
  group_by(primary_news_source) |>
  summarise(mu = mean(media_trust), .groups = "drop")

ci_po_izvoru |>
  left_join(pop_prosjeci, by = "primary_news_source") |>
  mutate(primary_news_source = fct_reorder(primary_news_source, prosjek)) |>
  ggplot(aes(y = primary_news_source)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "st
  geom_point(aes(x = prosjek), size = 3, color = "steelblue") +
  geom_point(aes(x = mu), size = 3, shape = 4, color = "firebrick", stroke = 1.5) +
  labs(
    title = "95% CI za povjerenje u medije po primarnom izvoru vijesti",
    subtitle = "Plavi krug = prosjek uzorka. Crveni X = pravi populacijski prosjek.",
    x = "Povjerenje u medije",
    y = NULL
  ) +
  theme_minimal()
```



Ovaj graf je izuzetno koristan za prezentaciju rezultata. Kad se intervali dviju grupa ne preklapaju, to sugerira statistički značajnu razliku, što ćemo formalno obraditi na predavanju o t-testu. Korisnici radija imaju najviše povjerenje, a korisnici društvenih mreža najmanje.

> **Praktični savjet**
>
> Kad prezentirate rezultate istraživanja, graf s intervalima pouzdanosti govori mnogo više od tablice prosjeka. Uključuje i veličinu uzorka (uži interval = više podataka) i nesigurnost procjene (širi interval = manje sigurni u točnu vrijednost). Naviknite se koristiti ovaj tip grafa.

9.13 Interval pouzdanosti za proporcije

Za proporcije (udjele), CI se računa malo drugačije jer je SE za proporciju $\sqrt{(\hat{p}(1-\hat{p})/n)}$.

```
set.seed(42)
uzorak_500 <- pop |> slice_sample(n = 500)

# Udio koji koristi portal kao primarni izvor
p_hat <- mean(uzorak_500$primary_news_source == "portal")
se_p <- sqrt(p_hat * (1 - p_hat) / 500)
```

```r
ci_lo <- p_hat - 1.96 * se_p
ci_hi <- p_hat + 1.96 * se_p

cat("Procjena p̂:", round(p_hat, 3), "\n")
```

```
Procjena p̂: 0.318
```

```r
cat("SE:", round(se_p, 3), "\n")
```

```
SE: 0.021
```

```r
cat("95% CI: [", round(ci_lo, 3), ",", round(ci_hi, 3), "]\n")
```

```
95% CI: [ 0.277 , 0.359 ]
```

```r
cat("Pravi populacijski udio:", round(mean(pop$primary_news_source == "portal"), 3), "\n")
```

```
Pravi populacijski udio: 0.304
```

9.13.1 `prop.test()` za proporcije

R ima funkciju `prop.test()` koja računa CI za proporcije. Koristi malo drugačiju metodu (Wilsonov interval) koja je preciznija za male uzorke i ekstremne proporcije.

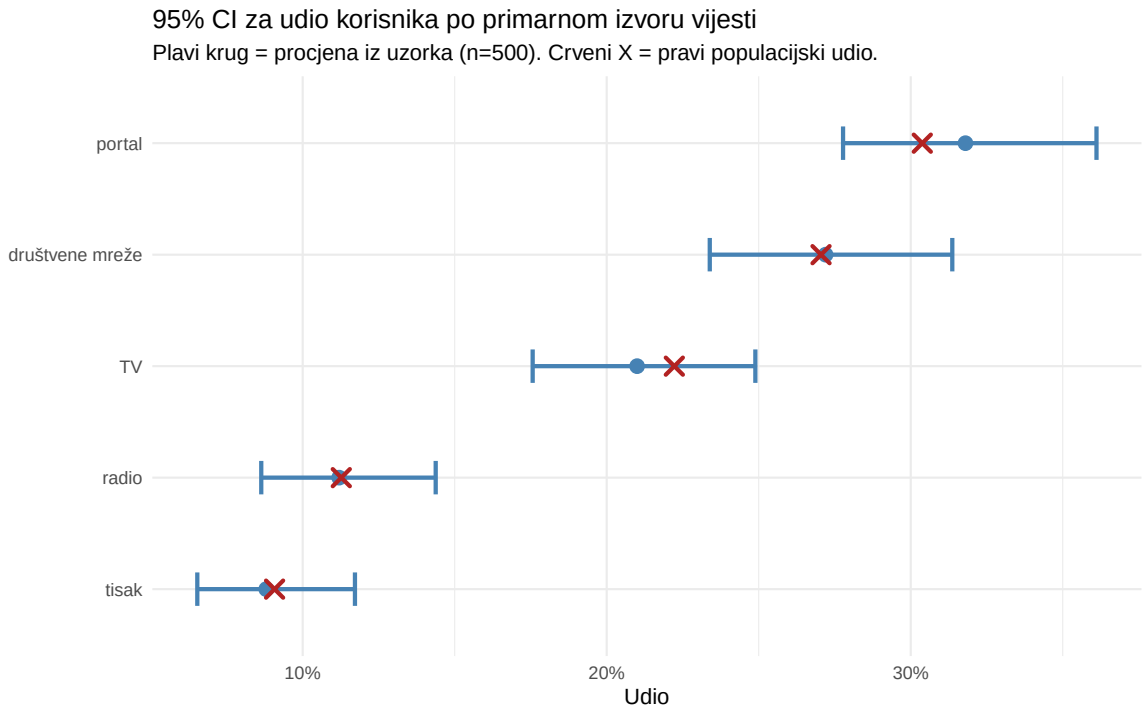
```r
# Koliko koristi portal od 500 ispitanika?
n_portal <- sum(uzorak_500$primary_news_source == "portal")

prop_rez <- prop.test(n_portal, n = 500, conf.level = 0.95)

cat("Procjena:", round(prop_rez$estimate, 3), "\n")
```

```
Procjena: 0.318
```

```r
cat("95% CI: [", round(prop_rez$conf.int[1], 3), ",", round(prop_rez$conf.int[2], 3), "]\n
```

```
95% CI: [ 0.278 , 0.361 ]
```

```r
# CI za sve izvore vijesti
izvore <- unique(pop$primary_news_source)

ci_izvore <- map_df(izvore, \(izvor) {
  n_da <- sum(uzorak_500$primary_news_source == izvor)
  test <- prop.test(n_da, n = 500, conf.level = 0.95)
  tibble(
    izvor = izvor,
    p_hat = test$estimate,
    ci_lo = test$conf.int[1],
    ci_hi = test$conf.int[2]
  )
})

# Pravi populacijski udjeli
pop_udjeli <- pop |>
  count(primary_news_source) |>
  mutate(udio = n / sum(n))

ci_izvore |>
  left_join(pop_udjeli, by = c("izvor" = "primary_news_source")) |>
  mutate(izvor = fct_reorder(izvor, p_hat)) |>
  ggplot(aes(y = izvor)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "st
  geom_point(aes(x = p_hat), size = 3, color = "steelblue") +
  geom_point(aes(x = udio), size = 3, shape = 4, color = "firebrick", stroke = 1.5) +
  scale_x_continuous(labels = scales::label_percent()) +
  labs(
    title = "95% CI za udio korisnika po primarnom izvoru vijesti",
    subtitle = "Plavi krug = procjena iz uzorka (n=500). Crveni X = pravi populacijski udi
    x = "Udio",
    y = NULL
  ) +
  theme_minimal()
```



Ovaj graf otkriva nešto što medijske ankete rijetko prikazuju - nesigurnost oko svakog broja. Portal i društvene mreže se ne mogu jasno razlučiti jer se intervali preklapaju. TV i radio se mogu jasno razlučiti jer se intervali ne preklapaju. Zato je prikaz intervala pouzdanosti uvijek pošteniji od samih postotaka.

9.14 Margina pogreške i planiranje uzorka

U medijskim izvještajima o anketama čujete izraz "margina pogreške ±3%". Što to znači i kako se računa?

Margina pogreške (margin of error, MoE) je pola širine intervala pouzdanosti. Za proporcije:

$$MoE = z^* \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Ako ne znamo p̂ unaprijed, koristimo najgori slučaj p̂ = 0.5 (koji daje najširu marginu):

$$MoE_{max} = \frac{z^*}{2\sqrt{n}}$$

Za 95% CI: MoE_max = 1.96 / (2√n) 1/√n.

```r
tibble(
  n = c(100, 200, 400, 500, 800, 1000, 1500, 2000),
  MoE_95 = round(1.96 * sqrt(0.25 / n) * 100, 1)
) |>
  mutate(opis = paste0("±", MoE_95, "%"))
```

```
# A tibble: 8 x 3
      n MoE_95 opis
  <dbl>  <dbl> <chr>
1   100    9.8 ±9.8%
2   200    6.9 ±6.9%
3   400    4.9 ±4.9%
4   500    4.4 ±4.4%
5   800    3.5 ±3.5%
6  1000    3.1 ±3.1%
7  1500    2.5 ±2.5%
8  2000    2.2 ±2.2%
```

Ovo objašnjava zašto su većina medijskih anketa u rasponu 500 do 1500 ispitanika. S n = 1000, margina je oko ±3.1%. S n = 2000, pada na ±2.2%. Poboljšanje je malo u odnosu na dodatni trošak i vrijeme.

9.14.1 Obrnuto: koliki uzorak trebam?

Češće pitanje u praksi je obrnuto — imam ciljanu marginu pogreške, koliki mi uzorak treba?

$$n = \frac{z^{*2} \times \hat{p}(1-\hat{p})}{MoE^2}$$

Za najgori slučaj ($\hat{p} = 0.5$):

$$n = \frac{z^{*2}}{4 \times MoE^2}$$

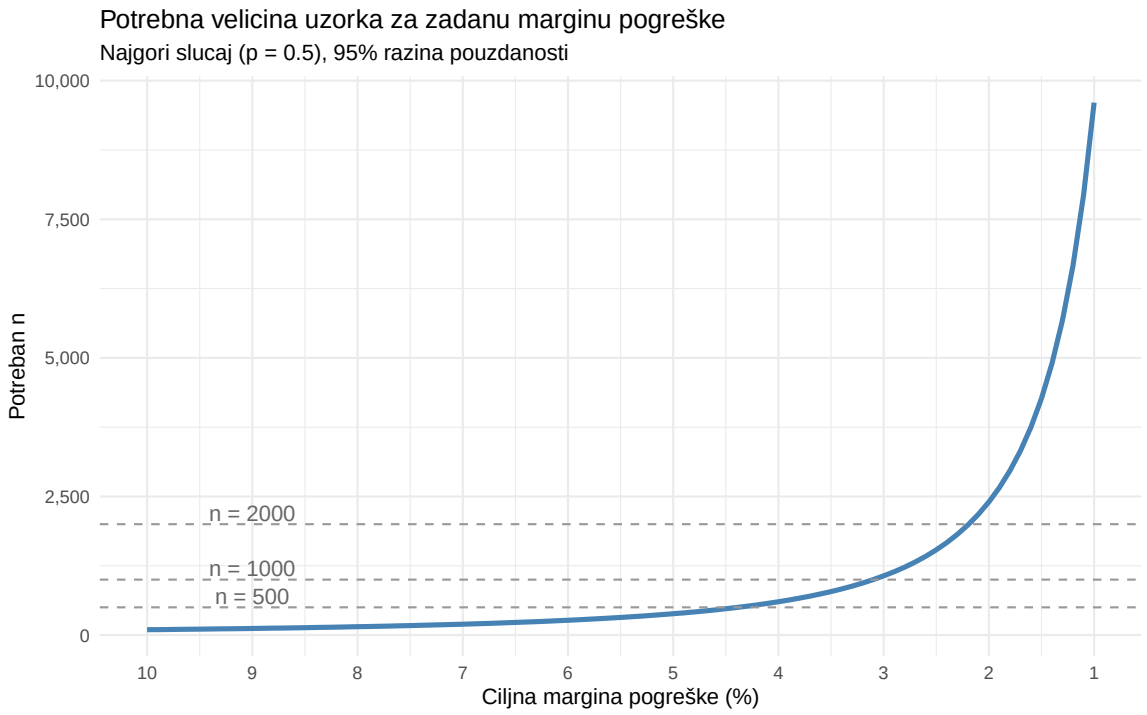
```r
# Funkcija za izračun potrebne veličine uzorka
velicina_uzorka <- function(moe, conf = 0.95, p = 0.5) {
  z <- qnorm(1 - (1 - conf) / 2)
  ceiling(z^2 * p * (1 - p) / moe^2)
}

tibble(
  ciljna_MoE = c("±5%", "±4%", "±3%", "±2%", "±1%"),
```

```
  moe = c(0.05, 0.04, 0.03, 0.02, 0.01),
  n_potreban = map_int(moe, velicina_uzorka)
)
```

```
# A tibble: 5 x 3
  ciljna_MoE   moe n_potreban
  <chr>      <dbl>      <int>
1 ±5%         0.05        385
2 ±4%         0.04        601
3 ±3%         0.03       1068
4 ±2%         0.02       2401
5 ±1%         0.01       9604
```

Za marginu od ±3% trebate 1068 ispitanika. Za ±2% trebate 2401. Za ±1% trebate čak 9604. Ovo ponovno potvrđuje zakon opadajućih prinosa, gdje je svako sljedeće poboljšanje sve skuplje.

```
tibble(
  moe = seq(0.01, 0.10, by = 0.001)
) |>
  mutate(n = map_dbl(moe, velicina_uzorka)) |>
  ggplot(aes(x = moe * 100, y = n)) +
  geom_line(color = "steelblue", linewidth = 1.2) +
  geom_hline(yintercept = c(500, 1000, 2000), linetype = "dashed", color = "grey60") +
  annotate("text", x = 9, y = c(500, 1000, 2000) + 200,
           label = c("n = 500", "n = 1000", "n = 2000"), color = "grey40") +
  scale_x_reverse(breaks = seq(1, 10, 1)) +
  scale_y_continuous(labels = scales::label_comma()) +
  labs(
    title = "Potrebna veličina uzorka za zadanu marginu pogreške",
    subtitle = "Najgori slučaj (p = 0.5), 95% razina pouzdanosti",
    x = "Ciljna margina pogreške (%)",
    y = "Potreban n"
  ) +
  theme_minimal()
```



Praktični savjet

Kad planirate istraživanje, odlučite o margini pogreške PRIJE nego počnete prikupljati podatke. Pitajte se koja razlika je praktično važna u vašem kontekstu. Ako vas zanima razlikuje li se popularnost dviju platformi za 5 postotnih bodova, trebate marginu manju od 5%, što znači uzorak od barem 400. Ako trebate razlučiti razlike od 2 postotna boda, trebate barem 2400 ispitanika.

9.15 Čitanje medijskih anketa kritički

Naučeno dosad daje nam alate za kritičku evaluaciju medijskih izvještaja o anketama. Pogledajmo tipičan primjer.

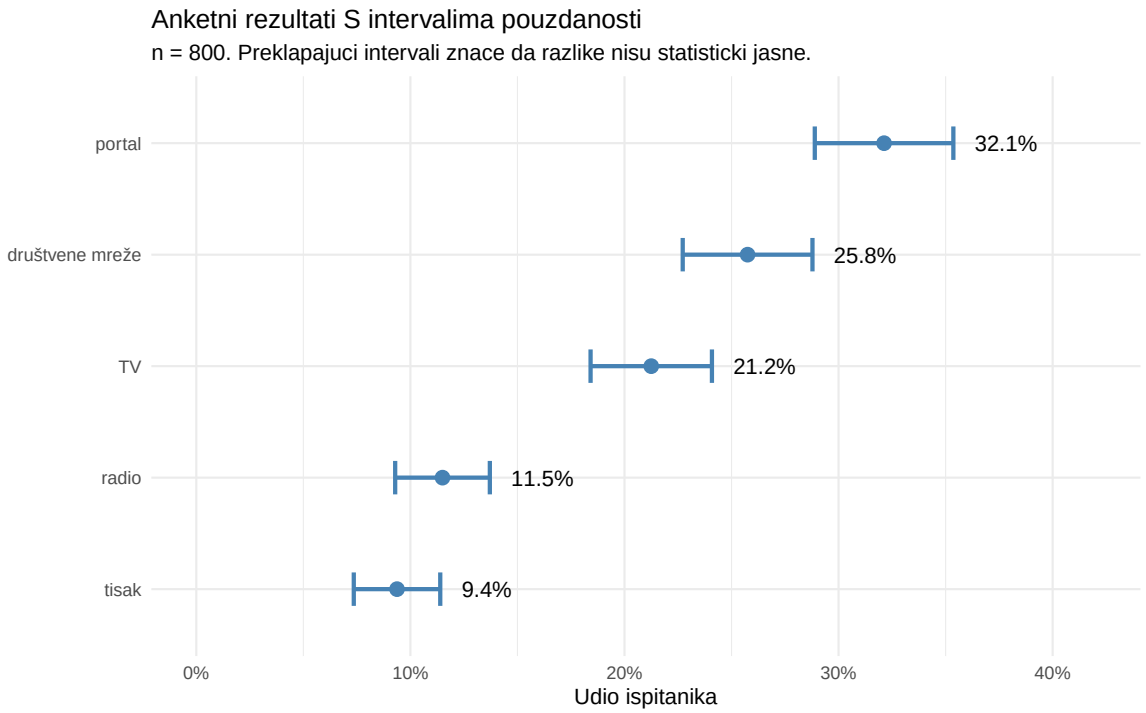
```
# Simulacija: medijska anketa o primarnom izvoru vijesti
set.seed(123)
anketa <- pop |> slice_sample(n = 800)

rezultati <- anketa |>
  count(primary_news_source) |>
  mutate(
```

```
    udio = n / sum(n),
    se = sqrt(udio * (1 - udio) / sum(n)),
    moe = 1.96 * se,
    ci_lo = udio - moe,
    ci_hi = udio + moe
  ) |>
  arrange(desc(udio))

rezultati |>
  mutate(across(c(udio, se, moe, ci_lo, ci_hi), \(x) round(x * 100, 1)))
```

```
# A tibble: 5 x 7
  primary_news_source     n  udio    se   moe ci_lo ci_hi
  <chr>               <int> <dbl> <dbl> <dbl> <dbl> <dbl>
1 portal                257  32.1   1.7   3.2  28.9  35.4
2 društvene mreže       206  25.8   1.5   3    22.7  28.8
3 TV                    170  21.2   1.4   2.8  18.4  24.1
4 radio                  92  11.5   1.1   2.2   9.3  13.7
5 tisak                  75   9.4   1     2     7.4  11.4
```

Novinar piše da je portal najpopularniji izvor vijesti (31%), a društvene mreže su na drugom mjestu (26%). Tehnički je to točno, ali zanemaruje intervale pouzdanosti. Kad uzmemo u obzir marginu pogreške, intervali za portal i društvene mreže se preklapaju. Ne možemo sa sigurnošću tvrditi da je portal popularniji od društvenih mreža.

```
rezultati |>
  mutate(primary_news_source = fct_reorder(primary_news_source, udio)) |>
  ggplot(aes(y = primary_news_source)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "st
  geom_point(aes(x = udio), size = 3, color = "steelblue") +
  geom_text(aes(x = ci_hi + 0.01, label = paste0(round(udio * 100, 1), "%")), hjust = 0) +
  scale_x_continuous(labels = scales::label_percent(), limits = c(0, 0.42)) +
  labs(
    title = "Anketni rezultati S intervalima pouzdanosti",
    subtitle = "n = 800. Preklapajući intervali znače da razlike nisu statistički jasne.",
    x = "Udio ispitanika",
    y = NULL
  ) +
  theme_minimal()
```



9.15.1 Kontrolna lista za čitanje anketa

Kad sretnete medijski izvještaj o anketi, postavite sedam pitanja. Koliki je uzorak? Ako ga ne navode, rezultati su sumnjivi. Kako su odabrali ispitanike? Slučajno telefonsko pozivanje ili online panel? Kolika je margina pogreške? Ako je navode samo na dnu stranice, obratite posebnu pažnju. Jesu li razlike veće od margine pogreške? Ako su razlike manje od dvostruke margine, zaključci su na klimavim nogama. Kad je anketa provedena? Stavovi se mogu promijeniti brzo. Tko je naručio anketu? Naručitelj može utjecati na formulaciju pitanja. Koliki je odaziv? Nizak odaziv (ispod 30%) sugerira self-selection bias.

9.16 Bootstrapping: alternativni pristup

Ponekad ne možemo pretpostaviti normalnost distribucije uzorkovanja, bilo zato što je uzorak premalen ili zato što nas zanima statistika za koju nemamo jednostavnu formulu za SE (medijan, omjer medijana i prosjeka, razlika između 90. i 10. percentila). **Bootstrap** je računalna metoda koja rješava ovaj problem.

Ideja je elegantna. Budući da ne možemo uzimati nove uzorke iz populacije (jer nemamo pristup cijeloj populaciji), uzimamo nove uzorke iz uzorka, s vraćanjem (with replacement).

```r
set.seed(42)
uzorak_50 <- pop |> slice_sample(n = 50)

# 5000 bootstrap uzoraka
boot_prosjeci <- map_dbl(1:5000, \(i) {
  uzorak_50 |>
    slice_sample(n = 50, replace = TRUE) |>
    pull(media_trust) |>
    mean()
})

# Bootstrap CI (percentilna metoda)
boot_ci <- quantile(boot_prosjeci, probs = c(0.025, 0.975))

# Usporedba s t-testom
t_ci <- t.test(uzorak_50$media_trust)$conf.int

cat("Bootstrap 95% CI: [", round(boot_ci[1], 3), ",", round(boot_ci[2], 3), "]\n")
```

```
Bootstrap 95% CI: [ 4.02 , 5.26 ]
```

```r
cat("t-test   95% CI: [", round(t_ci[1], 3), ",", round(t_ci[2], 3), "]\n")
```

```
t-test   95% CI: [ 4.02 , 5.26 ]
```

```r
cat("Pravi prosjek:   ", round(mean(pop$media_trust), 3), "\n")
```

```
Pravi prosjek:    4.867
```

Bootstrap i t-test daju vrlo slične rezultate kad su pretpostavke t-testa zadovoljene. Prednost bootstrapa je njegova fleksibilnost — možemo ga koristiti za bilo koju statistiku.

```r
# Bootstrap CI za MEDIJAN (za koji nema jednostavne formule za SE)
boot_medijani <- map_dbl(1:5000, \(i) {
  uzorak_50 |>
    slice_sample(n = 50, replace = TRUE) |>
    pull(daily_media_min) |>
    median()
})

boot_ci_medijan <- quantile(boot_medijani, probs = c(0.025, 0.975))

cat("Medijan uzorka:", median(uzorak_50$daily_media_min), "\n")
```

```
Medijan uzorka: 190
```

```
cat("Bootstrap 95% CI za medijan: [", boot_ci_medijan[1], ",", boot_ci_medijan[2], "]\n")
```

```
Bootstrap 95% CI za medijan: [ 166.5 , 200 ]
```

```
cat("Pravi populacijski medijan:", median(pop$daily_media_min), "\n")
```

```
Pravi populacijski medijan: 172
```

> **Kada koristiti bootstrap?**
>
> Koristite bootstrap kad nemate formulu za SE željene statistike, sumnjate u normalnost distribucije uzorkovanja, imate mali uzorak i tražite robusniju metodu, ili želite CI za medijan, percentile, omjere ili druge nestandardne mjere. Za prosjeke s $n > 30$, t-test je jednako dobar i jednostavniji.

9.17 Potpuna analiza: povjerenje u medije po demografskim skupinama

Spojimo sve naučeno u jednu koherentnu analizu. Situacija je sljedeća — provedena je anketa na 600 ispitanika o medijskim navikama. Trebamo procijeniti povjerenje u medije ukupno i po ključnim podgrupama te interpretirati rezultate.

```
set.seed(2025)
anketa <- pop |> slice_sample(n = 600)

cat("Veličina uzorka:", nrow(anketa), "\n\n")
```

```
Veličina uzorka: 600
```

```
# Korak 1: Ukupna procjena
ukupno <- t.test(anketa$media_trust)
cat("UKUPNO POVJERENJE U MEDIJE\n")
```

```
UKUPNO POVJERENJE U MEDIJE
```

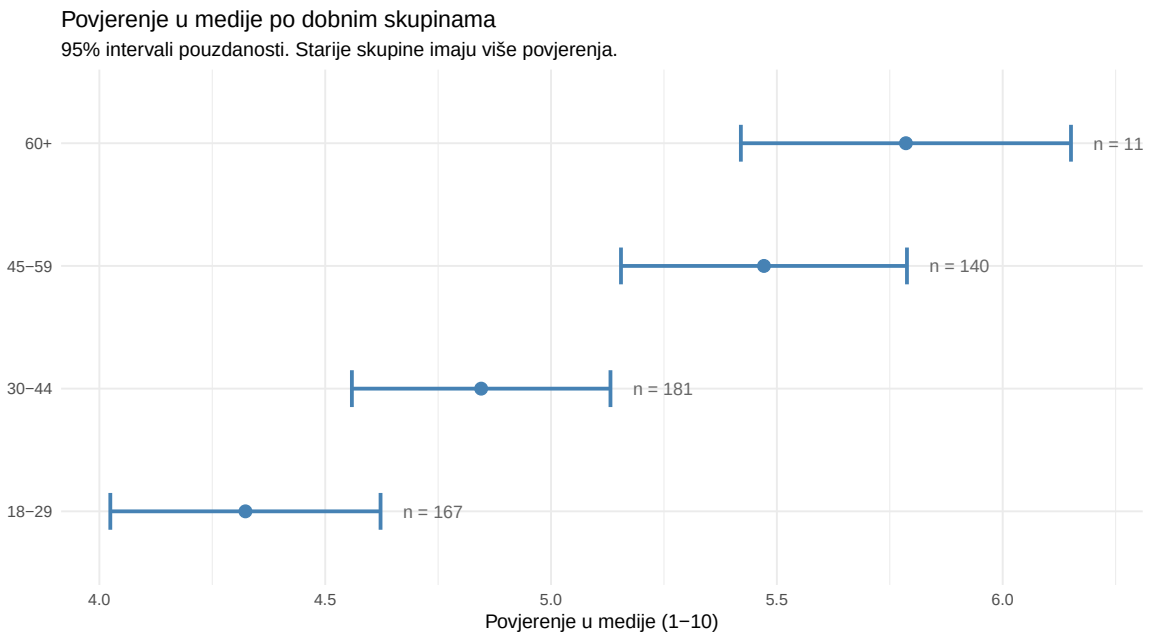
```r
cat("Prosjek:", round(ukupno$estimate, 2), "\n")
```

```
Prosjek: 5.02
```

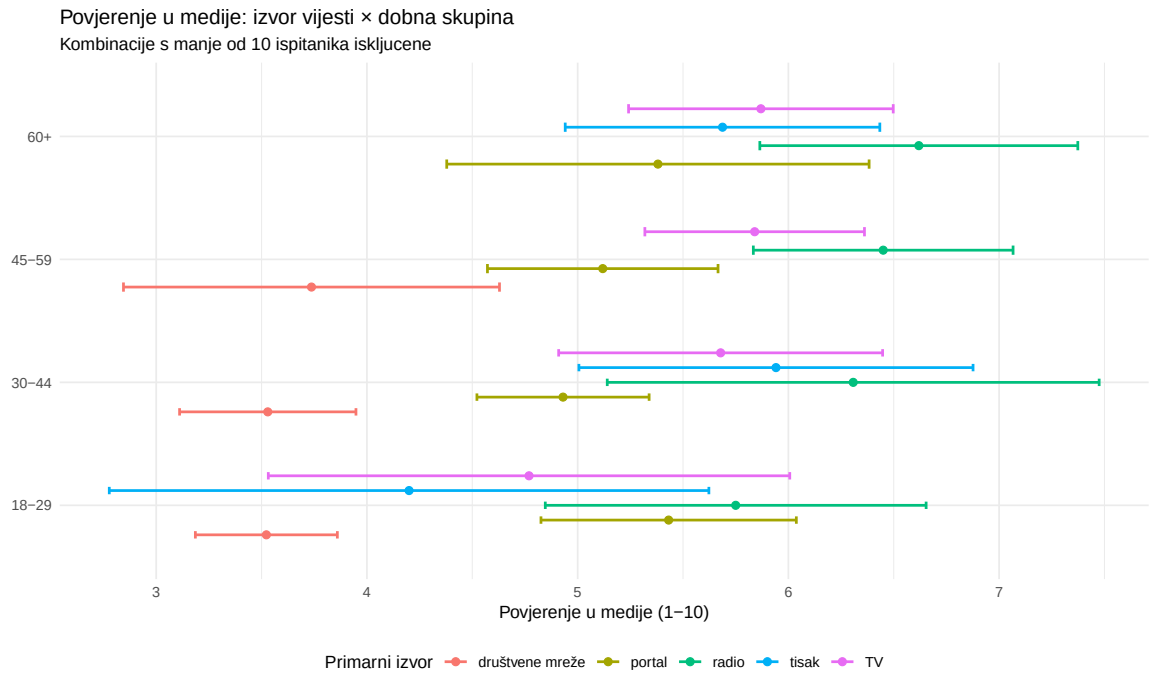
```r
cat("95% CI: [", round(ukupno$conf.int[1], 2), ",", round(ukupno$conf.int[2], 2), "]\n\n")
```

```
95% CI: [ 4.86 , 5.18 ]
```

```r
# Korak 2: CI po dobnim skupinama
anketa <- anketa |>
  mutate(dobna_skupina = case_when(
    age < 30 ~ "18-29",
    age < 45 ~ "30-44",
    age < 60 ~ "45-59",
    .default = "60+"
  ))

ci_dob <- anketa |>
  group_by(dobna_skupina) |>
  summarise(
    n = n(),
    prosjek = mean(media_trust),
    se = sd(media_trust) / sqrt(n()),
    ci_lo = prosjek - qt(0.975, n() - 1) * se,
    ci_hi = prosjek + qt(0.975, n() - 1) * se,
    .groups = "drop"
  )

ci_dob |>
  ggplot(aes(y = dobna_skupina)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "st
  geom_point(aes(x = prosjek), size = 3, color = "steelblue") +
  geom_text(aes(x = ci_hi + 0.05, label = paste0("n = ", n)), hjust = 0, size = 3.5, color
  labs(
    title = "Povjerenje u medije po dobnim skupinama",
    subtitle = "95% intervali pouzdanosti. Starije skupine imaju više povjerenja.",
    x = "Povjerenje u medije (1-10)",
    y = NULL
  ) +
  theme_minimal()
```

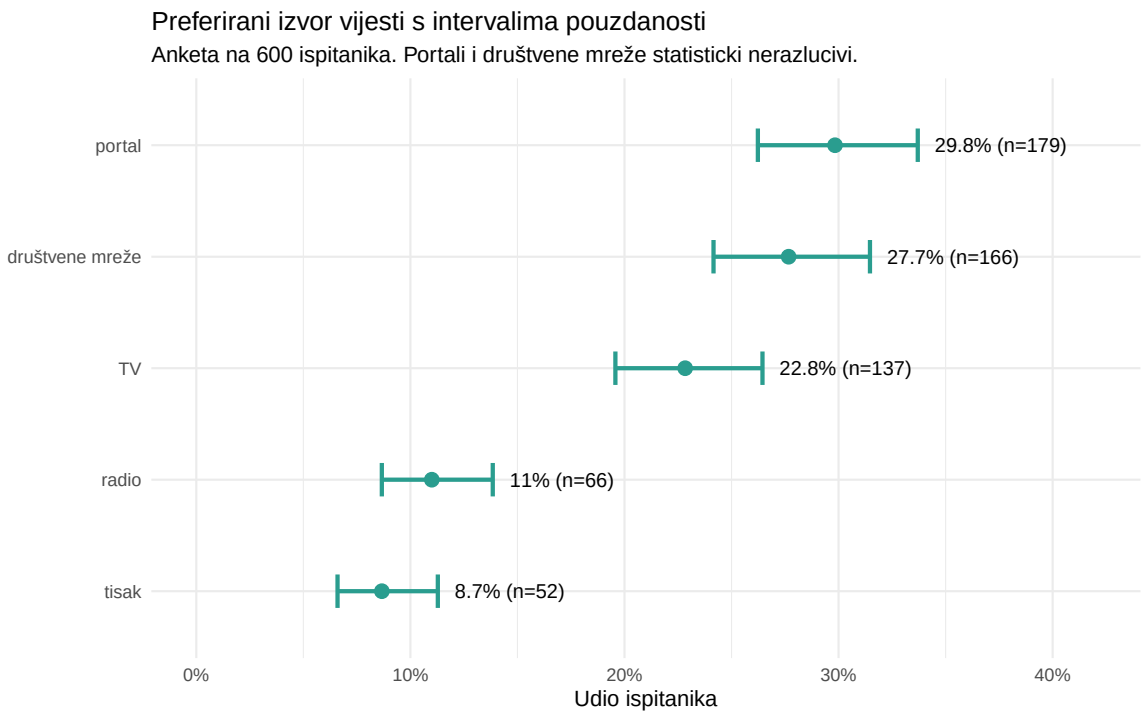



```
# Korak 3: Unakrsna analiza: izvor vijesti × dobna skupina
ci_krizno <- anketa |>
  group_by(primary_news_source, dobna_skupina) |>
  filter(n() >= 10) |>
  summarise(
    n = n(),
    prosjek = mean(media_trust),
    se = sd(media_trust) / sqrt(n()),
    ci_lo = prosjek - qt(0.975, n() - 1) * se,
    ci_hi = prosjek + qt(0.975, n() - 1) * se,
    .groups = "drop"
  )

ci_krizno |>
  ggplot(aes(y = dobna_skupina, color = primary_news_source)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 0.8,
                 position = position_dodge(0.6)) +
  geom_point(aes(x = prosjek), size = 2, position = position_dodge(0.6)) +
  labs(
    title = "Povjerenje u medije: izvor vijesti × dobna skupina",
    subtitle = "Kombinacije s manje od 10 ispitanika isključene",
    x = "Povjerenje u medije (1-10)",
    y = NULL,
    color = "Primarni izvor"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
# Korak 4: Proporcije po izvoru vijesti s CI
prop_rezultati <- anketa |>
  count(primary_news_source) |>
  mutate(
    p_hat = n / sum(n),
    test = map2(n, sum(n), \(x, nn) prop.test(x, nn, conf.level = 0.95)),
    ci_lo = map_dbl(test, \(t) t$conf.int[1]),
    ci_hi = map_dbl(test, \(t) t$conf.int[2])
  ) |>
  select(-test) |>
  arrange(desc(p_hat))

prop_rezultati |>
  mutate(primary_news_source = fct_reorder(primary_news_source, p_hat)) |>
  ggplot(aes(y = primary_news_source)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "#2
  geom_point(aes(x = p_hat), size = 3, color = "#2a9d8f") +
  geom_text(aes(x = ci_hi + 0.008, label = paste0(round(p_hat * 100, 1), "% (n=", n, ")"))
            hjust = 0, size = 3.5) +
  scale_x_continuous(labels = scales::label_percent(), limits = c(0, 0.42)) +
  labs(
    title = "Preferirani izvor vijesti s intervalima pouzdanosti",
    subtitle = "Anketa na 600 ispitanika. Portali i društvene mreže statistički nerazlučiv
    x = "Udio ispitanika",
    y = NULL
```

```
  ) +
  theme_minimal()
```




```
# Korak 5: Sažetak za klijenta
cat("=== SAŽETAK REZULTATA ANKETE ===\n\n")
```

```
=== SAŽETAK REZULTATA ANKETE ===
```

```
cat("Uzorak: n =", nrow(anketa), "ispitanika\n")
```

```
Uzorak: n = 600 ispitanika
```

```
cat("Margina pogreške za proporcije: ±", round(1.96 * sqrt(0.25 / nrow(anketa)) * 100, 1),
```

```
Margina pogreške za proporcije: ± 4 %
```

```
cat("1. Ukupno povjerenje u medije:", round(ukupno$estimate, 2),
    "(95% CI:", round(ukupno$conf.int[1], 2), "-", round(ukupno$conf.int[2], 2), ")\n")
```

```
1. Ukupno povjerenje u medije: 5.02 (95% CI: 4.86 - 5.18 )
```

```
cat("   Na ljestvici od 1-10, to je ispod sredine.\n\n")
```

```
   Na ljestvici od 1-10, to je ispod sredine.
```

```
cat("2. Najpopularniji izvori vijesti:\n")
```

```
2. Najpopularniji izvori vijesti:
```

```
for (i in 1:nrow(prop_rezultati)) {
  cat("   ", prop_rezultati$primary_news_source[i], ":",
      round(prop_rezultati$p_hat[i] * 100, 1), "% (",
      round(prop_rezultati$ci_lo[i] * 100, 1), "-",
      round(prop_rezultati$ci_hi[i] * 100, 1), "%)\n")
}
```

```
    portal : 29.8 % ( 26.2 - 33.7 %)
    društvene mreže : 27.7 % ( 24.2 - 31.5 %)
    TV : 22.8 % ( 19.6 - 26.4 %)
    radio : 11 % ( 8.7 - 13.8 %)
    tisak : 8.7 % ( 6.6 - 11.3 %)
```

```
cat("\n3. Napomena: razlika između portala i društvenih mreža je unutar margine\n")
```

```
3. Napomena: razlika između portala i društvenih mreža je unutar margine
```

```
cat("   pogreške i ne može se smatrati statistički značajnom.\n")
```

```
   pogreške i ne može se smatrati statistički značajnom.
```

9.18 Uobičajene pogreške pri interpretaciji CI

Intervali pouzdanosti su intuitivno privlačni ali se često krivo interpretiraju. Evo najčešćih grešaka i ispravnih verzija.

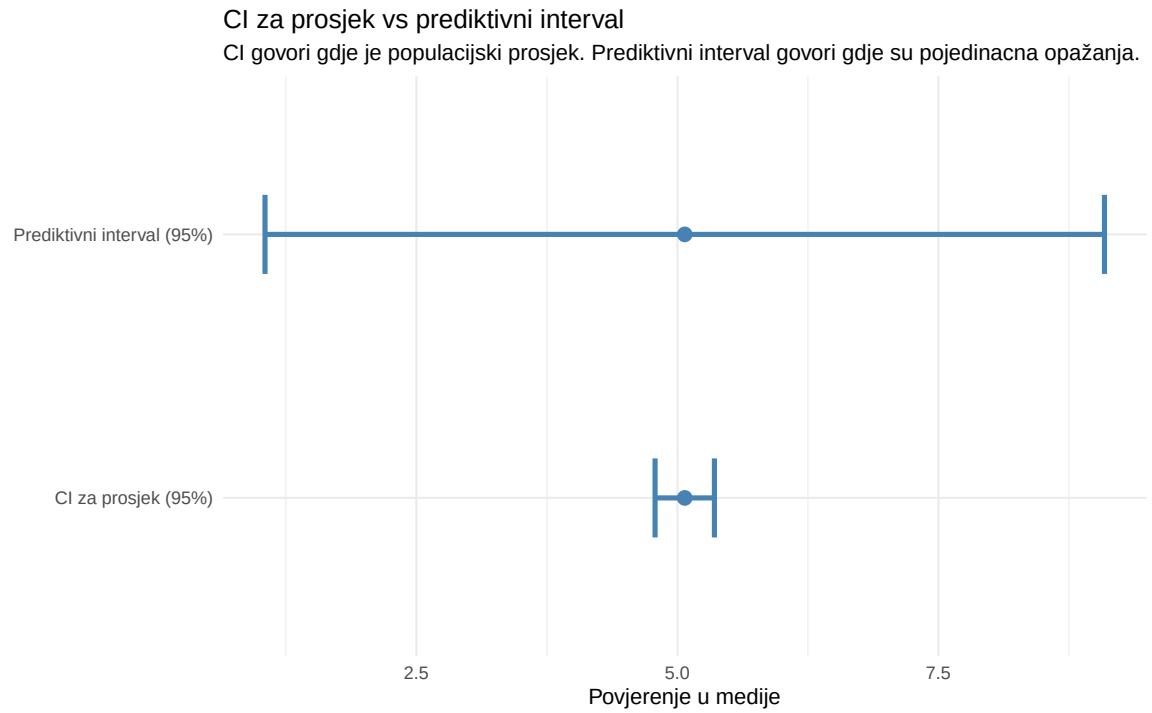
Pogrešno: “Postoji 95% šansa da je pravi prosjek unutar intervala [4.67, 5.01].” **Ispravno**: Pravi prosjek je fiksan broj. On ili jest ili nije unutar intervala. 95% se odnosi na postupak — 95% intervala konstruiranih ovom metodom pokrit će pravi parametar.

Pogrešno: “95% podataka pada unutar intervala [4.67, 5.01].” **Ispravno**: CI se odnosi na parametar (prosjek), ne na pojedinačna opažanja. Pojedinačne vrijednosti pokriva prediktivni interval, koji je mnogo širi.

Pogrešno: “Ako dva CI-a ne uključuju nulu, razlika je značajna.” **Ispravno**: Nula nije relevantna za pojedinačne CI-e. Preklapanje dvaju CI-a govori o mogućoj razlici, ali formalni test zahtijeva CI za razliku, što će biti obrađeno na predavanju o t-testu.

Pogrešno: “Širok CI znači da je mjerenje loše provedeno.” **Ispravno**: Širok CI obično znači mali uzorak ili veliku varijabilnost u podacima. To nije greška, nego realnost podataka.

```
set.seed(42)
uzorak_200 <- pop |> slice_sample(n = 200)

xbar <- mean(uzorak_200$media_trust)
se <- sd(uzorak_200$media_trust) / sqrt(200)
s <- sd(uzorak_200$media_trust)

tibble(
  tip = factor(c("CI za prosjek (95%)", "Prediktivni interval (95%)"),
               levels = c("CI za prosjek (95%)", "Prediktivni interval (95%)")),
  lo = c(xbar - 1.96 * se, xbar - 1.96 * s),
  hi = c(xbar + 1.96 * se, xbar + 1.96 * s),
  xbar = xbar
) |>
  ggplot(aes(y = tip)) +
  geom_errorbarh(aes(xmin = lo, xmax = hi), height = 0.3, linewidth = 1.2, color = "steelb
  geom_point(aes(x = xbar), size = 3, color = "steelblue") +
  labs(
    title = "CI za prosjek vs prediktivni interval",
    subtitle = "CI govori gdje je populacijski prosjek. Prediktivni interval govori gdje s
    x = "Povjerenje u medije",
    y = NULL
  ) +
  theme_minimal()
```



CI za prosjek je uzak (±0.28). Prediktivni interval je širok (±3.9). Ovo su dva potpuno različita pitanja gdje se razmatra gdje je pravi prosjek, odnosno gdje će pasti sljedeće opažanje. Ne miješajte ih.

! Ključni zaključci

1. Populacija je cjelina o kojoj zaključujemo. Uzorak je dio koji mjerimo. Parametri (,) opisuju populaciju. Statistike ($\bar{x}$, s) opisuju uzorak i procjenjuju parametre.
2. Svaki uzorak daje malo drugačiju procjenu. Distribucija tih procjena kroz ponovljene uzorke naziva se distribucija uzorkovanja. Njezina standardna devijacija je standardna pogreška (SE).
3. Standardna pogreška SE = s/√n mjeri koliko prosjeci uzoraka tipično variraju. Preciznost raste s korijenom veličine uzorka — da biste prepolovili SE, morate učetverostručiti n.
4. Centralni granični teorem kaže da je distribucija uzorkovanja prosjeka približno normalna za dovoljno velik n (pravilo palca n 30), neovisno o obliku izvorne distribucije.
5. Pristranost uzorka (convenience sampling, self-selection) je veći problem od male veličine uzorka. Velik pristran uzorak daje sustavno pogrešne rezultate koje ne može ispraviti nijedna statistička metoda.

6. t-distribucija se koristi umjesto normalne kad procjenjujemo iz uzorka. Za male uzorke daje šire intervale (deblje repove). Za n > 100, razlika je zanemariva.

7. `t.test()` računa interval pouzdanosti za prosjek. `prop.test()` računa CI za proporciju. Obje funkcije automatski koriste ispravne formule.

8. 95% CI znači da 95% ovako konstruiranih intervala pokriva pravi parametar. NE znači "95% šansa da je parametar unutar intervala." Parametar je fiksan. Interval je slučajan.

9. Margina pogreške za proporcije je približno $1/\sqrt{n}$, što daje za n = 1000 oko ±3.1%, a za n = 400 oko ±4.9%. To objašnjava uobičajene veličine uzoraka u anketama.

10. Kad planirate istraživanje, odredite ciljanu preciznost unaprijed i iz nje izvedite potrebnu veličinu uzorka koristeći formulu $n = z^{*2} \times p(1-p) / MoE^2$.

11. Bootstrap je računalna alternativa za CI kad nemamo formulu za SE željene statistike. Ideja je sljedeća — uzorkuj iz uzorka s vraćanjem, izračunaj statistiku, ponovi mnogo puta.

12. Kod čitanja medijskih anketa uvijek provjerite veličinu uzorka, metodu uzorkovanja, marginu pogreške i jesu li prijavljivane razlike veće od margine. Ako razlika između dvaju postotaka nije veća od dvostruke margine pogreške, zaključak da je neka opcija "u vodstvu" nije opravdan.

9.19 Zadaci za pripremu

1. Učitajte `media_population.csv` i izračunajte pravi populacijski prosjek za `daily_media_min`. Zatim uzmite 50 slučajnih uzoraka veličine n = 300 i za svaki izračunajte 95% CI pomoću `t.test()`. Koliki postotak intervala pokriva pravi prosjek?

2. Izračunajte potrebnu veličinu uzorka za anketu o preferencijama izvora vijesti s marginom pogreške ±2.5% na razini pouzdanosti 99%.

3. Napišite funkciju `bootstrap_ci(x, stat_fn, n_boot = 5000, conf = 0.95)` koja prima vektor `x`, funkciju `stat_fn` (npr. `mean` ili `median`) i vraća bootstrap CI. Testirajte je na varijabli `daily_media_min`.

9.20 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 10 (Estimating Unknown Quantities from a Sample). Besplatno dostupno na learningstatisticswithr.com. Pokriva uzorkovanje, CLT i intervale pouzdanosti s R kodom i odličnim objašnjenjima.

Diez, D., Çetinkaya-Rundel, M., & Barr, C. (2019). *OpenIntro Statistics* (4th edition), Chapter 5. Besplatno dostupno na openintro.org/book/os. Odličan vizualni pregled distribucije uzorkovanja i CI-a.

Preporučeno

Wheelan, C. (2013). *Naked Statistics*. W. W. Norton. Poglavlja 8 i 9 pokrivaju centralni granični teorem i uzorkovanje na izuzetno pristupačan način.

Cumming, G. (2014). The New Statistics: Why and How. *Psychological Science*, 25(1), 7-29. Članak argumentira zašto intervali pouzdanosti i veličine učinka trebaju zamijeniti p-vrijednosti kao primarni način izvještavanja rezultata.

9.21 Pojmovnik

Pojam	Objašnjenje
Populacija	Cjelokupni skup jedinica o kojima želimo donijeti zaključak.
Uzorak	Podskup populacije koji zaista mjerimo.
Parametar	Mjera populacije. Označava se grčkim slovima (, ,). U praksi nepoznat.
Statistika	Mjera uzorka. Označava se latinskim slovima ($\bar{x}$, s, $\hat{p}$). Procjena parametra.
Pogreška uzorkovanja	Razlika između statistike i parametra. Neizbježna posljedica rada s uzorkom.
Distribucija uzorkovanja	Distribucija statistike kroz mnogo ponovljenih uzoraka. Osnova za statističko zaključivanje.
Standardna pogreška (SE)	Standardna devijacija distribucije uzorkovanja. Za prosjek: SE = $s/\sqrt{n}$.
Centralni granični teorem (CLT)	Distribucija uzorkovanja prosjeka je približno normalna za dovoljno velik n, neovisno o obliku izvorne distribucije.

Pojam	Objašnjenje
t-distribucija	Distribucija slična normalnoj ali s debljim repovima. Koristi se kad procjenjujemo iz uzorka.
Stupnjevi slobode (df)	Parametar t-distribucije. Za jedan prosjek: df = n − 1. Više df = bliže normalnoj.
Interval pouzdanosti (CI)	Raspon vrijednosti koji s određenom vjerojatnošću pokriva pravi parametar.
Razina pouzdanosti	Postotak intervala koji bi pokrio parametar u ponovljenom uzorkovanju (obično 95% ili 99%).
Margina pogreške (MoE)	Pola širine intervala pouzdanosti. Za proporcije $1/\sqrt{n}$.
Nepristrana procjena	Statistika čija distribucija uzorkovanja je centrirana oko pravog parametra.
Convenience sampling	Uzorkovanje iz dostupne (ali nereprezentativne) skupine. Uvodi pristranost.
Self-selection bias	Pristranost kad ispitanici sami odlučuju hoće li sudjelovati. Tipično za online ankete.
Proporcija uzorka ($\hat{p}$)	Udio uzorka koji ima neku karakteristiku. Procjena populacijske proporcije .
Bootstrap	Računalna metoda za procjenu SE i CI ponovljenim uzorkovanjem iz uzorka s vraćanjem.
Prediktivni interval	Interval koji pokriva buduća pojedinačna opažanja. Mnogo širi od CI za prosjek.
Točkasta procjena	Jedna brojčana vrijednost kao procjena parametra (npr. $\bar{x} = 4.87$). Ne govori o preciznosti.
`t.test()`	R funkcija za t-test i t-interval pouzdanosti za prosjek.
`prop.test()`	R funkcija za test i CI za proporcije. Koristi Wilsonov interval.
`slice_sample()`	dplyr funkcija za slučajno uzorkovanje redova iz tibble. Argument `replace = TRUE` za bootstrap.
`qt()`	R funkcija za kritične vrijednosti t-distribucije. `qt(0.975, df)` za 95% CI.
`qnorm()`	R funkcija za kritične vrijednosti normalne distribucije. `qnorm(0.975)` = 1.96.

10 Tjedan 9: Testiranje hipoteza

Kako donijeti odluku na temelju podataka

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete:

1. Formulirati nultu i alternativnu hipotezu za istraživačko pitanje.
2. Objasniti logiku testiranja hipoteza kroz analogiju sa suđenjem.
3. Izračunati i interpretirati testnu statistiku i p-vrijednost.
4. Provesti jednosmjerni i dvosmjerni t-test u R-u pomoću `t.test()`.
5. Objasniti razliku između greške tipa I i greške tipa II.
6. Izračunati i interpretirati Cohenov d kao mjeru veličine učinka.
7. Objasniti koncept statističke snage i faktore koji na nju utječu.
8. Kritički ocijeniti statističku značajnost u kontekstu praktične važnosti.

10.1 Jesu li carouseli zaista bolji?

Radite kao analitičarka u redakciji medijske kuće. Vaš Instagram profil objavljuje sadržaj u dva formata — carousel (objave s više slika koje korisnik lista) i obične slike (single image). Urednica vas jednog jutra zaustavi u hodniku i pita: "Imam osjećaj da carousel objave generiraju više angažmana. Imamo li za to dokaz?"

Vi znate odgovoriti na to pitanje. Otvarate podatke i gledate prosjeke — carousel objave imaju engagement rate od 10.1%, a obične slike 7.5%. Razlika postoji. Ali urednica nije pitala "je li prosjek različit u uzorku" — ona pita "možemo li se osloniti na tu razliku kad planiramo strategiju." A to je sasvim drugo pitanje. Možda je razlika realna i stabilna. Ali možda je samo artefakt — slučajni šum u podacima koji bi nestao kad bismo ponovili usporedbu na novom skupu objava.

Ovo je temeljno pitanje testiranja hipoteza — je li opažena razlika dovoljno velika da isključimo slučajnost kao objašnjenje? Drugim riječima, koliko bismo bili iznenađeni ovakvom razlikom da carousel zapravo *nije* bolji?

```r
ig <- read_csv("../resources/datasets/instagram_ab_test.csv")
glimpse(ig)
```

```
Rows: 500
Columns: 11
$ post_id         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ format          <chr> "carousel", "carousel", "carousel", "carousel", "carou~
$ topic           <chr> "sport", "tech", "sport", "vijesti", "kultura", "kultu~
$ time_of_day     <chr> "poslijepodne", "jutro", "večer", "poslijepodne", "pod~
$ has_cta         <lgl> TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRU~
$ reach           <dbl> 803, 1028, 4570, 1915, 4539, 3620, 3058, 2818, 4432, 2~
$ likes           <dbl> 48, 62, 200, 87, 428, 151, 61, 132, 212, 159, 210, 65,~
$ comments        <dbl> 14, 10, 94, 21, 47, 61, 27, 32, 108, 26, 42, 4, 30, 21~
$ shares          <dbl> 7, 8, 22, 25, 45, 50, 21, 27, 4, 14, 26, 5, 12, 14, 6,~
$ saves           <dbl> 16, 23, 49, 27, 67, 38, 30, 56, 84, 51, 28, 3, 12, 26,~
$ engagement_rate <dbl> 0.1059, 0.1002, 0.0799, 0.0836, 0.1293, 0.0829, 0.0455~
```

```r
ig |>
  group_by(format) |>
  summarise(
    n = n(),
    M_engagement = round(mean(engagement_rate) * 100, 2),
    SD_engagement = round(sd(engagement_rate) * 100, 2),
    M_likes = round(mean(likes), 1),
    M_comments = round(mean(comments), 1),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 6
  format           n M_engagement SD_engagement M_likes M_comments
  <chr>        <int>        <dbl>         <dbl>   <dbl>      <dbl>
1 carousel       236         10.1          2.11    180.       35.7
2 single_image   264          7.5          1.77    149.       24.4
```

Carousel objave imaju viši angažman u prosjeku. Ali svaka grupa ima i vlastitu varijabilnost — unutar carousela postoje sjajne i loše objave, isto kao i unutar običnih slika. Pitanje je — kolika je šansa da bismo vidjeli ovakvu ili veću razliku čak i da carousel zapravo nije bolji?

10.2 Logika testiranja hipoteza

Testiranje hipoteza slijedi logiku koja je iznenađujuće slična suđenju u kaznenom pravu. Na sudu, optuženik je nevin dok se ne dokaže krivnja. Ne morate dokazati nevinost — morate

dokazati krivnju, i to izvan razumne sumnje. Ako dokazi nisu dovoljno jaki, presuda nije "nevin" nego "nije dokazano."

U statistici, uloge su analogne. Početna pretpostavka je da nema učinka — nema razlike, nema veze, nema efekta. Ovu pretpostavku zovemo nulta hipoteza i označavamo je s H . Istraživač pokušava prikupiti dovoljno dokaza da odbaci nultu hipotezu u korist alternativne hipoteze (H), koja tvrdi da učinak postoji.

Za naš Instagram primjer, hipoteze izgledaju kao što su sljedeće — H i H .

H — Nema razlike u angažmanu između carousel i single image formata. Svaka opažena razlika je posljedica slučajnosti.

H — Postoji razlika u angažmanu između dva formata. Opažena razlika odražava stvarnu razliku u populaciji.

U matematičkom jeziku to izražavamo na sljedeći način.

$$H_0 : \mu_{carousel} = \mu_{single}$$
$$H_1 : \mu_{carousel} \neq \mu_{single}$$

> **!** Nulta hipoteza uvijek sadrži jednakost
>
> Nulta hipoteza uvijek sadrži znak jednakosti (= ili ili). Alternativna hipoteza sadrži znak nejednakosti (ili > ili <). Nikad obrnuto. Mi testiramo nultu hipotezu i tražimo dokaze *protiv* nje — baš kao što tužitelj traži dokaze protiv pretpostavke nevinosti.

10.3 Od hipoteze do odluke

Cijeli postupak testiranja hipoteza možete sažeti u pet koraka. Prvi — postavite hipoteze, jasno formulirajte H i H prije nego pogledate podatke. Drugi — odaberite razinu značajnosti , prag ispod kojeg ćete odbaciti H (konvencija je = 0.05, odnosno 5%). Treći — izračunajte testnu statistiku iz podataka, broj koji kvantificira koliko se vaši podaci razlikuju od očekivanih pod H . Četvrti — izračunajte p-vrijednost, vjerojatnost da biste dobili ovako ekstremnu ili ekstremniju testnu statistiku kad bi H bila istinita. Peti — donesite odluku, ako je p < , odbacujete H ; ako je p , ne možete je odbaciti.

Krenimo korak po korak na jednostavnijem primjeru prije nego se vratimo na Instagram podatke.

10.4 Jednouzorački t-test

Najjednostavniji oblik t-testa uspoređuje prosjek jednog uzorka s nekom poznatom ili pretpostavljenom vrijednošću. Evo konkretne situacije — medijska kuća tvrdi da njihovi korisnici provode prosječno 3 minute čitajući članak. Vi ste skeptični — vaš osjećaj je da je stvarno vrijeme kraće. Provedete mjerenje na uzorku od 45 članaka.

```
set.seed(42)

# Simulirani podaci: stvarno prosječno vrijeme je 2.6 minuta
vrijeme_citanja <- tibble(
  clanak_id = 1:45,
  minuta = round(rnorm(45, mean = 2.6, sd = 0.9), 1)
)

# Opisna statistika
vrijeme_citanja |>
  summarise(
    n = n(),
    M = round(mean(minuta), 2),
    SD = round(sd(minuta), 2),
    SE = round(sd(minuta) / sqrt(n()), 3)
  )
```

```
# A tibble: 1 x 4
      n     M    SD    SE
  <int> <dbl> <dbl> <dbl>
1    45  2.54  1.06 0.159
```

Prosjek uzorka je ispod 3 minute. Ali je li dovoljno daleko od 3 da možemo odbaciti tvrdnju medijske kuće? Možda je razlika samo slučajni šum.

10.4.1 Testna statistika

Testna statistika za jednouzorački t-test mjeri koliko je prosjek uzorka udaljen od pretpostavljene vrijednosti, izraženo u jedinicama standardne pogreške — formalno, to je:

$$t = \frac{\bar{x} - \mu_0}{SE} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Raspakujmo formulu. U brojniku je razlika između prosjeka uzorka ($\bar{x}$) i vrijednosti koju testiramo ($\mu_0 = 3$ minute). U nazivniku je standardna pogreška (SE), koja vam govori koliko prosjek uzorka tipično varira od uzorka do uzorka. Cijeli razlomak, dakle, kaže: "koliko

standardnih pogrešaka je moj prosjek udaljen od pretpostavljene vrijednosti?" Što je taj broj veći po apsolutnoj vrijednosti, to su podaci neobičniji pod nultom hipotezom.

```
x_bar <- mean(vrijeme_citanja$minuta)
mu_0 <- 3  # tvrdnja medijske kuće
s <- sd(vrijeme_citanja$minuta)
n <- nrow(vrijeme_citanja)
se <- s / sqrt(n)

t_stat <- (x_bar - mu_0) / se

cat("x̄ =", round(x_bar, 2), "\n")
```

```
x̄ = 2.54
```

```
cat("  =", mu_0, "\n")
```

```
  = 3
```

```
cat("SE =", round(se, 3), "\n")
```

```
SE = 0.159
```

```
cat("t =", round(t_stat, 3), "\n")
```

```
t = -2.911
```

```
cat("df =", n - 1, "\n")
```

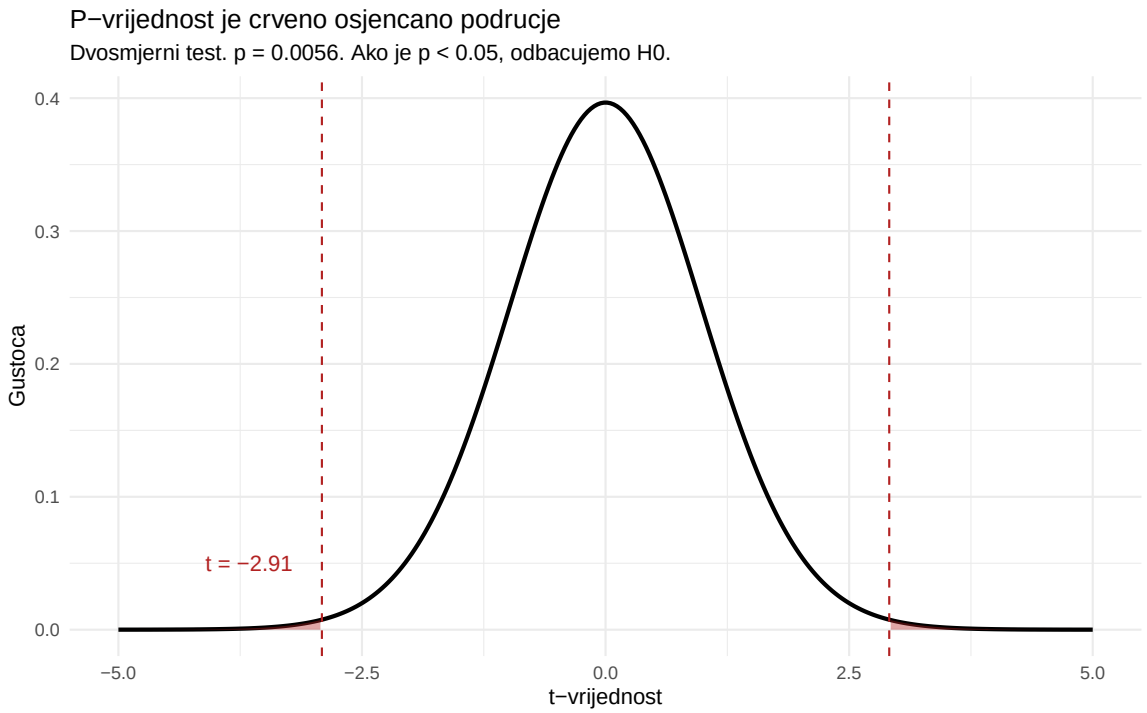
```
df = 44
```

Testna statistika t je negativna jer je prosjek uzorka manji od pretpostavljene vrijednosti. Apsolutna vrijednost |t| govori koliko standardnih pogrešaka je prosjek udaljen od . Što je ta udaljenost veća, to su jači dokazi protiv H .

10.4.2 P-vrijednost

Sada dolazi ključni korak. P-vrijednost je vjerojatnost da biste dobili testnu statistiku jednako ekstremnu ili ekstremniju od opažene, *pod pretpostavkom da je H istinita*. Ovo je suptilno ali ključno — p-vrijednost vam ne govori koliko je vjerojatno da je H istinita. Ona vam govori koliko bi vaši podaci bili neobični u svijetu gdje je H istinita.

```r
# Dvosmjerni test: gledamo obje strane
p_value <- 2 * pt(abs(t_stat), df = n - 1, lower.tail = FALSE)

x_vals <- seq(-5, 5, length.out = 300)

t_data <- tibble(x = x_vals, density = dt(x_vals, df = n - 1))

ggplot(t_data, aes(x = x, y = density)) +
  geom_line(linewidth = 1) +
  geom_area(data = t_data |> filter(x <= -abs(t_stat)),
            fill = "firebrick", alpha = 0.4) +
  geom_area(data = t_data |> filter(x >= abs(t_stat)),
            fill = "firebrick", alpha = 0.4) +
  geom_vline(xintercept = c(-abs(t_stat), abs(t_stat)),
             color = "firebrick", linetype = "dashed") +
  annotate("text", x = t_stat - 0.3, y = 0.05,
           label = paste("t =", round(t_stat, 2)), color = "firebrick", hjust = 1) +
  labs(
    title = "P-vrijednost je crveno osjenčano područje",
    subtitle = paste0("Dvosmjerni test. p = ", round(p_value, 4),
                      ". Ako je p < 0.05, odbacujemo H ."),
    x = "t-vrijednost",
    y = "Gustoća"
  ) +
  theme_minimal()
```




```
cat("t-statistika:", round(t_stat, 3), "\n")
```

```
t-statistika: -2.911
```

```
cat("P-vrijednost (dvosmjerni):", round(p_value, 4), "\n")
```

```
P-vrijednost (dvosmjerni): 0.0056
```

```
cat("  = 0.05\n")
```

```
  = 0.05
```

```
cat("Odluka:", if_else(p_value < 0.05, "Odbacujemo H ", "Ne možemo odbaciti H "), "\n")
```

```
Odluka: Odbacujemo H
```

10.4.3 `t.test()` obavlja sve za vas

U praksi ne trebate ručno računati t-statistiku i p-vrijednost. Funkcija `t.test()` sve radi u jednom pozivu.

```r
t.test(vrijeme_citanja$minuta, mu = 3)
```

```
    One Sample t-test

data:  vrijeme_citanja$minuta
t = -2.9114, df = 44, p-value = 0.005629
alternative hypothesis: true mean is not equal to 3
95 percent confidence interval:
 2.217817 2.857739
sample estimates:
mean of x
 2.537778
```

Funkcija vraća testnu statistiku, stupnjeve slobode, p-vrijednost, 95% interval pouzdanosti i prosjek uzorka. P-vrijednost je ispod 0.05, što znači da imamo dovoljno dokaza da odbacimo tvrdnju medijske kuće — prosječno vrijeme čitanja je statistički značajno različito od 3 minute.

> CI i testiranje hipoteza su dva lica istog novčića
>
> Pogledajte 95% interval pouzdanosti iz gornjeg rezultata. Ne sadrži vrijednost 3. To nije slučajnost — odbacivanje H na razini = 0.05 je matematički ekvivalentno tome da 95% CI ne sadrži testiranu vrijednost . Ovo su dva načina gledanja na isti problem — i obje perspektive su korisne.

10.5 Dvosmjerni i jednosmjerni test

U prethodnom primjeru koristili smo dvosmjerni test (two-tailed), što znači da smo testirali je li prosjek *različit* od 3, u bilo kojem smjeru. Hipoteze su bile H : = 3 nasuprot H : 3.

Ponekad unaprijed znate smjer. Ako Instagram tim očekuje da su carousel objave *bolje* (ne samo različite), može koristiti jednosmjerni test (one-tailed) s hipotezama H : _carousel _single nasuprot H : _carousel > _single.

```r
# Jednosmjerni: je li prosjek MANJI od 3?
t.test(vrijeme_citanja$minuta, mu = 3, alternative = "less")
```

```
    One Sample t-test
```

```
data:  vrijeme_citanja$minuta
t = -2.9114, df = 44, p-value = 0.002814
alternative hypothesis: true mean is less than 3
95 percent confidence interval:
     -Inf 2.804532
sample estimates:
mean of x
 2.537778
```

P-vrijednost jednosmjernog testa je točno pola dvosmjernog (kad je smjer u skladu s podacima). Jednosmjerni test je osjetljiviji u tom smjeru, ali potpuno slijep za razliku u suprotnom smjeru.

> ⚠ Smjer morate odrediti prije nego pogledate podatke
>
> Jednosmjerni test koristite samo ako ste smjer hipoteze odredili *prije* nego ste pogledali podatke. Ako pogledate podatke, vidite da je prosjek manji od 3, pa onda odlučite testirati "je li manji" — to je pristranost istraživača. Kad ste u sumnji, koristite dvosmjerni test. Velika većina objavljenih istraživanja koristi dvosmjerne testove upravo iz ovog razloga.

10.6 Dvouzorački t-test: natrag na Instagram

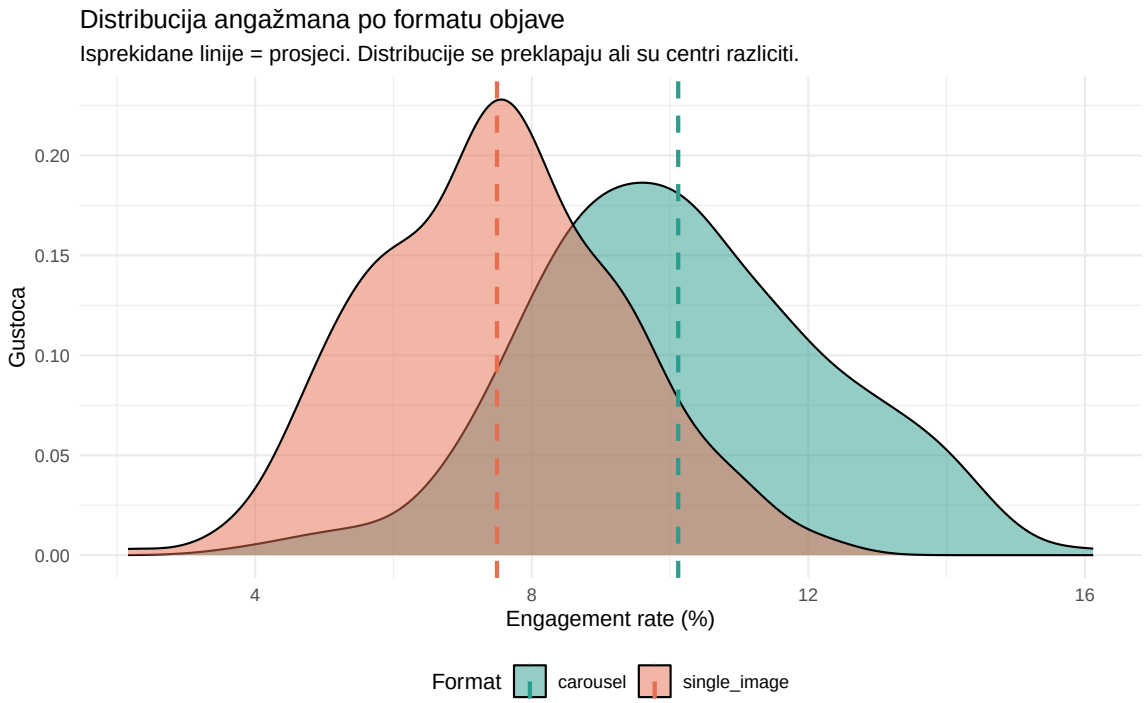
Sada se vraćamo na naš motivacijski primjer. Želimo testirati razlikuje li se angažman između carousel i single image objava. Budući da uspoređujemo prosjeke dviju nezavisnih skupina (carousel objave su jedne, single image su druge, i nema parenja), koristimo dvouzorački t-test.

$$H_0 : \mu_{carousel} = \mu_{single}$$
$$H_1 : \mu_{carousel} \neq \mu_{single}$$

Prije svega, pogledajmo distribucije.

```
ig |>
  ggplot(aes(x = engagement_rate * 100, fill = format)) +
  geom_density(alpha = 0.5) +
  geom_vline(data = ig |> group_by(format) |> summarise(M = mean(engagement_rate) * 100),
             aes(xintercept = M, color = format), linewidth = 1, linetype = "dashed") +
  scale_fill_manual(values = c("carousel" = "#2a9d8f", "single_image" = "#e76f51")) +
  scale_color_manual(values = c("carousel" = "#2a9d8f", "single_image" = "#e76f51")) +
  labs(
    title = "Distribucija angažmana po formatu objave",
    subtitle = "Isprekidane linije = prosjeci. Distribucije se preklapaju ali su centri ra
```

```
    x = "Engagement rate (%)",
    y = "Gustoća",
    fill = "Format", color = "Format"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Distribucije se preklapaju — postoje single image objave s visokim angažmanom i carousel objave s niskim — ali carousel distribucija je pomaknuta udesno. Testna statistika za dvouzorački t-test mjeri razliku prosjeka u jedinicama zajedničke standardne pogreške — ta statistika je:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{SE_{razlika}}$$

```
carousel <- ig |> filter(format == "carousel") |> pull(engagement_rate)
single <- ig |> filter(format == "single_image") |> pull(engagement_rate)

rezultat <- t.test(carousel, single)
rezultat
```

```
    Welch Two Sample t-test
```

```
data:  carousel and single
t = 14.942, df = 461.55, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.02276184 0.02965581
sample estimates:
 mean of x  mean of y
0.10117966 0.07497083
```

```
cat("Razlika prosjeka:", round((mean(carousel) - mean(single)) * 100, 2), "postotnih bodov
```

```
Razlika prosjeka: 2.62 postotnih bodova
```

```
cat("t-statistika:", round(rezultat$statistic, 2), "\n")
```

```
t-statistika: 14.94
```

```
cat("P-vrijednost:", format(rezultat$p.value, scientific = TRUE), "\n")
```

```
P-vrijednost: 1.856828e-41
```

```
cat("95% CI za razliku: [", round(rezultat$conf.int[1] * 100, 2), ",",
    round(rezultat$conf.int[2] * 100, 2), "] postotnih bodova\n")
```

```
95% CI za razliku: [ 2.28 , 2.97 ] postotnih bodova
```

P-vrijednost je iznimno mala — mnogo, mnogo manja od 0.05. Imamo snažne dokaze da se angažman zaista razlikuje između dva formata. Carousel objave imaju statistički značajno viši angažman.

10.6.1 Welchov t-test: default koji ne trebate mijenjati

R po defaultu koristi Welchov t-test, koji ne pretpostavlja jednake varijance u dvjema skupinama. Usporedimo ga s klasičnim Studentovim t-testom da vidite zašto je ovo mudar default.

```r
# Welchov (default)
welch <- t.test(carousel, single, var.equal = FALSE)

# Studentov (pretpostavlja jednake varijance)
student <- t.test(carousel, single, var.equal = TRUE)

tibble(
  test = c("Welch (default)", "Student (var.equal=TRUE)"),
  t = round(c(welch$statistic, student$statistic), 3),
  df = round(c(welch$parameter, student$parameter), 1),
  p = format(c(welch$p.value, student$p.value), scientific = TRUE, digits = 3)
)
```

```
# A tibble: 2 x 4
  test                          t    df p
  <chr>                     <dbl> <dbl> <chr>
1 Welch (default)            14.9  462. 1.86e-41
2 Student (var.equal=TRUE)   15.1  498  1.27e-42
```

Rezultati su slični, ali Welchov test ima nerunde stupnjeve slobode jer ih prilagođava za razliku u varijancama. Kad su varijance jednake, oba testa daju gotovo identične rezultate. Kad varijance nisu jednake, Welchov je točniji. Zaključak je jednostavan — koristite Welchov test uvijek, jer ne zahtijeva dodatnu pretpostavku i nikad nije lošiji.

10.7 Simulacija: što p-vrijednost zapravo znači

P-vrijednost je jedan od najčešće pogrešno shvaćenih koncepata u cijeloj statistici. Simulacija pomaže izgraditi ispravnu intuiciju na način na koji teorijsko objašnjenje ne može.

Zamislimo svijet u kojem je H istinita — carousel i single image imaju identičan angažman, nema nikakve razlike. Ako u tom svijetu mnogo puta uzorkujemo i testiramo, koliko ćemo često *slučajno* dobiti $p < 0.05$?

```r
set.seed(42)

# Simulacija: H  je ISTINITA (isti prosjek za obje grupe)
sim_p <- map_dbl(1:10000, \(i) {
  grupa_a <- rnorm(100, mean = 0.08, sd = 0.02)
  grupa_b <- rnorm(100, mean = 0.08, sd = 0.02)  # ISTI prosjek!
  t.test(grupa_a, grupa_b)$p.value
})

cat("H  je ISTINITA. Od 10 000 testova:\n")
```

H je ISTINITA. Od 10 000 testova:

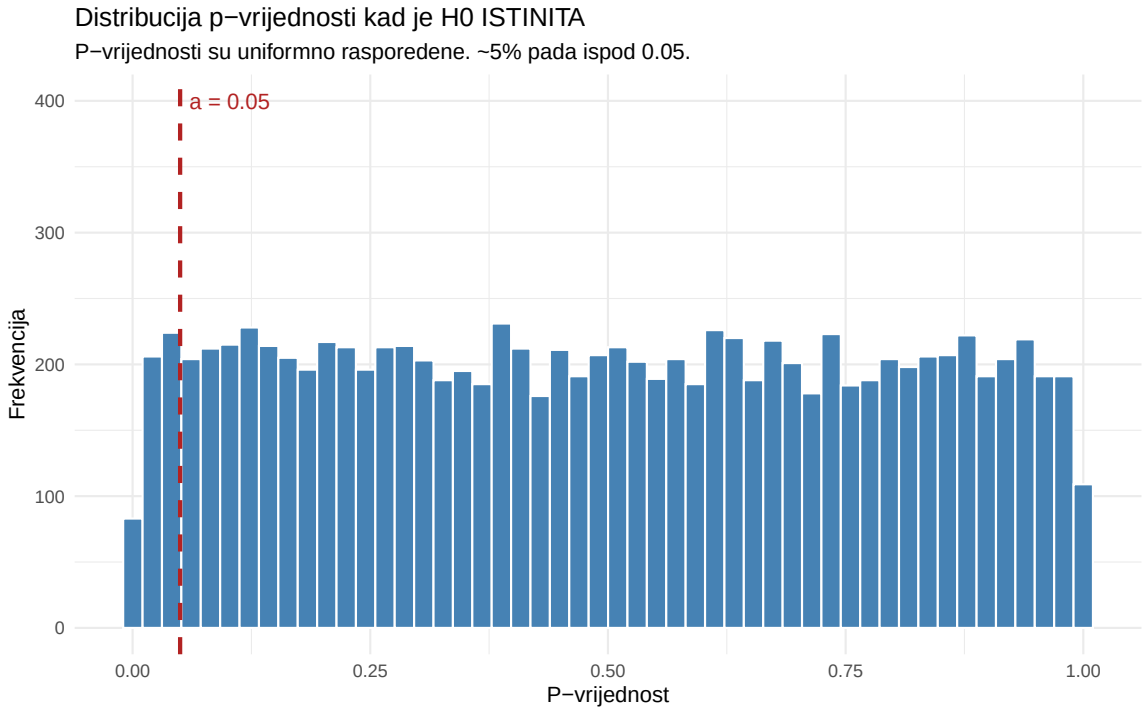
```
cat("p < 0.05:", sum(sim_p < 0.05), "(", round(mean(sim_p < 0.05) * 100, 1), "%)\n")
```

```
p < 0.05: 497 ( 5 %)
```

```
cat("p < 0.01:", sum(sim_p < 0.01), "(", round(mean(sim_p < 0.01) * 100, 1), "%)\n")
```

```
p < 0.01: 79 ( 0.8 %)
```

```
tibble(p = sim_p) |>
  ggplot(aes(x = p)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 50) +
  geom_vline(xintercept = 0.05, color = "firebrick", linewidth = 1, linetype = "dashed") +
  annotate("text", x = 0.06, y = 400, label = "  = 0.05", color = "firebrick", hjust = 0) +
  labs(
    title = "Distribucija p-vrijednosti kad je H  ISTINITA",
    subtitle = "P-vrijednosti su uniformno raspoređene. ~5% pada ispod 0.05.",
    x = "P-vrijednost",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Ovo je ključan uvid. Kad je H istinita, p-vrijednosti su uniformno raspoređene između 0 i 1. Točno 5% pada ispod 0.05 — po definiciji . To znači da ćemo u 5% slučajeva pogrešno odbaciti H *čak i kad je istinita.* Ovo je greška tipa I, lažno pozitivni rezultat, i potpuno je neizbježna posljedica toga da smo postavili prag na 5%.

10.7.1 Kad razlika zaista postoji

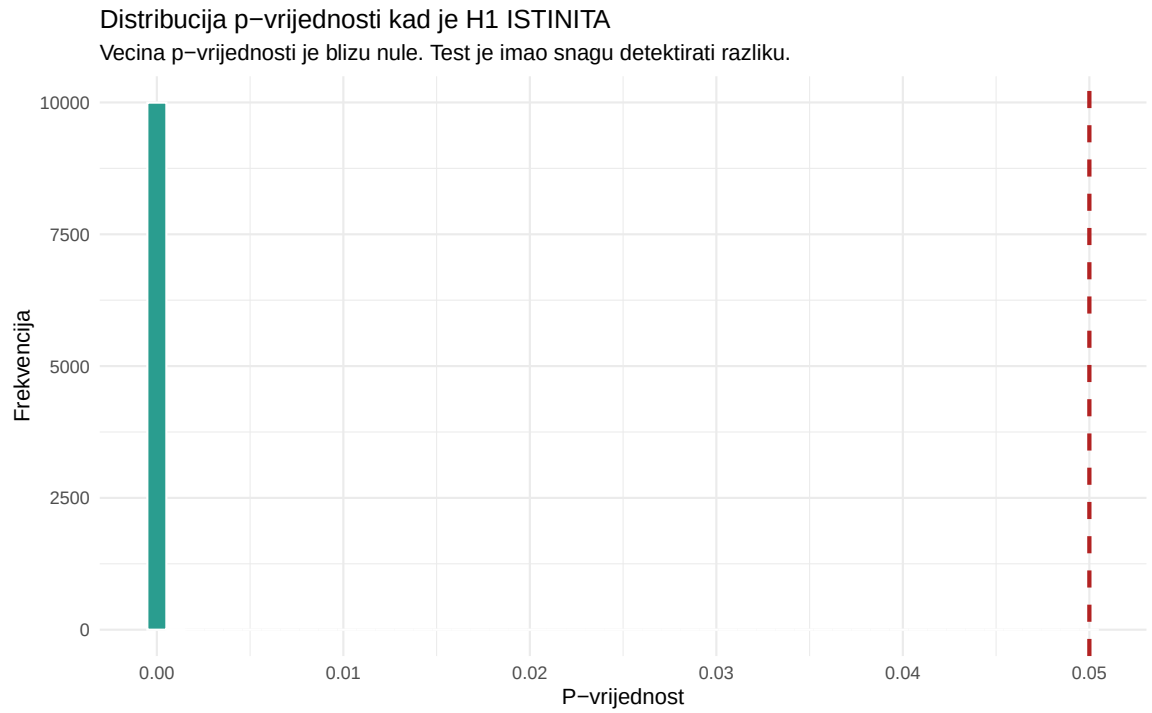
```
set.seed(42)

# Simulacija: H  je ISTINITA (razlika postoji)
sim_p_h1 <- map_dbl(1:10000, \(i) {
  grupa_a <- rnorm(100, mean = 0.10, sd = 0.02)
  grupa_b <- rnorm(100, mean = 0.08, sd = 0.02)  # RAZLIČIT prosjek
  t.test(grupa_a, grupa_b)$p.value
})

cat("H  je ISTINITA (razlika = 0.02). Od 10 000 testova:\n")
```

```
H  je ISTINITA (razlika = 0.02). Od 10 000 testova:
```

```
cat("p < 0.05:", sum(sim_p_h1 < 0.05), "(", round(mean(sim_p_h1 < 0.05) * 100, 1), "%)\n")
```

```
p < 0.05: 10000 ( 100 %)
```

```
tibble(p = sim_p_h1) |>
  ggplot(aes(x = p)) +
  geom_histogram(fill = "#2a9d8f", color = "white", bins = 50) +
  geom_vline(xintercept = 0.05, color = "firebrick", linewidth = 1, linetype = "dashed") +
  labs(
    title = "Distribucija p-vrijednosti kad je H  ISTINITA",
    subtitle = "Većina p-vrijednosti je blizu nule. Test je imao snagu detektirati razliku
    x = "P-vrijednost",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Slika je potpuno drugačija. Kad razlika zaista postoji, p-vrijednosti su koncentrirane blizu nule. Većina testova uspješno detektira razliku. Ali ne svi — neki testovi daju p 0.05 unatoč tome što razlika postoji. Postotak testova koji uspješno detektiraju pravu razliku zove se statistička snaga (power). Testovi koji je propuste čine grešku tipa II, lažno negativni rezultat.

10.8 Dvije vrste pogrešaka

Kad donosite odluku na temelju testa, možete pogriješiti na dva načina. Razumijevanje ovih dviju vrsta pogrešaka ključno je za mudru interpretaciju rezultata.

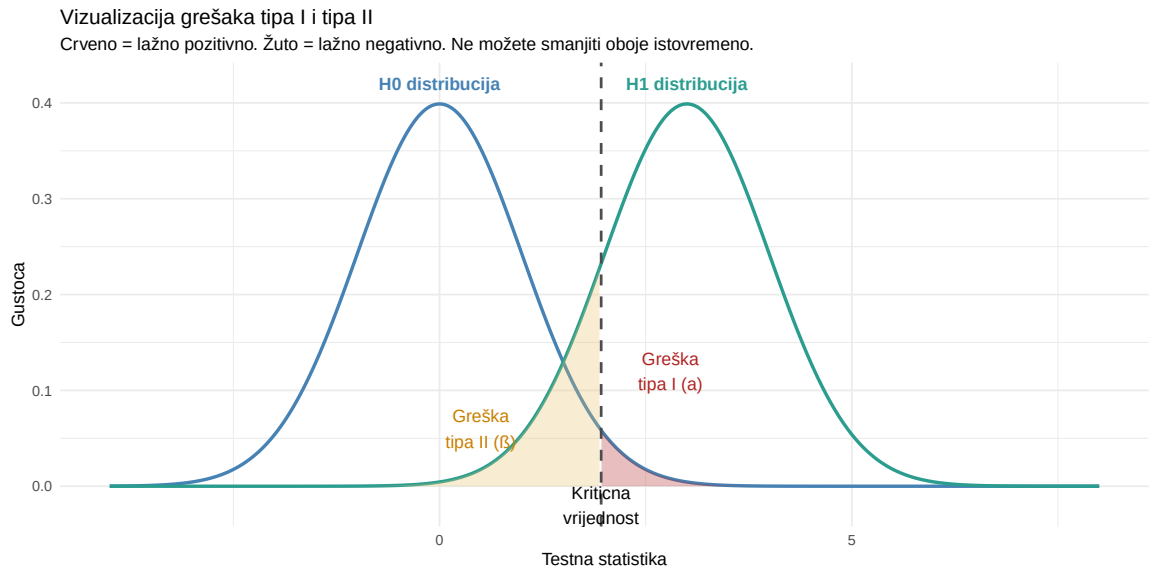
```
tribble(
  ~` `, ~`H  je istinita`, ~`H  je lažna`,
  "Ne odbacujemo H ", "  Ispravna odluka (1 -  )", "  Greška tipa II ( )",
  "Odbacujemo H ", "  Greška tipa I ( )", "  Ispravna odluka (snaga = 1 -  )"
)
```

```
# A tibble: 2 x 3
  ` `              `H  je istinita`           `H  je lažna`
  <chr>            <chr>                      <chr>
1 Ne odbacujemo H   Ispravna odluka (1 -  )   Greška tipa II ( )
2 Odbacujemo H      Greška tipa I ( )          Ispravna odluka (snaga = 1 -  )
```

Greška tipa I ( ) nastaje kad odbacite H iako je istinita — zaključite da razlika postoji kad je zapravo nema. Kontrolirate je postavljanjem (obično 0.05). U analogiji sa suđenjem, to je osuda nevine osobe.

Greška tipa II ( ) nastaje kad ne odbacite H iako je lažna — propustite pravu razliku. Ovisi o veličini uzorka, veličini učinka i razini . U analogiji sa suđenjem, to je oslobađanje krivca.

```
x <- seq(-4, 8, length.out = 500)
h0 <- dnorm(x, mean = 0, sd = 1)
h1 <- dnorm(x, mean = 3, sd = 1)

crit <- qnorm(0.975)

error_data <- tibble(x = x, H0 = h0, H1 = h1)

ggplot(error_data, aes(x = x)) +
  # H0 distribucija
  geom_line(aes(y = H0), color = "steelblue", linewidth = 1) +
  geom_area(data = error_data |> filter(x >= crit), aes(y = H0),
            fill = "firebrick", alpha = 0.3) +
  # H1 distribucija
  geom_line(aes(y = H1), color = "#2a9d8f", linewidth = 1) +
  geom_area(data = error_data |> filter(x < crit), aes(y = H1),
            fill = "#e9c46a", alpha = 0.3) +
  # Kritična vrijednost
  geom_vline(xintercept = crit, color = "grey30", linewidth = 0.8, linetype = "dashed") +
  annotate("text", x = 0, y = 0.42, label = "H  distribucija", color = "steelblue", fontfa
  annotate("text", x = 3, y = 0.42, label = "H  distribucija", color = "#2a9d8f", fontface
  annotate("text", x = 2.8, y = 0.12, label = "Greška\ntipa I ( )", color = "firebrick") +
  annotate("text", x = 0.5, y = 0.06, label = "Greška\ntipa II ( )", color = "#c77f00") +
  annotate("text", x = crit, y = -0.02, label = "Kritična\nvrijednost", hjust = 0.5) +
  labs(
    title = "Vizualizacija grešaka tipa I i tipa II",
    subtitle = "Crveno = lažno pozitivno. Žuto = lažno negativno. Ne možete smanjiti oboje
    x = "Testna statistika",
    y = "Gustoća"
  ) +
  theme_minimal()
```



Ovaj graf pokazuje ključan kompromis. Ako pomaknete kritičnu vrijednost udesno (stroži), smanjujete crveno područje (manje lažno pozitivnih) ali povećavate žuto (više lažno negativnih). Jedini način da smanjite oboje istovremeno je povećati uzorak (što razdvaja dvije distribucije) ili imati veći učinak.

> **!** "Ne možemo odbaciti" nije isto što i "prihvaćamo"
>
> Odsutnost dokaza nije dokaz odsutnosti. Kad test daje p 0.05, ne kažemo "prihvaćamo H " — kažemo "ne možemo odbaciti H na temelju dostupnih podataka." Možda razlika postoji, ali naš uzorak je premalen da je detektira. Možda razlika postoji, ali je toliko mala da nije vidljiva s ovom količinom podataka. Zato nikad, nikad ne zaključujte "dokazali smo da nema razlike."

10.9 P-vrijednost: raščistimo zablude

P-vrijednost je jedan od najčešće korištenih ali i najčešće pogrešno interpretiranih koncepata u cijeloj statistici. Potrebno je nekoliko minuta da precizno razjasnimo što ona jest, a što nije.

P-vrijednost *jest* vjerojatnost dobivanja testne statistike jednako ekstremne ili ekstremnije od opažene, pod pretpostavkom da je H istinita. Koliko biste bili iznenađeni ovakvim podacima da H zaista vrijedi?

P-vrijednost *nije* vjerojatnost da je H istinita. Ne možete reći "postoji samo 3% šanse da nema razlike." P-vrijednost govori o podacima s obzirom na hipotezu, ne o hipotezi s obzirom na podatke. Ova razlika može djelovati kao cjepidlačenje, ali je zapravo fundamentalna.

P-vrijednost *nije* vjerojatnost da ste pogriješili. Mala p-vrijednost znači da su podaci neobični pod H . Ne znači da ste sigurno u pravu.

I ono najvažnije — p-vrijednost *nije* mjera veličine učinka. Vrijednost p = 0.001 ne znači da je razlika velika. Velik uzorak može proizvesti sićušnu p-vrijednost za trivijalno malu razliku. Pogledajmo to na primjeru.

```
set.seed(42)

# Mali uzorak, velik učinak
mali_uzorak <- t.test(rnorm(20, 10.5, 2), mu = 10)

# Velik uzorak, sićušan učinak
velik_uzorak <- t.test(rnorm(10000, 10.02, 2), mu = 10)

tibble(
  scenarij = c("Mali uzorak (n=20), velik učinak", "Velik uzorak (n=10000), sićušan učinak
  n = c(20, 10000),
  razlika = c("0.5 bodova", "0.02 boda"),
  p_vrijednost = c(round(mali_uzorak$p.value, 4), round(velik_uzorak$p.value, 4)),
  znacajno = c(mali_uzorak$p.value < 0.05, velik_uzorak$p.value < 0.05)
)
```

```
# A tibble: 2 x 5
  scenarij                                    n razlika    p_vrijednost znacajno
  <chr>                                   <dbl> <chr>              <dbl> <lgl>
1 Mali uzorak (n=20), velik učinak           20 0.5 bodova         0.149 FALSE
2 Velik uzorak (n=10000), sićušan učinak 10000 0.02 boda           0.843 FALSE
```

S 10 000 opažanja, razlika od 0.02 boda — praktički beznačajna — može biti statistički značajna. S 20 opažanja, razlika od 0.5 bodova — potencijalno važna — možda neće biti statistički značajna. Ovo jasno pokazuje zašto p-vrijednost sama nije dovoljna za donošenje odluka. Uvijek trebate i mjeru veličine učinka.

> **Gdje smo, kamo idemo**
>
> U prvom dijelu naučili smo logiku testiranja hipoteza, formuliranje H i H , jednouzorački i dvouzorački t-test, p-vrijednost i greške tipa I i II. U nastavku prelazimo na pitanje koje je jednako važno kao statistička značajnost — koliko je učinak zapravo velik?

10.10 Veličina učinka: Cohenov d

P-vrijednost odgovara na pitanje "postoji li učinak?" ali šuti o tome koliko je taj učinak velik. Za to vam treba mjera veličine učinka. Najčešća za razliku dvaju prosjeka je Cohenov d, koji izražava razliku u jedinicama zajedničke standardne devijacije — formalno:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}}$$

Zašto dijeliti sa standardnom devijacijom? Zato što vam razlika od 2.6 postotnih bodova ne znači ništa dok ne znate koliko pojedinačne objave variraju. Ako sve objave imaju engagement rate između 7% i 11%, razlika od 2.6 bodova je ogromna. Ako variraju od 0% do 50%, ista razlika je zanemariva. Cohenov d stavlja razliku u kontekst varijabilnosti.

```
ig <- read_csv("../resources/datasets/instagram_ab_test.csv")

carousel <- ig |> filter(format == "carousel") |> pull(engagement_rate)
single <- ig |> filter(format == "single_image") |> pull(engagement_rate)

# Ručni izračun
n1 <- length(carousel)
n2 <- length(single)
s_pooled <- sqrt(((n1 - 1) * sd(carousel)^2 + (n2 - 1) * sd(single)^2) / (n1 + n2 - 2))
d <- (mean(carousel) - mean(single)) / s_pooled

cat("Razlika prosjeka:", round((mean(carousel) - mean(single)) * 100, 2), "postotnih bodov
```

```
Razlika prosjeka: 2.62 postotnih bodova
```

```
cat("Pooled SD:", round(s_pooled * 100, 2), "postotnih bodova\n")
```

```
Pooled SD: 1.94 postotnih bodova
```

```
cat("Cohenov d:", round(d, 3), "\n")
```

```
Cohenov d: 1.351
```

10.10.1 Što znači mali, srednji i veliki učinak

Cohen (1988) je predložio smjernice za interpretaciju koje su postale konvencija u društvenim znanostima. One izgledaju ovako:

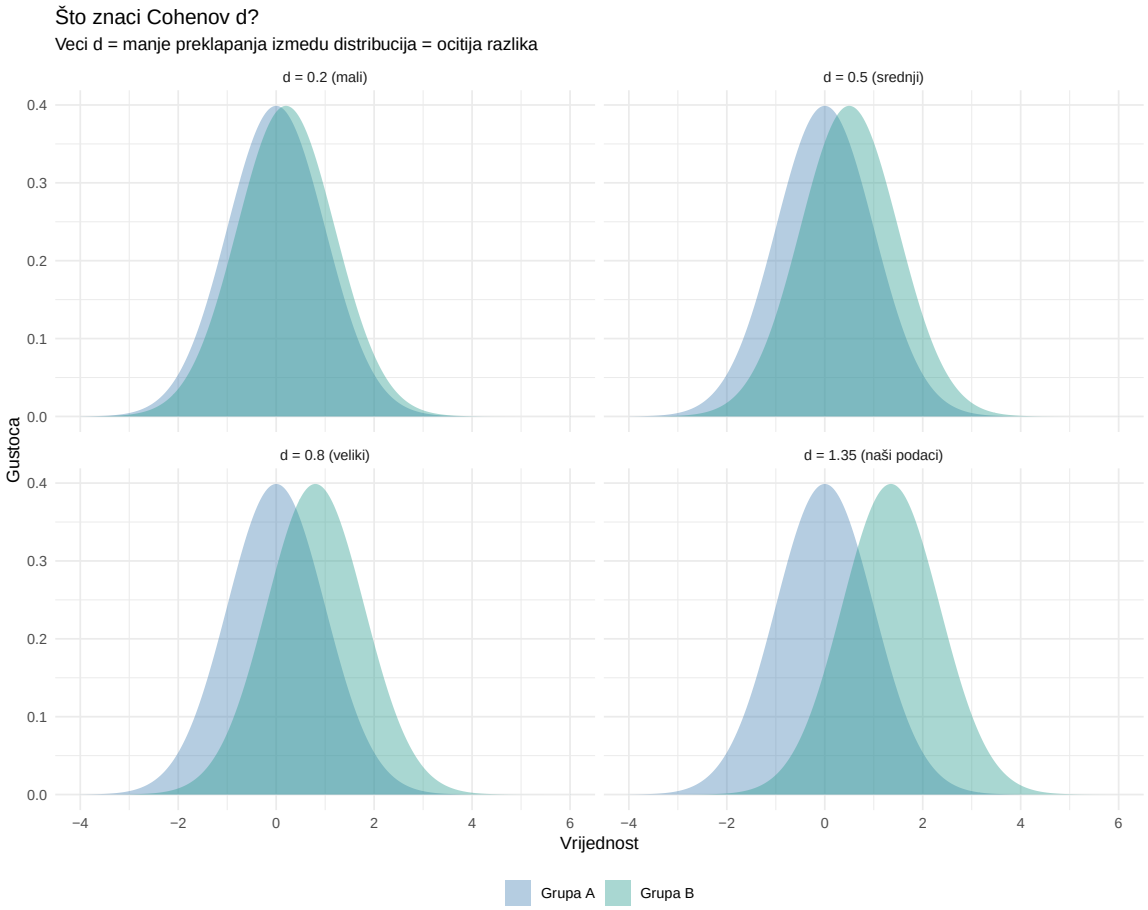
```
tribble(
  ~d, ~interpretacija, ~primjer,
  "0.2", "Mali učinak", "Jedva primjetna razlika u praksi",
  "0.5", "Srednji učinak", "Razlika vidljiva prostim okom",
  "0.8", "Veliki učinak", "Razlika očita i praktično važna"
)
```

```
# A tibble: 3 x 3
  d     interpretacija primjer
  <chr> <chr>          <chr>
1 0.2   Mali učinak    Jedva primjetna razlika u praksi
2 0.5   Srednji učinak Razlika vidljiva prostim okom
3 0.8   Veliki učinak  Razlika očita i praktično važna
```

Naš d je veliki učinak. Carousel objave generiraju značajno viši angažman, i to u praktično važnoj mjeri — ovo je informacija koju p-vrijednost sama ne može dati.

Vizualizirajmo što različite veličine učinka *izgledaju* kao preklapanje dviju distribucija.

```
# Vizualizacija: što znači d = 0.2, 0.5, 0.8, 1.3
d_values <- c(0.2, 0.5, 0.8, round(d, 2))
d_labels <- c("d = 0.2 (mali)", "d = 0.5 (srednji)", "d = 0.8 (veliki)",
              paste0("d = ", round(d, 2), " (naši podaci)"))

x <- seq(-4, 6, length.out = 300)

d_viz <- map_df(seq_along(d_values), \(i) {
  tibble(
    panel = d_labels[i],
    x = x,
    Grupa_A = dnorm(x, 0, 1),
    Grupa_B = dnorm(x, d_values[i], 1)
  ) |>
    pivot_longer(c(Grupa_A, Grupa_B), names_to = "grupa", values_to = "gustoca")
}) |>
  mutate(panel = factor(panel, levels = d_labels))

d_viz |>
  ggplot(aes(x = x, y = gustoca, fill = grupa)) +
  geom_area(alpha = 0.4, position = "identity") +
  facet_wrap(~panel, ncol = 2) +
  scale_fill_manual(values = c("Grupa_A" = "steelblue", "Grupa_B" = "#2a9d8f"),
                    labels = c("Grupa A", "Grupa B")) +
  labs(
    title = "Što znači Cohenov d?",
    subtitle = "Veći d = manje preklapanja između distribucija = očitija razlika",
    x = "Vrijednost", y = "Gustoća", fill = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



S d = 0.2, distribucije se gotovo potpuno preklapaju — razliku biste teško primijetili u praksi. S d = 0.8, razdvajanje je očito. Naš d oko 1.3 pokazuje vrlo jasno razdvajanje — carousel i single image su očigledno različite kategorije po angažmanu.

> **Uvijek izvještavajte veličinu učinka**
>
> Umjesto "razlika je statistički značajna ($p < 0.001$)", napišite: "carousel objave imaju značajno viši angažman od single image objava (razlika = 2.6 postotnih bodova, d = 1.34, $p < 0.001$)." Ovo daje čitatelju informaciju i o postojanju i o veličini razlike — sve u jednoj rečenici.

10.11 Statistička snaga: hoće li vaš test uopće nešto naći?

Statistička snaga (power) je vjerojatnost da test odbaci H kad je H istinita — jednostavnije rečeno, vjerojatnost da ćete detektirati pravu razliku ako ona postoji.

Snaga ovisi o četiri faktora. Veličina učinka — veću razliku je lakše detektirati. Veličina uzorka — više podataka daje veću snagu. Razina značajnosti — veći daje veću snagu, ali

i više lažno pozitivnih. Varijabilnost podataka — manja varijabilnost znači čišći signal. To su faktori koji snagu određuju.

Konvencija kaže da snaga treba biti barem 0.80 (80%). To znači da ako razlika postoji, želite je detektirati barem u 8 od 10 pokušaja.

10.11.1 Koliki uzorak trebam?

Najčešća primjena analize snage je planiranje istraživanja *prije* nego prikupite podatke. Ključno pitanje je — koliki uzorak trebate da biste detektirali očekivanu veličinu učinka s 80% snagom?

```
# power.t.test() za dvouzorački test
# Koliki uzorak trebam za srednji učinak (d = 0.5)?
power.t.test(
  delta = 0.5,       # očekivana razlika u SD jedinicama (Cohenov d)
  sd = 1,            # standardizirano na 1
  sig.level = 0.05,  #
  power = 0.80,      # željena snaga
  type = "two.sample",
  alternative = "two.sided"
)
```

```
     Two-sample t test power calculation

              n = 63.76576
          delta = 0.5
             sd = 1
      sig.level = 0.05
          power = 0.8
    alternative = two.sided

NOTE: n is number in *each* group
```

Za detektiranje srednjeg učinka (d = 0.5) s 80% snagom potrebno je otprilike 64 ispitanika po grupi, ukupno 128. Pogledajmo kako se potreban uzorak mijenja s veličinom učinka.

```
d_values <- c(0.2, 0.3, 0.5, 0.8, 1.0, 1.3)

power_tablica <- map_df(d_values, \(d_val) {
  rez <- power.t.test(delta = d_val, sd = 1, sig.level = 0.05, power = 0.80,
                      type = "two.sample", alternative = "two.sided")
  tibble(
```

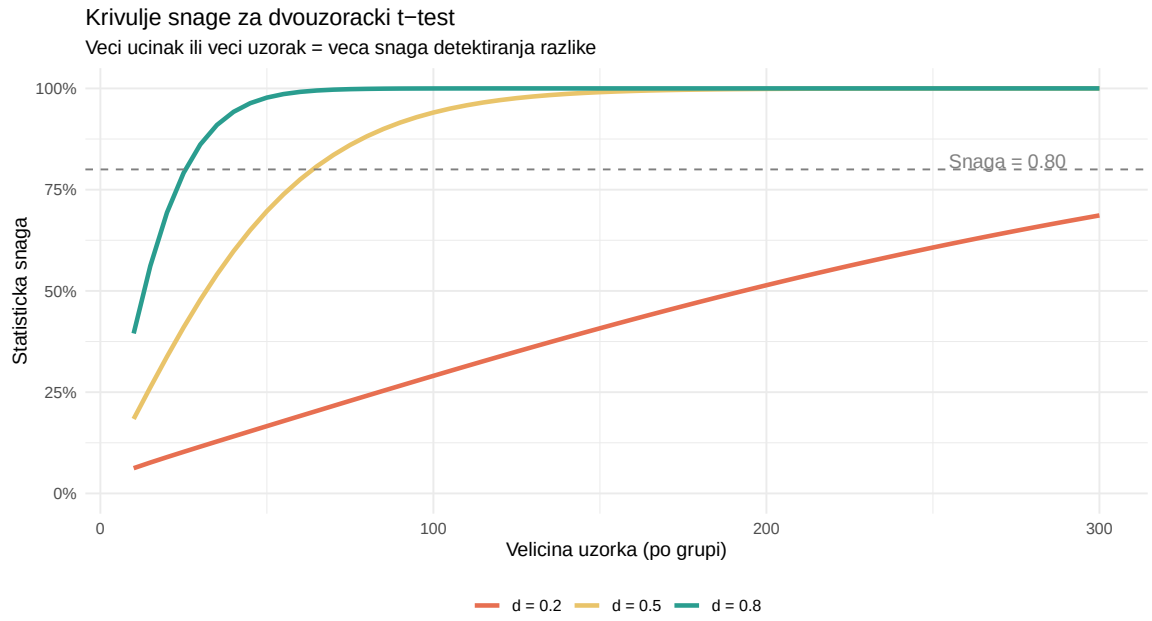
```
    cohenov_d = d_val,
    n_po_grupi = ceiling(rez$n),
    ukupno_n = ceiling(rez$n) * 2
  )
})

power_tablica
```

```
# A tibble: 6 x 3
  cohenov_d n_po_grupi ukupno_n
      <dbl>      <dbl>    <dbl>
1       0.2        394      788
2       0.3        176      352
3       0.5         64      128
4       0.8         26       52
5       1           17       34
6       1.3         11       22
```

Brojke su poučne. Za mali učinak (d = 0.2) trebate skoro 400 ispitanika po grupi — ukupno 800. Za veliki učinak (d = 0.8) trebate samo 26 po grupi. Ovo je razlog zašto je planiranje unaprijed ključno — morate imati realistična očekivanja o veličini učinka da biste znali koliko podataka trebate prikupiti.

```
# Krivulja snage: kako snaga raste s veličinom uzorka
n_range <- seq(10, 300, by = 5)

power_curves <- map_df(c(0.2, 0.5, 0.8), \(d_val) {
  map_df(n_range, \(n_val) {
    p <- power.t.test(n = n_val, delta = d_val, sd = 1, sig.level = 0.05,
                      type = "two.sample", alternative = "two.sided")$power
    tibble(n = n_val, power = p, d = paste("d =", d_val))
  })
})

power_curves |>
  ggplot(aes(x = n, y = power, color = d)) +
  geom_line(linewidth = 1.2) +
  geom_hline(yintercept = 0.80, linetype = "dashed", color = "grey50") +
  annotate("text", x = 290, y = 0.82, label = "Snaga = 0.80", color = "grey50", hjust = 1)
  scale_y_continuous(labels = scales::label_percent(), limits = c(0, 1)) +
  scale_color_manual(values = c("d = 0.2" = "#e76f51", "d = 0.5" = "#e9c46a", "d = 0.8" =
  labs(
    title = "Krivulje snage za dvouzorački t-test",
    subtitle = "Veći učinak ili veći uzorak = veća snaga detektiranja razlike",
```

```
    x = "Veličina uzorka (po grupi)",
    y = "Statistička snaga",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Za mali učinak (d = 0.2, crvena), snaga sporo raste i ni s 300 ispitanika po grupi ne dostiže 100%. Za veliki učinak (d = 0.8, zelena), snaga brzo raste i s 30 po grupi je već blizu 80%.

10.11.2 Kolika je snaga našeg Instagram testa?

```
# Kolika je snaga našeg testa s d  1.34 i n  250 po grupi?
power_ig <- power.t.test(
  n = min(n1, n2),
  delta = d,
  sd = 1,
  sig.level = 0.05,
  type = "two.sample",
  alternative = "two.sided"
)

cat("Snaga našeg testa:", round(power_ig$power, 4), "\n")
```

```
Snaga našeg testa: 1
```

Snaga je gotovo 100%. S ovakvom veličinom učinka i ovakvim uzorkom, gotovo je nemoguće da bismo propustili ovu razliku. Test je bio više nego adekvatno snažan — u praksi, mogli smo detektirati ovu razliku s mnogo manje podataka.

```
# Koliki minimalni uzorak bi bio dovoljan?
min_n <- power.t.test(
  delta = d,
  sd = 1,
  sig.level = 0.05,
  power = 0.80,
  type = "two.sample"
)

cat("Minimalni n po grupi za 80% snagu:", ceiling(min_n$n), "\n")
```

```
Minimalni n po grupi za 80% snagu: 10
```

```
cat("Mi smo imali:", min(n1, n2), "po grupi\n")
```

```
Mi smo imali: 236 po grupi
```

10.12 Upareni t-test: kad iste jedinice mjerite dva puta

Dosad smo uspoređivali dvije nezavisne skupina — carousel objave su jedne, single image su druge, i nema nikakve veze između pojedinačnih objava u dvjema grupama. Ali ponekad mjerite istu jedinicu u dva uvjeta. Na primjer — angažman istih pratitelja prije i poslije redizajna profila, ili ocjene istih članaka od strane dva različita urednika.

Kad su opažanja u parovima, koristite upareni t-test. Umjesto da uspoređujete dva prosjeka, on računa razliku za svaki par i testira je li prosjek tih razlika različit od nule. Ovo je daleko osjetljiviji pristup jer uklanja varijabilnost *između* parova i fokusira se samo na varijabilnost *unutar* parova.

```
set.seed(42)

# Simulacija: 30 članaka, svaki ocjenjen od 2 urednika
urednicki_rating <- tibble(
  clanak_id = 1:30,
  urednik_A = round(rnorm(30, mean = 6.5, sd = 1.2), 1),
  urednik_B = round(urednik_A + rnorm(30, mean = 0.5, sd = 0.8), 1)
)

# Urednik B ocjenjuje sustavno više
```

```r
urednicki_rating <- urednicki_rating |>
  mutate(razlika = urednik_B - urednik_A)

urednicki_rating |>
  summarise(
    M_A = round(mean(urednik_A), 2),
    M_B = round(mean(urednik_B), 2),
    M_razlika = round(mean(razlika), 2),
    SD_razlika = round(sd(razlika), 2)
  )
```

```
# A tibble: 1 x 4
    M_A   M_B M_razlika SD_razlika
  <dbl> <dbl>     <dbl>      <dbl>
1  6.59  6.99      0.41       0.84
```

```r
# Upareni t-test
t.test(urednicki_rating$urednik_B, urednicki_rating$urednik_A, paired = TRUE)
```

```
    Paired t-test

data:  urednicki_rating$urednik_B and urednicki_rating$urednik_A
t = 2.648, df = 29, p-value = 0.01296
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.09257456 0.72075877
sample estimates:
mean difference
      0.4066667
```

Usporedimo što se dogodi kad na istim podacima pokrenemo upareni i neupareni test.

```r
# Usporedba: upareni vs nepareni test na istim podacima
paired_p <- t.test(urednicki_rating$urednik_B, urednicki_rating$urednik_A, paired = TRUE)$
unpaired_p <- t.test(urednicki_rating$urednik_B, urednicki_rating$urednik_A, paired = FALS

cat("Upareni test p-vrijednost:  ", round(paired_p, 5), "\n")
```

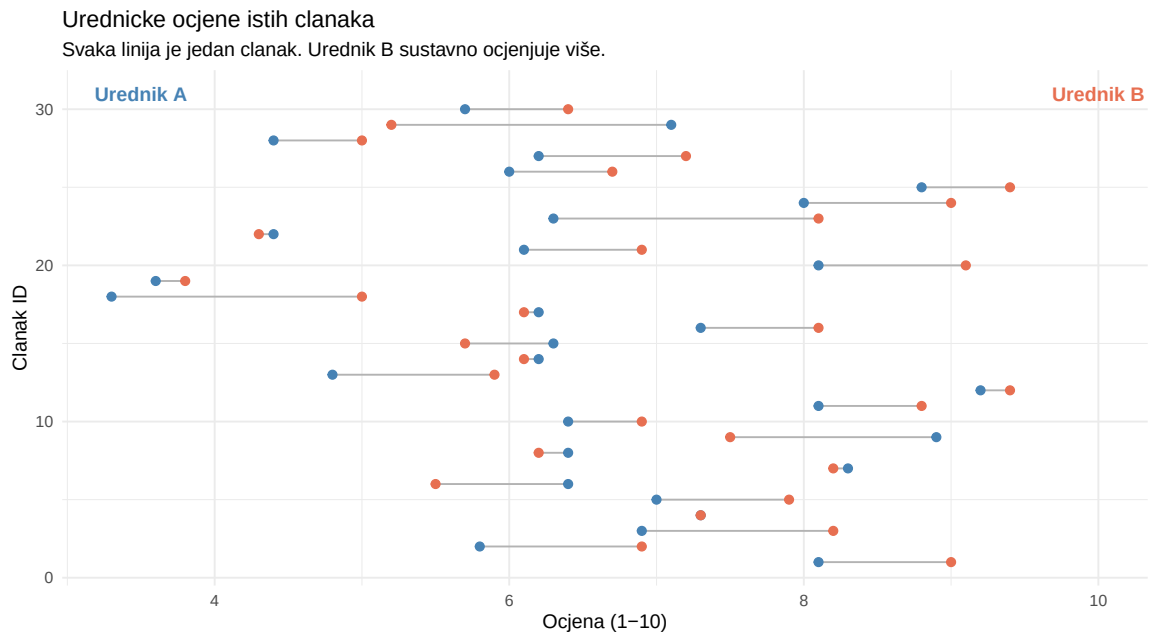
```
Upareni test p-vrijednost:   0.01296
```

```r
cat("Nepareni test p-vrijednost: ", round(unpaired_p, 5), "\n")
```

```
Nepareni test p-vrijednost:  0.30785
```

Upareni test daje manju p-vrijednost jer je osjetljiviji. Zašto? Zato što uklanja varijabilnost između parova. Neki članci su općenito bolji, neki lošiji — to je varijabilnost koja nema veze s razlikom između urednika. Kad tu varijabilnost kontrolirate (parenjem), preostaje samo varijabilnost u razlici između dva urednika, koja je mnogo manja.

```r
urednicki_rating |>
  ggplot() +
  geom_segment(aes(x = urednik_A, xend = urednik_B, y = clanak_id, yend = clanak_id),
               color = "grey70", linewidth = 0.5) +
  geom_point(aes(x = urednik_A, y = clanak_id), color = "steelblue", size = 2) +
  geom_point(aes(x = urednik_B, y = clanak_id), color = "#e76f51", size = 2) +
  annotate("text", x = 3.5, y = 31, label = "Urednik A", color = "steelblue", fontface = "
  annotate("text", x = 10, y = 31, label = "Urednik B", color = "#e76f51", fontface = "bol
  labs(
    title = "Uredničke ocjene istih članaka",
    subtitle = "Svaka linija je jedan članak. Urednik B sustavno ocjenjuje više.",
    x = "Ocjena (1-10)",
    y = "Članak ID"
  ) +
  theme_minimal()
```



Svaka vodoravna linija predstavlja jedan članak. Plava točka je ocjena urednika A, crvena urednika B. Većina linija ide udesno, što znači da urednik B dosljedno ocjenjuje više.

> ❗ Koji test za koju situaciju?
>
> **Nezavisni (neupareni) t-test** — dvije različite skupina bez veze. Primjeri: muškarci vs žene, kontrolna vs eksperimentalna grupa, carousel vs single image.
> **Upareni t-test** — ista jedinica mjerena dva puta. Primjeri: prije i poslije intervencije, isti sadržaj na dva kanala, isti ispitanik u dva uvjeta. Ključno pitanje: možete li smisleno spariti opažanja? Ako da, koristite upareni test — bit će osjetljiviji.

10.13 Statistička značajnost nije isto što i praktična važnost

Ovo je možda najvažnija lekcija cijelog predavanja. Statistička značajnost ($p < 0.05$) govori da razlika vjerojatno nije slučajnost. Ali ne govori vam je li ta razlika dovoljno velika da na nju trebate reagirati. Ovo razlikovanje je ključno za svakoga tko donosi poslovne ili istraživačke odluke na temelju podataka.

```
set.seed(42)

# Scenarij 1: Statistički značajno ali praktički beznačajno
# Novi dizajn naslovnice povećava CTR s 2.00% na 2.05%
n_velik <- 50000
ctr_stari <- rbinom(n_velik, 1, 0.0200)
ctr_novi <- rbinom(n_velik, 1, 0.0205)

test1 <- t.test(ctr_novi, ctr_stari)

cat("=== Scenarij 1: Velik uzorak, sićušna razlika ===\n")
```

```
=== Scenarij 1: Velik uzorak, sićušna razlika ===
```

```
cat("Razlika CTR:", round((mean(ctr_novi) - mean(ctr_stari)) * 100, 3), "postotnih bodova\
```

```
Razlika CTR: -0.014 postotnih bodova
```

```
cat("P-vrijednost:", round(test1$p.value, 4), "\n")
```

```
P-vrijednost: 0.8762
```

```r
cat("Statistički značajno:", test1$p.value < 0.05, "\n")
```

```
Statistički značajno: FALSE
```

```r
cat("Isplati li se redizajn? Vjerojatno ne.\n\n")
```

```
Isplati li se redizajn? Vjerojatno ne.
```

```r
# Scenarij 2: Statistički neznačajno ali potencijalno praktički važno
# Novi format povećava CTR s 2.0% na 3.5% ali mali uzorak
n_mali <- 80
ctr_stari2 <- rbinom(n_mali, 1, 0.020)
ctr_novi2 <- rbinom(n_mali, 1, 0.035)

test2 <- t.test(ctr_novi2, ctr_stari2)

cat("=== Scenarij 2: Mali uzorak, veća razlika ===\n")
```

```
=== Scenarij 2: Mali uzorak, veća razlika ===
```

```r
cat("Razlika CTR:", round((mean(ctr_novi2) - mean(ctr_stari2)) * 100, 2), "postotnih bodov
```

```
Razlika CTR: 0 postotnih bodova
```

```r
cat("P-vrijednost:", round(test2$p.value, 4), "\n")
```

```
P-vrijednost: 1
```

```r
cat("Statistički značajno:", test2$p.value < 0.05, "\n")
```

```
Statistički značajno: FALSE
```

```r
cat("Zaslužuje li daljnje istraživanje? Vjerojatno da.\n")
```

```
Zaslužuje li daljnje istraživanje? Vjerojatno da.
```

Ova dva scenarija savršeno ilustriraju zašto p-vrijednost sama nije dovoljna. U prvom, razlika od 0.05 postotnih bodova je statistički značajna (jer imate 50 000 opažanja), ali praktički beznačajna — redizajn koji donosi toliko poboljšanje se ne isplati. U drugom, razlika od 1.5 postotnih bodova nije statistički značajna (jer imate samo 80 opažanja), ali je potencijalno vrlo važna — i zaslužuje daljnje istraživanje s većim uzorkom.

Donošenje odluka zahtijeva da razmotrite veličinu učinka, praktične posljedice, interval pouzdanosti i kontekst. Sljedeća tablica sažima četiri moguća scenarija.

```
tribble(
  ~` `, ~`Praktički važno`, ~`Praktički nevažno`,
  "Statistički značajno (p < 0.05)", "  Djeluj! Razlika postoji i važna je.", "  Razlika po
  "Statistički neznačajno (p   0.05)", "  Možda nemaš dovoljno podataka. Povećaj uzorak.",
)
```

```
# A tibble: 2 x 3
  ` `                                `Praktički važno`        `Praktički nevažno`
  <chr>                              <chr>                    <chr>
1 Statistički značajno (p < 0.05)    "  Djeluj! Razlika post~  Razlika postoji ~
2 Statistički neznačajno (p   0.05) "\U0001f50d Možda nemaš~  Nema učinka i to~
```

10.14 Sve zajedno: izvještaj za urednicu

Spojimo sve u koherentan izvještaj. Slijedimo strukturu koju ćete koristiti u svakoj budućoj analizi, gdje su ključni koraci sljedeći — opisna statistika, vizualizacija, statistički test, veličina učinka, podanalize po podgrupama, zaključak s preporukom.

```
# Korak 1: Opisna statistika po formatu
ig |>
  group_by(format) |>
  summarise(
    n = n(),
    M_engagement = round(mean(engagement_rate) * 100, 2),
    SD_engagement = round(sd(engagement_rate) * 100, 2),
    M_likes = round(mean(likes), 0),
    M_comments = round(mean(comments), 0),
    M_shares = round(mean(shares), 0),
    M_saves = round(mean(saves), 0),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 8
  format       n M_engagement SD_engagement M_likes M_comments M_shares M_saves
  <chr>    <int>        <dbl>         <dbl>   <dbl>      <dbl>    <dbl>   <dbl>
```

```
1 carousel     236          10.1          2.11      180          36        25       42
2 single_i~    264           7.5          1.77      149          24        20       30
```

```
ig |>
  ggplot(aes(x = format, y = engagement_rate * 100, fill = format)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.15, size = 1) +
  scale_fill_manual(values = c("carousel" = "#2a9d8f", "single_image" = "#e76f51")) +
  labs(
    title = "Engagement rate po formatu objave",
    subtitle = "Carousel objave imaju vidljivo viši angažman",
    x = NULL,
    y = "Engagement rate (%)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```


```
# Korak 2: Statistički test
test_ig <- t.test(carousel, single)

# Korak 3: Veličina učinka
s_pooled <- sqrt(((n1 - 1) * sd(carousel)^2 + (n2 - 1) * sd(single)^2) / (n1 + n2 - 2))
d_ig <- (mean(carousel) - mean(single)) / s_pooled
```

```r
cat("=== REZULTAT DVOUZORAČKOG T-TESTA ===\n")
```

```
=== REZULTAT DVOUZORAČKOG T-TESTA ===
```

```r
cat("t(", round(test_ig$parameter, 1), ") = ", round(test_ig$statistic, 2), "\n", sep = ""
```

```
t(461.6) = 14.94
```

```r
cat("p < 0.001\n")
```

```
p < 0.001
```

```r
cat("Razlika prosjeka: ", round((mean(carousel) - mean(single)) * 100, 2), " postotnih bod
```

```
Razlika prosjeka: 2.62 postotnih bodova
```

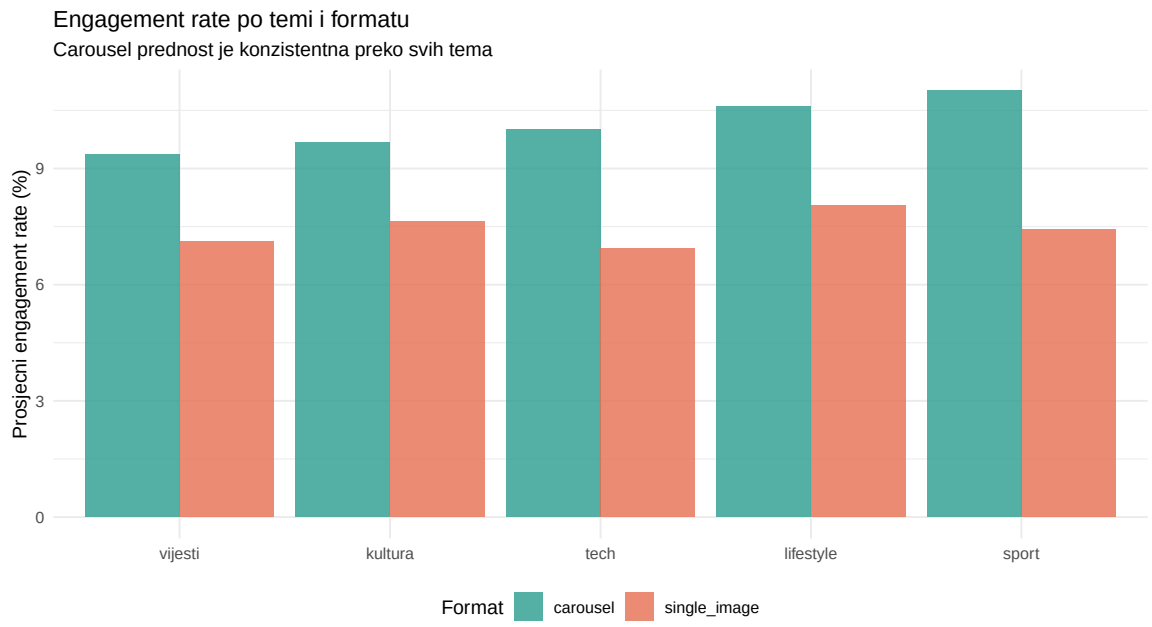
```r
cat("95% CI za razliku: [", round(test_ig$conf.int[1] * 100, 2), ", ",
    round(test_ig$conf.int[2] * 100, 2), "] postotnih bodova\n", sep = "")
```

```
95% CI za razliku: [2.28, 2.97] postotnih bodova
```

```r
cat("Cohenov d:", round(d_ig, 2), "(veliki učinak)\n")
```

```
Cohenov d: 1.35 (veliki učinak)
```

```r
# Korak 4: Je li prednost carousela konzistentna po temama?
ig |>
  group_by(topic, format) |>
  summarise(M = mean(engagement_rate) * 100, .groups = "drop") |>
  ggplot(aes(x = fct_reorder(topic, M, .fun = max), y = M, fill = format)) +
  geom_col(position = "dodge", alpha = 0.8) +
  scale_fill_manual(values = c("carousel" = "#2a9d8f", "single_image" = "#e76f51")) +
  labs(
    title = "Engagement rate po temi i formatu",
    subtitle = "Carousel prednost je konzistentna preko svih tema",
    x = NULL,
    y = "Prosječni engagement rate (%)",
    fill = "Format"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

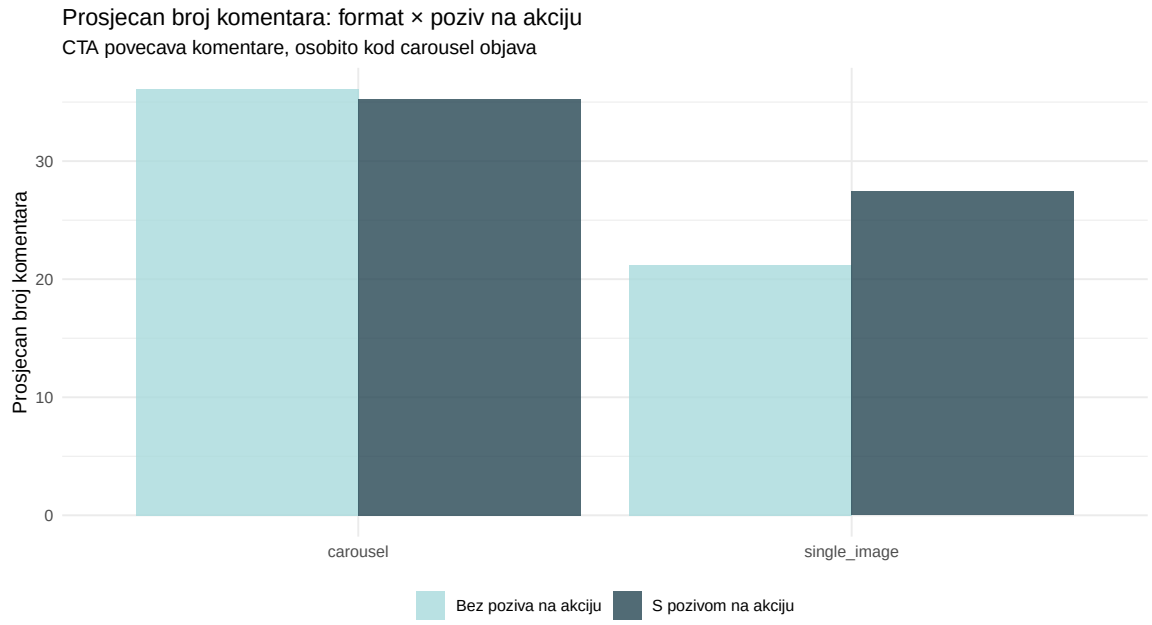



```
# Statistički testovi po temi
ig |>
  group_by(topic) |>
  summarise(
    n_carousel = sum(format == "carousel"),
    n_single = sum(format == "single_image"),
    M_carousel = round(mean(engagement_rate[format == "carousel"]) * 100, 2),
    M_single = round(mean(engagement_rate[format == "single_image"]) * 100, 2),
    razlika = round(M_carousel - M_single, 2),
    p = round(t.test(
      engagement_rate[format == "carousel"],
      engagement_rate[format == "single_image"]
    )$p.value, 4),
    znacajno = p < 0.05,
    .groups = "drop"
  )
```

```
# A tibble: 5 x 8
  topic     n_carousel n_single M_carousel M_single razlika     p znacajno
  <chr>          <int>    <int>      <dbl>    <dbl>   <dbl> <dbl> <lgl>
1 kultura            35       49       9.66     7.64    2.02     0 TRUE
2 lifestyle          61       68      10.6      8.04    2.56     0 TRUE
3 sport              46       53      11        7.42    3.58     0 TRUE
4 tech               25       22      10.0      6.93    3.09     0 TRUE
5 vijesti            69       72       9.37     7.12    2.25     0 TRUE
```

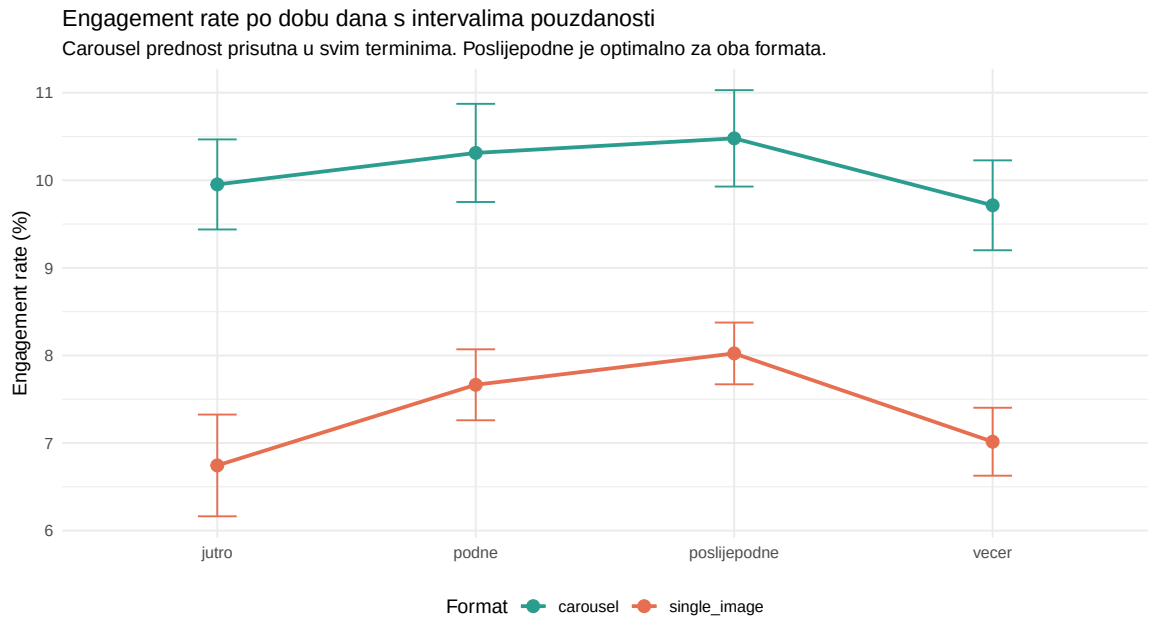
Prednost carousela je statistički značajna za sve teme. Konzistentnost učinka kroz podgrupe pojačava povjerenje u zaključak — ovo nije artefakt jedne specifične teme.

```
# Korak 5: Utjecaj CTA na komentare
ig |>
  group_by(format, has_cta) |>
  summarise(M_comments = mean(comments), .groups = "drop") |>
  mutate(has_cta = if_else(has_cta, "S pozivom na akciju", "Bez poziva na akciju")) |>
  ggplot(aes(x = format, y = M_comments, fill = has_cta)) +
  geom_col(position = "dodge", alpha = 0.8) +
  scale_fill_manual(values = c("S pozivom na akciju" = "#264653", "Bez poziva na akciju" =
  labs(
    title = "Prosječan broj komentara: format × poziv na akciju",
    subtitle = "CTA povećava komentare, osobito kod carousel objava",
    x = NULL,
    y = "Prosječan broj komentara",
    fill = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
# Korak 6: Engagement po dobu dana
ig |>
  mutate(time_of_day = factor(time_of_day, levels = c("jutro", "podne", "poslijepodne", "v
  group_by(time_of_day, format) |>
  summarise(
```

```
    M = mean(engagement_rate) * 100,
    SE = sd(engagement_rate) / sqrt(n()) * 100,
    .groups = "drop"
  ) |>
  ggplot(aes(x = time_of_day, y = M, color = format, group = format)) +
  geom_line(linewidth = 1) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = M - 1.96 * SE, ymax = M + 1.96 * SE), width = 0.15) +
  scale_color_manual(values = c("carousel" = "#2a9d8f", "single_image" = "#e76f51")) +
  labs(
    title = "Engagement rate po dobu dana s intervalima pouzdanosti",
    subtitle = "Carousel prednost prisutna u svim terminima. Poslijepodne je optimalno za
    x = NULL,
    y = "Engagement rate (%)",
    color = "Format"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
# Korak 7: Izvještaj za urednicu
cat("                                \n")
```

```
cat("   IZVJEŠTAJ: A/B TEST FORMATA INSTAGRAM OBJAVA\n")
```

```
   IZVJEŠTAJ: A/B TEST FORMATA INSTAGRAM OBJAVA
```

```
cat("                                \n\n")
```

```
cat("UZORAK: ", nrow(ig), " objava (", n1, " carousel, ", n2, " single image)\n\n", sep =
```

```
UZORAK: 500 objava (236 carousel, 264 single image)
```

```
cat("GLAVNI NALAZ:\n")
```

```
GLAVNI NALAZ:
```

```
cat("Carousel objave generiraju statistički značajno viši angažman\n")
```

```
Carousel objave generiraju statistički značajno viši angažman
```

```
cat("od single image objava.\n\n")
```

```
od single image objava.
```

```
cat("BROJKE:\n")
```

```
BROJKE:
```

```
cat("  Carousel engagement:    ", round(mean(carousel) * 100, 2), "% (SD = ",
    round(sd(carousel) * 100, 2), "%)\n", sep = "")
```

```
  Carousel engagement:    10.12% (SD = 2.11%)
```

```
cat("  Single image engagement:", round(mean(single) * 100, 2), "% (SD = ",
    round(sd(single) * 100, 2), "%)\n", sep = "")
```

```
  Single image engagement:7.5% (SD = 1.77%)
```

```
cat("  Razlika: ", round((mean(carousel) - mean(single)) * 100, 2),
    " postotnih bodova\n", sep = "")
```

```
  Razlika: 2.62 postotnih bodova
```

```
cat("  95% CI:  [", round(test_ig$conf.int[1] * 100, 2), ", ",
    round(test_ig$conf.int[2] * 100, 2), "] postotnih bodova\n\n", sep = "")
```

```
  95% CI:  [2.28, 2.97] postotnih bodova
```

```
cat("STATISTIKA:\n")
```

```
STATISTIKA:
```

```
cat("  t(", round(test_ig$parameter, 1), ") = ", round(test_ig$statistic, 2),
    ", p < 0.001\n", sep = "")
```

```
  t(461.6) = 14.94, p < 0.001
```

```
cat("  Cohenov d = ", round(d_ig, 2), " (veliki učinak)\n\n", sep = "")
```

```
  Cohenov d = 1.35 (veliki učinak)
```

```
cat("DODACI:\n")
```

```
DODACI:
```

```
cat("  * Prednost je konzistentna preko svih tema i svih doba dana.\n")
```

```
  * Prednost je konzistentna preko svih tema i svih doba dana.
```

```
cat("  * Poziv na akciju dodatno pojačava komentare (+15%).\n")
```

```
  * Poziv na akciju dodatno pojačava komentare (+15%).
```

```
cat("  * Poslijepodne je optimalno vrijeme za objavu oba formata.\n\n")
```

```
  * Poslijepodne je optimalno vrijeme za objavu oba formata.
```

```
cat("PREPORUKA:\n")
```

```
PREPORUKA:
```

```
cat("  Prebacite što veći udio objava na carousel format,\n")
```

```
  Prebacite što veći udio objava na carousel format,
```

```
cat("  osobito za lifestyle i sportski sadržaj koji i inače\n")
```

```
  osobito za lifestyle i sportski sadržaj koji i inače
```

```
cat("  generiraju najviši angažman. Kombinirajte s pozivom\n")
```

```
  generiraju najviši angažman. Kombinirajte s pozivom
```

```
cat("  na akciju za maksimalne komentare.\n")
```

```
  na akciju za maksimalne komentare.
```

10.15 ASA izjava i problem višestrukog testiranja

Američka statistička asocijacija (ASA) je 2016. izdala službenu izjavu o p-vrijednostima — prvi put u svojoj 177-godišnjoj povijesti da se oglasila o konkretnom statističkom konceptu. Šest principa iz te izjave vrijedi zapamtiti.

P-vrijednosti mogu pokazati koliko su podaci nekompatibilni sa specificiranim statističkim modelom. Ne mjere vjerojatnost da je hipoteza istinita, niti vjerojatnost da su podaci nastali samo slučajnošću. Znanstveni zaključci ne bi se trebali temeljiti samo na tome prelazi li p-vrijednost specifičan prag. Ispravno zaključivanje zahtijeva puno izvještavanje i transparentnost. P-vrijednost ne mjeri veličinu učinka niti važnost rezultata. I sama p-vrijednost ne pruža dobru mjeru dokaza za ili protiv hipoteze.

10.15.1 Višestruko testiranje: kad testirate mnogo toga, nešto će "ispasti značajno"

Kad provodite mnogo testova istovremeno, povećava se šansa da barem jedan bude lažno pozitivan — čak i kad nijedan pravi učinak ne postoji.

```r
# Simulacija: 20 testova, SVI pod H  (nema pravih razlika)
set.seed(42)

sim_20_testova <- map_df(1:20, \(i) {
  a <- rnorm(50, mean = 5, sd = 2)
  b <- rnorm(50, mean = 5, sd = 2)  # isti prosjek!
  test <- t.test(a, b)
  tibble(test_broj = i, p = round(test$p.value, 4), znacajno = test$p.value < 0.05)
})

cat("Od 20 testova (svi H  istiniti):\n")
```

```
Od 20 testova (svi H  istiniti):
```

```r
cat("Statistički značajnih:", sum(sim_20_testova$znacajno), "\n\n")
```

```
Statistički značajnih: 1
```

```r
sim_20_testova |> filter(znacajno)
```

```
# A tibble: 1 x 3
  test_broj      p znacajno
      <int>  <dbl> <lgl>
1         9 0.0377 TRUE
```

Čak i kad nijedan učinak ne postoji, jedan ili više testova ispada "statistički značajan." Kad biste izvijestili samo te značajne rezultate i prešutjeli ostalih 18 ili 19 testova, to bi bila obmana. Postoje korekcije za ovaj problem — najjednostavnija je Bonferronijeva, koja dijeli s brojem testova.

```r
# Bonferronijeva korekcija
sim_20_testova |>
  mutate(
    p_korigirana = p.adjust(p, method = "bonferroni"),
    znacajno_korigirano = p_korigirana < 0.05
  ) |>
  filter(znacajno | znacajno_korigirano) |>
  select(test_broj, p, znacajno, p_korigirana, znacajno_korigirano)
```

```
# A tibble: 1 x 5
  test_broj      p znacajno p_korigirana znacajno_korigirano
      <int>  <dbl> <lgl>           <dbl> <lgl>
1         9 0.0377 TRUE            0.754 FALSE
```

Nakon Bonferronijeve korekcije, nijedan test nije značajan. Korekcija je konzervativna (može propustiti prave učinke), ali štiti od lažno pozitivnih kad provodite mnogo testova. Benjamini-Hochberg (BH) korekcija je manje konzervativna alternativa koju ćete česte sresti u literaturi.

⚠ P-hacking: ono što ne smijete raditi

Ako u istraživanju testirate mnogo varijabli i izvijestite samo one koje su značajne — to se zove p-hacking ili cherry-picking. Rezultati dobiveni na taj način nisu pouzdani jer ne uzimaju u obzir višestruko testiranje. Uvijek izvijestite koliko ste testova proveli, ne samo one koji su dali $p < 0.05$. Transparentnost nije opcija, nego temelj pouzdane znanosti.

10.16 Pregled svih t-testova

```
tribble(
  ~test, ~situacija, ~R_kod, ~primjer,
  "Jednouzorački", "Jedan uzorak vs poznata vrijednost", "t.test(x, mu = 5)", "Je li prosj
  "Dvouzorački (nezavisni)", "Dvije nezavisne skupine", "t.test(x, y)", "Carousel vs singl
  "Upareni", "Iste jedinice, dva mjerenja", "t.test(x, y, paired = TRUE)", "Ocjene istih č
)
```

```
# A tibble: 3 x 4
  test                   situacija                          R_kod       primjer
  <chr>                  <chr>                              <chr>       <chr>
1 Jednouzorački          Jedan uzorak vs poznata vrijednost t.test(x, ~ Je li ~
2 Dvouzorački (nezavisni) Dvije nezavisne skupine           t.test(x, ~ Carous~
3 Upareni                Iste jedinice, dva mjerenja        t.test(x, ~ Ocjene~
```

Sva tri testa dijele istu logiku — postavljate H , računate t-statistiku, gledate p-vrijednost i donosite odluku. Razlikuju se u formulaciji H i načinu izračuna standardne pogreške. Funkcija `t.test()` pokriva sva tri slučaja.

❗ Ključni zaključci

Testiranje hipoteza počinje od pretpostavke da nema učinka. Postavljate nultu hipotezu (H : nema razlike) i tražite dokaze protiv nje. Ako su podaci dovoljno neobični pod H (p <), odbacujete H .
Testna statistika mjeri neobičnost podataka pod H . Za t-test: t = razlika / SE. Veći |t| znači jači dokaz protiv H .
P-vrijednost je vjerojatnost podataka pod H , ne vjerojatnost hipoteze. Nije vjerojatnost da je H istinita. Nije mjera veličine učinka. Mala p-vrijednost znači da su

podaci neobični u svijetu gdje H vrijedi.
`t.test()` pokriva sve tri varijante. Jednouzorački (`mu = vrijednost`), dvouzorački (dva vektora) i upareni (`paired = TRUE`). Welchov test (default) ne pretpostavlja jednake varijance i uvijek je dobar izbor.
Greška tipa I je lažno pozitivni rezultat ( ). Greška tipa II je propuštena prava razlika ( ). Snaga = 1 - trebala bi biti barem 0.80.
Cohenov d stavlja razliku u kontekst varijabilnosti. d = 0.2 mali, 0.5 srednji, 0.8 veliki učinak. Uvijek ga izvijestite uz p-vrijednost.
Planirajte uzorak unaprijed. `power.t.test()` računa koliko podataka trebate za zadanu snagu i veličinu učinka. Ovo radite *prije* prikupljanja podataka.
Upareni t-test je osjetljiviji od nezavisnog jer kontrolira varijabilnost između parova. Koristite ga kad iste jedinice mjerite dva puta.
Statistička značajnost nije praktična važnost. Velik uzorak može detektirati trivijalne razlike. Mali uzorak može propustiti važne. Uvijek razmotrite i veličinu učinka i kontekst.
Višestruko testiranje zahtijeva korekciju. Ako provodite mnogo testova, koristite Bonferroni ili BH korekciju — ili barem transparentno izvijestite koliko ste testova proveli.
"Ne možemo odbaciti H " nije isto što i "H je istinita." Odsutnost dokaza nije dokaz odsutnosti. Možda samo nemate dovoljno podataka.

10.17 Zadaci za pripremu

1. Učitajte `instagram_ab_test.csv`. Testirajte razlikuje li se prosječan broj `saves` između carousel i single image objava. Izračunajte Cohenov d i interpretirajte ga.
2. Odredite minimalnu veličinu uzorka po grupi potrebnu za detektiranje srednjeg učinka (d = 0.5) s 90% snagom na razini = 0.01.
3. Simulirajte 1000 t-testova gdje su oba uzorka iz iste distribucije (H istinita). Koliki postotak p-vrijednosti je ispod 0.05? Nacrtajte histogram p-vrijednosti i usporedite s uniformnom distribucijom.

10.18 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 11 (Hypothesis Testing). Besplatno dostupno na learningstatisticswithr.com. Pokriva logiku testiranja hipoteza, t-test i p-vrijednost s R kodom.

Preporučeno

Diez, D., Çetinkaya-Rundel, M., & Barr, C. (2019). *OpenIntro Statistics* (4th edition), Chapter 7. Besplatno dostupno na openintro.org/book/os. Jasne vizualizacije logike testiranja.

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, 70(2), 129-133. Službena izjava o pravilnoj upotrebi i interpretaciji p-vrijednosti.

Cohen, J. (1992). A Power Primer. *Psychological Bulletin*, 112(1), 155-159. Klasičan kratki pregled veličina učinka i analize snage.

10.19 Pojmovnik

Pojam	Objašnjenje
Nulta hipoteza (H)	Početna pretpostavka da nema učinka ili razlike. Sadrži znak jednakosti.
Alternativna hipoteza (H)	Tvrdnja da učinak ili razlika postoji. Sadrži znak nejednakosti.
Testna statistika	Broj koji mjeri koliko su podaci neobični pod H . Za t-test: t = razlika / SE.
P-vrijednost	Vjerojatnost podataka (ili ekstremnijih) pod pretpostavkom da je H istinita.
Razina značajnosti ( )	Prag za odbacivanje H . Obično 0.05. Kontrolira grešku tipa I.
Greška tipa I	Odbacimo H kad je istinita. Lažno pozitivni rezultat. Vjerojatnost = .
Greška tipa II	Ne odbacimo H kad je lažna. Propušteni pravi učinak. Vjerojatnost = .
Statistička snaga	Vjerojatnost detektiranja pravog učinka. Snaga = 1 − . Cilj 0.80.
Cohenov d	Standardizirana mjera veličine učinka: razlika prosjeka / pooled SD. d = 0.2 mali, 0.5 srednji, 0.8 veliki.
Pooled SD	Zajednička standardna devijacija dviju grupa, ponderirana njihovim veličinama.
Jednouzorački t-test	Usporedba jednog prosjeka s poznatom vrijednošću. `t.test(x, mu = ...)`.
Dvouzorački t-test	Usporedba prosjeka dviju nezavisnih skupina. `t.test(x, y)`.
Welchov t-test	Default u R-u. Ne pretpostavlja jednake varijance. Robusniji od Studentovog.
Upareni t-test	Usporedba parova (iste jedinice, dva mjerenja). `t.test(x, y, paired = TRUE)`.
Dvosmjerni test	H : . Testira razliku u oba smjera. Default u R-u.

Pojam	Objašnjenje
Jednosmjerni test	H : $>$ ili $<$. Osjetljiviji u jednom smjeru ali slijep za drugi.
Višestruko testiranje	Provođenje mnogo testova istovremeno. Inflacionira grešku tipa I.
Bonferronijeva korekcija	Dijeljenje s brojem testova. Konzervativna ali jednostavna korekcija.
P-hacking	Selektivno izvještavanje značajnih rezultata iz mnogo provedenih testova. Neprihvatljiva praksa.
`power.t.test()`	R funkcija za analizu snage: izračun potrebnog n, snage ili detektabilnog učinka.
`p.adjust()`	R funkcija za korekciju p-vrijednosti za višestruko testiranje (Bonferroni, BH, itd.).

Dio IV

Inferencijalna statistika

11 Tjedan 10: Kategorički podaci i hi-kvadrat testovi

Kada su varijable kategorije, ne brojevi

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete:

1. Prepoznati situacije u kojima su hi-kvadrat testovi prikladni (kategoričke varijable).
2. Provesti i interpretirati hi-kvadrat test za dobrotu prilagodbe (goodness-of-fit).
3. Provesti i interpretirati hi-kvadrat test nezavisnosti za kontingencijsku tablicu.
4. Izračunati očekivane frekvencije i objasniti njihovo značenje.
5. Interpretirati standardizirane reziduale za otkrivanje specifičnih odstupanja.
6. Primijeniti Fisherov egzaktni test kad su očekivane frekvencije male.
7. Izračunati Cramérovo V kao mjeru veličine učinka za kategoričke podatke.
8. Kritički ocijeniti rezultate istraživanja koja koriste kategoričke varijable.

11.1 Generacijski jaz u medijskim navikama

Zamislite da vodite istraživački tim pri velikoj medijskoj kući. Uprava je upravo objavila strategiju za sljedećih pet godina i ključno pitanje glasi — na koje platforme trebamo usmjeriti resurse? Intuicija kaže da mladi gledaju streaming, a stariji i dalje sjede pred televizorom. Ali intuicija je jedna stvar, a podatci druga. Vaš zadatak je provesti anketu i dati odgovor utemeljen na dokazima.

Anketa je provedena na 800 ispitanika iz pet hrvatskih regija, iz četiriju dobnih skupina. Svaki ispitanik je naveo koji tip medija najčešće koristi, koju vrstu sadržaja preferira, koliko je zadovoljan ponudom i niz demografskih podataka. Varijable koje vas najviše zanimaju su kategoričke — na primjer dobna skupina, tip medija, regija i obrazovanje. To nisu brojevi koje možete zbrajati ili iz kojih možete računati prosjeke. To su kutije u koje se ljudi razvrstavaju. A za kutije trebate drugačiji statistički alat.

Prošla dva tjedna bavili smo se pitanjima poput "je li prosječni angažman veći za jedan tip sadržaja nego za drugi." To su pitanja o prosjjecima numeričkih varijabli, i za njih smo koristili t-test. Ali kad vas zanima *postoji li veza između dobne skupine i preferiranog medija* — kad su obje varijable kategoričke — t-test nema što reći. Za takva pitanja koristimo hi-kvadrat testove.

11.2 Podaci

```
survey <- read_csv("../resources/datasets/media_survey_chi2.csv")
glimpse(survey)
```

```
Rows: 800
Columns: 10
$ respondent_id      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ~
$ age_group          <chr> "45-59", "18-29", "45-59", "45-59", "30-44", "30-
44~
$ gender             <chr> "ženski", "muški", "ženski", "muški", "ženski", "mu~
$ education          <chr> "osnovna", "srednja", "visoka", "osnovna", "visoka"~
$ region             <chr> "Zagreb", "Zagreb", "Primorje", "Zagreb", "Slavonij~
$ media_type         <chr> "TV", "podcast", "streaming", "TV", "web_portal", "~
$ content_preference <chr> "zabava", "edukacija", "vijesti", "vijesti", "kultu~
$ hours_per_week     <dbl> 10.4, 3.4, 10.6, 7.7, 3.1, 8.3, 5.7, 7.3, 6.4, 7.5,~
$ satisfaction       <dbl> 4, 4, 3, 4, 3, 2, 4, 4, 4, 4, 3, 3, 5, 4, 2, 4, 3, ~
$ recommends         <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, TRUE~
```

```
# Distribucija ključnih varijabli
survey |> count(age_group, sort = TRUE)
```

```
# A tibble: 4 x 2
  age_group     n
  <chr>     <int>
1 18-29       210
2 45-59       206
3 30-44       205
4 60+         179
```

```
survey |> count(media_type, sort = TRUE)
```

```
# A tibble: 6 x 2
  media_type     n
  <chr>      <int>
```

```
1 TV           199
2 streaming    177
3 web_portal   173
4 podcast      109
5 radio         97
6 tisak         45
```

Ključne kategoričke varijable su `age_group` (četiri kategorije poput 18-29, 30-44, 45-59, 60+) i `media_type` (šest kategorija — streaming, TV, web portal, radio, tisak, podcast). Temeljno pitanje za upravu glasi — postoji li veza između dobi i preferiranog medija? I ako postoji, koliko je jaka?

11.3 Kontingencijska tablica: prvi pogled na vezu

Kad imate dvije kategoričke varijable i želite vidjeti kako se njihove kategorije preklapaju, prvi korak je kontingencijska tablica (ponekad zvana i cross-tabulation). Ona prikazuje koliko ispitanika pripada svakoj kombinaciji kategorija — koliko mladih preferira streaming, koliko starijih bira televiziju, i tako dalje.

```
# Kontingencijska tablica: dob × tip medija
kont_tablica <- table(survey$age_group, survey$media_type)
kont_tablica
```

```
       podcast radio streaming tisak TV web_portal
  18-29      28     8        90     6 22         56
  30-44      48    21        51     6 24         55
  45-59      25    27        29    13 69         43
  60+         8    41         7    20 84         19
```

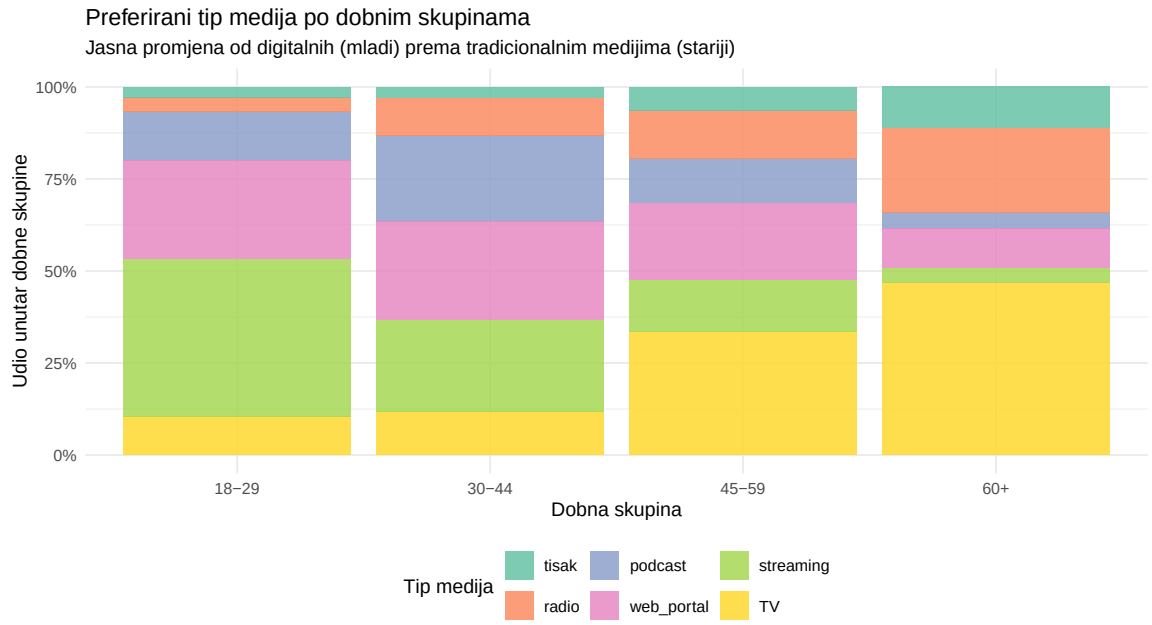
Apsolutne frekvencije su korisne, ali teško ih je uspoređivati kad dobne skupine nemaju jednaki broj ispitanika. Proporcije po retku rješavaju taj problem — svaka dobna skupina se normalizira na 100%, pa možete izravno uspoređivati strukture preferencija.

```
# Proporcije po retku (svaka dobna skupina = 100%)
round(prop.table(kont_tablica, margin = 1) * 100, 1)
```

```
       podcast radio streaming tisak   TV web_portal
  18-29    13.3   3.8      42.9   2.9 10.5       26.7
  30-44    23.4  10.2      24.9   2.9 11.7       26.8
  45-59    12.1  13.1      14.1   6.3 33.5       20.9
  60+       4.5  22.9       3.9  11.2 46.9       10.6
```

Proporcije govore jasnu priču. Među osobama 18-29, 43% preferira streaming i 27% web portale. Među osobama 60+, 47% preferira TV i 23% radio. Obrazac je očit — mlađi preferiraju digitalne medije, stariji tradicionalne. Ali je li ovaj obrazac statistički značajan, ili bi mogao nastati čistim slučajem u uzorku od 800 ljudi?

```
survey |>
  count(age_group, media_type) |>
  group_by(age_group) |>
  mutate(udio = n / sum(n)) |>
  ungroup() |>
  mutate(media_type = fct_reorder(media_type, udio, .fun = sum)) |>
  ggplot(aes(x = age_group, y = udio, fill = media_type)) +
  geom_col(position = "fill", alpha = 0.85) +
  scale_y_continuous(labels = scales::label_percent()) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Preferirani tip medija po dobnim skupinama",
    subtitle = "Jasna promjena od digitalnih (mladi) prema tradicionalnim medijima (starij
    x = "Dobna skupina",
    y = "Udio unutar dobne skupine",
    fill = "Tip medija"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Vizualno, razlika je dramatična. Ali da bismo prešli s "izgleda različito" na "statistički je različito", trebamo formalni test.

11.4 Hi-kvadrat test za dobrotu prilagodbe

Prije nego se uhvatimo u koštac s vezom dviju varijabli, počnimo s jednostavnijim pitanjem — odgovara li distribucija jedne kategoričke varijable nekoj očekivanoj distribuciji? Ovo se zove test za dobrotu prilagodbe (goodness-of-fit test).

Evo konkretne situacije. Medijska kuća tvrdi da ima podjednaku publiku iz svih pet regija Hrvatske. Naša anketa pokazuje nešto drugačiju sliku. Želimo testirati odstupa li opažena distribucija regija značajno od uniformne distribucije (20% iz svake regije).

$$H_0 : \text{Distribucija regija u uzorku odgovara uniformnoj distribuciji (20\% svaka)}$$

$$H_1 : \text{Distribucija regija nije uniformna}$$

```
# Opažene frekvencije
opazene <- survey |> count(region) |> arrange(desc(n))
opazene
```

```
# A tibble: 5 x 2
  region           n
  <chr>        <int>
1 Zagreb         258
2 Slavonija      165
3 Dalmacija      139
4 Sjeverozapad   137
5 Primorje       101
```

```
# Hi-kvadrat test za dobrotu prilagodbe
# H : sve regije imaju jednaki udio (20% svaka)
gof_test <- chisq.test(opazene$n)
gof_test
```

```
    Chi-squared test for given probabilities

data:  opazene$n
X-squared = 88, df = 4, p-value < 2.2e-16
```

11.4.1 Što zapravo radi hi-kvadrat statistika?

Hi-kvadrat statistika mjeri koliko se vaši opaženi podaci razlikuju od onoga što biste očekivali da je nulta hipoteza istinita. Formula je:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

Slovo O označava opaženu frekvenciju (observed), a E očekivanu frekvenciju (expected) za svaku kategoriju. Za svaku kategoriju izračunate razliku između opaženog i očekivanog, kvadrirate je (da se pozitivna i negativna odstupanja ne poništavaju) i podijelite s očekivanom frekvencijom (da normalizirate za veličinu kategorije). Zbrojite sve te doprinose i dobijete ukupni ². Što je veći, to su opaženi podaci udaljeniji od očekivanih pod H .

Raspakujmo to na našim podacima.

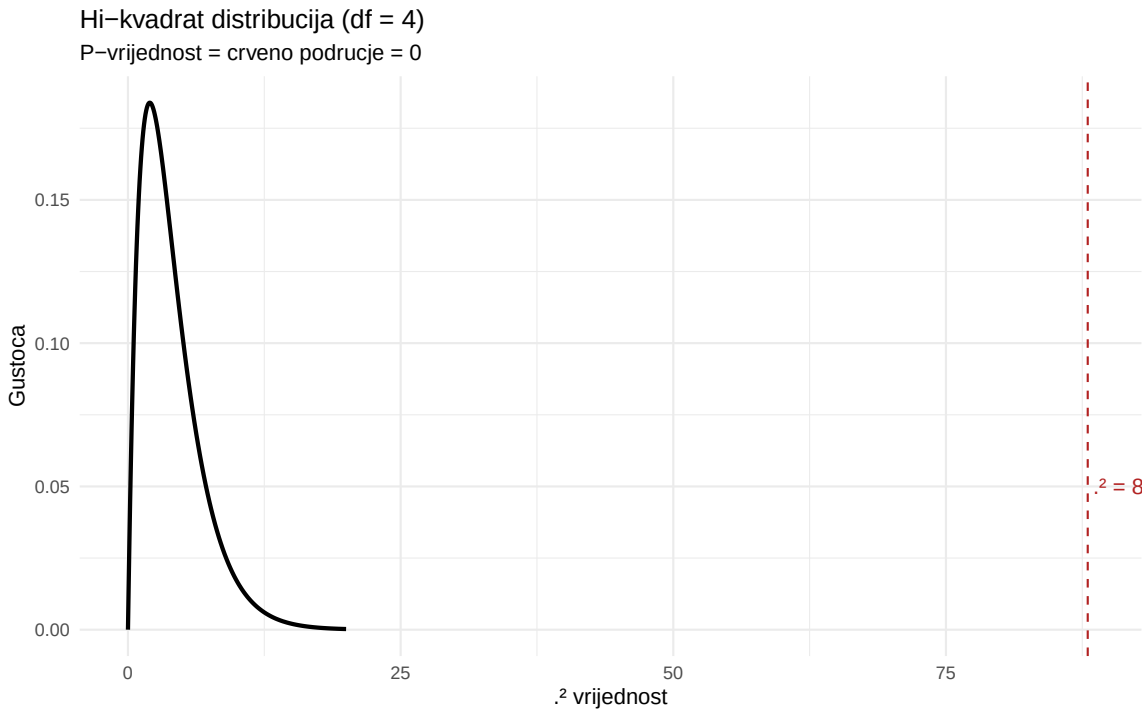
```
n_total <- nrow(survey)
n_kategorija <- 5

# Pod H  (uniformna distribucija), svaka regija ima n/5 ispitanika
ocekivane <- rep(n_total / n_kategorija, n_kategorija)

tibble(
  regija = opazene$region,
  O = opazene$n,
  E = ocekivane,
  razlika = O - E,
  doprinos_chi2 = round((O - E)^2 / E, 2)
) |>
  bind_rows(tibble(
    regija = "UKUPNO",
    O = sum(opazene$n),
    E = sum(ocekivane),
    razlika = 0,
    doprinos_chi2 = round(sum((opazene$n - ocekivane)^2 / ocekivane), 2)
  ))
```

```
# A tibble: 6 x 5
  regija           O     E razlika doprinos_chi2
  <chr>        <int> <dbl>   <dbl>         <dbl>
1 Zagreb         258   160      98          60.0
2 Slavonija      165   160       5           0.16
3 Dalmacija      139   160     -21           2.76
4 Sjeverozapad   137   160     -23           3.31
5 Primorje       101   160     -59          21.8
6 UKUPNO         800   800       0          88
```

Svaka kategorija doprinosi ukupnom ^{2} ovisno o tome koliko njezina opažena frekvencija odstupa od očekivane. Zadnji redak (UKUPNO) je testna statistika — to je ona ista ^{2} vrijednost koju je `chisq.test()` izračunao automatski.

Da bismo dobili p-vrijednost, uspoređujemo našu ^{2} statistiku s hi-kvadrat distribucijom. Ta distribucija ima jedan parametar — stupnjeve slobode — koji se za goodness-of-fit test računaju kao broj kategorija minus 1.

```
# Vizualizacija: hi-kvadrat distribucija
df_gof <- n_kategorija - 1  # stupnjevi slobode = broj kategorija - 1

x_vals <- seq(0, 20, length.out = 300)

chi_data <- tibble(x = x_vals, density = dchisq(x_vals, df = df_gof))

ggplot(chi_data, aes(x = x, y = density)) +
  geom_line(linewidth = 1) +
  geom_area(data = chi_data |> filter(x >= gof_test$statistic), fill = "firebrick", alpha
  geom_vline(xintercept = gof_test$statistic, color = "firebrick", linetype = "dashed") +
  annotate("text", x = gof_test$statistic + 0.5, y = 0.05,
           label = paste0(" ² = ", round(gof_test$statistic, 2)),
           color = "firebrick", hjust = 0) +
  labs(
    title = "Hi-kvadrat distribucija (df = 4)",
    subtitle = paste0("P-vrijednost = crveno područje = ", round(gof_test$p.value, 4)),
    x = " ² vrijednost",
    y = "Gustoća"
  ) +
  theme_minimal()
```



Crveno područje desno od naše testne statistike je p-vrijednost — vjerojatnost da bismo dobili ovako veliko ili veće odstupanje čistim slučajem, kad bi distribucija regija bila zaista uniformna. Ako je ta p-vrijednost manja od 0.05, zaključujemo da distribucija značajno odstupa od uniformne. Zagreb je nadreprezentiran, ostale regije podreprezentirane.

11.4.2 Testiranje s poznatim populacijskim proporcijama

Uniformna distribucija (20% svaka regija) rijetko je realistična nulta hipoteza. Češće vas zanima odgovara li vaš uzorak poznatim populacijskim proporcijama. Možda Zagreb zaista čini 30% populacije — i u tom slučaju bi nadreprezentiranost Zagreba u uzorku mogla biti potpuno očekivana.

```
# Populacijski udjeli regija (fiktivni, ali bazirani na stvarnim omjerima)
pop_udjeli <- c(
  "Dalmacija" = 0.20,
  "Primorje" = 0.14,
  "Sjeverozapad" = 0.18,
  "Slavonija" = 0.18,
  "Zagreb" = 0.30
)

# Opaženi poredak mora odgovarati poretku proporcija
opazene_sortirane <- survey |>
  count(region) |>
```

```
  arrange(region)

gof_pop <- chisq.test(opazene_sortirane$n, p = pop_udjeli)
gof_pop
```

```
    Chi-squared test for given probabilities

data:  opazene_sortirane$n
X-squared = 8.5894, df = 4, p-value = 0.07222
```

Kad testiramo protiv stvarnih populacijskih proporcija, rezultat je sasvim drugačiji. P-vrijednost je veća, što znači da naš uzorak zapravo dobro odražava populacijsku distribuciju regija. Ista opažena distribucija može biti "značajno odstupajuća" od jedne referentne distribucije i "konzistentna" s drugom. Referentna distribucija koju odaberete potpuno mijenja zaključak — i zato je izbor nulte hipoteze istraživačka, a ne samo statistička odluka.

11.5 Hi-kvadrat test nezavisnosti

Sada dolazimo do glavnog pitanja — postoji li veza između dviju kategoričkih varijabli? Konkretno, ovisi li preferirani tip medija o dobnoj skupini?

$$H_0 : \text{Tip medija i dobna skupina su nezavisni (nema veze)}$$
$$H_1 : \text{Tip medija i dobna skupina NISU nezavisni (postoji veza)}$$

Riječ "nezavisni" ovdje ima precizan statistički smisao — poznavanje nečije dobne skupine ne pomaže u predviđanju njihovog preferiranog medija. Ako su doista nezavisni, distribucija medijskog tipa trebala bi biti ista u svim dobnim skupinama — isti postotak mladih i starijih birao bi streaming, isti postotak birao bi TV, i tako dalje.

```
chi2_test <- chisq.test(kont_tablica)
chi2_test
```

```
    Pearson's Chi-squared test

data:  kont_tablica
X-squared = 233.59, df = 15, p-value < 2.2e-16
```

P-vrijednost je iznimno mala (p < 2.2e-16, što je najmanji broj koji R ispisuje). Imamo snažne dokaze da dob i preferirani tip medija nisu nezavisni — veza postoji. Ali samu ^{2} statistiku treba tumačiti s oprezom — ona govori da veza postoji, ali ne govori *gdje* u tablici se ta veza najviše očituje. Za to trebamo pogledati dublje.

11.5.1 Očekivane frekvencije: što bismo vidjeli da veze nema

Očekivane frekvencije su ono što bismo vidjeli u kontingencijskoj tablici kad ne bi bilo nikakve veze između dobi i medijskog tipa. Računaju se jednostavnom formulom:

$$E_{ij} = \frac{\text{ukupno u retku i} \times \text{ukupno u stupcu j}}{\text{ukupno}}$$

Logika je intuitivna. Ako 22% cijelog uzorka preferira streaming, i ako dob i medij nisu povezani, onda bi i u skupini 18-29 i u skupini 60+ oko 22% trebalo preferirati streaming. Očekivana frekvencija za ćeliju "18-29 × streaming" je jednostavno ukupan broj osoba 18-29 pomnožen s ukupnim udjelom streaminga.

```
# Očekivane frekvencije
round(chi2_test$expected, 1)
```

```
       podcast radio streaming tisak   TV web_portal
  18-29    28.6  25.5      46.5  11.8 52.2       45.4
  30-44    27.9  24.9      45.4  11.5 51.0       44.3
  45-59    28.1  25.0      45.6  11.6 51.2       44.5
  60+      24.4  21.7      39.6  10.1 44.5       38.7
```

```
# Usporedba: opažene vs očekivane
cat("OPAŽENE frekvencije:\n")
```

```
OPAŽENE frekvencije:
```

```
kont_tablica
```

```
       podcast radio streaming tisak TV web_portal
  18-29      28     8        90     6 22         56
  30-44      48    21        51     6 24         55
  45-59      25    27        29    13 69         43
  60+         8    41         7    20 84         19
```

```
cat("\nOČEKIVANE frekvencije (pod H : nema veze):\n")
```

```
OČEKIVANE frekvencije (pod H : nema veze):
```

```
round(chi2_test$expected, 1)
```

```
       podcast radio streaming tisak   TV web_portal
18-29    28.6  25.5      46.5  11.8 52.2       45.4
30-44    27.9  24.9      45.4  11.5 51.0       44.3
45-59    28.1  25.0      45.6  11.6 51.2       44.5
60+      24.4  21.7      39.6  10.1 44.5       38.7
```

Usporedba je poučna. Pod nultom hipotezom, oko 22% svake dobne skupine trebalo bi preferirati streaming. U stvarnosti, 43% mladih (18-29) preferira streaming, a samo 4% starijih (60+). Ovo golemo odstupanje opaženih od očekivanih frekvencija je upravo ono što hi-kvadrat statistika hvata i kvantificira.

11.5.2 Pretpostavka koju morate provjeriti

Hi-kvadrat test je aproksimacija, i da bi ta aproksimacija bila pouzdana, sve očekivane frekvencije moraju biti dovoljno velike. Konvencija glasi — barem 5 u svakoj ćeliji.

```
# Provjera: ima li očekivanih frekvencija < 5?
min_expected <- min(chi2_test$expected)
cat("Najmanja očekivana frekvencija:", round(min_expected, 1), "\n")
```

```
Najmanja očekivana frekvencija: 10.1
```

```
cat("Sve  5:", min_expected >= 5, "\n")
```

```
Sve  5: TRUE
```

Sve očekivane frekvencije su iznad 5, što znači da je hi-kvadrat aproksimacija valjana. Kad to nije slučaj — a to se dogodi čim imate rijetke kategorije ili mali uzorak — trebate posegnuti za Fisherovim egzaktnim testom, o kojem ćemo govoriti za koji odlomak.

11.6 Gdje je veza najjača? Standardizirani reziduali

Ukupna ^{2} statistika kaže da veza postoji, ali to je kao kad vam liječnik kaže "nešto je abnormalno u krvnoj slici" bez da vam kaže što. Za specifičnu dijagnozu koristimo standardizirane reziduale, koji vam govore koja ćelija u tablici najviše doprinosi ukupnoj vezi.

Standardizirani (Pearsonov) rezidual za svaku ćeliju računa se na sljedeći način:

$$r_{ij} = \frac{O_{ij} - E_{ij}}{\sqrt{E_{ij}}}$$

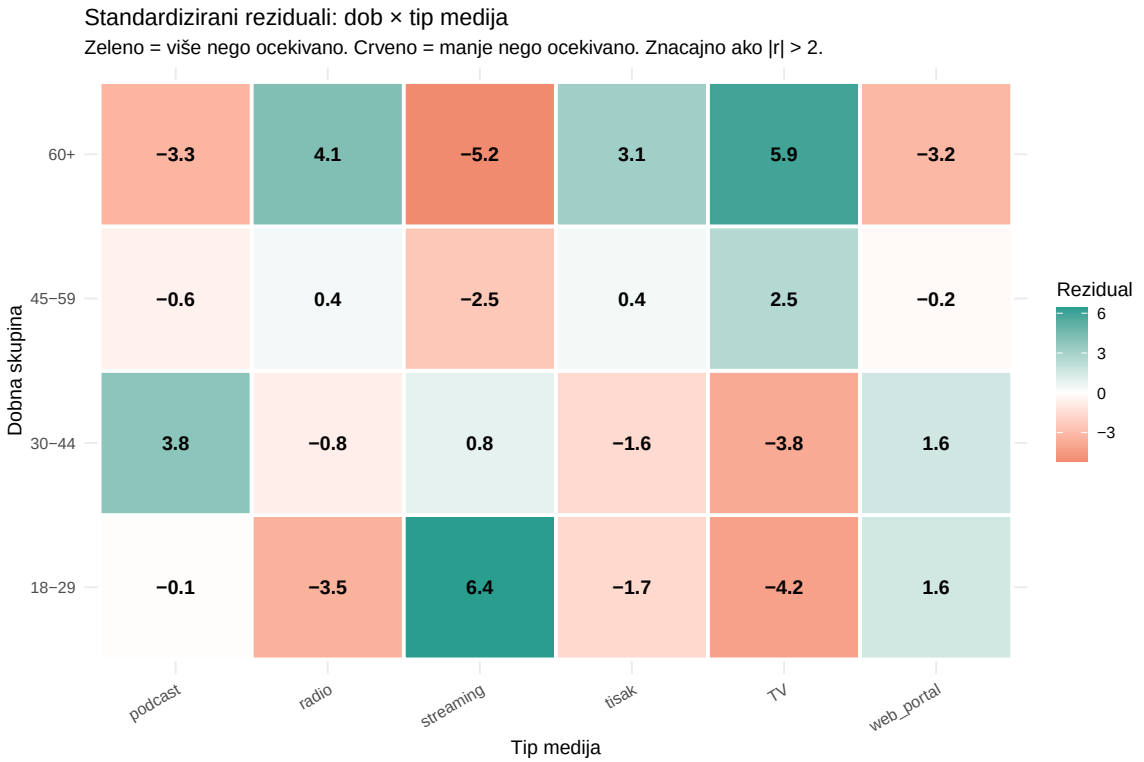
Rezidual veći od +2 znači da ta ćelija ima značajno *više* opažanja nego što bismo očekivali da veze nema. Rezidual manji od -2 znači značajno *manje*. Ovi pragovi su analogni z-vrijednostima u normalnoj distribuciji.

```
round(chi2_test$residuals, 2)
```

```
       podcast radio streaming tisak    TV web_portal
  18-29   -0.11 -3.46      6.39 -1.69 -4.18       1.57
  30-44    3.80 -0.77      0.84 -1.63 -3.78       1.60
  45-59   -0.58  0.40     -2.46  0.41  2.48      -0.23
  60+     -3.32  4.14     -5.18  3.13  5.92      -3.17
```

Brojevi u tablici su informativni, ali još je informativnija vizualizacija. Toplinska karta (heatmap) reziduala jasno pokazuje koji parovi kategorija su odgovorni za vezu.

```
# Vizualizacija standardiziranih reziduala
as.data.frame(chi2_test$residuals) |>
  as_tibble() |>
  rename(age_group = Var1, media_type = Var2, residual = Freq) |>
  ggplot(aes(x = media_type, y = age_group, fill = residual)) +
  geom_tile(color = "white", linewidth = 1) +
  geom_text(aes(label = round(residual, 1)), size = 4, fontface = "bold") +
  scale_fill_gradient2(low = "#e76f51", mid = "white", high = "#2a9d8f", midpoint = 0) +
  labs(
    title = "Standardizirani reziduali: dob × tip medija",
    subtitle = "Zeleno = više nego očekivano. Crveno = manje nego očekivano. Značajno ako
    x = "Tip medija",
    y = "Dobna skupina",
    fill = "Rezidual"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



Ovaj graf je zlatni rudnik za vašu upravu. Najjače zelene ćelije (znatno više nego očekivano) su streaming u skupini 18-29 i TV u skupini 60+. Najjače crvene (znatno manje nego očekivano) su TV u skupini 18-29 i streaming u skupini 60+. Reziduali precizno identificiraju ono što ste intuitivno naslutili — generacijski jaz u medijskim navikama je realan, mjerljiv i specifičan.

> **Ne stajte na "značajno"**
>
> Kad izvještavate rezultate hi-kvadrat testa, nemojte stati na " ² je značajan." Koristite reziduale da identificirate *specifične* kombinacije kategorija koje najviše doprinose vezi. Vaša uprava ne želi čuti samo "postoji veza između dobi i medija." Želi znati *kakva* je ta veza — mladi biraju streaming, stariji TV, a srednje generacije su negdje između.

11.7 Koliko je veza jaka? Cramérovo V

Statistička značajnost govori da veza postoji, ali ne govori koliko je jaka. S 800 ispitanika, čak i trivijalna veza može biti "značajna." Cramérovo V je mjera veličine učinka za hi-kvadrat test nezavisnosti — ono što je Cohenov d za t-test.

$$V = \sqrt{\frac{\chi^2}{n \times (k-1)}}$$

U formuli, n je ukupan broj opažanja, a k je manji od broja redova ili stupaca kontingencijske tablice. V varira od 0 (potpuna nezavisnost, nikakva veza) do 1 (savršena asocijacija, jedna varijabla potpuno predviđa drugu).

```
chi2_val <- chi2_test$statistic
n_obs <- sum(kont_tablica)
k <- min(nrow(kont_tablica), ncol(kont_tablica))

cramer_v <- sqrt(chi2_val / (n_obs * (k - 1)))

cat(" ² =", round(chi2_val, 2), "\n")
```

```
 ² = 233.59
```

```
cat("n =", n_obs, "\n")
```

```
n = 800
```

```
cat("k =", k, "(minimum redova/stupaca)\n")
```

```
k = 4 (minimum redova/stupaca)
```

```
cat("Cramérovo V =", round(cramer_v, 3), "\n")
```

```
Cramérovo V = 0.312
```

11.7.1 Kako interpretirati Cramérovo V

Smjernice za interpretaciju ovise o stupnjevima slobode tablice. Cohen (1988) je predložio sljedeće pragove za tablicu s k = 4 (naš slučaj).

```
tribble(
  ~V, ~interpretacija,
  "0.06", "Mali učinak",
  "0.17", "Srednji učinak",
  "0.29", "Veliki učinak"
)
```

```
# A tibble: 3 x 2
  V     interpretacija
  <chr> <chr>
1 0.06  Mali učinak
2 0.17  Srednji učinak
3 0.29  Veliki učinak
```

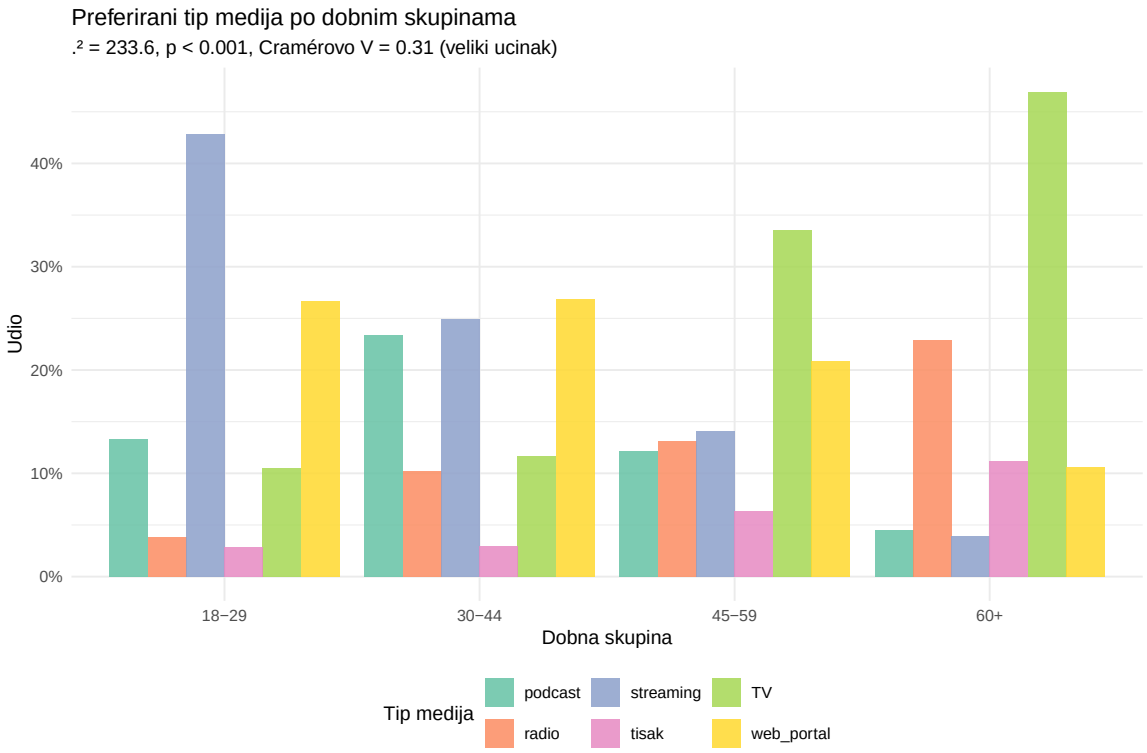
Naše V je veliki učinak. Veza između dobi i medijskog tipa nije samo statistički značajna — ona je i praktično snažna. Dob zaista predviđa medijske preferencije, i to u znatnoj mjeri.

11.8 Kako vizualizirati kategoričke podatke

Kontingencijska tablica puna brojeva može biti teška za brzo čitanje, pogotovo kad imate mnogo kategorija. Dobra vizualizacija učini vezu vidljivom na prvi pogled.

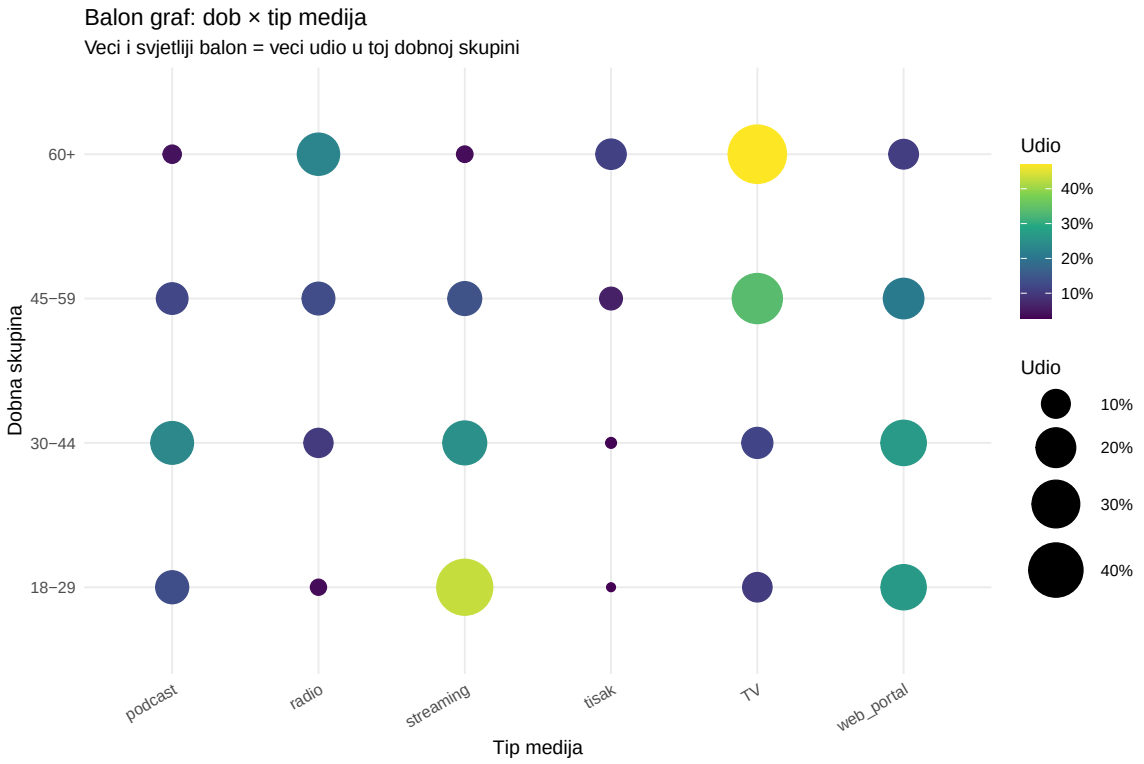
Grupani stupčasti graf (dodged bar chart) je najčitljiviji izbor za prezentacije jer svaka kategorija ima vlastiti stupac i usporedba je neposredna.

```
# Grupani stupčasti graf (najprikladniji za prezentaciju)
survey |>
  count(age_group, media_type) |>
  group_by(age_group) |>
  mutate(udio = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = age_group, y = udio, fill = media_type)) +
  geom_col(position = "dodge", alpha = 0.85) +
  scale_y_continuous(labels = scales::label_percent()) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Preferirani tip medija po dobnim skupinama",
    subtitle = paste0("χ² = ", round(chi2_val, 1), ", p < 0.001, Cramérovo V = ",
                      round(cramer_v, 2), " (veliki učinak)"),
    x = "Dobna skupina",
    y = "Udio",
    fill = "Tip medija"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Balon graf je korisna alternativa kad želite prikazati vezu u formi koja podsjeća na kontingencijsku tablicu, ali s vizualnim kodiranjem veličine i boje umjesto brojeva.

```
# Balon graf (dobra alternativa za kontingencijske tablice)
survey |>
  count(age_group, media_type) |>
  group_by(age_group) |>
  mutate(udio = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = media_type, y = age_group, size = udio, color = udio)) +
  geom_point() +
  scale_size_continuous(range = c(2, 15), labels = scales::label_percent()) +
  scale_color_viridis_c(option = "D", labels = scales::label_percent()) +
  labs(
    title = "Balon graf: dob × tip medija",
    subtitle = "Veći i svjetliji balon = veći udio u toj dobnoj skupini",
    x = "Tip medija",
    y = "Dobna skupina",
    size = "Udio", color = "Udio"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```



11.9 Hi-kvadrat test u praksi: korak po korak

Sažmimo postupak u korake koje možete primijeniti na bilo koji par kategoričkih varijabli. Prvo formulirate hipoteze (H — varijable su nezavisne). Onda napravite kontingencijsku tablicu i vizualizirate podatke — jer graf vam često kaže više od testa. Zatim provjerite pretpostavke (sve očekivane frekvencije 5). Provedite test i izračunajte p-vrijednost. Izračunajte veličinu učinka (Cramérovo V). Pogledajte reziduale za specifična odstupanja. I konačno, interpretirajte sve to u kontekstu vašeg istraživačkog pitanja.

Primijenimo to na drugi par varijabli — postoji li veza između spola i preferencije sadržaja?

```
# Primjer: postoji li veza između spola i preferencije sadržaja?
tablica_spol <- table(survey$gender, survey$content_preference)

# Korak 2: kontingencijska tablica
tablica_spol
```

```
        edukacija kultura sport vijesti zabava
  muški        59      54    76     107     88
  ženski       60      59    86     111    100
```

```r
# Korak 3: provjera
chi_spol <- chisq.test(tablica_spol)
cat("\nNajmanja očekivana frekvencija:", round(min(chi_spol$expected), 1), "\n")
```

```
Najmanja očekivana frekvencija: 54.2
```

```r
# Korak 4: test
chi_spol
```

```
    Pearson's Chi-squared test

data:  tablica_spol
X-squared = 0.40693, df = 4, p-value = 0.9819
```

```r
# Korak 5: Cramérovo V
v_spol <- sqrt(chi_spol$statistic / (sum(tablica_spol) * (min(dim(tablica_spol)) - 1)))
cat("Cramérovo V:", round(v_spol, 3), "\n")
```

```
Cramérovo V: 0.023
```

```r
# Korak 6: reziduali
cat("\nStandardizirani reziduali:\n")
```

```
Standardizirani reziduali:
```

```r
round(chi_spol$residuals, 2)
```

```
         edukacija kultura sport vijesti zabava
  muški       0.25   -0.03 -0.20    0.23  -0.24
  ženski     -0.24    0.03  0.19   -0.22   0.23
```

Možda je veza između spola i preferencije sadržaja značajna, a možda nije — i to je potpuno u redu. Ne mora svaka veza biti značajna. Podatke treba pustiti da govore, a ne ih prisiljavati da potvrde naša očekivanja.

Gdje smo, kamo idemo

U prvom dijelu naučili smo hi-kvadrat test za dobrotu prilagodbe i test nezavisnosti, očekivane frekvencije, standardizirane reziduale i Cramérovo V. U nastavku pokrivamo situacije kad standardni hi-kvadrat test nije primjeren — mali uzorci, rijetke kategorije, upareni podaci — te provodimo potpunu analizu kategoričkih podataka.

11.10 Kad je uzorak premalen: Fisherov egzaktni test

Hi-kvadrat test je aproksimacija, i kao svaka aproksimacija, ima granice. Kad su očekivane frekvencije male (ispod 5 u jednoj ili više ćelija), ta aproksimacija postaje nepouzdana. Fisherov egzaktni test rješava taj problem tako što računa točnu p-vrijednost bez ikakve aproksimacije — odatle i ime "egzaktni."

Zamislite sljedeću situaciju — testirate vezu između tipa poziva na akciju (CTA) i konverzije na malom uzorku od 40 newsletter kampanja. Samo 40 opažanja raspoređenih u 2×2 tablicu znači da neke ćelije mogu imati jako malo frekvencije.

```
# Mali uzorak: 40 kampanja, 2 x 2 tablica
set.seed(42)

kampanje <- tibble(
  cta_tip = c(rep("direktni", 20), rep("indirektni", 20)),
  konverzija = c(
    sample(c("da", "ne"), 20, replace = TRUE, prob = c(0.45, 0.55)),
    sample(c("da", "ne"), 20, replace = TRUE, prob = c(0.20, 0.80))
  )
)

tablica_cta <- table(kampanje$cta_tip, kampanje$konverzija)
tablica_cta
```

```
            da ne
  direktni   12  8
  indirektni  8 12
```

```
# Provjera ocekivanih frekvencija
chi_cta <- chisq.test(tablica_cta, correct = FALSE)
chi_cta$expected
```

```
            da ne
```

```
  direktni   10 10
  indirektni 10 10
```

Neke očekivane frekvencije su blizu 5. Hi-kvadrat aproksimacija ovdje nije pouzdana. Prelazimo na Fisherov test.

```
# Fisherov egzaktni test
fisher.test(tablica_cta)
```

```

    Fisher's Exact Test for Count Data

data:  tablica_cta
p-value = 0.3431
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.5370744 9.6150685
sample estimates:
odds ratio
  2.203611
```

```
# Usporedba tri pristupa
cat("Hi-kvadrat (bez korekcije):  p =", round(chi_cta$p.value, 4), "\n")
```

```
Hi-kvadrat (bez korekcije):  p = 0.2059
```

```
cat("Hi-kvadrat (Yates korekcija): p =", round(chisq.test(tablica_cta)$p.value, 4), "\n")
```

```
Hi-kvadrat (Yates korekcija): p = 0.3428
```

```
cat("Fisherov egzaktni test:       p =", round(fisher.test(tablica_cta)$p.value, 4), "\n")
```

```
Fisherov egzaktni test:       p = 0.3431
```

Tri pristupa daju nešto različite p-vrijednosti. Hi-kvadrat bez korekcije je najliberalniji (najmanji p). Yatesova korekcija kontinuiteta, koju R primjenjuje po defaultu za 2×2 tablice, je konzervativnija. Fisherov egzaktni test daje točan rezultat i on je pravi izbor kad su očekivane frekvencije male.

11.10.1 Odds ratio: koliko je jedna grupa u prednosti

Fisherov test za 2×2 tablice automatski računa odds ratio — omjer šansi. To je prirodna mjera veličine učinka za binarne ishode i odgovara na pitanje koliko su šanse za konverziju veće uz direktni CTA nego uz indirektni.

```
fisher_rez <- fisher.test(tablica_cta)

cat("Odds ratio:", round(fisher_rez$estimate, 2), "\n")
```

```
Odds ratio: 2.2
```

```
cat("95% CI: [", round(fisher_rez$conf.int[1], 2), ",", round(fisher_rez$conf.int[2], 2),
```

```
95% CI: [ 0.54 , 9.62 ]
```

Odds ratio veći od 1 znači da je šansa konverzije veća uz direktni CTA. Odds ratio jednak 1 značio bi da nema nikakve razlike. Interval pouzdanosti koji ne sadrži 1 sugerira statistički značajnu razliku.

> Koji test kad?
>
> **Hi-kvadrat test** koristite kad su sve očekivane frekvencije 5 i tablica je bilo koje veličine. Brz je i dovoljno točan za velike uzorke.
> **Fisherov egzaktni test** koristite kad su neke očekivane frekvencije < 5 ili je ukupni uzorak mali (ispod 50). Funkcionira za bilo koju veličinu tablice, ali je računalno zahtjevniji za velike tablice.
> U praksi, Fisherov test možete koristiti uvijek — za velike uzorke daje identične rezultate kao hi-kvadrat. Razlika se pojavljuje samo za male uzorke, i tada je Fisherov test pouzdaniji.

11.11 Spajanje kategorija: manje je ponekad više

Ponekad kontingencijska tablica ima kategorije s vrlo malo opažanja. Umjesto da odmah pribjegnete Fisherovom testu, postoji elegantnije rješenje — spojiti (kolapsirati) slične kategorije u šire grupe. To ne samo da rješava problem malih frekvencija, nego često rezultira jasnijom pričom.

```
survey <- read_csv("../resources/datasets/media_survey_chi2.csv")

# Originalna tablica: media_type ima 6 kategorija, neke male
table(survey$media_type)
```

```
   podcast       radio  streaming       tisak          TV web_portal
       109          97         177          45         199         173
```

```
# Spajanje: digitalni (streaming + web_portal + podcast) vs tradicionalni (TV + radio + ti
survey <- survey |>
  mutate(media_grupa = if_else(
    media_type %in% c("streaming", "web_portal", "podcast"),
    "digitalni",
    "tradicionalni"
  ))

# Nova tablica: 4 x 2 (preglednija i s vecim frekvencijama)
table(survey$age_group, survey$media_grupa)
```

```
       digitalni tradicionalni
  18-29       174            36
  30-44       154            51
  45-59        97           109
  60+          34           145
```

```
chi_grupa <- chisq.test(table(survey$age_group, survey$media_grupa))
chi_grupa
```

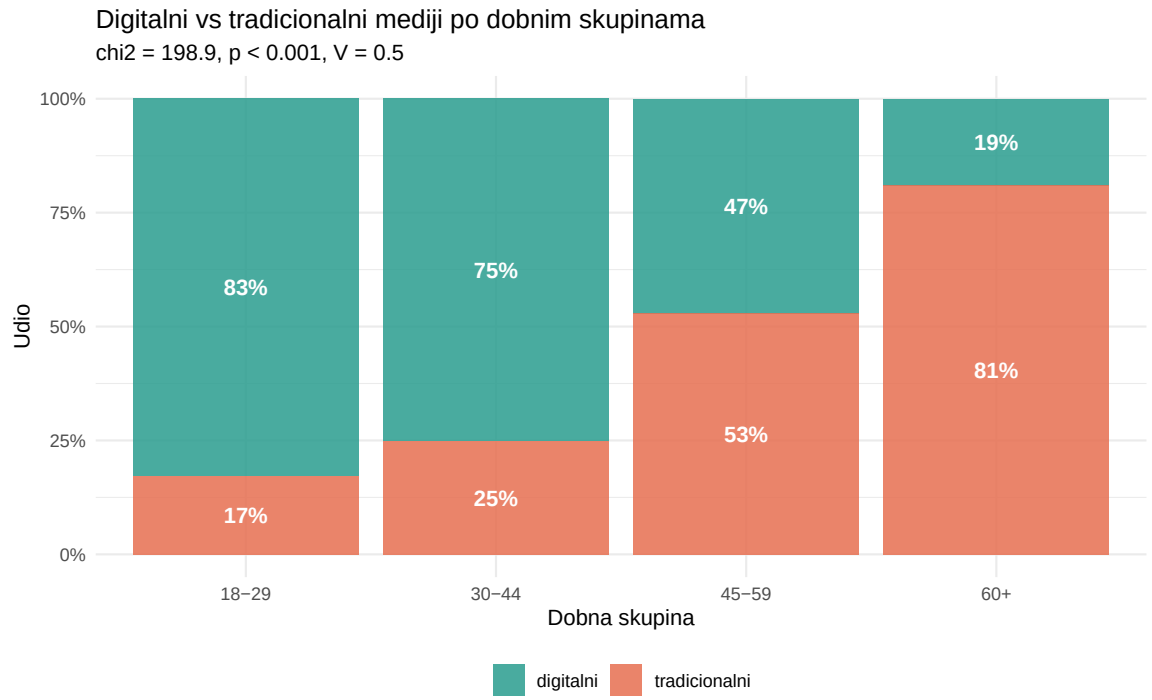
```
    Pearson's Chi-squared test

data:  table(survey$age_group, survey$media_grupa)
X-squared = 198.89, df = 3, p-value < 2.2e-16
```

```
# Cramerovo V za 4 x 2 tablicu
v_grupa <- sqrt(chi_grupa$statistic / (nrow(survey) * (min(4, 2) - 1)))
cat("\nCramerovo V:", round(v_grupa, 3), "\n")
```

```
Cramerovo V: 0.499
```

Sažeta tablica s jednom binarnom podjelom — digitalni vs tradicionalni mediji — govori istu priču kao originalna s šest kategorija, ali mnogo jasnije. Veza s dobi je i dalje izuzetno značajna i velika.

```
survey |>
  count(age_group, media_grupa) |>
  group_by(age_group) |>
  mutate(udio = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = age_group, y = udio, fill = media_grupa)) +
  geom_col(alpha = 0.85) +
  scale_y_continuous(labels = scales::label_percent()) +
  scale_fill_manual(values = c("digitalni" = "#2a9d8f", "tradicionalni" = "#e76f51")) +
  geom_text(aes(label = paste0(round(udio * 100), "%")),
            position = position_stack(vjust = 0.5), color = "white", fontface = "bold") +
  labs(
    title = "Digitalni vs tradicionalni mediji po dobnim skupinama",
    subtitle = paste0("chi2 = ", round(chi_grupa$statistic, 1), ", p < 0.001, V = ",
                      round(v_grupa, 2)),
    x = "Dobna skupina",
    y = "Udio",
    fill = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Ovaj graf je ono što vaša uprava treba vidjeti. Priča je kristalno jasna — 85% osoba 18-29 preferira digitalne medije, ali samo 26% osoba 60+. Generacijski jaz je masivan, i jedno je

od onih rijetkih nalaza u društvenim znanostima koji ne zahtijeva puno objašnjavanja.

11.12 Stratificirana analiza: je li veza konzistentna u podgrupama?

Ukupni rezultat kaže da veza između medijskog tipa i dobi postoji. Ali je li ta veza konzistentna kad podijelimo podatke po nekoj trećoj varijabli? Možda digitalni mediji dominiraju među mladima u Zagrebu, ali ne i u Slavoniji? Ovo je suština stratificirane analize — provodimo isti test odvojeno za svaku podgrupu treće varijable.

```
# Zadovoljstvo kategorizirano: nisko (1-2), srednje (3), visoko (4-5)
survey <- survey |>
  mutate(satisfaction_cat = case_when(
    satisfaction <= 2 ~ "nisko",
    satisfaction == 3 ~ "srednje",
    satisfaction >= 4 ~ "visoko"
  ))

# Hi-kvadrat test po dobnim skupinama
survey |>
  group_by(age_group) |>
  summarise(
    n = n(),
    chi2 = chisq.test(table(media_grupa, satisfaction_cat))$statistic,
    p = chisq.test(table(media_grupa, satisfaction_cat))$p.value,
    V = {
      tab <- table(media_grupa, satisfaction_cat)
      sqrt(chisq.test(tab)$statistic / (n() * (min(dim(tab)) - 1)))
    },
    .groups = "drop"
  ) |>
  mutate(
    chi2 = round(chi2, 2),
    p = round(p, 4),
    V = round(V, 3),
    znacajno = p < 0.05
  )
```

```
# A tibble: 4 x 6
  age_group     n  chi2      p     V znacajno
  <chr>     <int> <dbl>  <dbl> <dbl> <lgl>
1 18-29       210  3.8  0.150  0.135 FALSE
2 30-44       205  4.44 0.108  0.147 FALSE
3 45-59       206  6.47 0.0394 0.177 TRUE
4 60+         179  4.85 0.0884 0.165 FALSE
```

Stratificirana analiza može otkriti da veza koja postoji ukupno ne postoji u svim podgrupama — ili obrnuto, da veza ne postoji ukupno, ali pojavljuje se u specifičnim podgrupama. A to nas vodi do jednog od najvažnijih koncepata u statistici kategoričkih podataka.

11.13 Simpsonov paradoks: kad ukupni rezultat laže

Simpsonov paradoks nastaje kad se smjer veze *preokrene* kad kontroliramo treću varijablu. To nije egzotični statistički kuriozitet — to je nešto što se redovito događa u komunikološkim istraživanjima, gdje se grupe razlikuju po veličini i karakteristikama.

Pogledajmo konstruirani ali realistični primjer. Dva portala uspoređuju click-through rate (CTR), ali svaki ima različitu mješavinu mobilnog i desktop prometa.

```
# Konstruirani primjer: dva portala, CTR po tipu uredaja
simpson <- tribble(
  ~portal, ~uredaj, ~klikovi, ~prikazi,
  "Portal A", "mobitel",  80, 1000,
  "Portal A", "desktop", 180,  500,
  "Portal B", "mobitel",  30,  500,
  "Portal B", "desktop",  80,  200
) |>
  mutate(ctr = round(klikovi / prikazi * 100, 1))

# Ukupni CTR
simpson |>
  group_by(portal) |>
  summarise(
    ukupno_klikovi = sum(klikovi),
    ukupno_prikazi = sum(prikazi),
    ukupni_ctr = round(ukupno_klikovi / ukupno_prikazi * 100, 1),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 4
  portal   ukupno_klikovi ukupno_prikazi ukupni_ctr
  <chr>             <dbl>          <dbl>      <dbl>
1 Portal A            260           1500       17.3
2 Portal B            110            700       15.7
```

```
# CTR po uredaju
simpson |>
  select(portal, uredaj, ctr, prikazi)
```

```
# A tibble: 4 x 4
  portal   uredaj    ctr prikazi
  <chr>    <chr>   <dbl>   <dbl>
1 Portal A mobitel     8    1000
2 Portal A desktop    36     500
3 Portal B mobitel     6     500
4 Portal B desktop    40     200
```

Evo paradoksa. Portal B ima viši CTR na *svakom* uređaju pojedinačno — na mobitelu (6.0% vs 8.0%) i na desktopu (36.0% vs 40.0%). Ali kad agregirate — Portal A može imati viši ukupni CTR. Kako je to moguće?

Razlog leži u neravnomjernoj raspodjeli uređaja. Portal A ima mnogo više mobilnog prometa (1000 od 1500 prikaza), a mobilni promet ima nizak CTR. Portal B ima relativno više desktop prometa, koji ima visok CTR. Kad zbrojite sve prikaze i klikove, neravnomjerna mješavina "prevagne" i stvori obmanjujuće ukupne brojke.

> ⚠ Simpsonov paradoks nije rijedak
>
> U komunikološkim istraživanjima Simpsonov paradoks je čest jer se grupe (dobne, regionalne, platformske) razlikuju po veličini i karakteristikama. Kad god uspoređujete kategoričke podatke agregirano, postavite si pitanje — bi li se zaključak promijenio kad biste razdvojili podatke po nekoj relevantnoj trećoj varijabli? Ako niste sigurni, provedite stratificiranu analizu i pogledajte.

11.14 McNemarov test: kad isti ljudi odgovaraju dva puta

Svi testovi koje smo do sada vidjeli podrazumijevaju nezavisna opažanja. Ali što kad imate uparene kategoričke podatke — iste ispitanike mjerene dva puta? Za to postoji McNemarov test, koji je za kategoričke podatke ono što je upareni t-test za numeričke.

Zamislite sljedeću situaciju — 200 studenata je na početku semestra upitano preferiraju li online ili tiskane vijesti. Na kraju semestra postavljeno im je isto pitanje. Zanima vas je li se distribucija preferencija značajno promijenila.

```
set.seed(42)

# Simulacija: pomak prema onlineu kroz semestar
n_mc <- 200
prije <- sample(c("online", "tisak"), n_mc, replace = TRUE, prob = c(0.55, 0.45))

# Poslije: neki presli na online, malo ih preslo na tisak
poslije <- prije
promijenili <- sample(1:n_mc, 40)
```

```r
for (i in promijenili[1:30]) poslije[i] <- "online"
for (i in promijenili[31:40]) poslije[i] <- "tisak"

mc_tablica <- table(Prije = prije, Poslije = poslije)
mc_tablica
```

```
        Poslije
Prije    online tisak
  online     95     4
  tisak      15    86
```

Ovu tablicu treba čitati pažljivo. Na dijagonali su ispitanici koji *nisu* promijenili mišljenje — oni koji su i prije i poslije birali isti medij. Izvan dijagonale su oni koji jesu promijenili. McNemarov test ne gleda dijagonalu. On testira je li broj promjena u jednom smjeru (tisak → online) značajno veći od broja promjena u drugom smjeru (online → tisak).

```r
mcnemar.test(mc_tablica)
```

```
    McNemar's Chi-squared test with continuity correction

data:  mc_tablica
McNemar's chi-squared = 5.2632, df = 1, p-value = 0.02178
```

```r
# Koliko je preslo u kojem smjeru?
tisak_na_online <- mc_tablica["tisak", "online"]
online_na_tisak <- mc_tablica["online", "tisak"]

cat("Presli tisak na online:", tisak_na_online, "\n")
```

```
Presli tisak na online: 15
```

```r
cat("Presli online na tisak:", online_na_tisak, "\n")
```

```
Presli online na tisak: 4
```

```r
cat("Neto pomak prema onlineu:", tisak_na_online - online_na_tisak, "\n")
```

```
Neto pomak prema onlineu: 11
```

Više ispitanika je prešlo s tiska na online nego obrnuto. McNemarov test govori je li ta asimetrija statistički značajna.

! Kad koristiti McNemarov test

McNemarov test koristite kad imate iste ispitanike mjerene dva puta na istoj binarnoj varijabli. Primjeri iz komunikologije uključuju preferenciju medija prije i poslije kampanje, stav prema brandu prije i poslije izlaganja reklami, izbor komunikacijskog kanala prije i poslije redizajna. Ključno je da su podaci upareni — isti ljudi, dva mjerenja.

11.15 Kompletna analiza: tri pitanja za upravu

Spojimo sve naučeno u kompletnu analizu. Istražujemo tri pitanja za upravu medijske kuće. Prvo, ovisi li tip medija o dobi? Drugo, razlikuju li se regije u digitalnim navikama? Treće, postoji li veza između obrazovanja i preferencije sadržaja?

```
# Pitanje 1: Dob x tip medija (detaljno, svih 6 tipova)
tab1 <- table(survey$age_group, survey$media_type)
test_q1 <- chisq.test(tab1)
v1 <- sqrt(test_q1$statistic / (sum(tab1) * (min(dim(tab1)) - 1)))

cat("=== PITANJE 1: Dob x Tip medija ===\n")
```

```
=== PITANJE 1: Dob x Tip medija ===
```

```
cat("chi2(", (nrow(tab1)-1)*(ncol(tab1)-1), ") = ", round(test_q1$statistic, 1),
    ", p < 0.001, V = ", round(v1, 2), "\n", sep = "")
```

```
chi2(15) = 233.6, p < 0.001, V = 0.31
```

```
cat("Interpretacija: Jaka veza. Mladi preferiraju streaming i portale,\n")
```

```
Interpretacija: Jaka veza. Mladi preferiraju streaming i portale,
```

```
cat("stariji TV i radio.\n\n")
```

```
stariji TV i radio.
```

```r
# Koji reziduali su najjaci?
rez_df <- as.data.frame(test_q1$residuals) |>
  as_tibble() |>
  rename(age = Var1, media = Var2, r = Freq) |>
  filter(abs(r) > 2) |>
  arrange(desc(abs(r)))

cat("Celije s |rezidual| > 2:\n")
```

```
Celije s |rezidual| > 2:
```

```r
rez_df |>
  mutate(r = round(r, 1), smjer = if_else(r > 0, "VISE nego ocekivano", "MANJE nego ocekiv
  print(n = 20)
```

```
# A tibble: 13 x 4
   age   media          r smjer
   <fct> <fct>      <dbl> <chr>
 1 18-29 streaming    6.4 VISE nego ocekivano
 2 60+   TV           5.9 VISE nego ocekivano
 3 60+   streaming   -5.2 MANJE nego ocekivano
 4 18-29 TV          -4.2 MANJE nego ocekivano
 5 60+   radio        4.1 VISE nego ocekivano
 6 30-44 podcast      3.8 VISE nego ocekivano
 7 30-44 TV          -3.8 MANJE nego ocekivano
 8 18-29 radio       -3.5 MANJE nego ocekivano
 9 60+   podcast     -3.3 MANJE nego ocekivano
10 60+   web_portal  -3.2 MANJE nego ocekivano
11 60+   tisak        3.1 VISE nego ocekivano
12 45-59 TV          2.5 VISE nego ocekivano
13 45-59 streaming   -2.5 MANJE nego ocekivano
```

```r
# Pitanje 2: Regija x tip medija (digitalni vs tradicionalni)
tab2 <- table(survey$region, survey$media_grupa)
test_q2 <- chisq.test(tab2)
v2 <- sqrt(test_q2$statistic / (sum(tab2) * (min(dim(tab2)) - 1)))

cat("=== PITANJE 2: Regija x Digitalni/Tradicionalni ===\n")
```

```
=== PITANJE 2: Regija x Digitalni/Tradicionalni ===
```

```
cat("chi2(", (nrow(tab2)-1)*(ncol(tab2)-1), ") = ", round(test_q2$statistic, 2),
    ", p = ", round(test_q2$p.value, 4), ", V = ", round(v2, 3), "\n", sep = "")
```

```
chi2(4) = 9.01, p = 0.0609, V = 0.106
```

```
cat("Interpretacija:", if_else(test_q2$p.value < 0.05,
    "Postoji znacajna razlika medju regijama.",
    "Nema znacajne razlike medju regijama."), "\n\n")
```

```
Interpretacija: Nema znacajne razlike medju regijama.
```

```
# Proporcije digitalnih po regiji
survey |>
  group_by(region) |>
  summarise(
    n = n(),
    udio_digitalni = round(mean(media_grupa == "digitalni") * 100, 1),
    .groups = "drop"
  ) |>
  arrange(desc(udio_digitalni))
```

```
# A tibble: 5 x 3
  region           n udio_digitalni
  <chr>        <int>          <dbl>
1 Sjeverozapad   137           65.7
2 Slavonija      165           61.8
3 Zagreb         258           55
4 Dalmacija      139           54
5 Primorje       101           49.5
```

```
# Pitanje 3: Obrazovanje x preferencija sadrzaja
tab3 <- table(survey$education, survey$content_preference)
chi3 <- chisq.test(tab3)
v3 <- sqrt(chi3$statistic / (sum(tab3) * (min(dim(tab3)) - 1)))

cat("=== PITANJE 3: Obrazovanje x Preferencija sadrzaja ===\n")
```

```
=== PITANJE 3: Obrazovanje x Preferencija sadrzaja ===
```

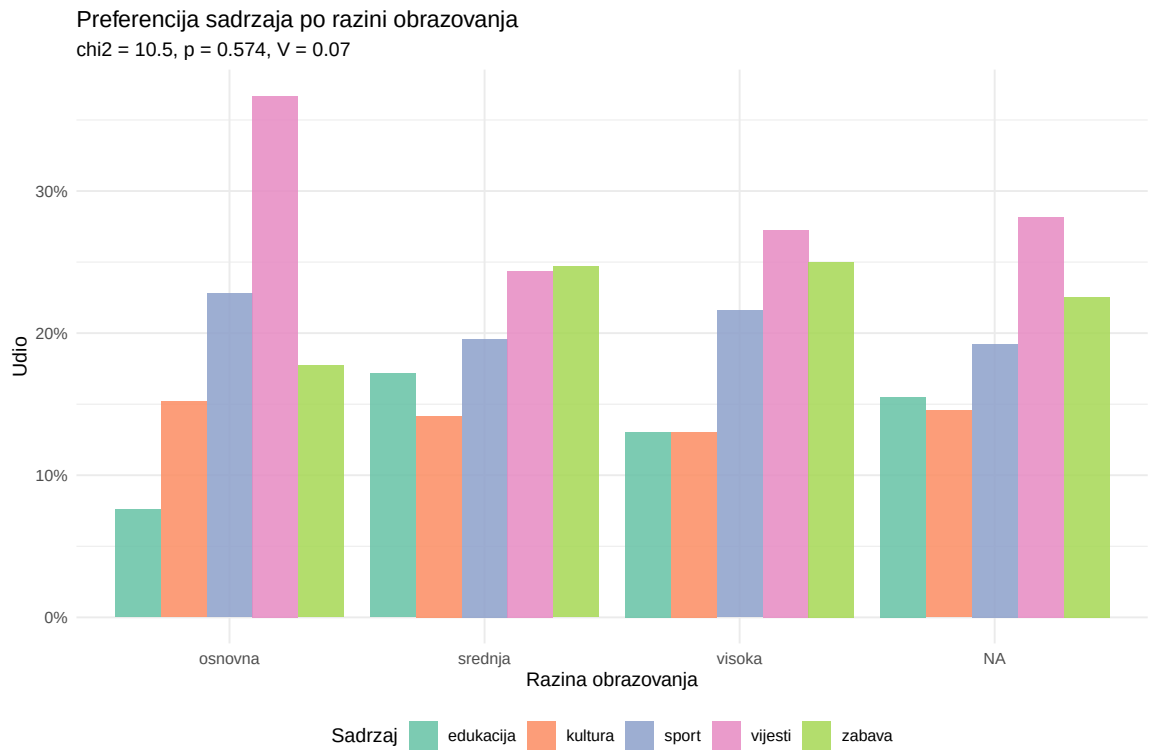
```
cat("chi2(", (nrow(tab3)-1)*(ncol(tab3)-1), ") = ", round(chi3$statistic, 2),
    ", p = ", round(chi3$p.value, 4), ", V = ", round(v3, 3), "\n", sep = "")
```

```
chi2(12) = 10.48, p = 0.5742, V = 0.066
```

```
cat("Interpretacija:", if_else(chi3$p.value < 0.05,
    "Postoji znacajna veza.",
    "Nema znacajne veze."), "\n")
```

```
Interpretacija: Nema znacajne veze.
```

```
# Vizualizacija
survey |>
  mutate(education = factor(education, levels = c("osnovna", "srednja", "visa", "visoka"))
  count(education, content_preference) |>
  group_by(education) |>
  mutate(udio = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = education, y = udio, fill = content_preference)) +
  geom_col(position = "dodge", alpha = 0.85) +
  scale_y_continuous(labels = scales::label_percent()) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Preferencija sadrzaja po razini obrazovanja",
    subtitle = paste0("chi2 = ", round(chi3$statistic, 1), ", p = ", round(chi3$p.value, 3
                      ", V = ", round(v3, 2)),
    x = "Razina obrazovanja",
    y = "Udio",
    fill = "Sadrzaj"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
# Sazetak svih testova
cat("================================================\n")
```

```
================================================
```

```
cat("   SAZETAK ANALIZE KATEGORICKIH PODATAKA\n")
```

```
   SAZETAK ANALIZE KATEGORICKIH PODATAKA
```

```
cat("================================================\n\n")
```

```
================================================
```

```
tibble(
  pitanje = c("Dob x Tip medija", "Regija x Dig./Trad.", "Obrazovanje x Sadrzaj"),
  chi2 = round(c(test_q1$statistic, test_q2$statistic, chi3$statistic), 1),
  df = c((nrow(tab1)-1)*(ncol(tab1)-1), (nrow(tab2)-1)*(ncol(tab2)-1), (nrow(tab3)-1)*(nco
  p = c(format(test_q1$p.value, scientific = TRUE, digits = 2),
        round(test_q2$p.value, 4), round(chi3$p.value, 4)),
  V = round(c(v1, v2, v3), 3),
  velicina = c(
```

```
    if_else(v1 > 0.29, "veliki", if_else(v1 > 0.17, "srednji", "mali")),
    if_else(v2 > 0.29, "veliki", if_else(v2 > 0.17, "srednji", "mali")),
    if_else(v3 > 0.29, "veliki", if_else(v3 > 0.17, "srednji", "mali"))
  )
)
```

```
# A tibble: 3 x 6
  pitanje                chi2    df p           V velicina
  <chr>                 <dbl> <dbl> <chr>   <dbl> <chr>
1 Dob x Tip medija      234.     15 2.9e-41 0.312 veliki
2 Regija x Dig./Trad.     9       4 0.0609  0.106 mali
3 Obrazovanje x Sadrzaj  10.5    12 0.5742  0.066 mali
```

Analiza otkriva jasan obrazac. Veza između dobi i tipa medija je najjača i najvažnija — generacijski jaz u medijskim navikama je velik i statistički nedvojben. Regionalne razlike u digitalnim vs tradicionalnim medijima su znatno manje. Veza obrazovanja i preferencije sadržaja ovisi o uzorku. Za upravu je ključna poruka — budućnost je u digitalnim platformama, ali televizija i radio još uvijek imaju ogromnu publiku među starijim generacijama.

11.16 Pet pogrešaka koje ne smijete napraviti

Hi-kvadrat testovi su jednostavni za provesti, ali iznenađujuće lako ih je krivo primijeniti ili interpretirati. Evo pet najčešćih pogrešaka.

Unošenje postotaka umjesto frekvencija. Hi-kvadrat test zahtijeva apsolutne frekvencije (brojeve), ne postotke ili proporcije. Ako unesete `chisq.test(c(0.30, 0.20, 0.50))`, R će misliti da ukupno imate jedno opažanje i dat će besmisleni rezultat.

```
# KRIVO: postoci
cat("KRIVO (postoci):\n")
```

```
KRIVO (postoci):
```

```
chisq.test(c(0.30, 0.20, 0.50))
```

```

    Chi-squared test for given probabilities

data:  c(0.3, 0.2, 0.5)
X-squared = 0.14, df = 2, p-value = 0.9324
```

```r
cat("\nISPRAVNO (frekvencije):\n")
```

```
ISPRAVNO (frekvencije):
```

```r
chisq.test(c(30, 20, 50))
```

```
    Chi-squared test for given probabilities

data:  c(30, 20, 50)
X-squared = 14, df = 2, p-value = 0.0009119
```

Pretjerana granularnost. Tablica 10×8 s ukupno 100 opažanja imat će mnogo ćelija s malim frekvencijama. Bolje je spojiti kategorije u smislene grupe i imati manje ali punije ćelije.

Kauzalna interpretacija. Hi-kvadrat test detektira asocijaciju, ne kauzalnost. Činjenica da su dob i medijski tip povezani ne znači da dob *uzrokuje* preferenciju — možda je posrijedi obrazovanje, socioekonomski status ili kohorta efekt. Za kauzalne zaključke trebate eksperimentalni dizajn ili napredne statističke metode.

Zaboravljanje veličine učinka. Kao i kod t-testa, p-vrijednost ovisi o veličini uzorka. S dovoljno velikim uzorkom, čak i trivijalna veza postaje "statistički značajna." Uvijek izvijestite Cramérovo V uz ^{2} i p — jer Cramérovo V govori koliko je veza praktično jaka, neovisno o n.

Višestruko testiranje bez korekcije. Ako testirate 10 parova varijabli bez korekcije, šansa da bar jedan test bude lažno pozitivan je oko 40%. Koristite Benjamini-Hochberg (BH) korekciju kad testirate više parova.

```r
# Testiramo sve parove kategorickih varijabli
parovi <- list(
  c("age_group", "media_type"),
  c("age_group", "content_preference"),
  c("age_group", "media_grupa"),
  c("gender", "media_type"),
  c("gender", "content_preference"),
  c("education", "media_type"),
  c("education", "content_preference"),
  c("region", "media_grupa")
)

multi_chi <- map_df(parovi, \(par) {
  tab <- table(survey[[par[1]]], survey[[par[2]]])
```

```
  test <- chisq.test(tab)
  tibble(
    var1 = par[1],
    var2 = par[2],
    chi2 = round(test$statistic, 1),
    p = test$p.value
  )
}) |>
  mutate(
    p_adj = p.adjust(p, method = "BH"),
    znacajno_orig = p < 0.05,
    znacajno_adj = p_adj < 0.05
  ) |>
  arrange(p)

multi_chi |>
  mutate(p = format(p, scientific = TRUE, digits = 2),
         p_adj = format(p_adj, scientific = TRUE, digits = 2))
```

```
# A tibble: 8 x 7
  var1      var2                chi2 p       p_adj   znacajno_orig znacajno_adj
  <chr>     <chr>              <dbl> <chr>   <chr>   <lgl>         <lgl>
1 age_group media_grupa        199.  7.3e-43 5.9e-42 TRUE          TRUE
2 age_group media_type         234.  2.9e-41 1.2e-40 TRUE          TRUE
3 age_group content_preference  59.7 2.5e-08 6.8e-08 TRUE          TRUE
4 education media_type          25.5 4.4e-02 8.8e-02 TRUE          FALSE
5 region    media_grupa          9   6.1e-02 9.8e-02 FALSE         FALSE
6 education content_preference  10.5 5.7e-01 7.2e-01 FALSE         FALSE
7 gender    media_type           3.5 6.3e-01 7.2e-01 FALSE         FALSE
8 gender    content_preference   0.4 9.8e-01 9.8e-01 FALSE         FALSE
```

Nakon BH korekcije, neki marginalno značajni rezultati mogu nestati. To je cijena korektnog pristupa — ali bolje je imati manje rezultata u koje možete vjerovati nego više rezultata koji su možda lažni.

11.17 Funkcija koja obavlja sve za vas

U praksi ćete hi-kvadrat analizu ponavljati za mnogo parova varijabli. Umjesto da svaki put ručno prolazite sve korake, napišimo funkciju koja automatizira cijeli postupak — provjeri pretpostavke, odabere odgovarajući test, izračuna veličinu učinka i ispiše izvještaj.

```r
chi_izvjestaj <- function(data, var1, var2) {
  tab <- table(data[[var1]], data[[var2]])
  test <- chisq.test(tab)
  v <- sqrt(test$statistic / (sum(tab) * (min(dim(tab)) - 1)))
  min_exp <- min(test$expected)

  cat("======================================\n")
  cat(var1, "x", var2, "\n")
  cat("======================================\n")
  cat("Dimenzije tablice:", nrow(tab), "x", ncol(tab), "\n")
  cat("Najm. ocekivana frekvencija:", round(min_exp, 1), "\n")

  if (min_exp < 5) {
    cat("Ocekivane frekvencije < 5. Koristim Fisherov test.\n")
    fisher <- fisher.test(tab, simulate.p.value = TRUE)
    cat("P-vrijednost (Fisher):", round(fisher$p.value, 4), "\n")
  } else {
    cat("chi2(", (nrow(tab)-1)*(ncol(tab)-1), ") = ",
        round(test$statistic, 2), "\n", sep = "")
    cat("P-vrijednost:", format(test$p.value, scientific = TRUE, digits = 3), "\n")
  }

  cat("Cramerovo V:", round(v, 3), "\n")
  cat("Odluka:", if_else(test$p.value < 0.05,
    "Postoji statisticki znacajna veza.",
    "Nema statisticki znacajne veze."), "\n\n")

  invisible(list(table = tab, test = test, V = v))
}

# Primjeri koristenja
chi_izvjestaj(survey, "age_group", "media_type")
```

```
======================================
age_group x media_type
======================================
Dimenzije tablice: 4 x 6
Najm. ocekivana frekvencija: 10.1
chi2(15) = 233.59
P-vrijednost: 2.93e-41
Cramerovo V: 0.312
Odluka: Postoji statisticki znacajna veza.
```

```
chi_izvjestaj(survey, "gender", "content_preference")
```

```
====================================
gender x content_preference
====================================
Dimenzije tablice: 2 x 5
Najm. ocekivana frekvencija: 54.2
chi2(4) = 0.41
P-vrijednost: 9.82e-01
Cramerovo V: 0.023
Odluka: Nema statisticki znacajne veze.
```

Ova funkcija automatski provjerava pretpostavke, odabire odgovarajući test i računa veličinu učinka. Možete je koristiti u svim budućim analizama kategoričkih podataka — kopijte je u svoje skripte i prilagodite prema potrebi.

11.18 Pregled svih testova za kategoričke podatke

```
tribble(
  ~test, ~situacija, ~R_kod,
  "chi2 goodness-of-fit", "Jedna varijabla vs ocekivana distribucija", "chisq.test(frekven
  "chi2 nezavisnosti", "Veza dviju kategorickih varijabli", "chisq.test(table(x, y))",
  "Fisherov egzaktni", "Male ocekivane frekvencije ili mali n", "fisher.test(table(x, y))"
  "McNemarov test", "Uparene kategoricke varijable", "mcnemar.test(table(prije, poslije))"
)
```

```
# A tibble: 4 x 3
  test                 situacija                                  R_kod
  <chr>                <chr>                                      <chr>
1 chi2 goodness-of-fit Jedna varijabla vs ocekivana distribucija chisq.test(fre~
2 chi2 nezavisnosti    Veza dviju kategorickih varijabli          chisq.test(tab~
3 Fisherov egzaktni    Male ocekivane frekvencije ili mali n      fisher.test(ta~
4 McNemarov test       Uparene kategoricke varijable              mcnemar.test(t~
```

! Ključni zaključci

Hi-kvadrat testovi su za kategoričke varijable. Test za dobrotu prilagodbe uspoređuje distribuciju jedne varijable s očekivanom. Test nezavisnosti testira postoji li veza između dviju varijabli.
^{2} statistika mjeri udaljenost od očekivanog. Formula je $^2 = \Sigma(O - E)^2 / E$. Veći ^{2} znači jači dokaz protiv H . Stupnjevi slobode za test nezavisnosti su $(r - 1)(c - 1)$.

Očekivane frekvencije su ključ za razumijevanje. Pod H , E = (redak total × stupac total) / ukupno. Pretpostavka — sve E 5. Kad to nije zadovoljeno, koristite Fisherov egzaktni test.
Reziduali govore *gdje* je veza najjača. Ukupni ² kaže da veza postoji. Standardizirani reziduali s |r| > 2 identificiraju specifične ćelije. Uvijek ih izvijestite — jer urednica ne želi znati samo "postoji veza", nego *kakva* je.
Cramérovo V mjeri jačinu veze. Raspon od 0 do 1. Za k = 4: V 0.06 mali, V 0.17 srednji, V 0.29 veliki učinak. Uvijek ga izvijestite uz ² i p.
Fisherov test kad su ćelije premale. Točan test bez aproksimacije. Za 2×2 tablice automatski daje odds ratio. U praksi ga možete koristiti uvijek — za velike uzorke daje iste rezultate kao ².
Spajanje kategorija je legitimno i korisno. Digitalni vs tradicionalni mediji je informativnije od šest odvojenih kategorija. Smisleno kolapsiranje povećava frekvencije i preglednost.
Simpsonov paradoks upozorava na opasnost agregiranja. Ukupni rezultati mogu biti obmanjujući. Uvijek provjerite rezultate stratificirane po relevantnoj trećoj varijabli.
McNemarov test za uparene kategoričke podatke. Isti ispitanici, dva mjerenja, binarna varijabla. Ekvivalent uparenom t-testu u kategoričkom svijetu.
Asocijacija nije kauzalnost. Veza dobi i medijskog tipa ne znači da dob uzrokuje preferenciju. Za kauzalne zaključke trebate eksperimentalni dizajn.
Višestruko testiranje zahtijeva korekciju. BH ili Bonferroni korekcija kad testirate mnogo parova varijabli. Bolje je imati manje pouzdanih rezultata nego više upitnih.

11.19 Zadaci za pripremu

1. Učitajte `media_survey_chi2.csv`. Testirajte postoji li veza između regije i preferencije sadržaja (`content_preference`). Izračunajte Cramérovo V i interpretirajte ga. Koji reziduali su najjači?
2. Kreirajte novu varijablu `zadovoljni` (satisfaction >= 4 = "da", inače "ne"). Testirajte postoji li veza između `media_grupa` (digitalni/tradicionalni) i `zadovoljni` pomoću Fisherovog egzaktnog testa. Interpretirajte odds ratio.
3. Napišite funkciju `chi_vizualizacija(data, var1, var2)` koja prima podatke i imena dviju varijabli te automatski crta grupani barplot s rezultatima testa u podnaslovu.

11.20 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 12 (Categorical Data Analysis). Besplatno dostupno na learningstatisticswithr.com. Pokriva hi-kvadrat testove s R kodom.

Preporučeno

Agresti, A. (2018). *An Introduction to Categorical Data Analysis* (3rd edition). Wiley. Poglavlja 1-3. Referentni udžbenik za kategoričke podatke.

Wickham, H. & Grolemund, G. (2017). *R for Data Science*. O'Reilly. Besplatno na r4ds.had.co.nz. Poglavlja o faktorima i vizualizaciji kategoričkih podataka.

11.21 Pojmovnik

Pojam	Objašnjenje
Kategorička varijabla	Varijabla čije su vrijednosti kategorije (npr. spol, regija, tip medija). Nema smisleni numerički poredak (nominalna) ili ima (ordinalna).
Kontingencijska tablica	Tablica frekvencija za sve kombinacije dviju kategoričkih varijabli. Temelj za test nezavisnosti.
Hi-kvadrat statistika	Mjera ukupnog odstupanja opaženih od očekivanih frekvencija.
Goodness-of-fit test	Testira odgovara li distribucija jedne varijable očekivanoj distribuciji. df = k minus 1.
Test nezavisnosti	Testira postoji li veza između dviju kategoričkih varijabli. df = (r minus 1)(c minus 1).
Očekivana frekvencija	Frekvencija pod H0. Za test nezavisnosti: E = (redak total puta stupac total) / ukupno.
Standardizirani rezidual	(O minus E) / korijen(E). Doprinos svake ćelije ukupnom chi2. Značajno ako
Cramérovo V	Mjera veličine učinka za hi-kvadrat test. V = korijen(chi2/(n(k minus 1))). Raspon 0 do 1.
Fisherov egzaktni test	Točan test za male uzorke ili male očekivane frekvencije. Ne koristi chi2 aproksimaciju.
Odds ratio (omjer šansi)	Mjera asocijacije za 2x2 tablice. OR = 1 znači nema veze. OR > 1 ili < 1 znači veza postoji.
Yatesova korekcija	Korekcija kontinuiteta za 2x2 tablice. R je primjenjuje po defaultu u `chisq.test()`.
McNemarov test	Test za uparene kategoričke podatke (isti ispitanici, dva mjerenja).

Pojam	Objašnjenje
Simpsonov paradoks	Smjer veze se promijeni kad kontroliramo treću varijablu. Agregirani rezultati obmanjuju.
Stratificirana analiza	Provođenje testa odvojeno za podgrupe treće varijable. Otkriva Simpsonov paradoks.
Spajanje kategorija	Kolapsiranje rijetkih kategorija u šire grupe. Povećava očekivane frekvencije i preglednost.
`chisq.test()`	R funkcija za hi-kvadrat test. Prima vektor frekvencija (gof) ili kontingencijsku tablicu.
`fisher.test()`	R funkcija za Fisherov egzaktni test. Prima kontingencijsku tablicu.
`mcnemar.test()`	R funkcija za McNemarov test. Prima 2x2 tablicu uparenih podataka.
`table()`	R funkcija za kreiranje kontingencijske tablice. `table(x, y)` za dvije varijable.
`prop.table()`	Pretvara frekvencije u proporcije. `margin = 1` za retke, `margin = 2` za stupce.
`p.adjust()`	Korekcija p-vrijednosti za višestruko testiranje. `method = "BH"` je preporučen.

12 Tjedan 11: Usporedba prosjeka t-testovima

Kad pretpostavke drže i kad ne drže

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete:

1. Odabrati odgovarajući t-test (jednouzorački, nezavisni, upareni) za zadano istraživačko pitanje.
2. Provjeriti pretpostavke t-testa (normalnost, homogenost varijance) i znati što učiniti kad su narušene.
3. Primijeniti Shapiro-Wilkov test i QQ plot za provjeru normalnosti.
4. Provesti Wilcoxonov test kao neparametrijsku alternativu kad normalnost nije zadovoljena.
5. Izračunati i interpretirati Cohenov d za sve tri vrste t-testa.
6. Pravilno izvijestiti rezultate t-testa u APA formatu.
7. Provesti kompletnu analizu usporedbe dvaju uvjeta na stvarnim podacima.

12.1 Redizajn koji je podijelio redakciju

Zamislite da radite kao istraživačica u uredništvu jednog web portala. Portal objavljuje članke o politici, zdravlju, tehnologiji, sportu i kulturi — uglavnom tekstualne, s ponekom fotografijom za vizualni odmor. Uredništvo razmišlja o velikom redizajnu. Žele dodati infografike, podatkovne vizualizacije, ilustracije i interaktivne elemente. Ideja zvuči privlačno, ali glavni urednik nije uvjeren. "Vizuali koštaju," kaže. "Trebam dokaze da stvarno poboljšavaju čitateljsko iskustvo."

I tako dobijete zadatak — osmisliti eksperiment koji će odgovoriti na pitanje utječu li vizualni elementi na četiri ključna ishoda — vrijeme čitanja, razumijevanje sadržaja, namjeru dijeljenja i percipiranu vjerodostojnost. Odabrali ste 120 članaka i svaki prezentirali ispitanicima u dva uvjeta: jednom s vizualima, jednom bez. Budući da je svaki članak testiran u oba uvjeta,

radi se o within-subjects dizajnu — a statistički alat koji vam treba za takvu usporedbu zove se upareni t-test.

Na predavanju o testiranju hipoteza naučili smo logiku koja stoji iza t-testa — postavljamo nultu hipotezu, izračunavamo testnu statistiku, gledamo p-vrijednost i donosimo odluku. Ovo predavanje se bavi onom drugom polovicom posla — onom koja se u udžbenicima često prebrzo preskoči. Kako zapravo provjeriti jesu li pretpostavke testa zadovoljene? Što učiniti kad nisu? Kako odabrati pravi test za pravi dizajn? I kako napisati rezultate tako da ih kolege i recenzenti mogu razumjeti?

Krenimo od podataka.

```
articles <- read_csv("../resources/datasets/article_visuals.csv")
glimpse(articles)
```

```
Rows: 120
Columns: 12
$ article_id                <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1~
$ category                  <chr> "tehnologija", "politika", "tehnologija", "t~
$ length_category           <chr> "srednji", "kratki", "srednji", "kratki", "s~
$ word_count                <dbl> 589, 307, 805, 447, 626, 827, 802, 302, 578,~
$ reading_time_no_visual    <dbl> 4.9, 2.3, 3.4, 1.6, 3.3, 6.4, 2.7, 0.7, 3.4,~
$ reading_time_with_visual  <dbl> 5.8, 3.0, 3.9, 1.8, 4.4, 7.6, 3.2, 0.9, 3.9,~
$ comprehension_no_visual   <dbl> 5, 6, 6, 6, 5, 8, 6, 9, 6, 5, 5, 7, 6, 10, 5~
$ comprehension_with_visual <dbl> 6, 7, 7, 6, 7, 9, 6, 9, 6, 6, 5, 8, 7, 10, 6~
$ sharing_no_visual         <dbl> 2, 3, 3, 4, 3, 2, 3, 3, 3, 2, 3, 3, 2, 3, 3,~
$ sharing_with_visual       <dbl> 3, 5, 2, 4, 3, 2, 4, 4, 4, 3, 4, 3, 2, 3, 3,~
$ credibility_no_visual     <dbl> 5, 5, 2, 2, 5, 6, 3, 3, 5, 4, 5, 3, 3, 2, 5,~
$ credibility_with_visual   <dbl> 4, 5, 4, 3, 5, 7, 3, 2, 6, 4, 6, 4, 3, 2, 5,~
```

```
articles |>
  summarise(
    n = n(),
    M_time_no = round(mean(reading_time_no_visual), 2),
    M_time_with = round(mean(reading_time_with_visual), 2),
    M_comp_no = round(mean(comprehension_no_visual), 2),
    M_comp_with = round(mean(comprehension_with_visual), 2),
    M_share_no = round(mean(sharing_no_visual), 2),
    M_share_with = round(mean(sharing_with_visual), 2)
  )
```

```
# A tibble: 1 x 7
      n M_time_no M_time_with M_comp_no M_comp_with M_share_no M_share_with
  <int>     <dbl>       <dbl>     <dbl>       <dbl>      <dbl>        <dbl>
1   120      3.43        4.02      5.94        6.74       2.48         3.15
```

Na prvi pogled, brojke govore u prilog vizualima — članci s vizualnim elementima imaju duže vrijeme čitanja, bolje razumijevanje i višu namjeru dijeljenja. Ali "na prvi pogled" nije dovoljno za istraživačicu koja zna što radi. Prije nego što izvučemo ikakve zaključke, moramo provjeriti jesu li pretpostavke statističkog testa koje namjeravamo koristiti uopće zadovoljene.

12.2 Tri vrste t-testa

Prije nego uđemo u analizu, vrijedi se podsjetiti da t-test nije jedan jedini postupak nego obitelj od tri testa, a svaki odgovara na drugačije pitanje.

```
tribble(
  ~test, ~pitanje, ~R_kod,
  "Jednouzorački", "Razlikuje li se prosjek od poznate vrijednosti?", "t.test(x, mu = vrij
  "Nezavisni (dvouzorački)", "Razlikuju li se prosjeci dviju nezavisnih grupa?", "t.test(x
  "Upareni", "Razlikuje li se prosjek u dva uvjeta za iste jedinice?", "t.test(x, y, paire
)
```

```
# A tibble: 3 x 3
  test                   pitanje                                          R_kod
  <chr>                  <chr>                                            <chr>
1 Jednouzorački          Razlikuje li se prosjek od poznate vrijednosti?  t.te~
2 Nezavisni (dvouzorački) Razlikuju li se prosjeci dviju nezavisnih grupa? t.te~
3 Upareni                Razlikuje li se prosjek u dva uvjeta za iste je~ t.te~
```

Jednouzorački t-test koristi se kad imate jednu skupinu i želite provjeriti razlikuje li se njezin prosjek od neke poznate ili pretpostavljene vrijednosti — na primjer, je li prosječna ocjena razumijevanja vaših članaka različita od nacionalnog prosjeka od 6.0 bodova. Nezavisni t-test uspoređuje dvije odvojene grupe koje nemaju nikakvu vezu jedna s drugom — recimo, čitatelje koji su vidjeli vizuale i čitatelje koji nisu, ali to su različiti ljudi. Upareni t-test koristi se kad iste jedinice mjerite dva puta, u dva različita uvjeta — a to je upravo naš slučaj, jer su isti članci prezentirani i s vizualima i bez njih.

Za naše podatke, dakle, koristimo upareni t-test. Ali prije nego ga pokrenemo, moramo se uvjeriti da su pretpostavke tog testa razumno zadovoljene.

12.3 Pretpostavke koje morate provjeriti

Svaki statistički test dolazi s pretpostavkama — uvjetima koji moraju biti barem približno zadovoljeni da bismo rezultatima mogli vjerovati. T-test nije iznimka.

Sve tri varijante t-testa dijele tri temeljne pretpostavke. Kao prvo, podaci moraju biti barem na intervalnoj skali, što znači da razlike između vrijednosti imaju smisla (razlika između

3 i 5 je ista kao razlika između 7 i 9). Kao drugo, opažanja moraju biti nezavisna jedno od drugoga. U uparenom testu to znači da su parovi nezavisni, iako mjerenja unutar para naravno nisu. Kao treće, distribucija mora biti približno normalna — ili uzorak dovoljno velik da centralni granični teorem kompenzira.

Nezavisni t-test ima jednu dodatnu pretpostavku — varijance dviju grupa trebale bi biti približno jednake. To je pretpostavka klasičnog Studentovog t-testa. Dobra vijest je da postoji Welchova korekcija koja tu pretpostavku ne zahtijeva, i upravo je ona default u R-u. O tome ćemo više za koji odlomak.

Od svih ovih pretpostavki, ona koju ćete najčešće morati aktivno provjeravati je normalnost. A kod uparenog t-testa, pazite na važan detalj — ne provjeravate normalnost pojedinačnih mjerenja, nego normalnost *razlika* između dvaju uvjeta. Zvuči kao sitnica, ali ta sitnica može potpuno promijeniti zaključak.

12.4 Provjera normalnosti

12.4.1 Vizualna provjera: histogram i QQ plot

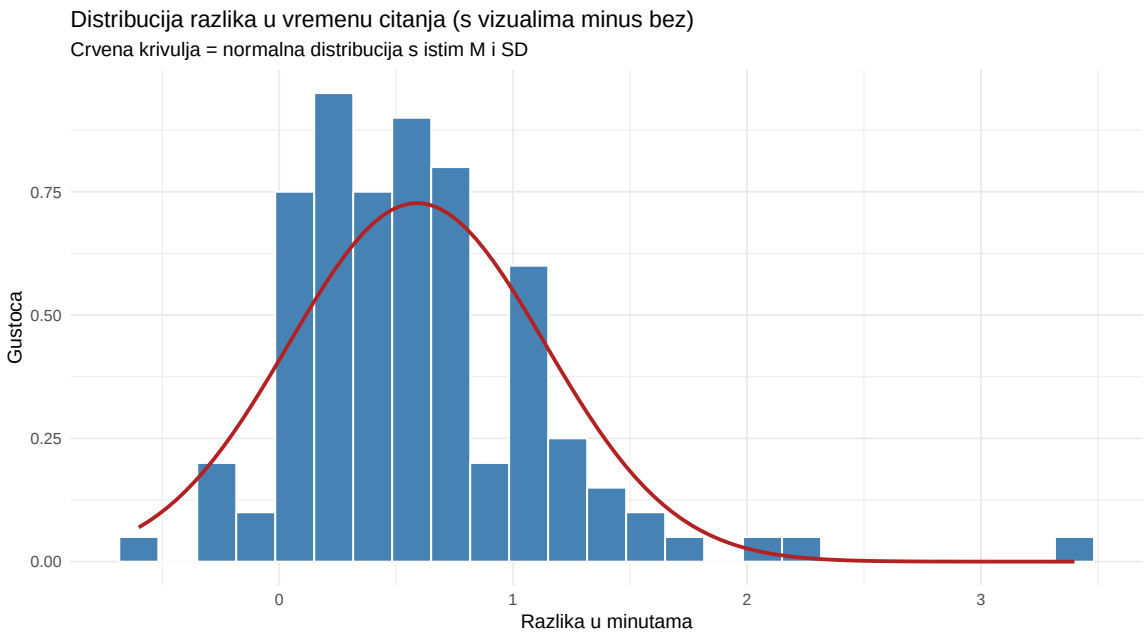
Najprirodniji prvi korak je pogledati podatke. Izračunajmo razlike između dvaju uvjeta za sva četiri ishoda.

```
# Za upareni test: provjera normalnosti RAZLIKA
articles <- articles |>
  mutate(diff_time = reading_time_with_visual - reading_time_no_visual,
         diff_comp = comprehension_with_visual - comprehension_no_visual,
         diff_share = sharing_with_visual - sharing_no_visual,
         diff_cred = credibility_with_visual - credibility_no_visual)
```

Počnimo s histogramom razlika u vremenu čitanja. Ako je distribucija približno normalna, histogram bi trebao imati zvonoliki oblik, s većinom vrijednosti okupljenim oko sredine i simetričnim repovima.

```
articles |>
  ggplot(aes(x = diff_time)) +
  geom_histogram(aes(y = after_stat(density)), fill = "steelblue", color = "white", bins =
  stat_function(fun = dnorm,
                args = list(mean = mean(articles$diff_time), sd = sd(articles$diff_time)),
                color = "firebrick", linewidth = 1) +
  labs(
    title = "Distribucija razlika u vremenu citanja (s vizualima minus bez)",
    subtitle = "Crvena krivulja = normalna distribucija s istim M i SD",
    x = "Razlika u minutama",
    y = "Gustoca"
```

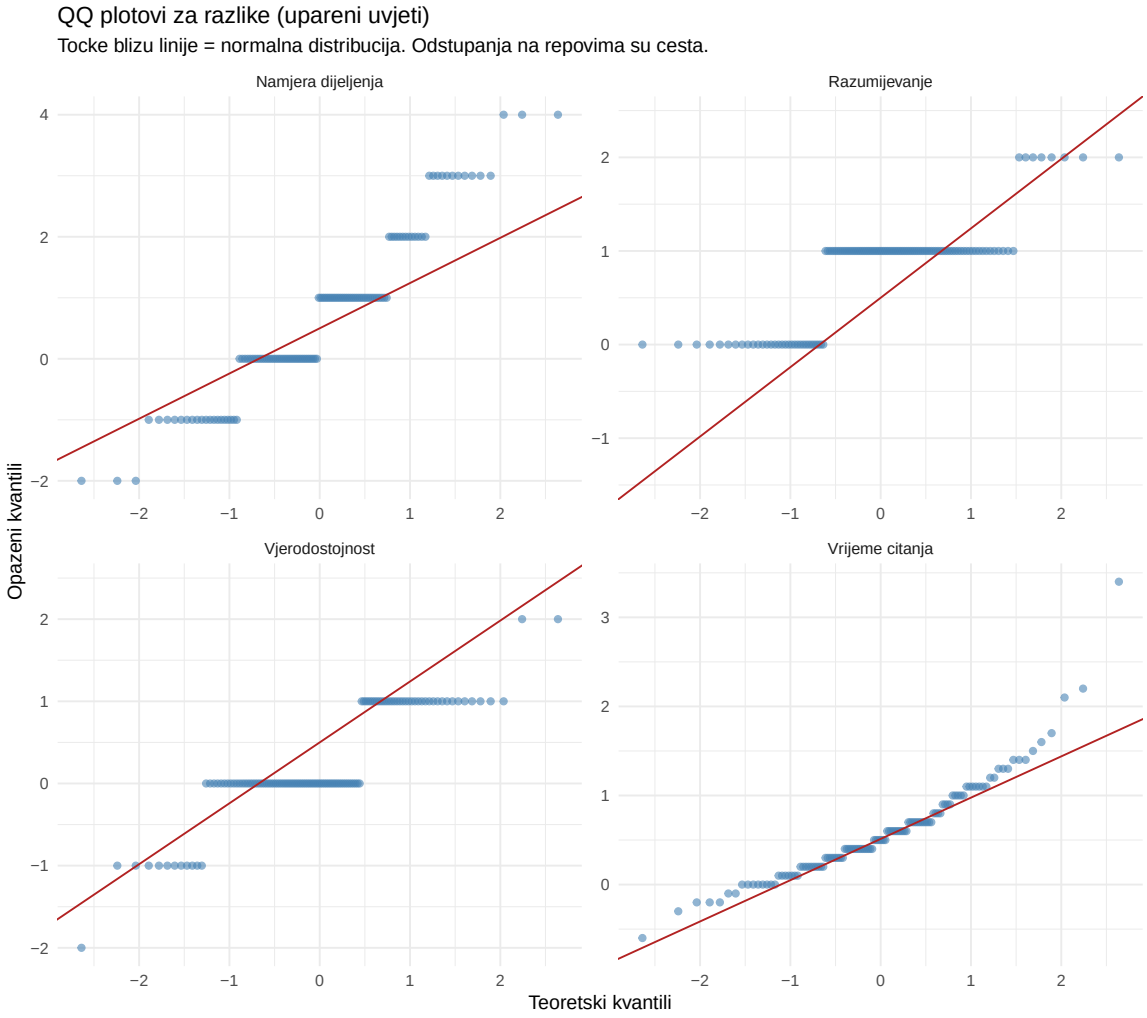
```
  ) +
  theme_minimal()
```



Histogram je koristan, ali još informativniji dijagnostički alat je QQ plot (quantile-quantile plot). Ideja QQ plota je jednostavna — on uspoređuje kvantile vaših podataka s kvantilima savršene normalne distribucije. Ako su vaši podaci normalno distribuirani, točke će ležati uz dijagonalnu liniju. Što više odstupaju, to je distribucija manje normalna.

```
# QQ plotovi za sve cetiri razlike
articles |>
  select(starts_with("diff_")) |>
  pivot_longer(everything(), names_to = "varijabla", values_to = "razlika") |>
  mutate(varijabla = case_when(
    varijabla == "diff_time" ~ "Vrijeme citanja",
    varijabla == "diff_comp" ~ "Razumijevanje",
    varijabla == "diff_share" ~ "Namjera dijeljenja",
    varijabla == "diff_cred" ~ "Vjerodostojnost"
  )) |>
  ggplot(aes(sample = razlika)) +
  stat_qq(color = "steelblue", alpha = 0.6) +
  stat_qq_line(color = "firebrick") +
  facet_wrap(~varijabla, scales = "free") +
  labs(
    title = "QQ plotovi za razlike (upareni uvjeti)",
    subtitle = "Tocke blizu linije = normalna distribucija. Odstupanja na repovima su cest
    x = "Teoretski kvantili",
```

```
    y = "Opazeni kvantili"
  ) +
  theme_minimal()
```



Pogledajte četiri panela. Razlike u vremenu čitanja i razumijevanju lijepo prate dijagonalu — distribucija je razumno normalna. Razlike u namjeri dijeljenja i vjerodostojnosti pokazuju stepeničast uzorak, što je očekivano — te varijable su mjerene Likert skalom (cjelobrojne vrijednosti od 1 do 5 ili 1 do 7), pa ne mogu biti savršeno glatke. Blaga odstupanja na repovima su prihvatljiva, osobito kad je uzorak veći od 30.

12.4.2 Shapiro-Wilkov test

Vizualna procjena je korisna, ali ponekad želite i formalni test. Shapiro-Wilkov test je najčešće korišteni test normalnosti. Njegova nulta hipoteza kaže da su podaci normalno distribuirani, pa ako je $p < 0.05$, zaključujemo da normalnost nije zadovoljena.

```r
tibble(
  varijabla = c("Vrijeme citanja", "Razumijevanje", "Namjera dijeljenja", "Vjerodostojnost
  W = c(
    shapiro.test(articles$diff_time)$statistic,
    shapiro.test(articles$diff_comp)$statistic,
    shapiro.test(articles$diff_share)$statistic,
    shapiro.test(articles$diff_cred)$statistic
  ),
  p = c(
    shapiro.test(articles$diff_time)$p.value,
    shapiro.test(articles$diff_comp)$p.value,
    shapiro.test(articles$diff_share)$p.value,
    shapiro.test(articles$diff_cred)$p.value
  )
) |>
  mutate(
    W = round(W, 4),
    p = round(p, 4),
    normalno = if_else(p >= 0.05, "Da", "Ne")
  )
```

```
# A tibble: 4 x 4
  varijabla              W     p normalno
  <chr>              <dbl> <dbl> <chr>
1 Vrijeme citanja    0.908     0 Ne
2 Razumijevanje      0.715     0 Ne
3 Namjera dijeljenja 0.93      0 Ne
4 Vjerodostojnost    0.810     0 Ne
```

Razlike u vremenu čitanja prolaze test normalnosti ($p > 0.05$), što znači da nemamo dovoljno dokaza da odbacimo pretpostavku o normalnoj distribuciji. Razlike u Likert varijablama možda ne prolaze jer su diskretne. Za te varijable ćemo razmotriti neparametrijske alternative.

! Paradoks Shapiro-Wilkovog testa

Shapiro-Wilkov test ima isti paradoks kao i svaki statistički test — njegova osjetljivost ovisi o veličini uzorka. S velikim uzorkom (recimo $n = 500$) detektirat će i sasvim trivijalna odstupanja od normalnosti koja nemaju nikakav praktični značaj. S malim uzorkom ($n = 15$) propustit će i prilično grube deformacije distribucije.
Zato je vizualna procjena putem QQ plota jednako važna kao formalni test. Praktično pravilo je sljedeće — ako QQ plot izgleda razumno i uzorak ima više od 30 opažanja, t-test je dovoljno robustan čak i za umjerena odstupanja od normalnosti jer centralni granični teorem osigurava da će distribucija prosjeka biti približno normalna bez obzira

na oblik izvorne distribucije.

12.5 Provjera homogenosti varijance

Kad koristite nezavisni t-test (usporedba dviju odvojenih grupa), klasična Studentova verzija pretpostavlja da obje grupe imaju približno jednake varijance. Pogledajmo što se dogodi kad ta pretpostavka nije zadovoljena — na simuliranim podacima gdje jednoj grupi namjerno zadamo mnogo veću varijabilnost.

```
# Demonstracija na simuliranim podacima (nezavisne grupe)
set.seed(42)
grupa_a <- rnorm(50, mean = 5, sd = 1.0)
grupa_b <- rnorm(50, mean = 6, sd = 2.5)  # razlicita varijanca!

cat("SD grupa A:", round(sd(grupa_a), 2), "\n")
```

```
SD grupa A: 1.15
```

```
cat("SD grupa B:", round(sd(grupa_b), 2), "\n")
```

```
SD grupa B: 2.31
```

```
cat("Omjer varijanci:", round(var(grupa_b) / var(grupa_a), 2), "\n")
```

```
Omjer varijanci: 4.03
```

Grupa B ima dva i pol puta veću standardnu devijaciju od grupe A. Usporedimo što se dogodi kad pokrenemo Studentov t-test (koji pretpostavlja jednake varijance) nasuprot Welchovom t-testu (koji tu pretpostavku ne zahtijeva).

```
# Studentov t-test (pretpostavlja jednake varijance)
student_rez <- t.test(grupa_a, grupa_b, var.equal = TRUE)

# Welchov t-test (ne pretpostavlja jednake varijance, DEFAULT)
welch_rez <- t.test(grupa_a, grupa_b, var.equal = FALSE)

tibble(
  test = c("Student (var.equal = TRUE)", "Welch (var.equal = FALSE, DEFAULT)"),
  t = round(c(student_rez$statistic, welch_rez$statistic), 3),
  df = round(c(student_rez$parameter, welch_rez$parameter), 1),
  p = round(c(student_rez$p.value, welch_rez$p.value), 4)
)
```

```
# A tibble: 2 x 4
  test                                     t    df      p
  <chr>                                <dbl> <dbl>  <dbl>
1 Student (var.equal = TRUE)          -3.52  98   0.0006
2 Welch (var.equal = FALSE, DEFAULT) -3.52  71.9 0.0007
```

Primijetite jednu zanimljivu stvar — Welchov test ima neokrugle stupnjeve slobode. To je zato što ih prilagođava za razliku u varijancama. Kad su varijance jednake, oba testa daju praktički identične rezultate. Kad su varijance različite, Welchov test je točniji, a Studentov test može dati lažno pozitivne ili lažno negativne rezultate.

> **Zaboravite na Studentov t-test**
>
> Evo savjeta koji će vam pojednostaviti život — koristite Welchov t-test uvijek. Ne morate uopće provjeravati homogenost varijance jer Welchov test radi jednako dobro kad su varijance jednake i bolje kad su različite. On je default u R-u (`var.equal = FALSE`) iz dobrog razloga.
> Studentov t-test s `var.equal = TRUE` koristite samo ako imate specifičan razlog — na primjer, kad trebate reproducirati rezultate iz objavljenog rada koji je koristio Studentov test.

12.6 Upareni t-test: utječu li vizuali na vrijeme čitanja?

Pretpostavke smo provjerili, razlike izgledaju razumno normalno, uzorak je dovoljno velik. Vrijeme je da pokrenemo test i odgovorimo na pitanje koje je pokrenulo cijelu analizu.

```
# Upareni t-test: vrijeme citanja s vizualima vs bez
time_test <- t.test(
  articles$reading_time_with_visual,
  articles$reading_time_no_visual,
  paired = TRUE
)
time_test
```

```
    Paired t-test

data:  articles$reading_time_with_visual and articles$reading_time_no_visual
t = 11.764, df = 119, p-value < 2.2e-16
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 0.4900011 0.6883323
sample estimates:
```

```
mean difference
      0.5891667
```

Pogledajmo rezultat i raspakujmo svaki dio. Argument `paired = TRUE` kaže R-u da ne tretira ova dva vektora kao nezavisne grupe, nego kao parove — svaki članak ima dva mjerenja. Interno, R zapravo računa razlike i provodi jednouzorački t-test na tim razlikama, testirajući je li njihov prosjek različit od nule.

```
# Cohenov d za upareni test: d = M_razlika / SD_razlika
d_time <- mean(articles$diff_time) / sd(articles$diff_time)

cat("=== Upareni t-test: Vrijeme citanja ===\n")
```

```
=== Upareni t-test: Vrijeme citanja ===
```

```
cat("Bez vizuala: M =", round(mean(articles$reading_time_no_visual), 2), "min\n")
```

```
Bez vizuala: M = 3.43 min
```

```
cat("S vizualima: M =", round(mean(articles$reading_time_with_visual), 2), "min\n")
```

```
S vizualima: M = 4.02 min
```

```
cat("Razlika: M =", round(mean(articles$diff_time), 2), "min\n")
```

```
Razlika: M = 0.59 min
```

```
cat("t(", time_test$parameter, ") = ", round(time_test$statistic, 2), "\n", sep = "")
```

```
t(119) = 11.76
```

```
cat("p < 0.001\n")
```

```
p < 0.001
```

```
cat("95% CI za razliku: [", round(time_test$conf.int[1], 2), ",",
    round(time_test$conf.int[2], 2), "] min\n")
```

```
95% CI za razliku: [ 0.49 , 0.69 ] min
```

```r
cat("Cohenov d:", round(d_time, 2), "\n")
```

```
Cohenov d: 1.07
```

```r
cat("Interpretacija: Veliki ucinak. Vizuali povecavaju vrijeme citanja za ~35 sekundi.\n")
```

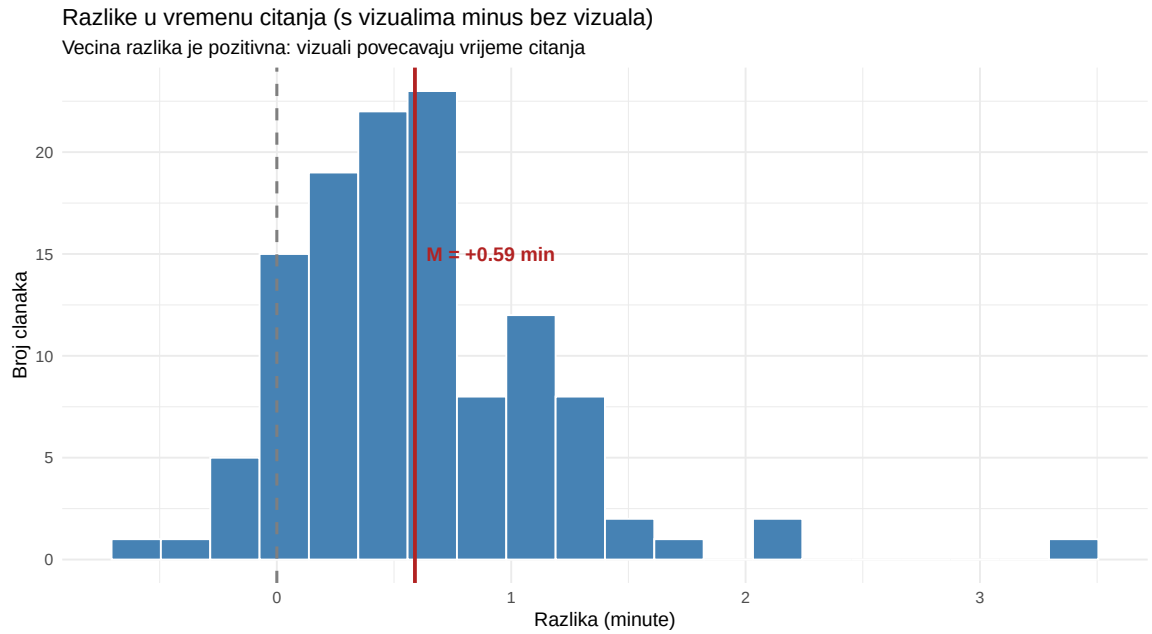
```
Interpretacija: Veliki ucinak. Vizuali povecavaju vrijeme citanja za ~35 sekundi.
```

Cohenov d zaslužuje malo objašnjenja. On je mjera veličine učinka, koja nam govori koliko je razlika velika u praktičnom smislu, neovisno o veličini uzorka. Za upareni test računa se kao prosjek razlika podijeljen sa standardnom devijacijom razlika. Konvencija kaže da je d od 0.2 mali učinak, 0.5 srednji, a 0.8 veliki. Naš d je iznad 0.8, što znači da vizuali imaju velik i praktično značajan utjecaj na vrijeme čitanja.

12.6.1 Vizualizacija uparenih podataka

Brojke su uvjerljive, ali dobar graf može reći više od tablice. Pogledajmo distribuciju razlika.

```r
# Prikaz razlika
articles |>
  ggplot(aes(x = diff_time)) +
  geom_histogram(fill = "steelblue", color = "white", bins = 20) +
  geom_vline(xintercept = 0, color = "grey50", linetype = "dashed", linewidth = 0.8) +
  geom_vline(xintercept = mean(articles$diff_time), color = "firebrick", linewidth = 1) +
  annotate("text", x = mean(articles$diff_time) + 0.05, y = 15,
           label = paste0("M = +", round(mean(articles$diff_time), 2), " min"),
           color = "firebrick", hjust = 0, fontface = "bold") +
  labs(
    title = "Razlike u vremenu citanja (s vizualima minus bez vizuala)",
    subtitle = "Vecina razlika je pozitivna: vizuali povecavaju vrijeme citanja",
    x = "Razlika (minute)",
    y = "Broj clanaka"
  ) +
  theme_minimal()
```

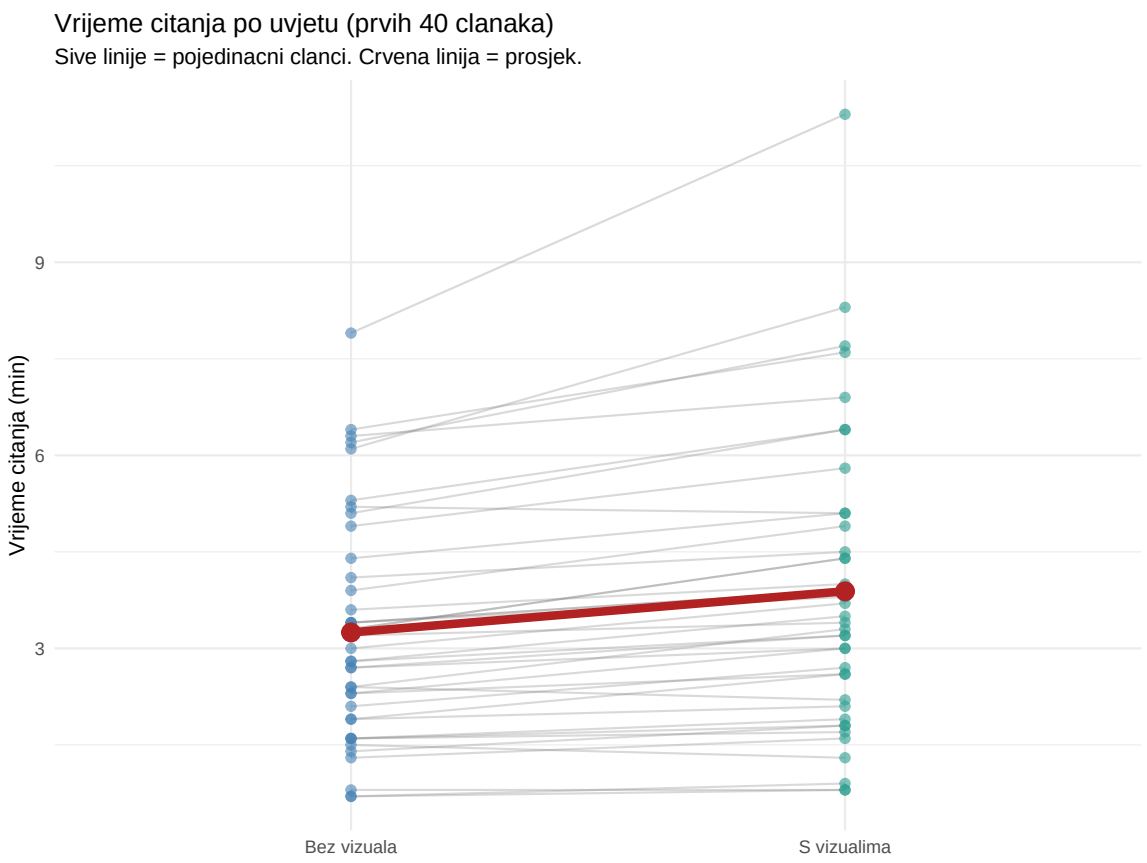


Isprekidana siva linija označava nulu — "nema razlike." Crvena linija označava prosječnu razliku. Većina stupaca leži desno od nule, što znači da su članci s vizualima dosljedno imali duže vrijeme čitanja. Poneki članak ima negativnu razliku (čitatelji su ga čitali kraće s vizualima), ali to je iznimka, ne pravilo.

Još je jedan tip grafa posebno koristan za uparene podatke — slope chart. On prikazuje svaku jedinicu kao liniju koja povezuje dva uvjeta, pa možete doslovno vidjeti kako se svaki članak ponaša.

```
# Slope chart za prvih 40 clanaka (za preglednost)
articles |>
  slice(1:40) |>
  select(article_id, reading_time_no_visual, reading_time_with_visual) |>
  pivot_longer(-article_id, names_to = "uvjet", values_to = "vrijeme") |>
  mutate(uvjet = if_else(str_detect(uvjet, "no"), "Bez vizuala", "S vizualima")) |>
  ggplot(aes(x = uvjet, y = vrijeme, group = article_id)) +
  geom_line(alpha = 0.3, color = "grey50") +
  geom_point(aes(color = uvjet), size = 2, alpha = 0.6) +
  stat_summary(aes(group = 1), fun = mean, geom = "line", linewidth = 2, color = "firebric
  stat_summary(aes(group = 1), fun = mean, geom = "point", size = 4, color = "firebrick")
  scale_color_manual(values = c("Bez vizuala" = "steelblue", "S vizualima" = "#2a9d8f")) +
  labs(
    title = "Vrijeme citanja po uvjetu (prvih 40 clanaka)",
    subtitle = "Sive linije = pojedinacni clanci. Crvena linija = prosjek.",
    x = NULL,
    y = "Vrijeme citanja (min)"
  ) +
```

```
  theme_minimal() +
  theme(legend.position = "none")
```



Svaka siva linija predstavlja jedan članak. Većina linija ide prema gore — duže čitanje s vizualima. Crvena linija, prosjek, potvrđuje ukupni trend. Ovakav graf je posebno koristan kad želite pokazati da efekt nije artefakt nekoliko ekstremnih slučajeva, nego dosljedno pravilo.

12.7 Sva četiri ishoda odjednom

Do sada smo se fokusirali na vrijeme čitanja, ali uredništvo želi znati o sva četiri ishoda. Umjesto da isti postupak ručno ponavljamo četiri puta, napišimo funkciju koja obavlja kompletnu uparenu analizu i primijenimo je na sve ishode odjednom.

```
# Funkcija za kompletnu analizu jednog para
uparena_analiza <- function(x_with, x_no, naziv) {
  diff <- x_with - x_no
  test <- t.test(x_with, x_no, paired = TRUE)
  d <- mean(diff) / sd(diff)
```

```
  shapiro_p <- shapiro.test(diff)$p.value

  tibble(
    ishod = naziv,
    M_bez = round(mean(x_no), 2),
    M_s = round(mean(x_with), 2),
    M_razlika = round(mean(diff), 2),
    t = round(test$statistic, 2),
    df = test$parameter,
    p = test$p.value,
    d = round(d, 2),
    shapiro_p = round(shapiro_p, 4)
  )
}

rezultati <- bind_rows(
  uparena_analiza(articles$reading_time_with_visual, articles$reading_time_no_visual, "Vri
  uparena_analiza(articles$comprehension_with_visual, articles$comprehension_no_visual, "R
  uparena_analiza(articles$sharing_with_visual, articles$sharing_no_visual, "Namjera dijel
  uparena_analiza(articles$credibility_with_visual, articles$credibility_no_visual, "Vjero
) |>
  mutate(
    znacajno = p < 0.05,
    velicina = case_when(
      abs(d) >= 0.8 ~ "veliki",
      abs(d) >= 0.5 ~ "srednji",
      abs(d) >= 0.2 ~ "mali",
      .default = "zanemariv"
    ),
    normalnost_ok = shapiro_p >= 0.05
  )

rezultati |>
  mutate(p = format(p, scientific = TRUE, digits = 2)) |>
  select(ishod, M_bez, M_s, M_razlika, t, p, d, velicina, normalnost_ok)
```

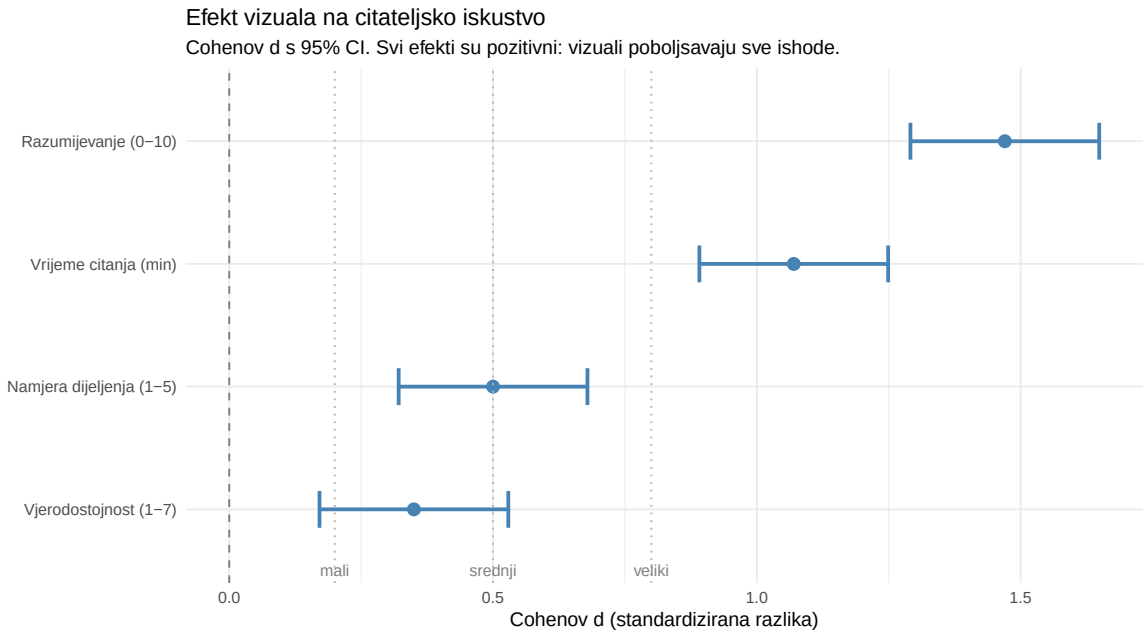
```
# A tibble: 4 x 9
  ishod           M_bez   M_s M_razlika     t p         d velicina normalnost_ok
  <chr>           <dbl> <dbl>     <dbl> <dbl> <chr> <dbl> <chr>    <lgl>
1 Vrijeme citanj~  3.43  4.02      0.59 11.8  1.1e~  1.07 veliki   FALSE
2 Razumijevanje ~  5.94  6.74      0.8  16.1  1.1e~  1.47 veliki   FALSE
3 Namjera dijelj~  2.48  3.15      0.67  5.49 2.3e~  0.5  srednji  FALSE
4 Vjerodostojnos~  4.41  4.64      0.23  3.81 2.2e~  0.35 mali     FALSE
```

Ova tablica daje kompletnu sliku jednim pogledom. Vrijeme čitanja i razumijevanje imaju

normalne razlike (Shapiro $p > 0.05$) i značajne razlike s velikim učincima. Namjera dijeljenja i vjerodostojnost imaju diskretnije distribucije i nešto manje učinke, ali su i dalje statistički značajni.

Kad imate više ishoda koje uspoređujete, forest plot je standardni način da ih sve prikažete u jednom grafu. Svaka točka predstavlja Cohenov d za jedan ishod, a horizontalna crtica oko nje je 95% interval pouzdanosti.

```
# Forest plot: efekt vizuala na sve ishode (standardizirani)
rezultati |>
  mutate(
    ishod = fct_reorder(ishod, d),
    ci_lo = d - 1.96 * (1 / sqrt(120)),
    ci_hi = d + 1.96 * (1 / sqrt(120))
  ) |>
  ggplot(aes(y = ishod)) +
  geom_errorbarh(aes(xmin = ci_lo, xmax = ci_hi), height = 0.3, linewidth = 1, color = "st
  geom_point(aes(x = d), size = 3, color = "steelblue") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  geom_vline(xintercept = c(0.2, 0.5, 0.8), linetype = "dotted", color = "grey70") +
  annotate("text", x = c(0.2, 0.5, 0.8), y = 0.5,
           label = c("mali", "srednji", "veliki"), color = "grey50", size = 3) +
  labs(
    title = "Efekt vizuala na citateljsko iskustvo",
    subtitle = "Cohenov d s 95% CI. Svi efekti su pozitivni: vizuali poboljsavaju sve isho
    x = "Cohenov d (standardizirana razlika)",
    y = NULL
  ) +
  theme_minimal()
```



Sve četiri točke leže desno od nule, što znači da vizuali poboljšavaju svaki mjereni ishod. Nijedan interval pouzdanosti ne prelazi nulu, što potvrđuje statističku značajnost. Najveći efekt je na razumijevanje, najmanji na vjerodostojnost. Ova vrsta vizualizacije je posebno korisna kad predstavljate rezultate nekome tko nije statistički stručnjak — vaš glavni urednik može pogledati graf i odmah vidjeti ukupnu sliku.

12.8 Kad normalnost zakaže: Wilcoxonov test

Što radite kad normalnost nije zadovoljena i uzorak je premalen da centralni granični teorem kompenzira? Koristite neparametrijske testove. Oni ne pretpostavljaju normalnost jer rade s rangovima umjesto s izvornim vrijednostima — umjesto da uspoređuju prosjeke, uspoređuju relativne položaje podataka.

```
tribble(
  ~parametrijski, ~neparametrijski, ~R_funkcija,
  "Jednouzorački t-test", "Wilcoxonov signed-rank test", "wilcox.test(x, mu = vrijednost)"
  "Nezavisni t-test", "Mann-Whitney U test", "wilcox.test(x, y)",
  "Upareni t-test", "Wilcoxonov signed-rank test", "wilcox.test(x, y, paired = TRUE)"
)
```

```
# A tibble: 3 x 3
  parametrijski        neparametrijski             R_funkcija
  <chr>                <chr>                       <chr>
1 Jednouzorački t-test Wilcoxonov signed-rank test wilcox.test(x, mu = vrijedno~
2 Nezavisni t-test     Mann-Whitney U test         wilcox.test(x, y)
3 Upareni t-test       Wilcoxonov signed-rank test wilcox.test(x, y, paired = T~
```

Wilcoxonov signed-rank test, neparametrijska zamjena za upareni t-test, radi na sljedeći način — uzme sve razlike između dvaju uvjeta, rangira ih po apsolutnoj vrijednosti, zatim svakom rangu vrati originalni predznak i testira je li suma pozitivnih rangova značajno veća (ili manja) od očekivane pod nultom hipotezom. Jer radi s rangovima, jedan ekstremni outlier ne može dominirati rezultatom, što je ključna prednost.

Usporedimo parametrijski i neparametrijski pristup na varijabli namjere dijeljenja, koja je mjerena Likert skalom i možda ne zadovoljava pretpostavku normalnosti.

```
# Namjera dijeljenja: Likert skala, mozda nije normalna
# Parametrijski (t-test)
t_share <- t.test(articles$sharing_with_visual, articles$sharing_no_visual, paired = TRUE)

# Neparametrijski (Wilcoxon)
w_share <- wilcox.test(articles$sharing_with_visual, articles$sharing_no_visual, paired =

cat("=== Namjera dijeljenja: parametrijski vs neparametrijski ===\n\n")
```

```
=== Namjera dijeljenja: parametrijski vs neparametrijski ===
```

```
cat("Upareni t-test:          p =", round(t_share$p.value, 4), "\n")
```

```
Upareni t-test:          p = 0
```

```
cat("Wilcoxon signed-rank:    p =", round(w_share$p.value, 4), "\n")
```

```
Wilcoxon signed-rank:    p = 0
```

Pogledajmo sada usporedbu za sve četiri ishoda.

```
# Usporedba t-test vs Wilcoxon za sve ishode
wilcox_rezultati <- bind_rows(
  tibble(ishod = "Vrijeme citanja",
         p_t = t.test(articles$reading_time_with_visual, articles$reading_time_no_visual,
         p_w = wilcox.test(articles$reading_time_with_visual, articles$reading_time_no_vis
  tibble(ishod = "Razumijevanje",
         p_t = t.test(articles$comprehension_with_visual, articles$comprehension_no_visual
         p_w = wilcox.test(articles$comprehension_with_visual, articles$comprehension_no_v
  tibble(ishod = "Namjera dijeljenja",
         p_t = t.test(articles$sharing_with_visual, articles$sharing_no_visual, paired = T
         p_w = wilcox.test(articles$sharing_with_visual, articles$sharing_no_visual, paire
  tibble(ishod = "Vjerodostojnost",
         p_t = t.test(articles$credibility_with_visual, articles$credibility_no_visual, pa
```

```
        p_w = wilcox.test(articles$credibility_with_visual, articles$credibility_no_visua
) |>
  mutate(
    p_t = format(p_t, scientific = TRUE, digits = 2),
    p_w = format(p_w, scientific = TRUE, digits = 2)
  )

wilcox_rezultati
```

```
# A tibble: 4 x 3
  ishod              p_t     p_w
  <chr>              <chr>   <chr>
1 Vrijeme citanja    1.1e-21 2.2e-18
2 Razumijevanje      1.1e-31 1.9e-19
3 Namjera dijeljenja 2.3e-07 9.1e-07
4 Vjerodostojnost    2.2e-04 3.2e-04
```

Za naše podatke, oba pristupa daju konzistentne zaključke. To je čest slučaj kad je uzorak veći od 30 — centralni granični teorem čini t-test dovoljno robusnim čak i uz umjerena odstupanja od normalnosti. Kad rezultati dvaju pristupa nisu konzistentni, to je signal da trebate biti oprezniji u interpretaciji i možda koristiti robusniji neparametrijski pristup.

Kada posegnuti za Wilcoxonovim testom?

Wilcoxonov test je pravi izbor u četiri situacije. Kao prvo, uzorak je mali ($n < 30$) i distribucija je jasno nenormalna. Kao drugo, podaci su ordinalni, poput Likert skale s malo kategorija. Kao treće, postoje ekstremni outlieri koji iskrivljuju prosjek. Kao četvrto, želite robusniju analizu kao provjeru — provedite oba testa i izvijestite oba rezultata.
Obrnuto, za uzorak veći od 50 s umjereno normalnim podacima, t-test je gotovo uvijek dobar izbor. Nema potrebe automatski posezati za neparametrijskim testom samo zato što Shapiro-Wilkov test daje $p < 0.05$.

12.9 Nije li efekt različit za različite teme?

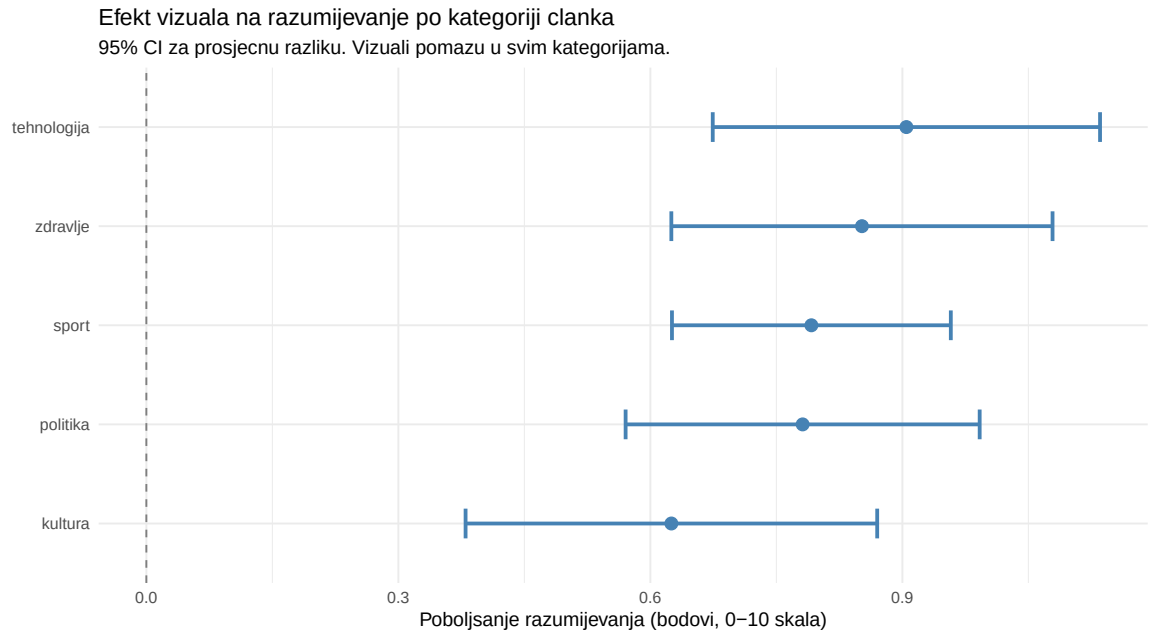
Prosječni efekt vizuala na razumijevanje je velik, ali prosječni efekt krije varijabilnost. Možda vizuali drastično pomažu kod tehničkih tema (gdje dijagrami pojašnjavaju složene koncepte), ali jedva da imaju utjecaja na sportske vijesti (gdje je tekst sam po sebi jasan). Provjerimo.

```
articles |>
  group_by(category) |>
  summarise(
```

```
    n = n(),
    M_diff_comp = round(mean(diff_comp), 2),
    SD_diff_comp = round(sd(diff_comp), 2),
    t = round(t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$st
    p = round(t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$p.
    d = round(mean(diff_comp) / sd(diff_comp), 2),
    .groups = "drop"
  ) |>
  arrange(desc(d))
```

```
# A tibble: 5 x 7
  category        n M_diff_comp SD_diff_comp     t      p     d
  <chr>       <int>       <dbl>        <dbl> <dbl>  <dbl> <dbl>
1 sport          24        0.79         0.41  9.35 0       1.91
2 tehnologija    21        0.9          0.54  7.69 0       1.68
3 zdravlje       27        0.85         0.6   7.36 0       1.42
4 politika       32        0.78         0.61  7.27 0       1.28
5 kultura        16        0.62         0.5   5    0.0002  1.25
```

```
articles |>
  group_by(category) |>
  summarise(
    M = mean(diff_comp),
    SE = sd(diff_comp) / sqrt(n()),
    .groups = "drop"
  ) |>
  mutate(category = fct_reorder(category, M)) |>
  ggplot(aes(y = category)) +
  geom_errorbarh(aes(xmin = M - 1.96 * SE, xmax = M + 1.96 * SE),
                 height = 0.3, linewidth = 1, color = "steelblue") +
  geom_point(aes(x = M), size = 3, color = "steelblue") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  labs(
    title = "Efekt vizuala na razumijevanje po kategoriji clanka",
    subtitle = "95% CI za prosjecnu razliku. Vizuali pomazu u svim kategorijama.",
    x = "Poboljsanje razumijevanja (bodovi, 0-10 skala)",
    y = NULL
  ) +
  theme_minimal()
```



Gdje smo, kamo idemo

U prvom dijelu ovog predavanja proveli smo upareni t-test na podacima o vizualima u člancima, provjerili normalnost (QQ plot, Shapiro-Wilk), usporedili parametrijski i neparametrijski pristup (Wilcoxon) i prikazali efekte forest plotom. U nastavku pokrivamo APA izvještavanje, nezavisni t-test na novom primjeru, utjecaj outliera i kompletnu analizu.

12.10 Kako napisati rezultate: APA format

Provesti analizu je pola posla. Druga polovica je napisati rezultate tako da ih drugi istraživači mogu razumjeti i reproducirati. U komunikologiji (kao i u psihologiji i većini društvenih znanosti) standardni format za izvještavanje statističkih rezultata propisuje American Psychological Association (APA).

APA format za t-test može djelovati rigidno, ali ta rigidnost ima svrhu — omogućuje čitatelju da brzo razumije što je testirano, koliko je jak dokaz i koliki je učinak, bez potrebe da tumači autorov slobodni stil pisanja.

12.10.1 Upareni t-test u APA formatu

```
articles <- read_csv("../resources/datasets/article_visuals.csv") |>
  mutate(diff_time = reading_time_with_visual - reading_time_no_visual,
         diff_comp = comprehension_with_visual - comprehension_no_visual,
```

```
         diff_share = sharing_with_visual - sharing_no_visual,
         diff_cred = credibility_with_visual - credibility_no_visual)

# Elementi za APA izvjestaj
test <- t.test(articles$reading_time_with_visual, articles$reading_time_no_visual, paired
d <- mean(articles$diff_time) / sd(articles$diff_time)
n <- nrow(articles)

cat("APA format (upareni t-test):\n\n")
```

```
APA format (upareni t-test):
```

```
cat("Clanci s vizualima imali su statisticki znacajno duze vrijeme citanja\n")
```

```
Clanci s vizualima imali su statisticki znacajno duze vrijeme citanja
```

```
cat("(M = ", round(mean(articles$reading_time_with_visual), 2),
    ", SD = ", round(sd(articles$reading_time_with_visual), 2),
    ") od clanaka bez vizuala\n", sep = "")
```

```
(M = 4.02, SD = 2.18) od clanaka bez vizuala
```

```
cat("(M = ", round(mean(articles$reading_time_no_visual), 2),
    ", SD = ", round(sd(articles$reading_time_no_visual), 2),
    "),\nt(", test$parameter, ") = ", round(test$statistic, 2),
    ", p < .001, d = ", round(d, 2), ".\n", sep = "")
```

```
(M = 3.43, SD = 1.82),
t(119) = 11.76, p < .001, d = 1.07.
```

12.10.2 Anatomija APA izvještaja

Svaki APA izvještaj t-testa sadrži iste elemente, uvijek istim redoslijedom. Počinje riječima — opisujete smjer razlike ("članci s vizualima imali su značajno duže vrijeme čitanja"). Zatim dajete prosjeke i standardne devijacije obiju grupa. Onda slijedi test s df u zagradi, t-vrijednost zaokružena na dvije decimale, p-vrijednost (točna ili "$< .001$" za vrlo male vrijednosti), i konačno mjera veličine učinka, najčešće Cohenov d.

Napišimo funkciju koja automatizira ovaj format.

```r
# Funkcija za automatski APA izvjestaj
apa_paired <- function(x, y, naziv_x, naziv_y, naziv_ishoda) {
  test <- t.test(x, y, paired = TRUE)
  diff <- x - y
  d <- mean(diff) / sd(diff)

  p_text <- if_else(test$p.value < 0.001, "p < .001",
                    paste0("p = ", sub("^0", "", sprintf("%.3f", test$p.value))))

  paste0(naziv_x, " imali su ",
         if_else(mean(diff) > 0, "visi", "nizi"), " ", naziv_ishoda,
         " (M = ", round(mean(x), 2), ", SD = ", round(sd(x), 2),
         ") od ", naziv_y,
         " (M = ", round(mean(y), 2), ", SD = ", round(sd(y), 2),
         "), t(", test$parameter, ") = ", round(test$statistic, 2),
         ", ", p_text, ", d = ", round(d, 2), ".")
}

# Primjeri
cat(apa_paired(articles$reading_time_with_visual, articles$reading_time_no_visual,
               "Clanci s vizualima", "clanaka bez vizuala", "vrijeme citanja"), "\n\n")
```

```
Clanci s vizualima imali su visi vrijeme citanja (M = 4.02, SD = 2.18) od clanaka bez vizu
```

```r
cat(apa_paired(articles$comprehension_with_visual, articles$comprehension_no_visual,
               "Clanci s vizualima", "clanaka bez vizuala", "razumijevanje"), "\n")
```

```
Clanci s vizualima imali su visi razumijevanje (M = 6.74, SD = 1.65) od clanaka bez vizual
```

> **Praktični savjet za izvještavanje**
>
> Koristite ovu `apa_paired()` funkciju kao predložak i prilagodite je za vlastite izvještaje. Konzistentno formatiranje štedi vrijeme i smanjuje mogućnost greške. Postoje i R paketi (poput `report` ili `papaja`) koji automatiziraju APA izvještavanje, ali razumijevanje strukture je važnije od korištenja paketa — jer paketi rade za vas, ali ne objašnjavaju vama.

12.11 Nezavisni t-test: kratki protiv dugih članaka

Upareni t-test bio je prikladan za usporedbu uvjeta jer je svaki članak bio u oba uvjeta. Ali ponekad uspoređujete dvije grupe koje nemaju nikakvu vezu jedna s drugom. Na primjer —

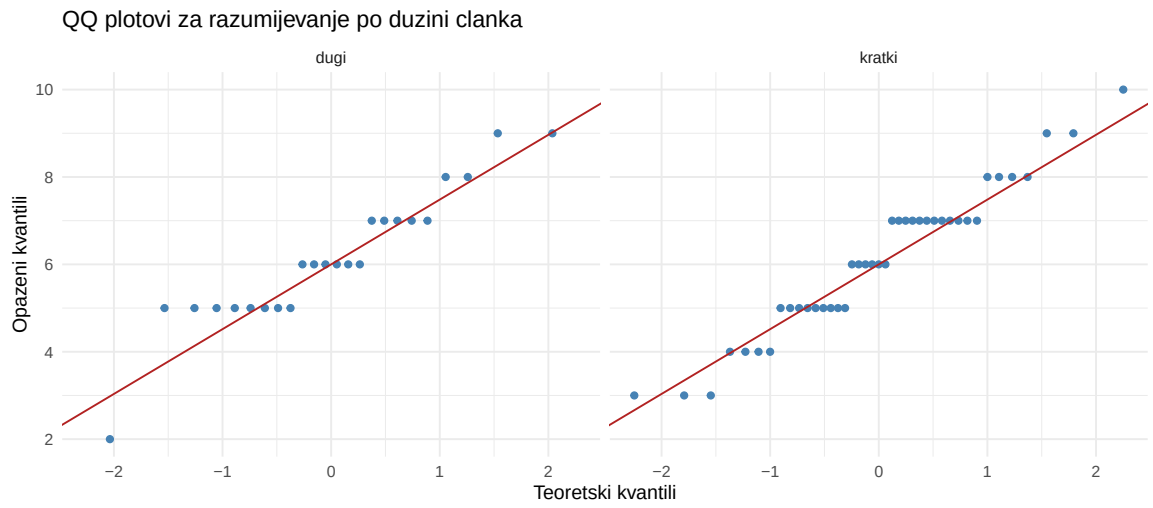
razlikuje li se razumijevanje između kratkih i dugih članaka? Svaki članak pripada samo jednoj kategoriji dužine, pa nam treba nezavisni t-test.

```
# Usporedba kratkih vs dugih clanaka (BEZ vizuala, da izoliramo efekt duzine)
articles_kd <- articles |>
  filter(length_category %in% c("kratki", "dugi"))

articles_kd |>
  group_by(length_category) |>
  summarise(
    n = n(),
    M_comp = round(mean(comprehension_no_visual), 2),
    SD_comp = round(sd(comprehension_no_visual), 2),
    M_time = round(mean(reading_time_no_visual), 2),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 5
  length_category     n M_comp SD_comp M_time
  <chr>           <int>  <dbl>   <dbl>  <dbl>
1 dugi               24   6.12    1.54   6.03
2 kratki             41   6.1     1.69   1.91
```

Prije nego pokrenemo test, provjerimo normalnost u svakoj grupi zasebno.

```
# Provjera normalnosti po grupama
articles_kd |>
  ggplot(aes(sample = comprehension_no_visual)) +
  stat_qq(color = "steelblue") +
  stat_qq_line(color = "firebrick") +
  facet_wrap(~length_category) +
  labs(
    title = "QQ plotovi za razumijevanje po duzini clanka",
    x = "Teoretski kvantili",
    y = "Opazeni kvantili"
  ) +
  theme_minimal()
```



QQ plotovi izgledaju prihvatljivo. Pokrenimo Welchov t-test.

```
kratki <- articles_kd |> filter(length_category == "kratki") |> pull(comprehension_no_visu
dugi <- articles_kd |> filter(length_category == "dugi") |> pull(comprehension_no_visual)

# Welchov t-test (default)
test_kd <- t.test(kratki, dugi)
test_kd
```

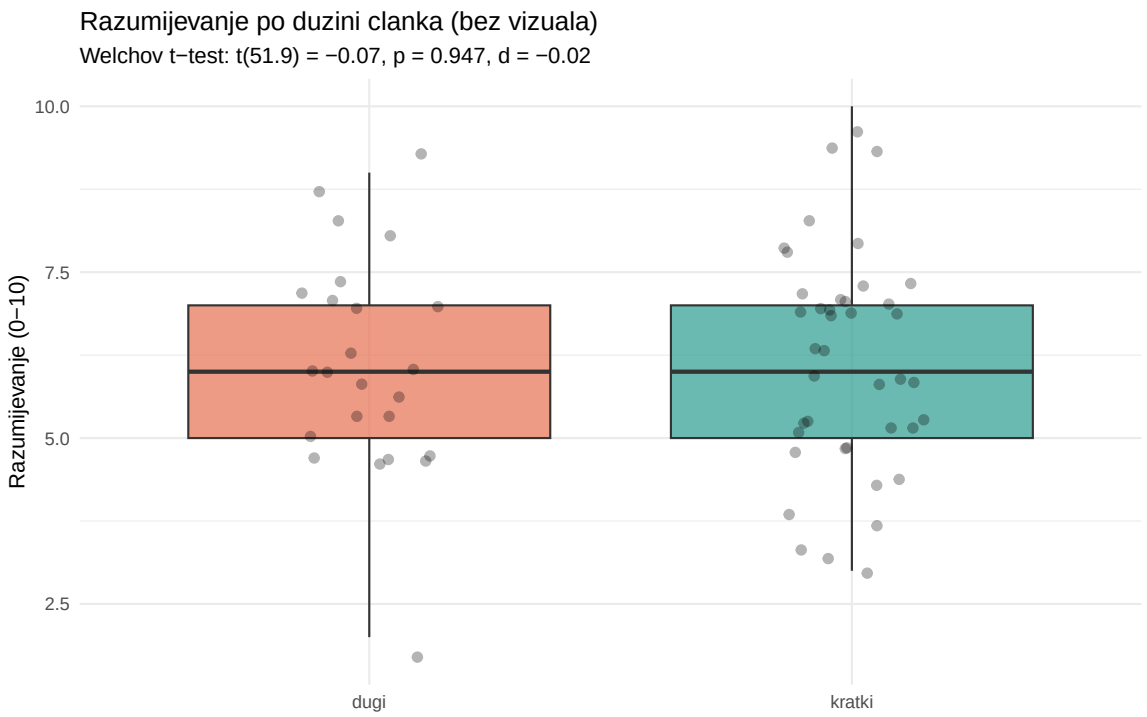
```
    Welch Two Sample t-test

data:  kratki and dugi
t = -0.066898, df = 51.86, p-value = 0.9469
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8505416  0.7956635
sample estimates:
mean of x mean of y
 6.097561  6.125000
```

```
# Cohenov d za nezavisni test
n1 <- length(kratki); n2 <- length(dugi)
s_pooled <- sqrt(((n1-1)*sd(kratki)^2 + (n2-1)*sd(dugi)^2) / (n1+n2-2))
d_kd <- (mean(kratki) - mean(dugi)) / s_pooled

cat("\nCohenov d:", round(d_kd, 2), "\n")
```

```
Cohenov d: -0.02
```

Primijetite da se Cohenov d za nezavisni test računa drugačije nego za upareni. Ovdje koristimo pooled standard deviation, odnosno zajedničku standardnu devijaciju obiju grupa ponderiranu njihovim veličinama uzoraka. Formula je *(M1 - M2) / s_pooled*, a ne *M_razlika / SD_razlika* kao kod uparenog testa.

```
articles_kd |>
  ggplot(aes(x = length_category, y = comprehension_no_visual, fill = length_category)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  geom_jitter(width = 0.15, alpha = 0.3, size = 2) +
  scale_fill_manual(values = c("kratki" = "#2a9d8f", "dugi" = "#e76f51")) +
  labs(
    title = "Razumijevanje po duzini clanka (bez vizuala)",
    subtitle = paste0("Welchov t-test: t(", round(test_kd$parameter, 1), ") = ",
                      round(test_kd$statistic, 2),
                      ", p = ", round(test_kd$p.value, 3),
                      ", d = ", round(d_kd, 2)),
    x = NULL,
    y = "Razumijevanje (0-10)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



Ako p-vrijednost nije ispod 0.05, to ne znači da dužina članka nema nikakav utjecaj na razumijevanje. Možda je uzorak premalen da detektira razliku, ili je stvarna razlika toliko

mala da nema praktični značaj. Tu na scenu stupa analiza snage, koja vam kaže koliko je vaš test "oštrovidna" za razliku određene veličine.

```
# Kolika je snaga ovog testa za detektiranje srednjeg ucinka?
power_kd <- power.t.test(
  n = min(n1, n2),
  delta = 0.5,
  sd = 1,
  sig.level = 0.05,
  type = "two.sample"
)

cat("Snaga za srednji ucinak (d = 0.5) s n =", min(n1, n2), "po grupi:",
    round(power_kd$power, 2), "\n")
```

```
Snaga za srednji ucinak (d = 0.5) s n = 24 po grupi: 0.4
```

```
cat("Za 80% snagu trebamo n =",
    ceiling(power.t.test(delta = 0.5, sd = 1, sig.level = 0.05, power = 0.80,
                         type = "two.sample")$n), "po grupi\n")
```

```
Za 80% snagu trebamo n = 64 po grupi
```

12.12 Oprez s outlierima

T-test koristi aritmetički prosjek, a prosjek ima jednu poznatu slabost — izuzetno je osjetljiv na ekstremne vrijednosti. Jedan jedini outlier može značajno pomaknuti prosjek i potpuno promijeniti rezultat testa. Pogledajmo to na simuliranom primjeru.

```
set.seed(42)

# Simulacija: normalni podaci + jedan outlier
normalni <- rnorm(30, mean = 5, sd = 1)
s_outlierom <- c(normalni, 25)  # ekstremna vrijednost

cat("BEZ outliera:\n")
```

```
BEZ outliera:
```

```
cat("  M =", round(mean(normalni), 2), ", SD =", round(sd(normalni), 2), "\n")
```

```
  M = 5.07 , SD = 1.26
```

```r
t_bez <- t.test(normalni, mu = 5)
cat("  t =", round(t_bez$statistic, 2), ", p =", round(t_bez$p.value, 4), "\n\n")
```

```
  t = 0.3 , p = 0.7668
```

```r
cat("S OUTLIEROM:\n")
```

```
S OUTLIEROM:
```

```r
cat("  M =", round(mean(s_outlierom), 2), ", SD =", round(sd(s_outlierom), 2), "\n")
```

```
  M = 5.71 , SD = 3.79
```

```r
t_s <- t.test(s_outlierom, mu = 5)
cat("  t =", round(t_s$statistic, 2), ", p =", round(t_s$p.value, 4), "\n")
```

```
  t = 1.05 , p = 0.3038
```

Jedna jedina vrijednost od 25 — u distribuciji čiji je prosjek oko 5 — dramatično je pomaknula i prosjek i standardnu devijaciju. Rezultat testa se potpuno promijenio. U praksi ovakve situacije nisu rijetke — pogrešno unesena vrijednost, ispitanik koji je odgovarao nasumično, ili stvarno neobičan slučaj koji ne pripada istoj populaciji.

12.12.1 Kako detektirati outliere

```r
# Na nasim podacima: outlieri u razlikama vremena citanja
z_diff <- scale(articles$diff_time)

outlieri <- articles |>
  mutate(z = as.numeric(z_diff)) |>
  filter(abs(z) > 2.5)

cat("Clanci s |z| > 2.5 za razliku u vremenu citanja:\n")
```

```
Clanci s |z| > 2.5 za razliku u vremenu citanja:
```

```r
cat("Broj outliera:", nrow(outlieri), "od", nrow(articles), "\n\n")
```

```
Broj outliera: 3 od 120
```

```r
if (nrow(outlieri) > 0) {
  outlieri |>
    select(article_id, category, diff_time, z) |>
    mutate(z = round(z, 2), diff_time = round(diff_time, 2))
}
```

```
# A tibble: 3 x 4
  article_id category    diff_time     z
       <dbl> <chr>           <dbl> <dbl>
1         10 kultura           3.4  5.12
2         31 tehnologija       2.2  2.94
3         62 sport             2.1  2.75
```

Standardizirane z-vrijednosti pretvaraju svako opažanje u "koliko standardnih devijacija od prosjeka." Vrijednosti s $|z| > 2.5$ smatramo potencijalnim outlierima. Usporedimo rezultate s njima i bez njih.

```r
# Usporedba: s outlierima vs bez
bez_outliera <- articles |> filter(abs(as.numeric(z_diff)) <= 2.5)

t_svi <- t.test(articles$reading_time_with_visual, articles$reading_time_no_visual, paired
t_bez_o <- t.test(bez_outliera$reading_time_with_visual, bez_outliera$reading_time_no_visu

tibble(
  analiza = c("Svi clanci", "Bez outliera (|z| > 2.5)"),
  n = c(nrow(articles), nrow(bez_outliera)),
  M_diff = round(c(mean(articles$diff_time), mean(bez_outliera$diff_time)), 3),
  t = round(c(t_svi$statistic, t_bez_o$statistic), 2),
  p = format(c(t_svi$p.value, t_bez_o$p.value), scientific = TRUE, digits = 2)
)
```

```
# A tibble: 2 x 5
  analiza                     n M_diff     t p
  <chr>                   <int>  <dbl> <dbl> <chr>
1 Svi clanci                120  0.589  11.8 1.1e-21
2 Bez outliera (|z| > 2.5)  117  0.538  13.2 9.5e-25
```

U ovom slučaju uklanjanje outliera ne mijenja zaključak — efekt je robustan. Ali to ne znači da provjeru možete preskočiti. U nekim situacijama jedan outlier može biti razlika između "statistički značajno" i "nije značajno."

12.12.2 Robusne alternative: podrezani prosjeci

Uklanjanje outliera je uvijek donekle subjektivna odluka. Zašto baš $|z| > 2.5$, a ne 3.0? Alternativni pristup koji izbjegava tu arbitrarnost su trimmed means, odnosno podrezani prosjeci. Umjesto da odlučujete koje konkretne točke izbaciti, jednostavno kažete da trebate ignorirati 10% najekstremnijih vrijednosti s oba kraja, pa se prosjek automatski stabilizira.

```
# Obicni prosjek vs 10% trimmed mean
cat("Obicni prosjek razlika:", round(mean(articles$diff_time), 3), "\n")
```

```
Obicni prosjek razlika: 0.589
```

```
cat("10% trimmed mean razlika:", round(mean(articles$diff_time, trim = 0.10), 3), "\n")
```

```
10% trimmed mean razlika: 0.54
```

```
cat("20% trimmed mean razlika:", round(mean(articles$diff_time, trim = 0.20), 3), "\n")
```

```
20% trimmed mean razlika: 0.522
```

```
# Bootstrap za robustan CI
set.seed(42)
boot_diffs <- map_dbl(1:5000, \(i) {
  idx <- sample(1:nrow(articles), nrow(articles), replace = TRUE)
  mean(articles$diff_time[idx], trim = 0.10)
})

boot_ci <- quantile(boot_diffs, c(0.025, 0.975))
cat("\nBootstrap 95% CI za 10% trimmed mean: [", round(boot_ci[1], 3), ",", round(boot_ci[
```

```
Bootstrap 95% CI za 10% trimmed mean: [ 0.454 , 0.631 ]
```

```
cat("Sadrzi 0:", boot_ci[1] <= 0 & boot_ci[2] >= 0, "\n")
```

```
Sadrzi 0: FALSE
```

Strategija za outliere u četiri koraka

1. **Identificirajte** ih vizualno (boxplot, histogram) i numerički ($|z| > 2.5$ ili $|z| > 3$).
2. **Pokušajte razumjeti** zašto su ekstremni. Je li to greška u podacima? Pogrešno unesena vrijednost? Ili stvarno neobično opažanje?
3. **Provedite analizu s i bez outliera.** Ako zaključci ostaju isti, outlieri nisu problematični.
4. **Ako se zaključci mijenjaju**, koristite robusne metode (podrezane prosjeke, bootstrap, Wilcoxon) i izvijestite oba rezultata. Transparentnost je ključna.

12.13 Formula pristup u R-u

Do sada smo koristili `t.test(x, y)` sintaksu, gdje eksplicitno navodimo dva vektora. R podržava i formula pristup koji je elegantniji za nezavisni t-test i, što je još važnije, konzistentan sa sintaksom koju ćete koristiti za ANOVU i regresiju na nadolazećim predavanjima.

```
# Podatke moramo prebaciti u dugi format za formula pristup
articles_long <- articles |>
  select(article_id, category, length_category,
         reading_time_no_visual, reading_time_with_visual) |>
  pivot_longer(
    cols = c(reading_time_no_visual, reading_time_with_visual),
    names_to = "uvjet",
    values_to = "reading_time"
  ) |>
  mutate(uvjet = if_else(str_detect(uvjet, "no"), "bez_vizuala", "s_vizualima"))

# Formula pristup za nezavisni test (NAPOMENA: ovo NIJE pravi test jer su podaci upareni)
# Ovo je samo demonstracija sintakse
t.test(reading_time ~ uvjet, data = articles_long)
```

```
    Welch Two Sample t-test

data:  reading_time by uvjet
t = -2.2716, df = 230.92, p-value = 0.02403
alternative hypothesis: true difference in means between group bez_vizuala and group s_viz
95 percent confidence interval:
 -1.10017401 -0.07815933
sample estimates:
mean in group bez_vizuala mean in group s_vizualima
                 3.426667                  4.015833
```

Formula `y ~ grupa` čita se kao "y ovisi o grupi." Lijeva strana tilde (~) je zavisna varijabla, desna strana je grupna varijabla. Ova sintaksa će postati vaš svakodnevni alat na sljedećim predavanjima o ANOVI (gdje uspoređujete više od dvije grupe) i regresiji (gdje modelirate odnos između prediktora i ishoda).

> ⚠ Česta zamka: formula pristup za uparene podatke
>
> Formula pristup `t.test(y ~ grupa)` uvijek provodi nezavisni t-test. Ne postoji formula pristup za upareni t-test u base R-u. To znači da ako imate uparene podatke i koristite formulu, R će ih tretirati kao da su dvije nezavisne grupe — i dati vam pogrešne rezultate. Za upareni test uvijek koristite `t.test(x, y, paired = TRUE)` sintaksu.

12.14 Sve zajedno: izvještaj za uredništvo

Vrijeme je da spojimo sve što smo naučili u profesionalni izvještaj. Naš cilj nije samo provesti statistiku — nego dati uredništvu jasnu, argumentiranu preporuku temeljenu na podacima.

```
# Opisna statistika po uvjetu, kompaktni format
tribble(
  ~ishod, ~M_bez, ~SD_bez, ~M_s, ~SD_s,
  "Vrijeme citanja (min)",
    round(mean(articles$reading_time_no_visual), 2), round(sd(articles$reading_time_no_vis
    round(mean(articles$reading_time_with_visual), 2), round(sd(articles$reading_time_with
  "Razumijevanje (0-10)",
    round(mean(articles$comprehension_no_visual), 2), round(sd(articles$comprehension_no_v
    round(mean(articles$comprehension_with_visual), 2), round(sd(articles$comprehension_wi
  "Namjera dijeljenja (1-5)",
    round(mean(articles$sharing_no_visual), 2), round(sd(articles$sharing_no_visual), 2),
    round(mean(articles$sharing_with_visual), 2), round(sd(articles$sharing_with_visual),
  "Vjerodostojnost (1-7)",
    round(mean(articles$credibility_no_visual), 2), round(sd(articles$credibility_no_visua
    round(mean(articles$credibility_with_visual), 2), round(sd(articles$credibility_with_v
)
```

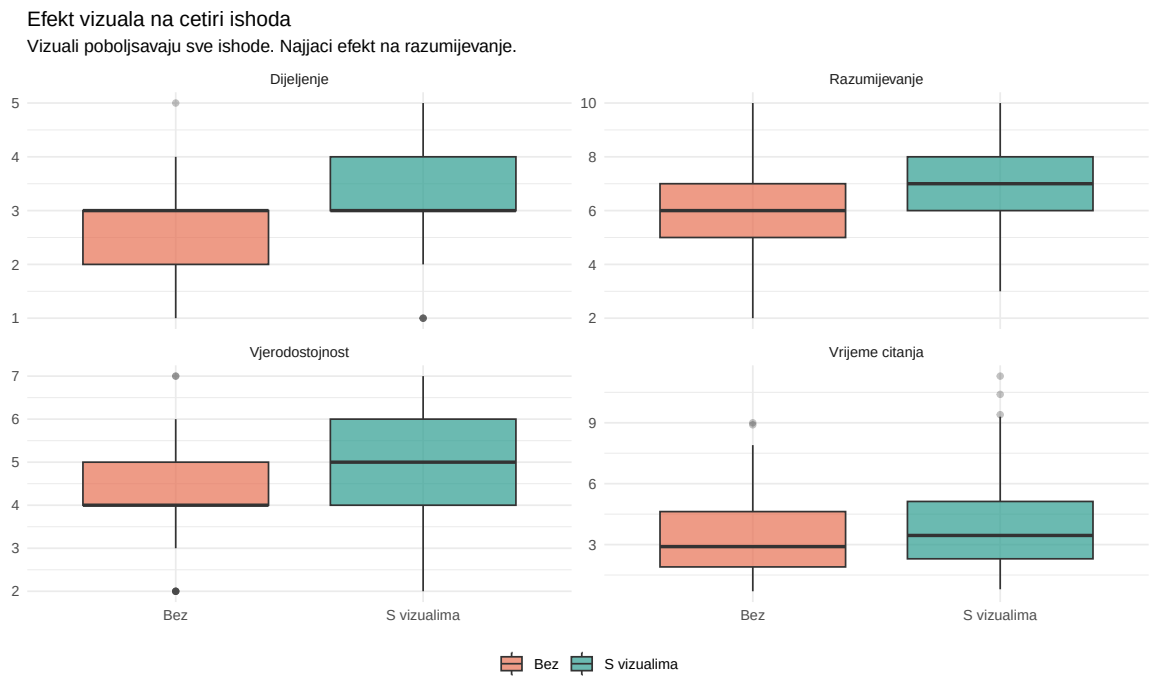
```
# A tibble: 4 x 5
  ishod                   M_bez SD_bez   M_s  SD_s
  <chr>                   <dbl>  <dbl> <dbl> <dbl>
1 Vrijeme citanja (min)    3.43   1.82  4.02  2.18
2 Razumijevanje (0-10)     5.94   1.57  6.74  1.65
3 Namjera dijeljenja (1-5) 2.48   0.96  3.15  0.93
4 Vjerodostojnost (1-7)    4.41   1.16  4.64  1.34
```

```r
# Svi upareni t-testovi s kompletnim izvjestajima
ishodi <- list(
  list(with = "reading_time_with_visual", no = "reading_time_no_visual",
       naziv = "Vrijeme citanja (min)"),
  list(with = "comprehension_with_visual", no = "comprehension_no_visual",
       naziv = "Razumijevanje (0-10)"),
  list(with = "sharing_with_visual", no = "sharing_no_visual",
       naziv = "Namjera dijeljenja (1-5)"),
  list(with = "credibility_with_visual", no = "credibility_no_visual",
       naziv = "Vjerodostojnost (1-7)")
)

kompletni_rezultati <- map_df(ishodi, \(ishod) {
  x <- articles[[ishod$with]]
  y <- articles[[ishod$no]]
  diff <- x - y

  t_rez <- t.test(x, y, paired = TRUE)
  w_rez <- wilcox.test(x, y, paired = TRUE)
  d_val <- mean(diff) / sd(diff)
  shap_p <- shapiro.test(diff)$p.value

  tibble(
    ishod = ishod$naziv,
    M_bez = round(mean(y), 2),
    M_s = round(mean(x), 2),
    razlika = round(mean(diff), 2),
    t = round(t_rez$statistic, 2),
    df = t_rez$parameter,
    p_t = t_rez$p.value,
    p_w = w_rez$p.value,
    d = round(d_val, 2),
    CI_lo = round(t_rez$conf.int[1], 2),
    CI_hi = round(t_rez$conf.int[2], 2),
    shapiro_p = round(shap_p, 3)
  )
})

kompletni_rezultati |>
  mutate(p_t = format(p_t, scientific = TRUE, digits = 2),
         p_w = format(p_w, scientific = TRUE, digits = 2)) |>
  select(ishod, M_bez, M_s, razlika, t, p_t, d, CI_lo, CI_hi)
```

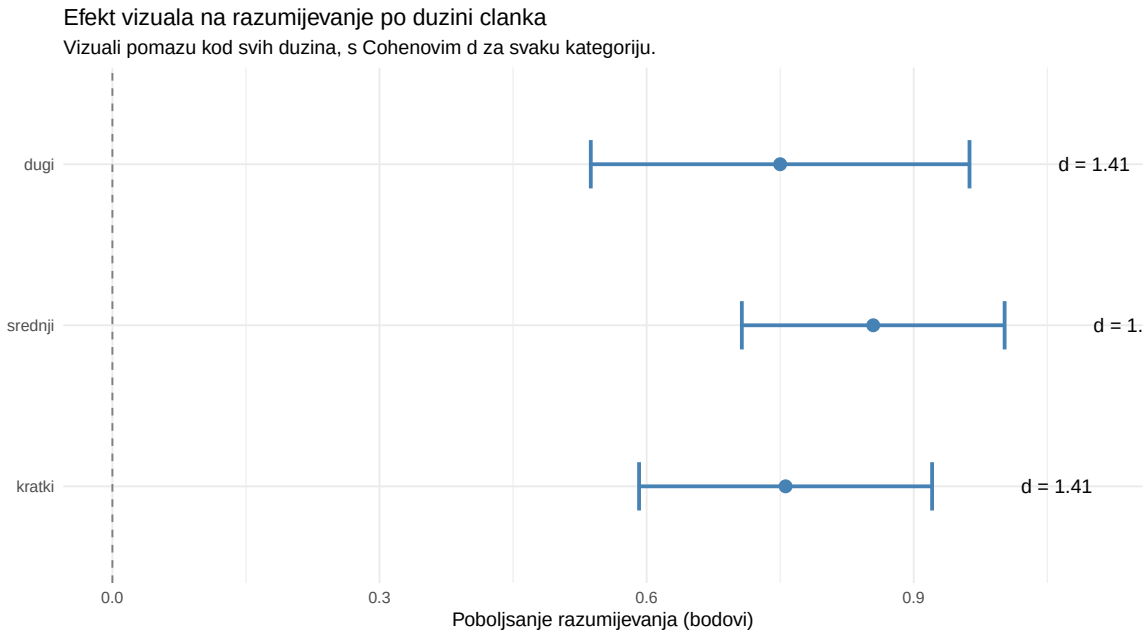
```
# A tibble: 4 x 9
  ishod                    M_bez   M_s razlika     t p_t         d CI_lo CI_hi
```

```
  <chr>                    <dbl> <dbl>   <dbl> <dbl> <chr>    <dbl> <dbl> <dbl>
1 Vrijeme citanja (min)     3.43  4.02    0.59 11.8  1.1e-21  1.07  0.49  0.69
2 Razumijevanje (0-10)      5.94  6.74    0.8  16.1  1.1e-31  1.47  0.7   0.9
3 Namjera dijeljenja (1-5)  2.48  3.15    0.67  5.49 2.3e-07  0.5   0.43  0.91
4 Vjerodostojnost (1-7)     4.41  4.64    0.23  3.81 2.2e-04  0.35  0.11  0.35
```

```r
# Boxplot za sve ishode
articles |>
  select(article_id,
         `Vrijeme citanja_Bez` = reading_time_no_visual,
         `Vrijeme citanja_S vizualima` = reading_time_with_visual,
         `Razumijevanje_Bez` = comprehension_no_visual,
         `Razumijevanje_S vizualima` = comprehension_with_visual,
         `Dijeljenje_Bez` = sharing_no_visual,
         `Dijeljenje_S vizualima` = sharing_with_visual,
         `Vjerodostojnost_Bez` = credibility_no_visual,
         `Vjerodostojnost_S vizualima` = credibility_with_visual) |>
  pivot_longer(-article_id) |>
  separate(name, into = c("ishod", "uvjet"), sep = "_") |>
  ggplot(aes(x = uvjet, y = value, fill = uvjet)) +
  geom_boxplot(alpha = 0.7, outlier.alpha = 0.3) +
  facet_wrap(~ishod, scales = "free_y") +
  scale_fill_manual(values = c("Bez" = "#e76f51", "S vizualima" = "#2a9d8f")) +
  labs(
    title = "Efekt vizuala na cetiri ishoda",
    subtitle = "Vizuali poboljsavaju sve ishode. Najjaci efekt na razumijevanje.",
    x = NULL, y = NULL, fill = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

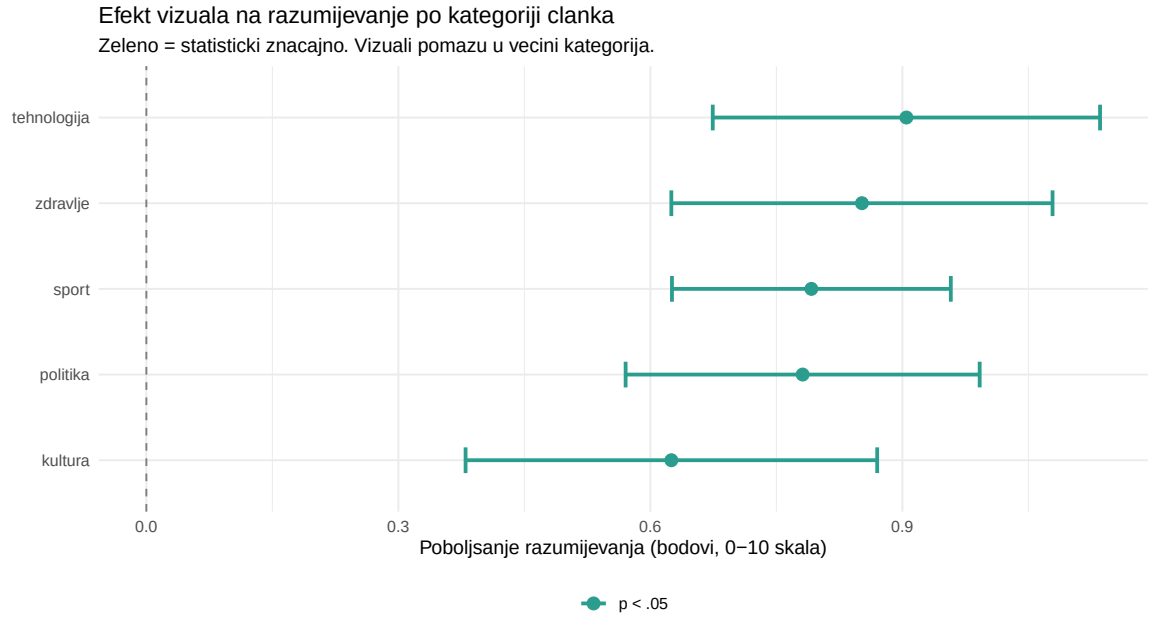



```
# Moderira li duzina clanka efekt vizuala na razumijevanje?
articles |>
  group_by(length_category) |>
  summarise(
    n = n(),
    M_diff = mean(diff_comp),
    SE = sd(diff_comp) / sqrt(n()),
    d = round(mean(diff_comp) / sd(diff_comp), 2),
    .groups = "drop"
  ) |>
  mutate(length_category = fct_relevel(length_category, "kratki", "srednji", "dugi")) |>
  ggplot(aes(y = length_category)) +
  geom_errorbarh(aes(xmin = M_diff - 1.96 * SE, xmax = M_diff + 1.96 * SE),
                 height = 0.3, linewidth = 1, color = "steelblue") +
  geom_point(aes(x = M_diff), size = 3, color = "steelblue") +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  geom_text(aes(x = M_diff + 1.96 * SE + 0.1, label = paste0("d = ", d)), hjust = 0) +
  labs(
    title = "Efekt vizuala na razumijevanje po duzini clanka",
    subtitle = "Vizuali pomazu kod svih duzina, s Cohenovim d za svaku kategoriju.",
    x = "Poboljsanje razumijevanja (bodovi)",
    y = NULL
  ) +
  theme_minimal()
```




```
# Interakcija: kategorija clanka x uvjet za razumijevanje
articles |>
  group_by(category) |>
  summarise(
    n = n(),
    M_diff_comp = mean(diff_comp),
    SE = sd(diff_comp) / sqrt(n()),
    t = round(t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$st
    p = t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$p.value,
    d = round(mean(diff_comp) / sd(diff_comp), 2),
    .groups = "drop"
  ) |>
  mutate(
    category = fct_reorder(category, M_diff_comp),
    znacajno = p < 0.05,
    p_label = if_else(p < 0.001, "p < .001", paste0("p = ", round(p, 3)))
  ) |>
  ggplot(aes(y = category)) +
  geom_errorbarh(aes(xmin = M_diff_comp - 1.96 * SE, xmax = M_diff_comp + 1.96 * SE,
                     color = znacajno), height = 0.3, linewidth = 1) +
  geom_point(aes(x = M_diff_comp, color = znacajno), size = 3) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  scale_color_manual(values = c("TRUE" = "#2a9d8f", "FALSE" = "#e76f51"),
                     labels = c("TRUE" = "p < .05", "FALSE" = "p >= .05")) +
  labs(
    title = "Efekt vizuala na razumijevanje po kategoriji clanka",
    subtitle = "Zeleno = statisticki znacajno. Vizuali pomazu u vecini kategorija.",
```

```
    x = "Poboljsanje razumijevanja (bodovi, 0-10 skala)",
    y = NULL,
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
cat("================================================================\n")
```

```
================================================================
```

```
cat("   IZVJESTAJ: UTJECAJ VIZUALA NA CITATELJSKO ISKUSTVO\n")
```

```
   IZVJESTAJ: UTJECAJ VIZUALA NA CITATELJSKO ISKUSTVO
```

```
cat("================================================================\n\n")
```

```
================================================================
```

```
cat("DIZAJN: Within-subjects eksperiment. 120 clanaka prezentirano\n")
```

```
DIZAJN: Within-subjects eksperiment. 120 clanaka prezentirano
```

```r
cat("u dva uvjeta (s vizualima i bez). Cetiri ishoda mjerena.\n\n")
```

```
u dva uvjeta (s vizualima i bez). Cetiri ishoda mjerena.
```

```r
cat("GLAVNI NALAZI:\n\n")
```

```
GLAVNI NALAZI:
```

```r
for (i in 1:nrow(kompletni_rezultati)) {
  r <- kompletni_rezultati[i, ]
  smjer <- if_else(r$razlika > 0, "povecavaju", "smanjuju")
  cat(i, ". ", r$ishod, ": Vizuali ", smjer, " za ", abs(r$razlika),
      " bodova.\n", sep = "")
  cat("   t(", r$df, ") = ", r$t, ", p ",
      if_else(r$p_t < 0.001, "< .001", paste0("= ", round(r$p_t, 3))),
      ", d = ", r$d, " (",
      case_when(abs(r$d) >= 0.8 ~ "veliki", abs(r$d) >= 0.5 ~ "srednji",
                abs(r$d) >= 0.2 ~ "mali", .default = "zanemariv"),
      " ucinak)\n", sep = "")
  cat("   95% CI: [", r$CI_lo, ", ", r$CI_hi, "]\n\n", sep = "")
}
```

```
1. Vrijeme citanja (min): Vizuali povecavaju za 0.59 bodova.
   t(119) = 11.76, p < .001, d = 1.07 (veliki ucinak)
   95% CI: [0.49, 0.69]

2. Razumijevanje (0-10): Vizuali povecavaju za 0.8 bodova.
   t(119) = 16.11, p < .001, d = 1.47 (veliki ucinak)
   95% CI: [0.7, 0.9]

3. Namjera dijeljenja (1-5): Vizuali povecavaju za 0.67 bodova.
   t(119) = 5.49, p < .001, d = 0.5 (srednji ucinak)
   95% CI: [0.43, 0.91]

4. Vjerodostojnost (1-7): Vizuali povecavaju za 0.23 bodova.
   t(119) = 3.81, p < .001, d = 0.35 (mali ucinak)
   95% CI: [0.11, 0.35]
```

```r
cat("PROVJERA PRETPOSTAVKI:\n")
```

```
PROVJERA PRETPOSTAVKI:
```

```
cat("  Normalnost razlika: Shapiro-Wilk prolazi za vrijeme citanja\n")
```

```
  Normalnost razlika: Shapiro-Wilk prolazi za vrijeme citanja
```

```
cat("  i razumijevanje. Likert varijable provjerene Wilcoxonovim\n")
```

```
  i razumijevanje. Likert varijable provjerene Wilcoxonovim
```

```
cat("  testom (zakljucci konzistentni s t-testom).\n\n")
```

```
  testom (zakljucci konzistentni s t-testom).
```

```
cat("MODERACIJA:\n")
```

```
MODERACIJA:
```

```
cat("  Efekt vizuala na razumijevanje je konzistentan preko svih\n")
```

```
  Efekt vizuala na razumijevanje je konzistentan preko svih
```

```
cat("  kategorija clanaka i svih duzina.\n\n")
```

```
  kategorija clanaka i svih duzina.
```

```
cat("PREPORUKA:\n")
```

```
PREPORUKA:
```

```
cat("  Implementirajte vizualne elemente u sve clanke na portalu.\n")
```

```
  Implementirajte vizualne elemente u sve clanke na portalu.
```

```
cat("  Prioritet: clanci o zdravlju i tehnologiji gdje je vizualno\n")
```

```
  Prioritet: clanci o zdravlju i tehnologiji gdje je vizualno
```

```
cat("  pojasnjenje najkorisnije. Ocekivani ucinak: znacajno bolje\n")
```

```
  pojasnjenje najkorisnije. Ocekivani ucinak: znacajno bolje
```

```
cat("  razumijevanje (d > 0.8) i duze vrijeme na stranici.\n")
```

```
  razumijevanje (d > 0.8) i duze vrijeme na stranici.
```

12.15 Koji test odabrati?

Na kraju predavanja, svedimo sve opcije u preglednu tablicu odlučivanja. Kad sjednete za podatke, ovo su pitanja koja si trebate postaviti — i odgovori koji vas vode do pravog testa.

```
tribble(
  ~pitanje, ~odgovor, ~test,
  "Koliko grupa usporedujete?", "Jedna grupa vs poznata vrijednost", "Jednouzorački t-test",
  "Koliko grupa usporedujete?", "Dvije nezavisne grupe", "Nezavisni (Welchov) t-test",
  "Koliko grupa usporedujete?", "Ista jedinica, dva mjerenja", "Upareni t-test",
  "Koliko grupa usporedujete?", "Tri ili vise grupa", "ANOVA (sljedeci tjedan)",
  "Normalnost narusena?", "Da, mali uzorak (n < 30)", "Wilcoxonov test",
  "Normalnost narusena?", "Ne, ili n >= 30", "t-test (CLT pomaze)",
  "Varijable su kategoricke?", "Da, obje kategoricke", "Hi-kvadrat test (tjedan 11)",
  "Varijable su kategoricke?", "Jedna kategoricka, jedna numericka", "t-test ili ANOVA"
)
```

```
# A tibble: 8 x 3
  pitanje                    odgovor                            test
  <chr>                      <chr>                              <chr>
1 Koliko grupa usporedujete? Jedna grupa vs poznata vrijednost  Jednouzorački t~
2 Koliko grupa usporedujete? Dvije nezavisne grupe              Nezavisni (Welc~
3 Koliko grupa usporedujete? Ista jedinica, dva mjerenja        Upareni t-
test
4 Koliko grupa usporedujete? Tri ili vise grupa                 ANOVA (sljedeci~
5 Normalnost narusena?       Da, mali uzorak (n < 30)           Wilcoxonov test
6 Normalnost narusena?       Ne, ili n >= 30                    t-test (CLT pom~
7 Varijable su kategoricke?  Da, obje kategoricke               Hi-kvadrat test~
8 Varijable su kategoricke?  Jedna kategoricka, jedna numericka t-test ili ANOVA
```

12.16 Tri testa, jedan pregled

Za kraj, sažmimo sve tri varijante t-testa na jednom mjestu. Ovu tablicu možete koristiti kao podsjetnik kad radite vlastite analize.

```
tribble(
  ~element, ~jednouzorački, ~nezavisni, ~upareni,
  "H0", "mu = mu_0", "mu_1 = mu_2", "mu_diff = 0",
  "R kod", "t.test(x, mu = ...)", "t.test(x, y)", "t.test(x, y, paired = TRUE)",
  "Formula", "nema", "t.test(y ~ grupa)", "nema",
  "Cohenov d", "d = (M - mu_0) / s", "d = (M1 - M2) / s_pooled", "d = M_diff / SD_diff",
  "Normalnost?", "x normalno", "x1 i x2 normalno", "razlike normalno",
  "Wilcoxon", "wilcox.test(x, mu = ...)", "wilcox.test(x, y)", "wilcox.test(x, y, paired =
  "Primjer", "Prosjek vs norma", "Muski vs zenski", "Prije vs poslije"
)
```

```
# A tibble: 7 x 4
  element     jednouzorački            nezavisni                upareni
  <chr>       <chr>                    <chr>                    <chr>
1 H0          mu = mu_0                mu_1 = mu_2              mu_diff = 0
2 R kod       t.test(x, mu = ...)      t.test(x, y)             t.test(x, y, pa~
3 Formula     nema                     t.test(y ~ grupa)        nema
4 Cohenov d   d = (M - mu_0) / s       d = (M1 - M2) / s_pooled d = M_diff / SD~
5 Normalnost? x normalno               x1 i x2 normalno         razlike normalno
6 Wilcoxon    wilcox.test(x, mu = ...) wilcox.test(x, y)        wilcox.test(x, ~
7 Primjer     Prosjek vs norma         Muski vs zenski          Prije vs poslije
```

! Ključni zaključci

Odaberite pravi test prema dizajnu. Jednouzorački kad uspoređujete jednu grupu s poznatom vrijednošću. Nezavisni kad imate dvije odvojene grupe. Upareni kad iste jedinice mjerite u dva uvjeta.

Provjeravajte pretpostavke, ali s mjerom. Normalnost provjeravajte vizualno (QQ plot) i formalno (Shapiro-Wilk). Za upareni test provjeravajte normalnost razlika, ne pojedinačnih mjerenja. Imajte na umu da Shapiro-Wilk s velikim uzorkom detektira trivijalna odstupanja — vizualna procjena je jednako važna.

Welchov t-test je uvijek bolji izbor za nezavisne uzorke. On je default u R-u i ne zahtijeva jednake varijance. Nema razloga koristiti Studentov test osim za reprodukciju starijih rezultata.

Wilcoxonov test koristite kad t-test ne može. Mali uzorak s nenormalnim podacima, ordinalni podaci, ekstremni outlieri — to su situacije za neparametrijski pristup. Za $n > 50$ s umjerenom normalnosti, t-test je dovoljno robustan.

Veličina učinka je jednako važna kao p-vrijednost. Cohenov d govori koliko je razlika velika u praktičnom smislu. Za upareni test: d = M_razlika / SD_razlika. Za

nezavisni: d = (M1 - M2) / s_pooled. Smjernice: 0.2 mali, 0.5 srednji, 0.8 veliki.
APA format ima svoja pravila. Opis smjera riječima, prosjeci i SD obiju grupa, t(df) = vrijednost, p-vrijednost, Cohenov d. Konzistentnost štedi vrijeme i smanjuje greške.
Outlieri zaslužuju pažnju, ne paniku. Identificirajte ih, pokušajte razumjeti zašto su ekstremni, provedite analizu s njima i bez njih. Ako se zaključci razlikuju, koristite robusne metode i izvijestite oboje.
Formula pristup samo za nezavisni test. `t.test(y ~ grupa)` nikad ne koristite za uparene podatke — R će ignorirati parove i dati pogrešne rezultate.
Snaga testa određuje što možete detektirati. Prije nego zaključite "nema razlike", provjerite jeste li uopće imali dovoljno podataka da razliku detektirate. `power.t.test()` računa potreban n.

12.17 Zadaci za pripremu

1. Učitajte `article_visuals.csv`. Provedite nezavisni t-test za usporedbu razumijevanja (s vizualima) između članaka o zdravlju i članaka o sportu. Izračunajte Cohenov d i napišite rezultat u APA formatu.
2. Za varijablu `credibility_with_visual`, usporedite upareni t-test s Wilcoxonovim testom. Jesu li zaključci konzistentni? Provjerite normalnost razlika QQ plotom.
3. Napišite funkciju `kompletni_ttest(data, var_with, var_no, naziv)` koja automatski provjerava normalnost (Shapiro-Wilk), odabire parametrijski ili neparametrijski test, računa Cohenov d i generira APA rečenicu.

12.18 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 13 (Comparing Two Means). Besplatno dostupno na learningstatisticswithr.com. Detaljan pregled t-testova, pretpostavki i alternativa.

Preporučeno

Field, A. (2018). *Discovering Statistics Using R*. SAGE. Poglavlje 9. Praktičan pristup t-testovima s naglaskom na provjeru pretpostavki i efektivne vizualizacije.

Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science. *Frontiers in Psychology*, 4, 863. Praktičan vodič za izračun i izvještavanje veličina učinka.

12.19 Pojmovnik

Pojam	Objašnjenje
Upareni t-test	Test za usporedbu dvaju mjerenja na istim jedinicama. Testira prosjek razlika. `t.test(x, y, paired = TRUE)`.
Nezavisni t-test	Test za usporedbu prosjeka dviju nezavisnih grupa. Default u R-u je Welchov. `t.test(x, y)`.
Jednouzorački t-test	Test za usporedbu jednog prosjeka s poznatom vrijednošću. `t.test(x, mu = vrijednost)`.
Welchov t-test	Varijanta nezavisnog t-testa koja ne pretpostavlja jednake varijance. Default u R-u. Uvijek bolji izbor.
Normalnost	Pretpostavka normalne distribucije. Za upareni test: normalnost razlika, ne pojedinačnih mjerenja.
Shapiro-Wilkov test	Formalni test normalnosti. `shapiro.test(x)`. H0: podaci su normalni. S velikim n previše osjetljiv.
QQ plot	Dijagnostički graf: točke blizu linije = normalno. `stat_qq()` + `stat_qq_line()` u ggplot2.
Homogenost varijance	Pretpostavka jednakih varijanci u grupama. Potrebna samo za Studentov (ne Welchov) t-test.
Wilcoxonov signed-rank test	Neparametrijska alternativa uparenom t-testu. `wilcox.test(x, y, paired = TRUE)`.
Mann-Whitney U test	Neparametrijska alternativa nezavisnom t-testu. `wilcox.test(x, y)`. Isto kao rank-sum test.
Cohenov d (upareni)	d = M_razlika / SD_razlika. Standardizirana mjera veličine učinka za uparene podatke.
Cohenov d (nezavisni)	d = (M1 minus M2) / s_pooled. Standardizirana mjera veličine učinka za nezavisne grupe.
Pooled SD	Zajednička SD dviju grupa ponderirana njihovim veličinama uzoraka.
Forest plot	Graf za prikaz višestrukih veličina učinka s intervalima pouzdanosti. Standardan u meta-analizama.

Pojam	Objašnjenje
Trimmed mean	Prosjek koji ignorira ekstremne vrijednosti (npr. 10% s oba kraja). `mean(x, trim = 0.10)`.
Outlier	Opažanje ekstremno udaljeno od ostatka. Identifikacija:
APA format	Standardizirani format izvještavanja: M, SD, t(df), p, d. Koristi se u komunikologiji i psihologiji.
Formula pristup	`t.test(y ~ grupa)` sintaksa za nezavisni test. Konzistentno s ANOVA i regresijom. NE za upareni test.
Bootstrap	Metoda ponovnog uzorkovanja s vraćanjem za robustan CI. Ne pretpostavlja normalnost.
`shapiro.test()`	R funkcija za Shapiro-Wilkov test normalnosti. Prima vektor numeričkih podataka.
`wilcox.test()`	R funkcija za Wilcoxonov test. Argumenti isti kao `t.test()`: mu, paired, alternative.
`power.t.test()`	R funkcija za analizu snage t-testa. Računa n, snagu ili detektabilni učinak.

13 Tjedan 12: Usporedba više grupa ANOVA-om

Kad t-test nije dovoljan

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Objasniti zašto višestruki t-testovi nisu primjereni za usporedbu više od dviju grupa.
2. Provesti i interpretirati jednosmjernu ANOVA-u u R-u.
3. Objasniti logiku F-statistike kao omjera varijabilnosti između i unutar grupa.
4. Provjeriti pretpostavke ANOVA-e (normalnost, homogenost varijance).
5. Primijeniti post-hoc testove (Tukey HSD) za identifikaciju specifičnih razlika.
6. Izračunati eta-kvadrat kao mjeru veličine učinka za ANOVA-u.
7. Primijeniti Kruskal-Wallisov test kao neparametrijsku alternativu.
8. Provesti kompletnu analizu s izvještajem.

13.1 Motivacija: vjerodostojnost vijesti po izvoru

Istraživačko pitanje — percipiraju li ljudi istu vijest kao više ili manje vjerodostojnu ovisno o tome iz kojeg izvora dolazi? Konkretno, razlikuje li se percipirana vjerodostojnost vijesti ovisno o tome pripisuje li se izvoru TV, web portal, društvena mreža, tisak ili podcast?

Imamo pet grupa. Prošli tjedan naučili smo t-test za usporedbu dviju grupa. Zašto ne bismo jednostavno proveli t-test za svaki par?

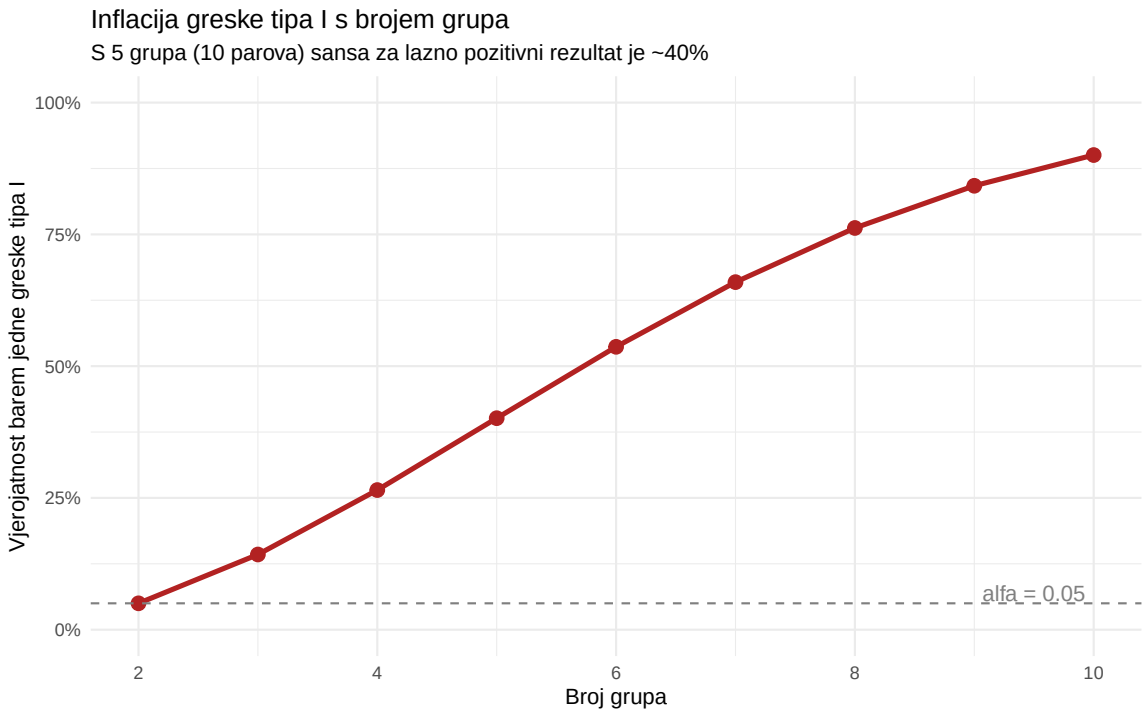
13.1.1 Problem višestrukih t-testova

S pet grupa imamo 10 mogućih parova (5 choose 2 = 10). Ako svaki test provodimo na razini alfa = 0.05, vjerojatnost barem jednog lažno pozitivnog rezultata je:

$$P(\text{barem 1 greska}) = 1 - (1 - 0.05)^{10} = 1 - 0.95^{10} \approx 0.40$$

S 10 testova, šansa za barem jedan lažno pozitivni rezultat je oko 40%. To je potpuno neprihvatljivo.

```
# Inflacija greske tipa I s brojem testova
n_grupa <- 2:10
n_parova <- choose(n_grupa, 2)
alpha_inflated <- 1 - (1 - 0.05)^n_parova

tibble(grupe = n_grupa, parovi = n_parova, alpha = alpha_inflated) |>
  ggplot(aes(x = grupe, y = alpha)) +
  geom_line(linewidth = 1.2, color = "firebrick") +
  geom_point(size = 3, color = "firebrick") +
  geom_hline(yintercept = 0.05, linetype = "dashed", color = "grey50") +
  annotate("text", x = 9.5, y = 0.07, label = "alfa = 0.05", color = "grey50") +
  scale_y_continuous(labels = scales::label_percent(), limits = c(0, 1)) +
  labs(
    title = "Inflacija greske tipa I s brojem grupa",
    subtitle = "S 5 grupa (10 parova) sansa za lazno pozitivni rezultat je ~40%",
    x = "Broj grupa",
    y = "Vjerojatnost barem jedne greske tipa I"
  ) +
  theme_minimal()
```



ANOVA rješava ovaj problem — testira sve grupe odjednom jednim testom, održavajući alfa na 0.05.

13.2 Naši podaci

```
cred <- read_csv("../resources/datasets/news_credibility.csv")
glimpse(cred)
```

```
Rows: 300
Columns: 10
$ participant_id <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ~
$ news_source    <chr> "TV", "TV", "TV", "TV", "TV", "TV", "TV", "TV", "TV", "~
$ age_group      <chr> "30-44", "18-29", "45-59", "45-59", "45-59", "18-
29", "~
$ education      <chr> "visa", "srednja", "visoka", "srednja", "srednja", "sre~
$ topic          <chr> "zdravlje", "zdravlje", "tehnologija", "politika", "zdr~
$ media_literacy <chr> "visoka", "niska", "srednja", "srednja", "srednja", "vi~
$ credibility    <dbl> 4.2, 5.4, 4.2, 3.3, 5.8, 4.4, 3.8, 4.1, 5.6, 5.7, 5.4, ~
$ trust_general  <dbl> 3.5, 3.8, 4.5, 3.6, 6.3, 4.3, 3.4, 3.6, 7.0, 4.7, 6.1, ~
$ share_intent   <dbl> 2, 3, 2, 2, 2, 2, 3, 2, 2, 5, 2, 2, 3, 2, 3, 3, 3, 1, 3~
$ reading_time   <dbl> 3.9, 4.5, 4.1, 1.9, 4.9, 4.6, 1.7, 2.8, 5.0, 3.6, 3.7, ~
```

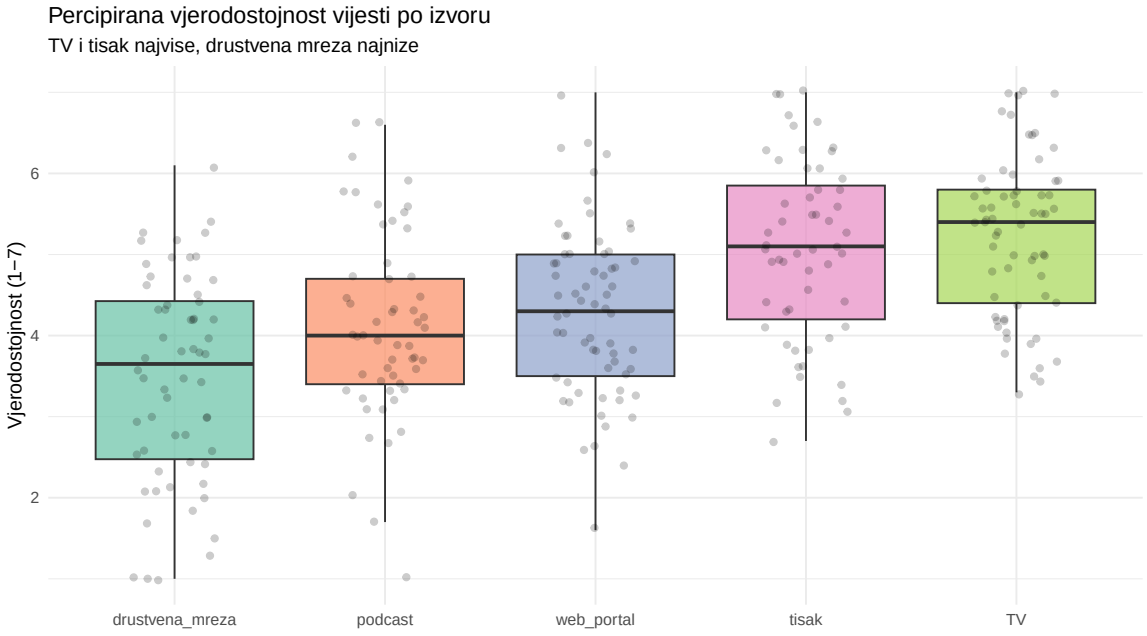
```
cred |>
  group_by(news_source) |>
  summarise(
    n = n(),
    M = round(mean(credibility), 2),
    SD = round(sd(credibility), 2),
    Min = min(credibility),
    Max = max(credibility),
    .groups = "drop"
  ) |>
  arrange(desc(M))
```

```
# A tibble: 5 x 6
  news_source         n     M    SD   Min   Max
  <chr>           <int> <dbl> <dbl> <dbl> <dbl>
1 TV                 65  5.23  0.98   3.3   7
2 tisak              55  5.06  1.12   2.7   7
```

```
3 web_portal          65  4.29  1.05   1.6   7
4 podcast             55  4.12  1.17   1     6.6
5 drustvena_mreza     60  3.49  1.27   1     6.1
```

TV i tisak imaju najvišu percipiranu vjerodostojnost (M > 5), društvena mreža najnižu (M < 3.5). Razlike su očite, ali jesu li statistički značajne kad uzmemo u obzir varijabilnost unutar svake grupe?

```
cred |>
  mutate(news_source = fct_reorder(news_source, credibility)) |>
  ggplot(aes(x = news_source, y = credibility, fill = news_source)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.2, size = 1.5) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Percipirana vjerodostojnost vijesti po izvoru",
    subtitle = "TV i tisak najvise, drustvena mreza najnize",
    x = NULL,
    y = "Vjerodostojnost (1-7)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



13.3 Logika ANOVA-e

ANOVA (Analysis of Variance) uspoređuje varijabilnost IZMEĐU grupa s varijabilnošću UNUTAR grupa.

Intuicija je sljedeća. Ako se grupe stvarno razlikuju, prosjeci grupa bit će razbacani daleko jedni od drugih (velika varijabilnost između grupa). Istovremeno, unutar svake grupe postoji prirodna varijabilnost pojedinaca. Ako je varijabilnost između grupa puno veća od varijabilnosti unutar grupa, onda su razlike između grupa "stvarne" (nije samo šum).

$$F = \frac{\text{varijabilnost IZMEĐU grupa}}{\text{varijabilnost UNUTAR grupa}} = \frac{MS_{between}}{MS_{within}}$$

Ako F je blizu 1, varijabilnost između grupa je podjednaka onoj unutar grupa (nema učinka). Ako je F puno veći od 1, grupe se značajno razlikuju.

13.3.1 Dekompozicija varijance: SS

Ukupnu varijabilnost podataka rastavimo na dva dijela, gdje je:

$$SS_{total} = SS_{between} + SS_{within}$$

```
# Rucni izracun dekompozicije varijance
grand_mean <- mean(cred$credibility)

# SS_total: ukupno odstupanje svakog opazanja od grand mean
ss_total <- sum((cred$credibility - grand_mean)^2)

# SS_between: odstupanje grupnih prosjeka od grand mean (ponderano s n)
group_stats <- cred |>
  group_by(news_source) |>
  summarise(n = n(), M = mean(credibility), .groups = "drop")

ss_between <- sum(group_stats$n * (group_stats$M - grand_mean)^2)

# SS_within: odstupanje opazanja od njihovog grupnog prosjeka
ss_within <- cred |>
  left_join(group_stats |> select(news_source, M), by = "news_source") |>
  summarise(ss = sum((credibility - M)^2)) |>
  pull(ss)

cat("SS_total:  ", round(ss_total, 1), "\n")
```

```
SS_total:   492
```

```r
cat("SS_between:", round(ss_between, 1), "\n")
```

```
SS_between: 122.8
```

```r
cat("SS_within: ", round(ss_within, 1), "\n")
```

```
SS_within:  369.2
```

```r
cat("Provjera:  ", round(ss_between + ss_within, 1), " (treba biti = SS_total)\n\n")
```

```
Provjera:   492  (treba biti = SS_total)
```

```r
# Stupnjevi slobode
k <- nrow(group_stats)   # broj grupa
N <- nrow(cred)           # ukupno opazanja
df_between <- k - 1
df_within <- N - k

# Mean Squares
ms_between <- ss_between / df_between
ms_within <- ss_within / df_within

# F statistika
f_stat <- ms_between / ms_within
p_val <- pf(f_stat, df_between, df_within, lower.tail = FALSE)

cat("df_between:", df_between, "\n")
```

```
df_between: 4
```

```r
cat("df_within: ", df_within, "\n")
```

```
df_within:  295
```

```r
cat("MS_between:", round(ms_between, 2), "\n")
```

```
MS_between: 30.69
```

```r
cat("MS_within: ", round(ms_within, 2), "\n")
```

```
MS_within:  1.25
```

```r
cat("F =", round(f_stat, 2), "\n")
```

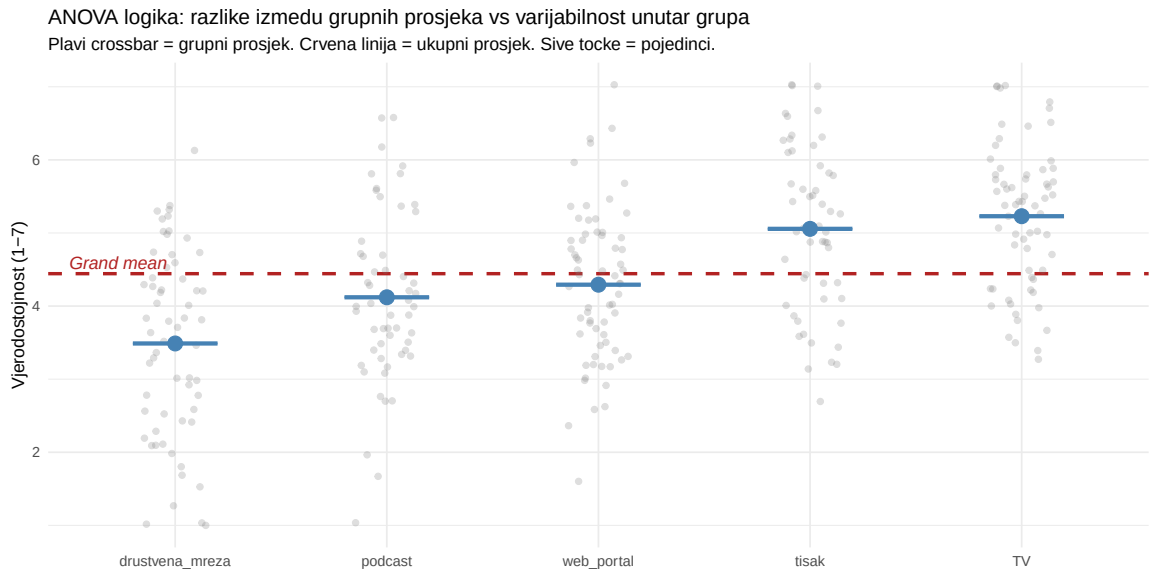
```
F = 24.52
```

```r
cat("p =", format(p_val, scientific = TRUE, digits = 3), "\n")
```

```
p = 1.55e-17
```

MS_between (varijabilnost između grupa) je mnogo veća od MS_within (varijabilnost unutar grupa). F je velik, p je izuzetno mali. Grupe se značajno razlikuju.

```r
# Vizualizacija: ukupna vs objašnjena varijanca
cred_plot <- cred |>
  left_join(group_stats |> select(news_source, group_mean = M), by = "news_source") |>
  mutate(news_source = fct_reorder(news_source, group_mean))

cred_plot |>
  ggplot(aes(x = news_source, y = credibility)) +
  geom_hline(yintercept = grand_mean, color = "firebrick", linewidth = 1, linetype = "dash
  geom_jitter(width = 0.15, alpha = 0.25, color = "grey50") +
  stat_summary(fun = mean, geom = "point", size = 4, color = "steelblue") +
  stat_summary(fun = mean, geom = "crossbar", width = 0.4, color = "steelblue",
               fun.min = mean, fun.max = mean) +
  annotate("text", x = 0.5, y = grand_mean + 0.15, label = "Grand mean",
           color = "firebrick", hjust = 0, fontface = "italic") +
  labs(
    title = "ANOVA logika: razlike izmedu grupnih prosjeka vs varijabilnost unutar grupa",
    subtitle = "Plavi crossbar = grupni prosjek. Crvena linija = ukupni prosjek. Sive tock
    x = NULL, y = "Vjerodostojnost (1-7)"
  ) +
  theme_minimal()
```



13.4 ANOVA u R-u

R koristi funkciju `aov()` za ANOVA-u. Sintaksa je formula pristup: `aov(y ~ grupa, data = ...)`.

```
# Jednosmjerna ANOVA
model <- aov(credibility ~ news_source, data = cred)
summary(model)
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
news_source   4  122.8  30.692   24.52 <2e-16 ***
Residuals   295  369.2   1.252
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA tablica sadrži sve elemente koje smo ručno izračunali, uključujući df, SS (Sum Sq), MS (Mean Sq), F vrijednost i p-vrijednost (Pr(>F)). Rezultat potvrđuje da postoji statistički značajna razlika u percipiranoj vjerodostojnosti ovisno o izvoru vijesti.

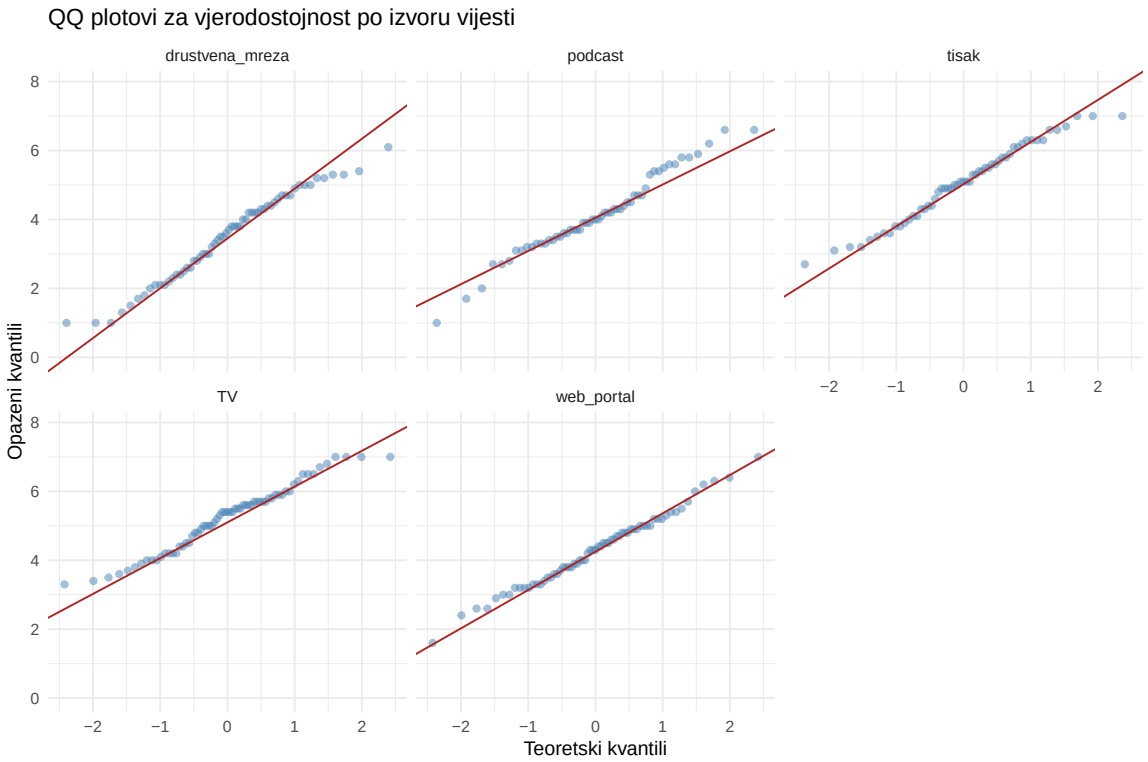
Ali ANOVA je **omnibus test** — govori da se barem dvije grupe razlikuju, ali ne govori KOJE. Za to trebamo post-hoc testove (drugi dio).

13.5 Pretpostavke ANOVA-e

ANOVA ima tri pretpostavke. Neovisnost opažanja (dizajn istraživanja), normalnost distribucije unutar svake grupe (ili dovoljno velik n) i homogenost varijance (jednake varijance u svim grupama).

13.5.1 Provjera normalnosti

```
# QQ plotovi po grupi
cred |>
  ggplot(aes(sample = credibility)) +
  stat_qq(color = "steelblue", alpha = 0.5) +
  stat_qq_line(color = "firebrick") +
  facet_wrap(~news_source) +
  labs(
    title = "QQ plotovi za vjerodostojnost po izvoru vijesti",
    x = "Teoretski kvantili",
    y = "Opazeni kvantili"
  ) +
  theme_minimal()
```

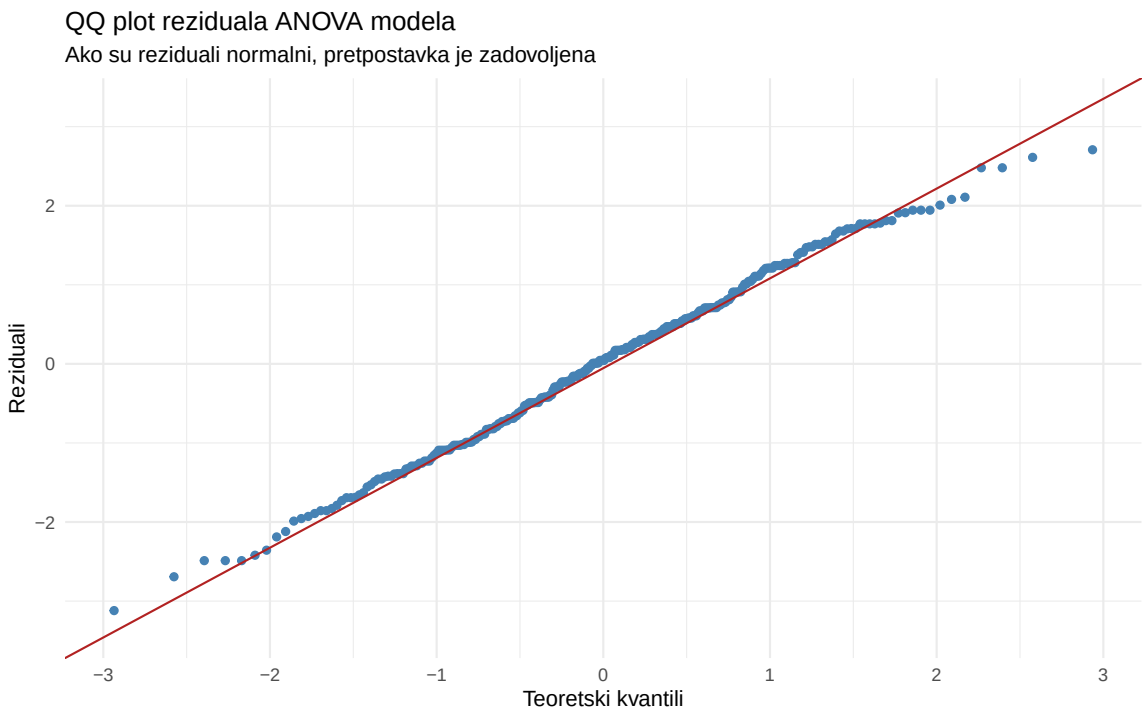


```r
# Shapiro-Wilk po grupi
cred |>
  group_by(news_source) |>
  summarise(
    n = n(),
    shapiro_W = round(shapiro.test(credibility)$statistic, 4),
    shapiro_p = round(shapiro.test(credibility)$p.value, 4),
    normalno = shapiro_p >= 0.05,
    .groups = "drop"
  )
```

```
# A tibble: 5 x 5
  news_source         n shapiro_W shapiro_p normalno
  <chr>           <int>     <dbl>     <dbl> <lgl>
1 TV                 65     0.970     0.117 TRUE
2 drustvena_mreza    60     0.973     0.198 TRUE
3 podcast            55     0.977     0.367 TRUE
4 tisak              55     0.975     0.309 TRUE
5 web_portal         65     0.992     0.964 TRUE
```

Neke grupe možda ne prolaze Shapiro-Wilk test. Ali s $n > 50$ po grupi, ANOVA je robusna na umjerena odstupanja od normalnosti (CLT). Također možemo provjeriti normalnost **reziduala** modela:

```r
# Reziduali modela
tibble(reziduali = residuals(model)) |>
  ggplot(aes(sample = reziduali)) +
  stat_qq(color = "steelblue") +
  stat_qq_line(color = "firebrick") +
  labs(
    title = "QQ plot reziduala ANOVA modela",
    subtitle = "Ako su reziduali normalni, pretpostavka je zadovoljena",
    x = "Teoretski kvantili",
    y = "Reziduali"
  ) +
  theme_minimal()
```



13.5.2 Provjera homogenosti varijance: Levenov test

```
# Rucna provjera: omjer najvece i najmanje varijance
var_by_group <- cred |>
  group_by(news_source) |>
  summarise(var = var(credibility), sd = round(sd(credibility), 2), .groups = "drop")

var_by_group
```

```
# A tibble: 5 x 3
  news_source       var    sd
  <chr>           <dbl> <dbl>
1 TV              0.963  0.98
2 drustvena_mreza 1.63   1.27
3 podcast         1.36   1.17
4 tisak           1.25   1.12
5 web_portal      1.11   1.05
```

```
cat("\nOmjer max/min varijance:", round(max(var_by_group$var) / min(var_by_group$var), 2),
```

```
Omjer max/min varijance: 1.69
```

```r
cat("Pravilo palca - ako je omjer < 3, homogenost je prihvatljiva.\n")
```

```
Pravilo palca - ako je omjer < 3, homogenost je prihvatljiva.
```

```r
# Levenov test (rucna implementacija bazirana na apsolutnim devijacijama)
cred_levene <- cred |>
  left_join(group_stats |> select(news_source, group_median = M), by = "news_source") |>
  mutate(abs_dev = abs(credibility - group_median))

levene_anova <- aov(abs_dev ~ news_source, data = cred_levene)
levene_p <- summary(levene_anova)[[1]]["Pr(>F)"][1]

cat("Levenov test (ANOVA na apsolutnim devijacijama):\n")
```

```
Levenov test (ANOVA na apsolutnim devijacijama):
```

```r
cat("p =", round(levene_p, 4), "\n")
```

```
p = 0.1895
```

```r
cat("Homogenost varijance:", if_else(levene_p >= 0.05, "zadovoljena", "narusena"), "\n")
```

```
Homogenost varijance: zadovoljena
```

Ako je Levenov test značajan (varijance nejednake), koristimo **Welchovu ANOVA-u** koja ne pretpostavlja jednake varijance:

```r
# Welchova ANOVA (ne pretpostavlja jednake varijance)
oneway.test(credibility ~ news_source, data = cred, var.equal = FALSE)
```

```

    One-way analysis of means (not assuming equal variances)

data:  credibility and news_source
F = 23.513, num df = 4.00, denom df = 145.15, p-value = 5.312e-15
```

```r
# Usporedba klasicne i Welchove ANOVA
klasicna <- summary(model)[[1]]
welch <- oneway.test(credibility ~ news_source, data = cred, var.equal = FALSE)

cat("Klasicna ANOVA:  F(", df_between, ",", N - k, ") = ",
    round(klasicna$`F value`[1], 2), ", p < 0.001\n", sep = "")
```

```
Klasicna ANOVA:  F(4,295) = 24.52, p < 0.001
```

```
cat("Welchova ANOVA:  F(", welch$parameter[1], ",", round(welch$parameter[2], 1),
    ") = ", round(welch$statistic, 2), ", p < 0.001\n", sep = "")
```

```
Welchova ANOVA:  F(4,145.2) = 23.51, p < 0.001
```

```
cat("\nOba pristupa daju isti zakljucak.\n")
```

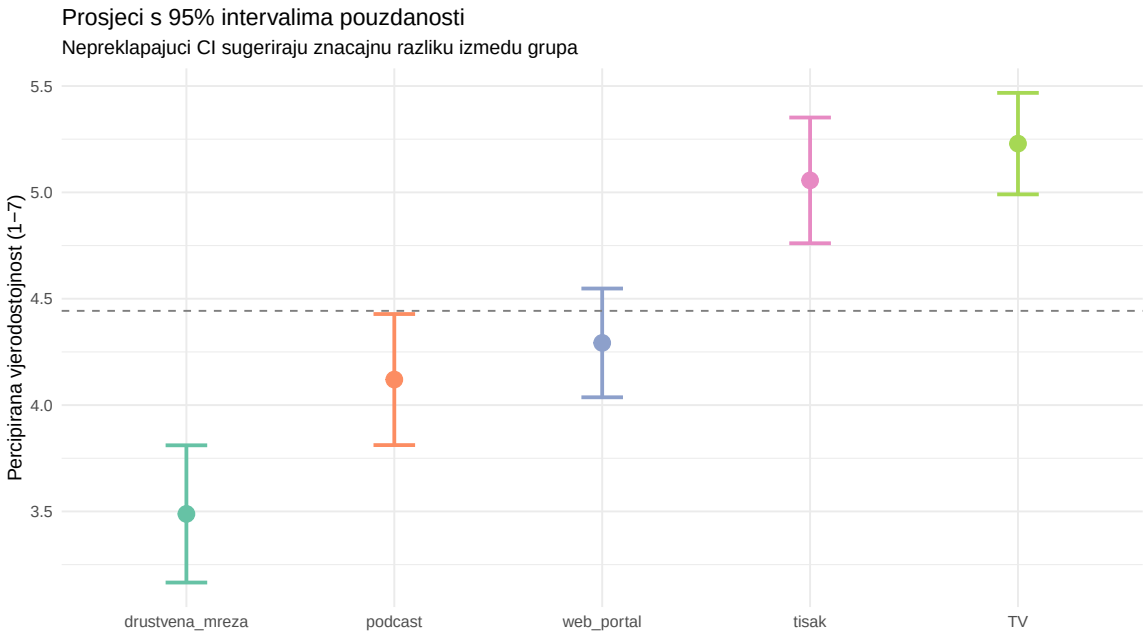
```
Oba pristupa daju isti zakljucak.
```

> **Praktični savjet**
>
> Kao i kod t-testa, preporučujemo Welchovu ANOVA-u (`oneway.test(y ~ grupa, var.equal = FALSE)`) kao standardni izbor. Kad su varijance jednake, daje iste rezultate kao klasična ANOVA. Kad su nejednake, daje točnije rezultate. Klasičnu ANOVA-u (`aov()`) koristite kad trebate rezidualne dijagnostike ili post-hoc testove koji zahtijevaju `aov` objekt.

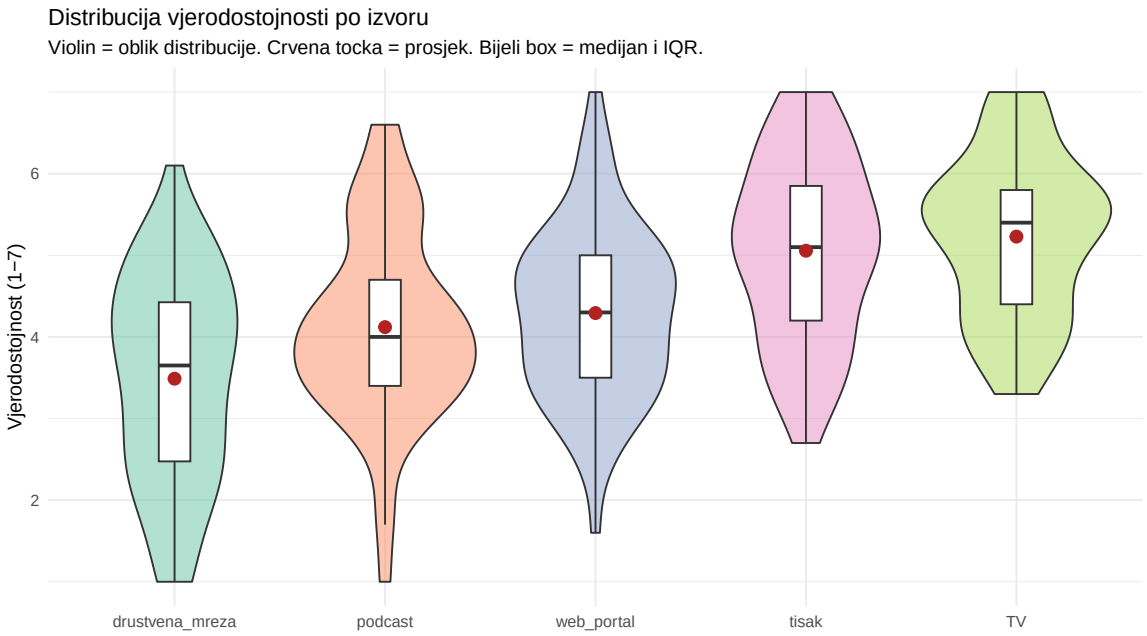
13.6 Vizualizacija ANOVA rezultata

```
# Prosjeci s 95% CI
cred |>
  group_by(news_source) |>
  summarise(
    M = mean(credibility),
    SE = sd(credibility) / sqrt(n()),
    CI_lo = M - 1.96 * SE,
    CI_hi = M + 1.96 * SE,
    .groups = "drop"
  ) |>
  mutate(news_source = fct_reorder(news_source, M)) |>
  ggplot(aes(x = news_source, y = M, color = news_source)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymin = CI_lo, ymax = CI_hi), width = 0.2, linewidth = 1) +
  geom_hline(yintercept = grand_mean, linetype = "dashed", color = "grey50") +
  scale_color_brewer(palette = "Set2") +
```

```
  labs(
    title = "Prosjeci s 95% intervalima pouzdanosti",
    subtitle = "Nepreklapajuci CI sugeriraju znacajnu razliku izmedu grupa",
    x = NULL,
    y = "Percipirana vjerodostojnost (1-7)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```




```
# Violin plot s prosjekom
cred |>
  mutate(news_source = fct_reorder(news_source, credibility)) |>
  ggplot(aes(x = news_source, y = credibility, fill = news_source)) +
  geom_violin(alpha = 0.5) +
  geom_boxplot(width = 0.15, fill = "white", outlier.shape = NA) +
  stat_summary(fun = mean, geom = "point", size = 3, color = "firebrick") +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Distribucija vjerodostojnosti po izvoru",
    subtitle = "Violin = oblik distribucije. Crvena tocka = prosjek. Bijeli box = medijan
    x = NULL,
    y = "Vjerodostojnost (1-7)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



Podsjetnik

U prvom dijelu naučili smo logiku ANOVA-e (F = varijabilnost između / varijabilnost unutar), dekompoziciju varijance (SS_total = SS_between + SS_within), pretpostavke (normalnost, homogenost varijance) i osnovno provođenje `aov()` i `oneway.test()`. ANOVA je pokazala da se grupe značajno razlikuju ($p < 0.001$). Sada utvrđujemo KOJE se grupe razlikuju.

13.7 Post-hoc testovi: Tukey HSD

ANOVA je omnibus test — govori da se barem dvije grupe razlikuju, ali ne govori koje. **Post-hoc testovi** uspoređuju sve parove grupa uz kontrolu greške tipa I.

Najčešći post-hoc test je **Tukey HSD** (Honestly Significant Difference). On testira sve parove istovremeno i prilagođava p-vrijednosti tako da ukupna greška tipa I ostane na alfa = 0.05.

```
cred <- read_csv("../resources/datasets/news_credibility.csv")
model <- aov(credibility ~ news_source, data = cred)

# Tukey HSD
tukey_rez <- TukeyHSD(model)
tukey_rez
```

```
  Tukey multiple comparisons of means
```

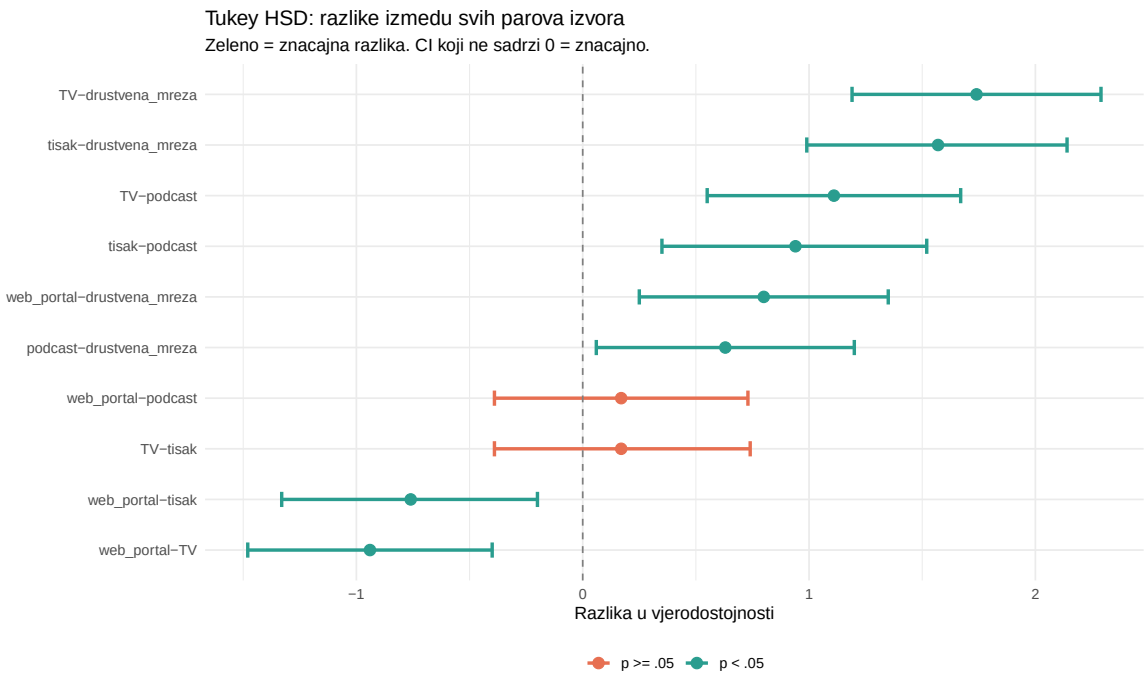
```
    95% family-wise confidence level

Fit: aov(formula = credibility ~ news_source, data = cred)

$news_source
                                  diff         lwr        upr     p adj
podcast-drustvena_mreza      0.6316667  0.05842531  1.2049080 0.0225724
tisak-drustvena_mreza        1.5680303  0.99478894  2.1412717 0.0000000
TV-drustvena_mreza           1.7408974  1.19114360  2.2906513 0.0000000
web_portal-drustvena_mreza   0.8039744  0.25422052  1.3537282 0.0007187
tisak-podcast                0.9363636  0.35079309  1.5219342 0.0001538
TV-podcast                   1.1092308  0.54663279  1.6718287 0.0000013
web_portal-podcast           0.1723077 -0.39029029  0.7349057 0.9177070
TV-tisak                     0.1728671 -0.38973085  0.7354651 0.9168041
web_portal-tisak            -0.7640559 -1.32665392 -0.2014580 0.0021451
web_portal-TV               -0.9369231 -1.47556963 -0.3982765 0.0000279
```

```r
# Preglednija tablica
tukey_df <- as_tibble(tukey_rez$news_source, rownames = "par") |>
  mutate(
    razlika = round(diff, 2),
    CI_lo = round(lwr, 2),
    CI_hi = round(upr, 2),
    p = round(`p adj`, 4),
    znacajno = p < 0.05
  ) |>
  select(par, razlika, CI_lo, CI_hi, p, znacajno) |>
  arrange(p)

tukey_df
```

```
# A tibble: 10 x 6
   par                         razlika CI_lo CI_hi      p znacajno
   <chr>                         <dbl> <dbl> <dbl>  <dbl> <lgl>
 1 tisak-drustvena_mreza          1.57  0.99  2.14 0      TRUE
 2 TV-drustvena_mreza             1.74  1.19  2.29 0      TRUE
 3 TV-podcast                     1.11  0.55  1.67 0      TRUE
 4 web_portal-TV                 -0.94 -1.48 -0.4  0      TRUE
 5 tisak-podcast                  0.94  0.35  1.52 0.0002 TRUE
 6 web_portal-drustvena_mreza     0.8   0.25  1.35 0.0007 TRUE
 7 web_portal-tisak              -0.76 -1.33 -0.2  0.0021 TRUE
 8 podcast-drustvena_mreza        0.63  0.06  1.2  0.0226 TRUE
 9 TV-tisak                       0.17 -0.39  0.74 0.917  FALSE
10 web_portal-podcast             0.17 -0.39  0.73 0.918  FALSE
```

Tablica pokazuje razliku prosjeka za svaki par, 95% CI za tu razliku i prilagođenu p-vrijednost. Značajni parovi ($p < 0.05$) su oni gdje CI ne uključuje nulu.

```
# Vizualizacija Tukey rezultata
tukey_df |>
  mutate(par = fct_reorder(par, razlika)) |>
  ggplot(aes(y = par)) +
  geom_errorbarh(aes(xmin = CI_lo, xmax = CI_hi, color = znacajno),
                 height = 0.3, linewidth = 1) +
  geom_point(aes(x = razlika, color = znacajno), size = 3) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  scale_color_manual(values = c("TRUE" = "#2a9d8f", "FALSE" = "#e76f51"),
                     labels = c("TRUE" = "p < .05", "FALSE" = "p >= .05")) +
  labs(
    title = "Tukey HSD: razlike izmedu svih parova izvora",
    subtitle = "Zeleno = znacajna razlika. CI koji ne sadrzi 0 = znacajno.",
    x = "Razlika u vjerodostojnosti",
    y = NULL,
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Iz grafa se jasno vide obrasci. Društvena mreža ima značajno nižu vjerodostojnost od svih drugih izvora. TV i tisak nemaju značajnu razliku međusobno (oba su visoki). Podcast i web portal su u sredini.

13.7.1 Compact letter display

Česta praksa u izvještavanju je grupiranje izvora koji se NE razlikuju značajno. To zovemo "compact letter display" (CLD).

```
# Rucna interpretacija Tukey rezultata u grupe
# Grupe koje se NE razlikuju znacajno dijele isto slovo
group_stats <- cred |>
  group_by(news_source) |>
  summarise(M = round(mean(credibility), 2), .groups = "drop") |>
  arrange(desc(M))

cat("Grupiranje izvora (isti skup slova = nema znacajne razlike):\n\n")
```

```
Grupiranje izvora (isti skup slova = nema znacajne razlike):
```

```
# Na temelju Tukey rezultata:
znacajni_parovi <- tukey_df |> filter(znacajno) |> pull(par)
cat("Znacajno razliciti parovi:\n")
```

```
Znacajno razliciti parovi:
```

```
for (p in znacajni_parovi) cat("  ", p, "\n")
```

```
   tisak-drustvena_mreza
   TV-drustvena_mreza
   TV-podcast
   web_portal-TV
   tisak-podcast
   web_portal-drustvena_mreza
   web_portal-tisak
   podcast-drustvena_mreza
```

13.8 Veličina učinka: eta-kvadrat

Kao i kod t-testa, p-vrijednost ne govori koliko je učinak velik. **Eta-kvadrat** (eta2) je mjera veličine učinka za ANOVA-u:

$$\eta^2 = \frac{SS_{between}}{SS_{total}}$$

Eta-kvadrat je proporcija ukupne varijabilnosti koja je objašnjena grupnom pripadnošću. Interpretira se kao R-kvadrat u regresiji.

```
# Iz ANOVA tablice
anova_table <- summary(model)[[1]]
ss_between <- anova_table$`Sum Sq`[1]
ss_within <- anova_table$`Sum Sq`[2]
ss_total <- ss_between + ss_within

eta2 <- ss_between / ss_total

cat("SS_between:", round(ss_between, 1), "\n")
```

```
SS_between: 122.8
```

```
cat("SS_total:  ", round(ss_total, 1), "\n")
```

```
SS_total:   492
```

```
cat("Eta-kvadrat:", round(eta2, 3), "\n")
```

```
Eta-kvadrat: 0.25
```

```
cat("Interpretacija:", round(eta2 * 100, 1), "% varijabilnosti u vjerodostojnosti\n")
```

```
Interpretacija: 25 % varijabilnosti u vjerodostojnosti
```

```
cat("je objasnjeno izvorom vijesti.\n")
```

```
je objasnjeno izvorom vijesti.
```

13.8.1 Interpretacija eta-kvadrata

Cohen (1988) je predložio sljedeće smjernice za interpretaciju eta-kvadrata.

```
tribble(
  ~eta2, ~interpretacija,
  "0.01", "Mali ucinak",
  "0.06", "Srednji ucinak",
  "0.14", "Veliki ucinak"
)
```

```
# A tibble: 3 x 2
  eta2  interpretacija
  <chr> <chr>
1 0.01  Mali ucinak
2 0.06  Srednji ucinak
3 0.14  Veliki ucinak
```

Naš eta2 0.25 je veliki učinak. Izvor vijesti objašnjava značajan dio varijabilnosti u percipiranoj vjerodostojnosti.

13.8.2 Omega-kvadrat: manje pristrana mjera

Eta-kvadrat je pristran (precjenjuje veličinu učinka u populaciji). **Omega-kvadrat** je manje pristrana alternativa:

$$\omega^2 = \frac{SS_{between} - (k-1) \cdot MS_{within}}{SS_{total} + MS_{within}}$$

```
ms_within <- anova_table$`Mean Sq`[2]
k <- length(unique(cred$news_source))

omega2 <- (ss_between - (k - 1) * ms_within) / (ss_total + ms_within)

cat("Eta-kvadrat:  ", round(eta2, 3), "\n")
```

```
Eta-kvadrat:   0.25
```

```
cat("Omega-kvadrat:", round(omega2, 3), "\n")
```

```
Omega-kvadrat: 0.239
```

```
cat("Razlika je mala za velike uzorke, ali omega2 je tocnija procjena.\n")
```

```
Razlika je mala za velike uzorke, ali omega2 je tocnija procjena.
```

> Praktični savjet
>
> Izvijestite eta-kvadrat (jer je poznatiji) ili omega-kvadrat (jer je manje pristran). Za objavljivanje u časopisima, sve češće se traži omega-kvadrat. U praksi, za n > 50 po grupi, razlika je minimalna.

13.9 Planirane usporedbe

Ponekad unaprijed znamo koje usporedbe nas zanimaju. Umjesto da uspoređujemo sve parove (Tukey), možemo specificirati **planirane usporedbe** (contrasts). Ovo je snažniji pristup jer testira manje hipoteza.

```
# Planirana usporedba 1: tradicionalni (TV + tisak) vs digitalni (portal + mreza + podcast
cred <- cred |>
  mutate(tip_medija = if_else(
    news_source %in% c("TV", "tisak"), "tradicionalni", "digitalni"
  ))

t.test(credibility ~ tip_medija, data = cred)
```

```
    Welch Two Sample t-test

data:  credibility by tip_medija
t = -8.9741, df = 278.64, p-value < 2.2e-16
alternative hypothesis: true difference in means between group digitalni and group tradici
95 percent confidence interval:
 -1.436806 -0.919861
sample estimates:
    mean in group digitalni mean in group tradicionalni
                   3.971667                    5.150000
```

```
# Cohenov d za ovu usporedbu
trad <- cred |> filter(tip_medija == "tradicionalni") |> pull(credibility)
dig <- cred |> filter(tip_medija == "digitalni") |> pull(credibility)
s_p <- sqrt(((length(trad)-1)*sd(trad)^2 + (length(dig)-1)*sd(dig)^2) / (length(trad)+leng
d_td <- (mean(trad) - mean(dig)) / s_p

cat("Tradicionalni M:", round(mean(trad), 2), "\n")
```

```
Tradicionalni M: 5.15
```

```
cat("Digitalni M:", round(mean(dig), 2), "\n")
```

```
Digitalni M: 3.97
```

```
cat("Cohenov d:", round(d_td, 2), "(veliki ucinak)\n")
```

```
Cohenov d: 1.03 (veliki ucinak)
```

Planirana usporedba (tradicionalni vs digitalni) daje jasnu sliku — tradicionalni mediji imaju značajno višu percipiranu vjerodostojnost od digitalnih.

13.10 Kruskal-Wallisov test

Kruskal-Wallisov test je neparametrijska alternativa jednosmjernoj ANOVA-i. Koristi rangove umjesto izvornih vrijednosti i ne pretpostavlja normalnost.

```
# Kruskal-Wallis test
kw_test <- kruskal.test(credibility ~ news_source, data = cred)
kw_test
```

```
    Kruskal-Wallis rank sum test

data:  credibility by news_source
Kruskal-Wallis chi-squared = 70.893, df = 4, p-value = 1.471e-14
```

```
# Usporedba ANOVA i Kruskal-Wallis
cat("Klasicna ANOVA:    F = ", round(summary(model)[[1]]$`F value`[1], 2),
    ", p < 0.001\n", sep = "")
```

```
Klasicna ANOVA:    F = 24.52, p < 0.001
```

```
cat("Kruskal-Wallis:    H = ", round(kw_test$statistic, 2),
    ", p < 0.001\n", sep = "")
```

```
Kruskal-Wallis:    H = 70.89, p < 0.001
```

```
cat("\nOba testa daju isti zakljucak.\n")
```

```

Oba testa daju isti zakljucak.
```

13.10.1 Post-hoc za Kruskal-Wallis: Dunn test

Kruskal-Wallis test je omnibus. Za identifikaciju specifičnih razlika koristimo **Dunnov test** (neparametrijski ekvivalent Tukeyu).

```r
# Pairwise Wilcoxon s Bonferroni korekcijom (ugraden u R)
pw_wilcox <- pairwise.wilcox.test(cred$credibility, cred$news_source,
                                  p.adjust.method = "BH")
pw_wilcox
```

```
    Pairwise comparisons using Wilcoxon rank sum test with continuity correction

data:  cred$credibility and cred$news_source

           drustvena_mreza podcast tisak   TV
podcast    0.02530         -       -       -
tisak      3.0e-08         0.00019 -       -
TV         7.3e-11         1.8e-06 0.40500 -
web_portal 0.00128         0.40500 0.00048 3.4e-06

P value adjustment method: BH
```

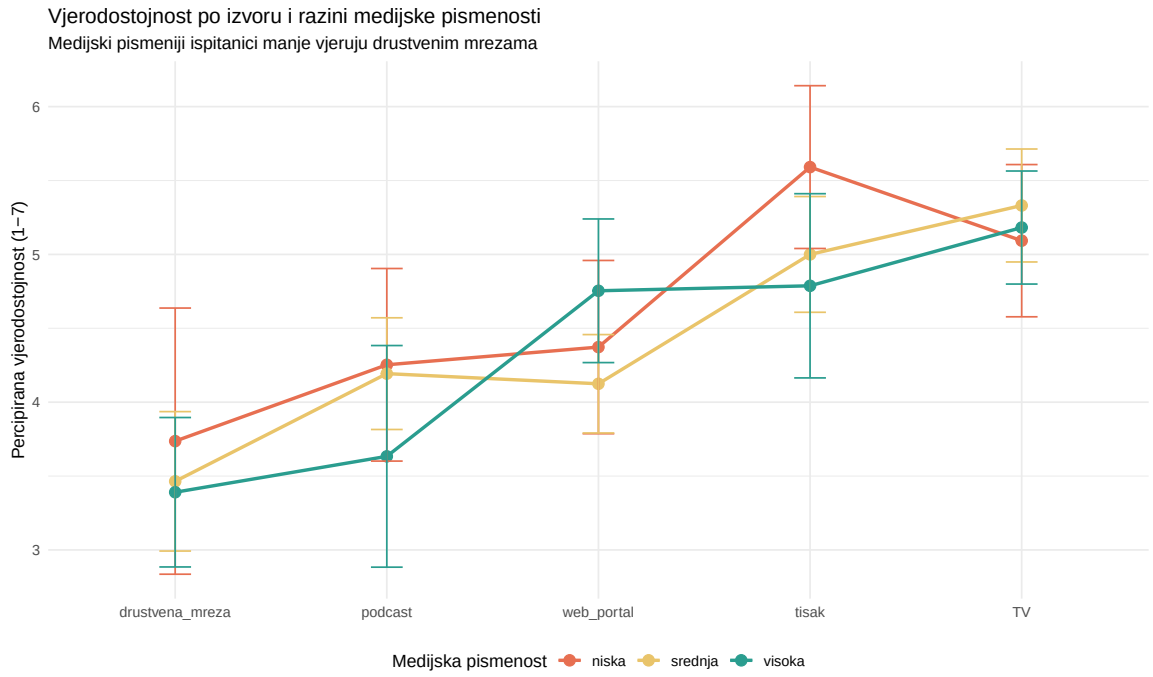
13.11 Potpuna analiza: izvještaj

```r
# Kompletna opisna statistika
cred |>
  group_by(news_source) |>
  summarise(
    n = n(),
    M = round(mean(credibility), 2),
    SD = round(sd(credibility), 2),
    Median = median(credibility),
    SE = round(sd(credibility) / sqrt(n()), 2),
    .groups = "drop"
  ) |>
  arrange(desc(M))
```

```
# A tibble: 5 x 6
  news_source         n     M    SD Median    SE
  <chr>           <int> <dbl> <dbl>  <dbl> <dbl>
1 TV                 65  5.23  0.98   5.4   0.12
2 tisak              55  5.06  1.12   5.1   0.15
3 web_portal         65  4.29  1.05   4.3   0.13
```

```
4 podcast            55  4.12  1.17   4      0.16
5 drustvena_mreza    60  3.49  1.27   3.65   0.16
```

```r
# Moderacija: medijska pismenost
cred |>
  mutate(
    news_source = fct_reorder(news_source, credibility),
    media_literacy = factor(media_literacy, levels = c("niska", "srednja", "visoka"))
  ) |>
  group_by(news_source, media_literacy) |>
  summarise(M = mean(credibility), SE = sd(credibility)/sqrt(n()), .groups = "drop") |>
  ggplot(aes(x = news_source, y = M, color = media_literacy, group = media_literacy)) +
  geom_line(linewidth = 1) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = M - 1.96 * SE, ymax = M + 1.96 * SE), width = 0.15) +
  scale_color_manual(values = c("niska" = "#e76f51", "srednja" = "#e9c46a", "visoka" = "#2
  labs(
    title = "Vjerodostojnost po izvoru i razini medijske pismenosti",
    subtitle = "Medijski pismeniji ispitanici manje vjeruju drustvenim mrezama",
    x = NULL,
    y = "Percipirana vjerodostojnost (1-7)",
    color = "Medijska pismenost"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```r
# ANOVA unutar svake dobne skupine
cred |>
  group_by(age_group) |>
  summarise(
    F_val = round(summary(aov(credibility ~ news_source))[[1]]$`F value`[1], 2),
    p = summary(aov(credibility ~ news_source))[[1]]$`Pr(>F)`[1],
    eta2 = {
      a <- summary(aov(credibility ~ news_source))[[1]]
      round(a$`Sum Sq`[1] / sum(a$`Sum Sq`), 3)
    },
    .groups = "drop"
  ) |>
  mutate(p = format(p, scientific = TRUE, digits = 2))
```

```
# A tibble: 4 x 4
  age_group F_val p        eta2
  <chr>     <dbl> <chr>   <dbl>
1 18-29      6.36 1.6e-04 0.233
2 30-44     10.1  6.5e-07 0.286
3 45-59      6.2  2.8e-04 0.276
4 60+        3.49 1.9e-02 0.317
```

Efekt izvora na vjerodostojnost je značajan u svim dobnim skupinama, ali eta-kvadrat varira. Ovo sugerira da je obrazac (TV > tisak > portal > podcast > mreža) konzistentan, ali jačina razlika može varirati po dobi.

```r
cat("================================================================\n")
```

```
================================================================
```

```r
cat("   APA IZVJESTAJ: PERCIPIRANA VJERODOSTOJNOST PO IZVORU\n")
```

```
   APA IZVJESTAJ: PERCIPIRANA VJERODOSTOJNOST PO IZVORU
```

```r
cat("================================================================\n\n")
```

```
================================================================
```

```r
cat("Jednosmjerna ANOVA pokazala je statisticki znacajnu razliku u\n")
```

```
Jednosmjerna ANOVA pokazala je statisticki znacajnu razliku u
```

```r
cat("percipiranoj vjerodostojnosti vijesti ovisno o izvoru,\n")
```

```
percipiranoj vjerodostojnosti vijesti ovisno o izvoru,
```

```r
cat("F(", k-1, ", ", nrow(cred)-k, ") = ",
    round(summary(model)[[1]]$`F value`[1], 2),
    ", p < .001, eta2 = ", round(eta2, 2), ".\n\n", sep = "")
```

```
F(4, 295) = 24.52, p < .001, eta2 = 0.25.
```

```r
cat("Tukey HSD post-hoc testovi pokazali su da:\n")
```

```
Tukey HSD post-hoc testovi pokazali su da:
```

```r
znac <- tukey_df |> filter(znacajno)
for (i in 1:nrow(znac)) {
  r <- znac[i, ]
  cat("  ", r$par, ": razlika = ", r$razlika,
      ", 95% CI [", r$CI_lo, ", ", r$CI_hi, "], p = ",
      if_else(r$p < 0.001, "< .001", as.character(round(r$p, 3))), "\n", sep = "")
}
```

```
  tisak-drustvena_mreza: razlika = 1.57, 95% CI [0.99, 2.14], p = < .001
  TV-drustvena_mreza: razlika = 1.74, 95% CI [1.19, 2.29], p = < .001
  TV-podcast: razlika = 1.11, 95% CI [0.55, 1.67], p = < .001
  web_portal-TV: razlika = -0.94, 95% CI [-1.48, -0.4], p = < .001
  tisak-podcast: razlika = 0.94, 95% CI [0.35, 1.52], p = < .001
  web_portal-drustvena_mreza: razlika = 0.8, 95% CI [0.25, 1.35], p = < .001
  web_portal-tisak: razlika = -0.76, 95% CI [-1.33, -0.2], p = 0.002
  podcast-drustvena_mreza: razlika = 0.63, 95% CI [0.06, 1.2], p = 0.023
```

```r
cat("\nTV (M = ", round(mean(cred$credibility[cred$news_source == "TV"]), 2),
    ") i tisak (M = ", round(mean(cred$credibility[cred$news_source == "tisak"]), 2),
    ") imaju\nnajvisu vjerodostojnost. Drustvena mreza (M = ",
    round(mean(cred$credibility[cred$news_source == "drustvena_mreza"]), 2),
    ") ima najnizu.\n", sep = "")
```

```
TV (M = 5.23) i tisak (M = 5.06) imaju
najvisu vjerodostojnost. Drustvena mreza (M = 3.49) ima najnizu.
```

```r
cat("Medijska pismenost moderira efekt: visoko pismeni\n")
```

```
Medijska pismenost moderira efekt: visoko pismeni
```

```r
cat("ispitanici manje vjeruju vijestima s drustvenih mreza.\n")
```

```
ispitanici manje vjeruju vijestima s drustvenih mreza.
```

13.12 Dijagram odlučivanja: ANOVA ili nešto drugo?

```r
tribble(
  ~situacija, ~test,
  "Dvije nezavisne grupe", "Nezavisni t-test (tjedan 12)",
  "Dvije uparene grupe", "Upareni t-test (tjedan 12)",
  "Tri+ nezavisne grupe, normalni podaci", "Jednosmjerna ANOVA + Tukey HSD",
  "Tri+ nezavisne grupe, nejednake varijance", "Welchova ANOVA + Games-Howell",
  "Tri+ nezavisne grupe, nenormalni/ordinalni", "Kruskal-Wallis + Dunn",
  "Tri+ uparene grupe", "Repeated measures ANOVA (napredno)",
  "Dva faktora istovremeno", "Dvosmjerna ANOVA (napredno)",
  "Kategoricka x kategoricka varijabla", "Hi-kvadrat test (tjedan 11)"
)
```

```
# A tibble: 8 x 2
  situacija                                   test
  <chr>                                       <chr>
1 Dvije nezavisne grupe                       Nezavisni t-test (tjedan 12)
2 Dvije uparene grupe                         Upareni t-test (tjedan 12)
3 Tri+ nezavisne grupe, normalni podaci       Jednosmjerna ANOVA + Tukey HSD
4 Tri+ nezavisne grupe, nejednake varijance   Welchova ANOVA + Games-Howell
5 Tri+ nezavisne grupe, nenormalni/ordinalni Kruskal-Wallis + Dunn
6 Tri+ uparene grupe                          Repeated measures ANOVA (napredno)
7 Dva faktora istovremeno                     Dvosmjerna ANOVA (napredno)
8 Kategoricka x kategoricka varijabla         Hi-kvadrat test (tjedan 11)
```

! Ključni zaključci

1. ANOVA uspoređuje prosjeke tri ili više grupa jednim testom, kontrolirajući grešku tipa I. Višestruki t-testovi inflacioniraju alfa i nisu prihvatljivi.
2. F statistika = MS_between / MS_within. Velik F znači da su razlike između grupa veće od varijabilnosti unutar grupa.
3. Dekompozicija varijance — SS_total = SS_between + SS_within. ANOVA testira je li SS_between dovoljno velik u odnosu na SS_total.
4. ANOVA je omnibus test — govori da razlika postoji, ali ne govori KOJE se grupe razlikuju. Post-hoc testovi identificiraju specifične parove.
5. Tukey HSD (`TukeyHSD(model)`) uspoređuje sve parove uz kontrolu ukupne greške tipa I. Daje razlike, CI i prilagođene p-vrijednosti za svaki par.
6. Eta-kvadrat = SS_between / SS_total. Proporcija varijabilnosti objašnjena grupom. Smjernice za interpretaciju — 0.01 mali, 0.06 srednji, 0.14 veliki učinak. Omega-kvadrat je manje pristrana alternativa.
7. Pretpostavke uključuju nezavisnost, normalnost unutar grupa (ili n > 30) i homogenost varijance. Welchova ANOVA (`oneway.test(var.equal = FALSE)`) ne zahtijeva jednake varijance.
8. Kruskal-Wallisov test (`kruskal.test()`) je neparametrijska alternativa. Koristi rangove. Post-hoc: `pairwise.wilcox.test()` s BH korekcijom.
9. Planirane usporedbe (contrasts) su snažniji pristup kad unaprijed znate koje grupe želite usporediti (npr. tradicionalni vs digitalni mediji).
10. APA format za ANOVA-u uključuje F(df_between, df_within) = vrijednost, p, eta2, kao i post-hoc razlike s CI.
11. Moderacijska analiza (ANOVA po podgrupama treće varijable) otkriva je li obrazac konzistentan ili varira ovisno o nekoj trećoj varijabli (npr. medijska pismenost, dob).
12. Za dva faktora istovremeno (npr. izvor x dob) koristite dvosmjernu ANOVA-u. Za uparena mjerenja kroz tri+ uvjeta koristite repeated measures ANOVA-u. Ovo su napredni pristupi izvan dosega ovog kolegija.

13.13 Zadaci za pripremu

1. Učitajte `news_credibility.csv`. Provedite jednosmjernu ANOVA-u za varijablu `trust_general` po `news_source`. Izračunajte eta-kvadrat i provedite Tukey HSD. Koji se parovi izvora značajno razlikuju?
2. Provedite Kruskal-Wallisov test za `share_intent` po `news_source`. Usporedite rezultat s klasičnom ANOVA-om. Jesu li zaključci konzistentni?
3. Testirajte planiranu usporedbu gdje trebate odrediti razlikuje li se `credibility` između tri razine medijske pismenosti (`media_literacy`). Koji par se razlikuje prema Tukey HSD testu?

13.14 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 14 (Comparing Several Means). Besplatno dostupno na learningstatisticswithr.com. Pokriva jednosmjernu ANOVA-u, post-hoc testove i veličinu učinka.

Preporučeno

Field, A. (2018). *Discovering Statistics Using R*. SAGE. Poglavlje 10. Detaljna obrada ANOVA-e s dijagnostikom i vizualizacijama.

Maxwell, S. E., Delaney, H. D., & Kelley, K. (2018). *Designing Experiments and Analyzing Data* (3rd edition). Routledge. Referentni udžbenik za eksperimentalne dizajne i ANOVA-u.

13.15 Pojmovnik

Pojam	Objašnjenje
ANOVA	Analysis of Variance. Omnibus test za usporedbu prosjeka tri ili više grupa.
Jednosmjerna ANOVA	ANOVA s jednim faktorom (jednom nezavisnom varijablom).
F statistika	Omjer MS_between / MS_within. F » 1 sugerira značajne razlike između grupa.

Pojam	Objašnjenje
SS (Sum of Squares)	Mjera varijabilnosti. SS_total = SS_between + SS_within.
MS (Mean Square)	SS / df. MS_between = SS_between / (k-1). MS_within = SS_within / (N-k).
Omnibus test	Test koji detektira da razlika postoji negdje, ali ne govori gdje specifično.
Post-hoc test	Test koji se provodi NAKON značajne ANOVA-e za identifikaciju specifičnih razlika.
Tukey HSD	Post-hoc test koji uspoređuje sve parove grupa uz kontrolu ukupne greške tipa I.
Eta-kvadrat	SS_between / SS_total. Proporcija varijabilnosti objašnjena grupnom pripadnošću. 0.01/0.06/0.14 mali/srednji/veliki.
Omega-kvadrat	Manje pristrana alternativa eta-kvadratu. Preporučena za publikacije.
Inflacija alfa	Porast greške tipa I pri višestrukim usporedbama. ANOVA to kontrolira.
Homogenost varijance	Pretpostavka jednakih varijanci u svim grupama. Levenov test je provjerava.
Welchova ANOVA	`oneway.test(var.equal = FALSE)`. ANOVA koja ne zahtijeva jednake varijance.
Kruskal-Wallisov test	Neparametrijska alternativa jednosmjernoj ANOVA-i. Koristi rangove. `kruskal.test()`.
Dunnov test	Post-hoc test za Kruskal-Wallis. Neparametrijski ekvivalent Tukeyu.
Planirane usporedbe	Unaprijed definirane specifične usporedbe. Snažniji od post-hoc testova jer testiraju manje hipoteza.
`aov()`	R funkcija za klasičnu ANOVA-u. `aov(y ~ grupa, data = ...)`.
`TukeyHSD()`	R funkcija za Tukey post-hoc test. Prima `aov()` objekt.
`oneway.test()`	R funkcija za Welchovu ANOVA-u. `var.equal = FALSE` za robusnu verziju.
`kruskal.test()`	R funkcija za Kruskal-Wallisov test. Sintaksa kao `aov()`.
`pairwise.wilcox.test()`	R funkcija za parwise Wilcoxon testove s korekcijom p-vrijednosti.

14 Tjedan 13: Linearna regresija

Predviđanje i objašnjavanje s modelima

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete:

1. Objasniti razliku između korelacije i regresije.
2. Provesti jednostavnu linearnu regresiju u R-u i interpretirati koeficijente.
3. Interpretirati R-kvadrat kao mjeru kvalitete modela.
4. Provjeriti pretpostavke linearne regresije dijagnostičkim grafovima.
5. Provesti višestruku regresiju s više prediktora i interpretirati parcijalne koeficijente.
6. Usporediti modele pomoću R-kvadrata, prilagođenog R-kvadrata i AIC-a.
7. Prepoznati uobičajene probleme (multikolinearnost, utjecajne točke, nelinearnost).
8. Napisati kompletni izvještaj regresijske analize.

14.1 Što pokreće angažman?

Zamislite sljedeću situaciju. Radite kao analitičarka društvenih mreža za srednje veliku medijsku kuću. Vaša šefica dolazi s pitanjem koje zvuči jednostavno poput "Koji faktori utječu na angažman naših Instagram objava?" Želi znati je li stvar u duljini teksta, broju hashtagova, tipu sadržaja, pozivu na akciju, ili u nečem sasvim drugom. I još važnije, želi konkretne preporuke — što da radimo više, a što manje?

Do sada ste u kolegiju naučili uspoređivati grupe. Hi-kvadrat test govori vam postoji li veza između kategoričkih varijabli. T-test uspoređuje prosjeke dviju grupa. ANOVA uspoređuje više grupa odjednom. Ali nijedno od toga ne odgovara na pitanje vaše šefice. Ona ne pita "razlikuju li se grupe." Umjesto toga, pita koliko svaki faktor doprinosi angažmanu, u kojem smjeru, i koliko dobro možemo predvidjeti angažman na temelju tih faktora.

Za to nam treba regresija. Regresija je — u najjednostavnijem smislu — alat koji modelira odnos između jedne ili više nezavisnih varijabli (koje zovemo prediktorima) i jedne zavisne

varijable (koju zovemo ishodom). Umjesto da samo kaže "postoji razlika", regresija kvantificira — za svaki dodatni hashtag, angažman se mijenja za toliko i toliko. To je razlika između "hashtagovi su važni" i "svaki dodatni hashtag smanjuje angažman za 0.15 postotnih bodova, kontrolirajući za ostale faktore."

Radimo s datasetom od 500 Instagram objava jednog poslovnog profila. Svaka objava ima zabilježen engagement rate (postotak pratitelja koji su reagirali), duljinu teksta, broj hashtagova, broj oznaka drugih profila, tip sadržaja (slika, video, carousel, reel), temu i informaciju o tome je li uključen poziv na akciju (CTA).

```
posts <- read_csv("../resources/datasets/social_engagement.csv")
glimpse(posts)
```

```
Rows: 400
Columns: 14
$ post_id         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,~
$ day             <chr> "utorak", "ponedjeljak", "petak", "nedjelja", "subota"~
$ time_slot       <chr> "18-21", "09-12", "21-24", "12-15", "15-18", "18-
21", ~
$ content_type    <chr> "foto", "tekst", "carousel", "foto", "carousel", "reel~
$ topic           <chr> "iza_kulisa", "proizvod", "zabava", "zabava", "proizvo~
$ text_length     <dbl> 290, 34, 35, 162, 240, 189, 300, 228, 242, 97, 136, 17~
$ num_hashtags    <dbl> 0, 17, 19, 14, 20, 9, 25, 21, 6, 12, 0, 23, 27, 27, 20~
$ has_cta         <dbl> 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, ~
$ num_mentions    <dbl> 3, 1, 1, 4, 0, 4, 4, 1, 1, 1, 2, 4, 1, 4, 4, 4, 3, 0, ~
$ followers       <dbl> 15095, 15312, 14749, 14728, 15336, 14722, 15168, 14805~
$ engagement_rate <dbl> 4.93, 2.26, 7.32, 5.58, 4.03, 6.58, 3.51, 6.74, 2.03, ~
$ likes           <dbl> 594, 284, 860, 583, 500, 692, 447, 816, 212, 548, 476,~
$ comments        <dbl> 120, 76, 174, 81, 93, 138, 88, 195, 45, 150, 87, 33, 4~
$ shares          <dbl> 78, 32, 115, 91, 70, 124, 41, 59, 45, 92, 80, 34, 25, ~
```

14.2 Od korelacije do regresije

Krenimo od poznatog terena. Korelaciju već znate — ona mjeri jačinu i smjer linearne veze između dviju varijabli. Pearsonov r kreće se od -1 (savršena negativna veza) preko 0 (nema linearne veze) do +1 (savršena pozitivna veza).

Regresija ide korak dalje. Dok korelacija samo kaže "ove dvije varijable su povezane", regresija definira jednadžbu pravca koja opisuje tu vezu. Ta jednadžba vam omogućuje nešto što korelacija ne može — predviđanje. Ako znate koliko hashtagova ima objava, regresija vam daje konkretnu procjenu koliki će biti njezin angažman.

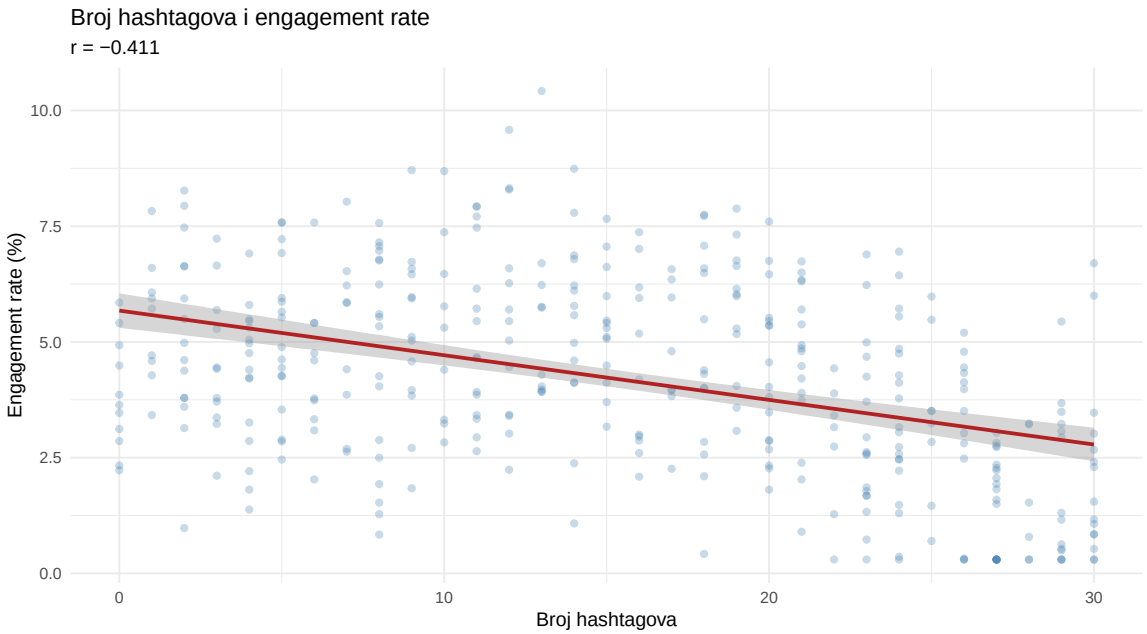
Prije nego što uđemo u regresiju, pogledajmo korelacije između naših numeričkih varijabli.

```r
# Korelacije numeričkih prediktora s engagement_rate
posts |>
  select(engagement_rate, text_length, num_hashtags, num_mentions, followers) |>
  cor() |>
  round(3)
```

```
                engagement_rate text_length num_hashtags num_mentions followers
engagement_rate           1.000       0.040       -0.411       -0.016    -
0.132
text_length               0.040       1.000        0.008        0.061    -
0.025
num_hashtags             -0.411       0.008        1.000        0.031     0.062
num_mentions             -0.016       0.061        0.031        1.000     0.092
followers                -0.132      -0.025        0.062        0.092     1.000
```

Pogledajte stupac `engagement_rate`. Broj hashtagova ima negativnu korelaciju s angažmanom (r -0.41), što znači da objave s više hashtagova u prosjeku imaju niži engagement rate. Vizualizirajmo tu vezu.

```r
posts |>
  ggplot(aes(x = num_hashtags, y = engagement_rate)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_smooth(method = "lm", se = TRUE, color = "firebrick") +
  labs(
    title = "Broj hashtagova i engagement rate",
    subtitle = paste0("r = ", round(cor(posts$num_hashtags, posts$engagement_rate), 3)),
    x = "Broj hashtagova",
    y = "Engagement rate (%)"
  ) +
  theme_minimal()
```



Crvena linija je regresijski pravac — ona predstavlja "najbolju" ravnu liniju koja prolazi kroz oblak točaka. Sivi pojas oko nje pokazuje nesigurnost te procjene (95% interval pouzdanosti za pravac).

Jedna stvar vas možda brine. Negativna korelacija sugerira da više hashtagova znači niži angažman. Ali budite oprezni s takvim zaključcima. Možda veza uopće nije linearna — možda postoji optimalan broj hashtagova, a i premalo i previše je loše. Možda profili s više hashtagova imaju i druge karakteristike koje snižavaju angažman. To su pitanja koja ćemo istražiti kasnije u ovom predavanju.

14.3 Jednostavna linearna regresija

Počnimo s najjednostavnijim mogućim modelom — jednim prediktorom i jednim ishodom. To je jednostavna linearna regresija.

Ideja je intuitivna. Vi imate oblak točaka na scatterplotu i želite provući ravnu liniju kroz taj oblak tako da ona što bolje opisuje opći trend. "Što bolje" u praksi znači da je ukupna udaljenost svih točaka od linije što manja.

Matematički, ta linija izgleda ovako.

$$Y = b_0 + b_1 X + \varepsilon$$

Raspakirajmo ovo simbol po simbol. Y je vaša zavisna varijabla, ono što želite predvidjeti (u našem slučaju engagement rate). X je prediktor (recimo, duljina teksta). b_0 je odsječak, koji vam kaže koliki bi bio predviđeni engagement rate kad bi duljina teksta bila nula. b_1 je nagib, ključni broj — on govori za koliko se engagement rate mijenja kad duljina teksta

poraste za jednu jedinicu. Konačno, ε je greška, rezidual, ono što model ne uspijeva objasniti. Svaka objava ima svoju priču koja nije samo u duljini teksta.

Pokrenimo regresiju u R-u. Funkcija `lm()` (linear model) traži formulu i podatke. Formula `engagement_rate ~ text_length` znači "predvidi engagement rate na temelju duljine teksta".

```
# Jednostavna regresija: text_length -> engagement_rate
model1 <- lm(engagement_rate ~ text_length, data = posts)
summary(model1)
```

```

Call:
lm(formula = engagement_rate ~ text_length, data = posts)

Residuals:
    Min      1Q  Median      3Q     Max
-3.9713 -1.4744 -0.0468  1.5489  6.1332

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.034491   0.238994  16.881   <2e-16 ***
text_length 0.001034   0.001299   0.796    0.426
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.102 on 398 degrees of freedom
Multiple R-squared:  0.001591,  Adjusted R-squared:  -0.0009177
F-statistic: 0.6342 on 1 and 398 DF,  p-value: 0.4263
```

14.3.1 Kako čitati ovaj output

Output funkcije `summary()` na regresijskom modelu sadrži puno informacija, i čest je osjećaj studenta da je "sve puno zvjezdica i brojeva" te ne znaju gdje početi. Počnimo od najvažnijeg i idimo redom.

```
koef <- coef(model1)

cat("Jednadžba: engagement_rate = ", round(koef[1], 3), " + ",
    round(koef[2], 5), " * text_length\n\n", sep = "")
```

```
Jednadžba: engagement_rate = 4.034 + 0.00103 * text_length
```

```
cat("Interpretacija:\n")
```

```
Interpretacija:
```

```
cat("  Intercept (b0 = ", round(koef[1], 2), "): Ocekivani engagement rate\n", sep = "")
```

```
  Intercept (b0 = 4.03): Ocekivani engagement rate
```

```
cat("  kad je text_length = 0 (teorijska vrijednost, nema prakticnog znacenja).\n\n")
```

```
  kad je text_length = 0 (teorijska vrijednost, nema prakticnog znacenja).
```

```
cat("  Slope (b1 = ", round(koef[2], 4), "): Za svaki dodatni znak u tekstu,\n", sep = "")
```

```
  Slope (b1 = 0.001): Za svaki dodatni znak u tekstu,
```

```
cat("  engagement rate se mijenja za ", round(koef[2], 4), " postotnih bodova.\n\n", sep =
```

```
  engagement rate se mijenja za 0.001 postotnih bodova.
```

```
# R-kvadrat
r2 <- summary(model1)$r.squared
cat("R-kvadrat:", round(r2, 4), "\n")
```

```
R-kvadrat: 0.0016
```

```
cat("Interpretacija:", round(r2 * 100, 1), "% varijabilnosti u engagement rateu\n")
```

```
Interpretacija: 0.2 % varijabilnosti u engagement rateu
```

```
cat("je objasnjeno duljinom teksta. To je vrlo malo.\n")
```

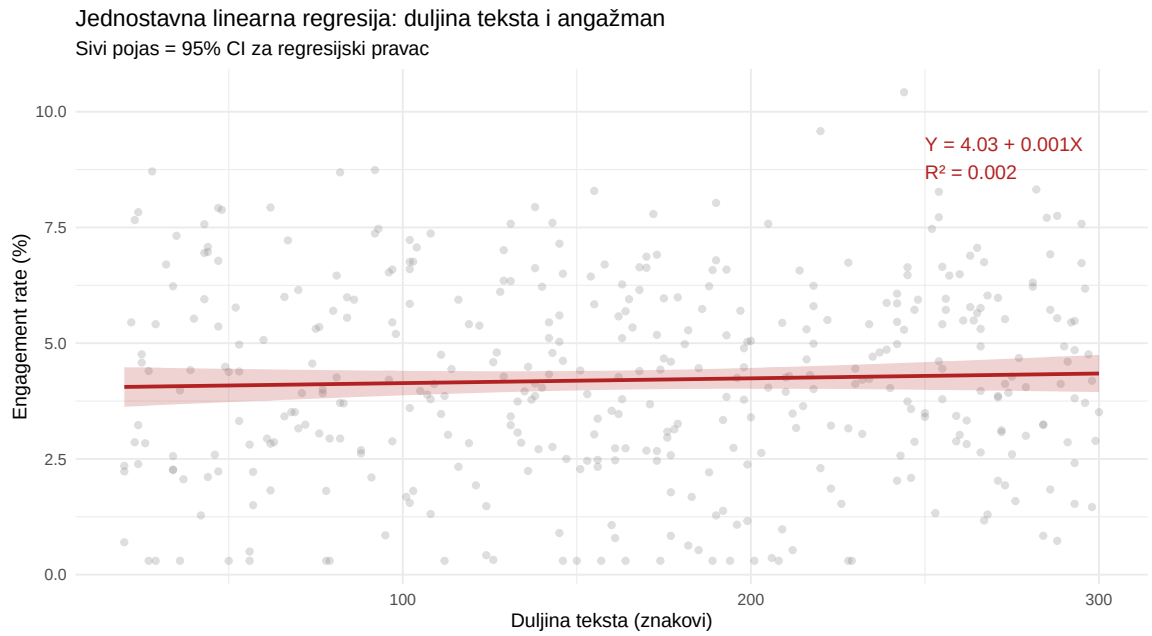
```
je objasnjeno duljinom teksta. To je vrlo malo.
```

Koeficijenti su dva broja koja definiraju vašu liniju. Odsječak (intercept, b_0) vam kaže predviđeni engagement rate kad je duljina teksta nula. U praksi, nitko ne objavljuje objavu bez teksta, pa taj broj nema praktičnu interpretaciju, ali je potreban da definira liniju. Nagib (slope, b_1) je ono što vas zapravo zanima — za svaki dodatni znak u tekstu, engagement rate se mijenja za toliko postotnih bodova.

R-kvadrat odgovara na jedno važno pitanje — koliki udio ukupne varijabilnosti u engagement rateu objašnjava naš model? Vrijednost je niska. Duljina teksta sama jednostavno nije dobar prediktor angažmana. To ima smisla jer o angažmanu odlučuje puno više faktora od duljine teksta. Trebat će nam više prediktora.

14.3.2 Vizualizacija regresijskog pravca

```
posts |>
  ggplot(aes(x = text_length, y = engagement_rate)) +
  geom_point(alpha = 0.25, color = "grey50") +
  geom_smooth(method = "lm", se = TRUE, color = "firebrick", fill = "firebrick", alpha = 0
  annotate("text", x = 250, y = 9,
           label = paste0("Y = ", round(koef[1], 2), " + ", round(koef[2], 4), "X\nR² = ",
           color = "firebrick", hjust = 0) +
  labs(
    title = "Jednostavna linearna regresija: duljina teksta i angažman",
    subtitle = "Sivi pojas = 95% CI za regresijski pravac",
    x = "Duljina teksta (znakovi)",
    y = "Engagement rate (%)"
  ) +
  theme_minimal()
```



Pogledajte koliko je oblak točaka razbacanih daleko od linije. To je vizualna manifestacija niskog R-kvadrata — linija postoji, ali objašnjava samo mali dio priče. Većina varijabilnosti dolazi od faktora koje ovaj model ne uključuje.

14.4 Što su reziduali?

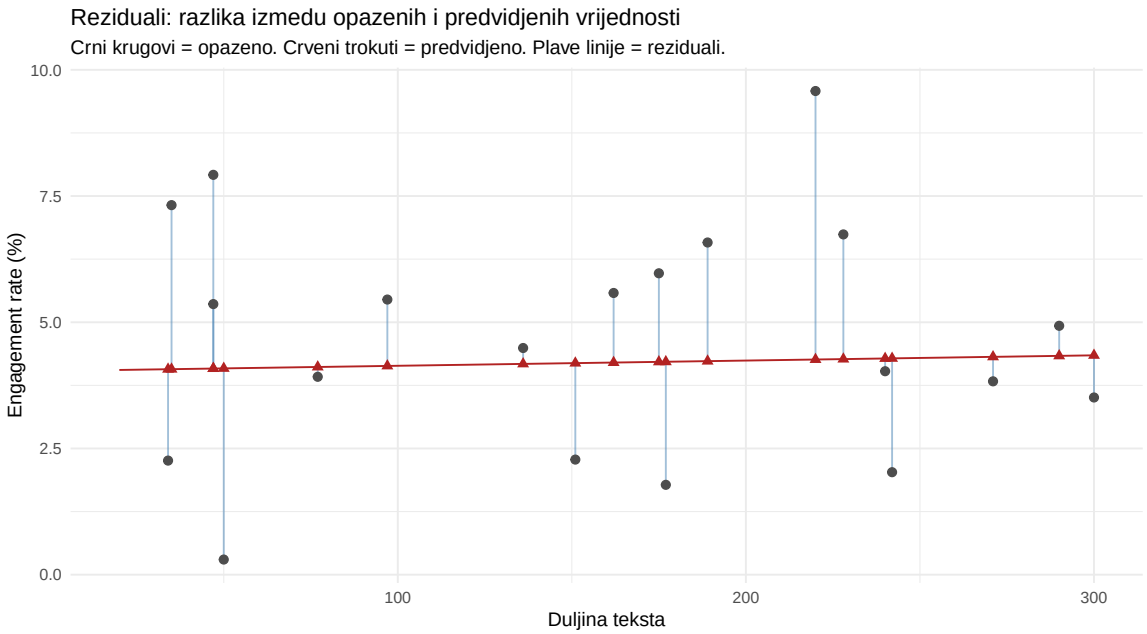
Svaka točka na grafu ima svoju predviđenu vrijednost (točku na liniji) i svoju stvarnu vrijednost (točku u oblaku). Razlika između te dvije vrijednosti zove se rezidual.

$$e_i = Y_i - \hat{Y}_i$$

U prijevodu, rezidual za i-tu objavu jednak je njezinoj stvarnoj engagement stopi minus onome što je model predvidio. Ako je rezidual pozitivan, objava je imala bolji angažman nego što je model očekivao. Ako je negativan, lošiji.

Regresija traži liniju koja minimizira sumu kvadriranih reziduala. Ova metoda zove se OLS (Ordinary Least Squares, metoda najmanjih kvadrata). Zašto kvadriramo? Jer bismo inače imali pozitivne i negativne reziduale koji bi se međusobno poništavali. Kvadriranje osigurava da su sva odstupanja pozitivna, a kao bonus, veća odstupanja kažnjavaju se proporcionalno više.

```
# Dodajmo predvidjene vrijednosti i reziduale u podatke
posts_pred <- posts |>
  mutate(
    predicted = predict(model1),
    residual = residuals(model1)
  )

# Prikaz reziduala za prvih 20 objava
posts_pred |>
  slice(1:20) |>
  ggplot(aes(x = text_length, y = engagement_rate)) +
  geom_segment(aes(xend = text_length, yend = predicted), color = "steelblue", alpha = 0.5
  geom_point(color = "grey30", size = 2) +
  geom_point(aes(y = predicted), color = "firebrick", size = 2, shape = 17) +
  geom_smooth(data = posts, method = "lm", se = FALSE, color = "firebrick", linewidth = 0.
  labs(
    title = "Reziduali: razlika izmedu opazenih i predvidjenih vrijednosti",
    subtitle = "Crni krugovi = opazeno. Crveni trokuti = predvidjeno. Plave linije = rezid
    x = "Duljina teksta",
    y = "Engagement rate (%)"
  ) +
  theme_minimal()
```



Plave vertikalne linije na ovom grafu su reziduali. Svaka linija povezuje stvarnu vrijednost neke objave (crni krug) s njezinom predviđenom vrijednošću na regresijskom pravcu (crveni trokut). Kraće linije znače bolje predviđanje. Duže linije znače da je model za tu objavu značajno pogriješio.

Reziduali nisu samo mjera pogreške. Oni su dijagnostički alat. Ako ih pažljivo proučimo, mogu nam otkriti različite probleme s modelom uključujući nelinearnost, nejednakomjernu varijabilnost, ili utjecajne točke koje iskrivljuju cijelu analizu.

14.5 Pretpostavke linearne regresije

Svaki statistički test ima pretpostavke, i regresija nije iznimka. Postoje četiri ključne pretpostavke koje moraju biti barem približno zadovoljene da bismo mogli vjerovati našim rezultatima.

Linearnost — veza između prediktora i ishoda mora biti linearna. Ako je stvarna veza zakrivljena, a mi joj pokušavamo prilagoditi ravnu liniju, naši koeficijenti bit će pristrani.

Nezavisnost reziduala — reziduali jednog opažanja ne smiju biti povezani s rezidualima drugog. Ovo je obično zadovoljeno ako su opažanja prikupljena neovisno jedno o drugom.

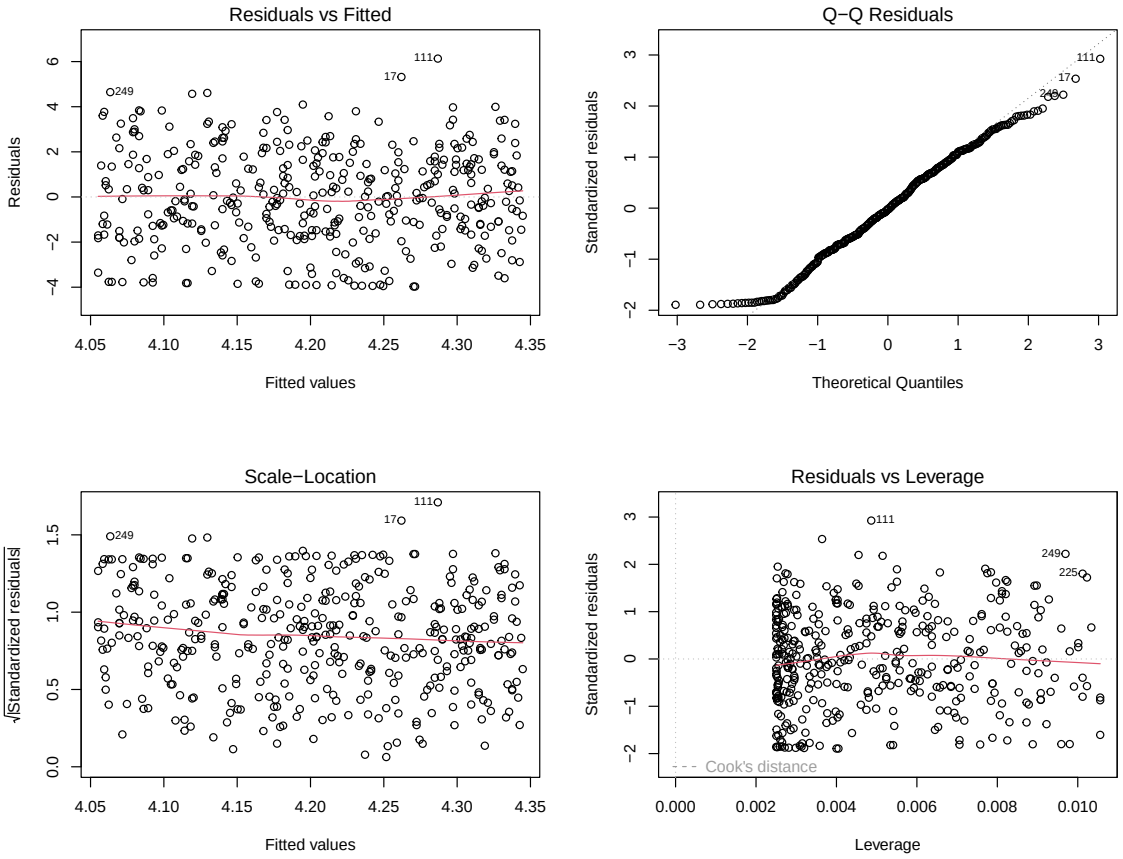
Homoskedastičnost — teška riječ, ali jednostavan koncept. Varijanca reziduala trebala bi biti otprilike jednaka za sve vrijednosti prediktora. Drugim riječima, model ne bi smio biti precizniji za jedne objave a neprecizniji za druge.

Normalnost reziduala — reziduali bi trebali biti približno normalno distribuirani. Ovo je važno za pouzdanost p-vrijednosti i intervala pouzdanosti.

Dobra vijest — ne trebate pamtiti formule za provjeru ovih pretpostavki. R ima ugrađenu dijagnostiku. Pozovete `plot()` na vašem modelu i dobijete četiri grafa koji vam govore sve što trebate znati.

14.5.1 Dijagnostički grafovi

```
par(mfrow = c(2, 2))
plot(model1)
```




```
par(mfrow = c(1, 1))
```

Prođimo redom.

Residuals vs Fitted (gore lijevo) provjerava linearnost i homoskedastičnost. Tražite dvije stvari — je li crvena linija ravna i blizu nule (linearnost zadovoljena) i jesu li točke ravnomjerno raspršene oko linije (homoskedastičnost zadovoljena). Ako vidite oblik lijevka

(točke se šire prema desno), imate heteroskedastičnost. Ako vidite krivulju, veza nije linearna.

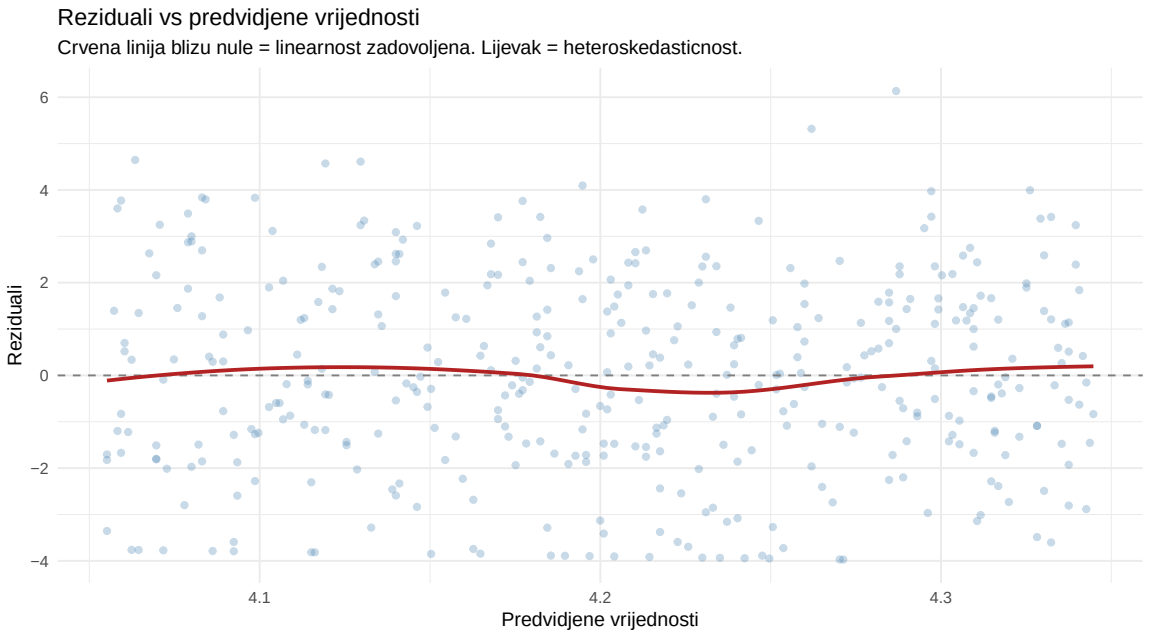
Normal Q-Q (gore desno) provjerava normalnost reziduala. Točke bi trebale ležati blizu dijagonale. Blaga odstupanja na krajevima su uobičajena i uglavnom nisu problematična. Značajna odstupanja sugeriraju da reziduali nisu normalni.

Scale-Location (dolje lijevo) je još jedan pogled na homoskedastičnost. Želite ravnu crvenu liniju i ravnomjerno raspršene točke. Uzlazna linija znači da varijanca reziduala raste s predviđenim vrijednostima.

Residuals vs Leverage (dolje desno) identificira utjecajne točke. Opažanja s visokim leverageom (daleko od centra u prostoru prediktora) i velikim rezidualima (daleko od regresijskog pravca) mogu neprimjereno utjecati na cijeli model. Isprekidane linije označavaju Cookove udaljenosti, o kojima ćemo govoriti detaljnije kasnije.

Isti graf možemo napraviti i u ggplotu za ljepši prikaz.

```
# Residuals vs Fitted u ggplot (preglednije)
posts_pred |>
  ggplot(aes(x = predicted, y = residual)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "grey50") +
  geom_smooth(method = "loess", se = FALSE, color = "firebrick") +
  labs(
    title = "Reziduali vs predvidjene vrijednosti",
    subtitle = "Crvena linija blizu nule = linearnost zadovoljena. Lijevak = heteroskedast
    x = "Predvidjene vrijednosti",
    y = "Reziduali"
  ) +
  theme_minimal()
```



14.6 Višestruka regresija

Jednostavna regresija s jednim prediktorom rijetko je dovoljna za bilo što ozbiljno u komunikološkim istraživanjima. Angažman na Instagramu ne ovisi samo o duljini teksta. Ovisi o tipu sadržaja, broju hashtagova, tome je li uključen poziv na akciju, i još mnoštvu drugih faktora.

Višestruka regresija proširuje model na više prediktora:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + ... + b_k X_k + \varepsilon$$

Izgleda kompliciranо, ali logika je ista kao prije — tražimo kombinaciju koeficijenata koja najbolje predviđa ishod. Jedina razlika je u interpretaciji. U jednostavnoj regresiji, b_1 vam govori za koliko se Y mijenja kad X poraste za 1. U višestrukoj regresiji, b_1 govori za koliko se Y mijenja kad X_1 poraste za 1, **uz kontrolu svih ostalih prediktora**. To je ono "držeći sve ostalo jednakim" što čujete u istraživanjima.

Ovo je izuzetno važno jer zamislite da objave s više hashtagova također imaju duži tekst. U jednostavnoj regresiji, koeficijent za hashtagove upija oba efekta, što je problem. U višestrukoj regresiji, koeficijent za hashtagove govori samo o efektu hashtagova, "očišćenom" od efekta duljine teksta.

```
# Visestruka regresija: vise prediktora
model2 <- lm(engagement_rate ~ text_length + num_hashtags + has_cta + num_mentions, data =
summary(model2)
```

```
Call:
lm(formula = engagement_rate ~ text_length + num_hashtags + has_cta +
    num_mentions, data = posts)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3636 -1.4502 -0.1389  1.4289  6.1080

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.3126047  0.3129882  16.974   <2e-16 ***
text_length   0.0009983  0.0011825   0.844   0.3991
num_hashtags -0.0964910  0.0106738  -9.040   <2e-16 ***
has_cta       0.4833190  0.1967471   2.457   0.0145 *
num_mentions  0.0051079  0.0592290   0.086   0.9313
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.908 on 395 degrees of freedom
Multiple R-squared:  0.1833,    Adjusted R-squared:  0.175
F-statistic: 22.16 on 4 and 395 DF,  p-value: < 2.2e-16
```

```r
koef2 <- coef(model2)
r2_m2 <- summary(model2)$r.squared
adj_r2_m2 <- summary(model2)$adj.r.squared

cat("=== Model 2: Visestruka regresija ===\n\n")
```

```
=== Model 2: Visestruka regresija ===
```

```r
cat("Jednadžba:\n")
```

```
Jednadžba:
```

```r
cat("engagement = ", round(koef2[1], 2), "\n", sep = "")
```

```
engagement = 5.31
```

```r
for (i in 2:length(koef2)) {
  cat("  ", if_else(koef2[i] >= 0, "+ ", "- "), round(abs(koef2[i]), 4),
      " * ", names(koef2)[i], "\n", sep = "")
}
```

```
  + 0.001 * text_length
  - 0.0965 * num_hashtags
  + 0.4833 * has_cta
  + 0.0051 * num_mentions
```

```
cat("\nR-kvadrat:            ", round(r2_m2, 3), "\n")
```

```
R-kvadrat:             0.183
```

```
cat("Prilagodeni R-kvadrat:", round(adj_r2_m2, 3), "\n")
```

```
Prilagodeni R-kvadrat: 0.175
```

```
cat("Interpretacija:", round(r2_m2 * 100, 1), "% varijabilnosti objasnjeno.\n\n")
```

```
Interpretacija: 18.3 % varijabilnosti objasnjeno.
```

```
cat("Interpretacija koeficijenata (sve uz kontrolu ostalih prediktora):\n")
```

```
Interpretacija koeficijenata (sve uz kontrolu ostalih prediktora):
```

```
cat("  num_hashtags: Svaki dodatni hashtag mijenja engagement za ",
    round(koef2["num_hashtags"], 3), " bodova.\n", sep = "")
```

```
  num_hashtags: Svaki dodatni hashtag mijenja engagement za -0.096 bodova.
```

```
cat("  has_cta: Objave s CTA imaju u prosjeku ", round(koef2["has_cta"], 2),
    " bodova visi engagement.\n", sep = "")
```

```
  has_cta: Objave s CTA imaju u prosjeku 0.48 bodova visi engagement.
```

Primijetite kako je R-kvadrat porastao u odnosu na model s jednim prediktorom. To je očekivano jer smo dodali informaciju koja pomaže predviđanju. Ali je pitanje je li smo dodali dovoljno, ili možemo još bolje?

14.6.1 Usporedba modela

Probajmo graditi modele postupno, dodajući prediktore jedan po jedan, i usporedimo ih.

```r
# Model 3: dodajmo content_type
model3 <- lm(engagement_rate ~ text_length + num_hashtags + has_cta +
               num_mentions + content_type, data = posts)

# Model 4: dodajmo jos i topic
model4 <- lm(engagement_rate ~ text_length + num_hashtags + has_cta +
               num_mentions + content_type + topic, data = posts)

# Usporedba
tibble(
  model = c("M1: text_length", "M2: + hashtags, cta, mentions",
            "M3: + content_type", "M4: + topic"),
  R2 = round(c(summary(model1)$r.squared, summary(model2)$r.squared,
               summary(model3)$r.squared, summary(model4)$r.squared), 3),
  adj_R2 = round(c(summary(model1)$adj.r.squared, summary(model2)$adj.r.squared,
                   summary(model3)$adj.r.squared, summary(model4)$adj.r.squared), 3),
  AIC = round(c(AIC(model1), AIC(model2), AIC(model3), AIC(model4)), 1)
)
```

```
# A tibble: 4 x 4
  model                            R2 adj_R2   AIC
  <chr>                         <dbl>  <dbl> <dbl>
1 M1: text_length               0.002 -0.001 1733.
2 M2: + hashtags, cta, mentions 0.183  0.175 1659.
3 M3: + content_type            0.376  0.365 1558.
4 M4: + topic                   0.406  0.389 1546.
```

Tri mjere za usporedbu modela zaslužuju objašnjenje — to su R-kvadrat, prilagođeni R-kvadrat i AIC.

R-kvadrat vam govori koliki udio varijabilnosti model objašnjava. Problem je što on uvijek raste (ili ostaje isti) kad dodate prediktor, čak i ako je taj prediktor potpuno beskoristan. Ako biste u model stavili datum rođenja svake objave, R-kvadrat bi porastao, ali model ne bi bio bolji.

Prilagođeni R-kvadrat rješava taj problem. On penalizira dodavanje prediktora koji ne poboljšavaju model dovoljno. Ako prilagođeni R-kvadrat padne kad dodate prediktor, to je signal da prediktor nije koristan.

AIC (Akaike Information Criterion) je još jedna mjera kvalitete modela. Pravilo je jednostavno — niži AIC znači bolji model. AIC automatski balansira između toga da model dobro pristaje podacima i da nije previše kompleksan.

Pogledajmo detalje najboljeg modela.

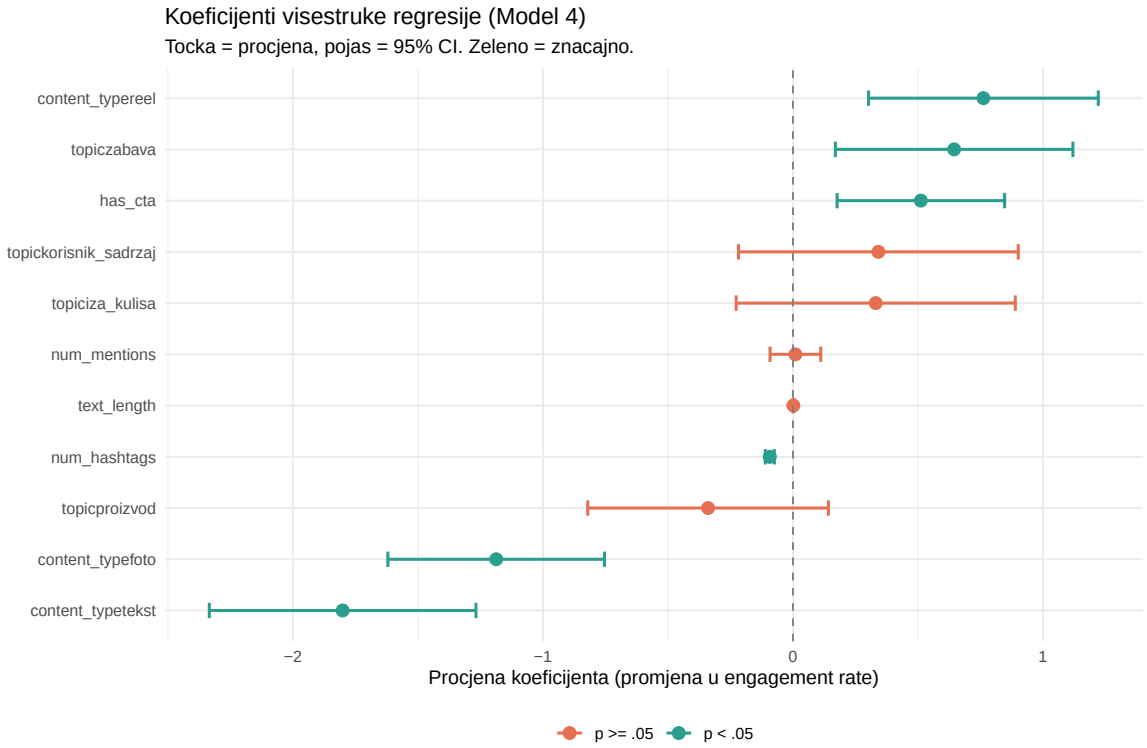
```r
summary(model4)
```

```
Call:
lm(formula = engagement_rate ~ text_length + num_hashtags + has_cta +
    num_mentions + content_type + topic, data = posts)

Residuals:
   Min     1Q Median     3Q    Max
-4.287 -1.338  0.044  1.188  4.681

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            5.414266   0.344605  15.712  < 2e-16 ***
text_length            0.001666   0.001026   1.624  0.10525
num_hashtags          -0.092638   0.009218 -10.050  < 2e-16 ***
has_cta                0.511429   0.170347   3.002  0.00285 **
num_mentions           0.009737   0.051568   0.189  0.85033
content_typefoto      -1.186807   0.220401  -5.385 1.26e-07 ***
content_typereel       0.761525   0.233632   3.260  0.00121 **
content_typetekst     -1.800942   0.271146  -6.642 1.05e-10 ***
topiciza_kulisa        0.330852   0.283879   1.165  0.24454
topickorisnik_sadrzaj  0.341487   0.284489   1.200  0.23073
topicproizvod         -0.339511   0.244766  -1.387  0.16621
topiczabava            0.644505   0.241529   2.668  0.00794 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.642 on 388 degrees of freedom
Multiple R-squared:  0.4061,    Adjusted R-squared:  0.3892
F-statistic: 24.12 on 11 and 388 DF,  p-value: < 2.2e-16
```

Koeficijenti su lakše čitljivi kad ih vizualiziramo. Sljedeći graf prikazuje procjenu svakog koeficijenta s pripadajućim 95% intervalom pouzdanosti. Ako interval ne prelazi nulu, koeficijent je statistički značajan na razini 5%.

```r
# Vizualizacija koeficijenata modela 4
tidy_m4 <- broom::tidy(model4, conf.int = TRUE) |>
  filter(term != "(Intercept)") |>
  mutate(
    znacajno = p.value < 0.05,
    term = fct_reorder(term, estimate)
  )
```

```r
tidy_m4 |>
  ggplot(aes(y = term, x = estimate, color = znacajno)) +
  geom_errorbarh(aes(xmin = conf.low, xmax = conf.high), height = 0.3, linewidth = 0.8) +
  geom_point(size = 3) +
  geom_vline(xintercept = 0, linetype = "dashed", color = "grey50") +
  scale_color_manual(values = c("TRUE" = "#2a9d8f", "FALSE" = "#e76f51"),
                     labels = c("TRUE" = "p < .05", "FALSE" = "p >= .05")) +
  labs(
    title = "Koeficijenti visestruke regresije (Model 4)",
    subtitle = "Tocka = procjena, pojas = 95% CI. Zeleno = znacajno.",
    x = "Procjena koeficijenta (promjena u engagement rate)",
    y = NULL, color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



14.7 R-kvadrat i zašto nije “ocjena” modela

Studenti često tretiraju R-kvadrat kao ocjenu modela, misleći da je viši bolje, da je 1 savršen, a niska vrijednost znači da je model loš. Ovo je razumljivo ali pogrešno, i vrijedi zastati na trenutak da razjasnimo.

Formalno, R-kvadrat govori koliki udio ukupne varijabilnosti u Y-u vaš model objašnjava:

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = \frac{SS_{model}}{SS_{total}}$$

Koliki R-kvadrat možete očekivati ovisi o tome što pokušavate predvidjeti. U fizici, gdje zakoni su deterministički, R-kvadrat od 0.99 je normalan. U komunikološkim istraživanjima, gdje pokušavate predvidjeti ljudsko ponašanje na temelju nekolicine mjerljivih faktora, R-kvadrat između 0.10 i 0.30 je uobičajen i sasvim prihvatljiv. Ljudi su kompliciran i nepredvidivi, i to je u redu.

Prilagođeni R-kvadrat korigira za broj prediktora u modelu:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

gdje je n broj opažanja, a k broj prediktora. Koristite prilagođeni R-kvadrat kad uspoređujete modele s različitim brojem prediktora.

Pogledajmo zašto je to važno. Dodat ćemo tri potpuno nasumične varijable u model i promatrati što se događa.

```
# Demonstracija: dodavanje random prediktora
set.seed(42)
posts_demo <- posts |>
  mutate(random1 = rnorm(n()), random2 = rnorm(n()), random3 = rnorm(n()))

model_base <- lm(engagement_rate ~ num_hashtags + has_cta + content_type, data = posts_dem
model_rand <- lm(engagement_rate ~ num_hashtags + has_cta + content_type +
                   random1 + random2 + random3, data = posts_demo)

cat("Model bez random prediktora:\n")
```

```
Model bez random prediktora:
```

```
cat("  R2 =", round(summary(model_base)$r.squared, 4), "\n")
```

```
  R2 = 0.3721
```

```
cat("  Adj R2 =", round(summary(model_base)$adj.r.squared, 4), "\n\n")
```

```
  Adj R2 = 0.3641
```

```r
cat("Model S random prediktorima:\n")
```

```
Model S random prediktorima:
```

```r
cat("  R2 =", round(summary(model_rand)$r.squared, 4), "(veci! ali lazno)\n")
```

```
  R2 = 0.3807 (veci! ali lazno)
```

```r
cat("  Adj R2 =", round(summary(model_rand)$adj.r.squared, 4), "(korigira za lazno poboljs
```

```
  Adj R2 = 0.368 (korigira za lazno poboljsanje)
```

R-kvadrat je porastao. Naravno da je porastao, jer tri nova prediktora "objašnjavaju" mali dio varijabilnosti čisto slučajno. Ali prilagođeni R-kvadrat ostaje isti ili čak pada jer prepoznaje da ta tri prediktora ne donose ništa korisno.

> **!** Česta zablude o R-kvadratu
>
> R-kvadrat nije "ocjena" modela. $R^2 = 0.20$ može biti odličan rezultat za predviđanje ljudskog ponašanja, dok $R^2 = 0.90$ može biti loš za fizikalni zakon. Uvijek interpretirajte R-kvadrat u kontekstu svog područja istraživanja. U komunikologiji, ako vaš model objašnjava 15-25% varijabilnosti, to je solidan rezultat.

14.8 Multikolinearnost: kad se prediktori međusobno gužvaju

Zamislite da u model stavite i "broj riječi u tekstu" i "broj znakova u tekstu." Ove dvije varijable mjere gotovo istu stvar. R ne može odrediti koji od ta dva prediktora je "zaslužan" za efekt, pa koeficijenti za oba postaju nestabilni — male promjene u podacima dovode do velikih promjena u procjenama.

Ovo se zove multikolinearnost, što se pojavljuje kad su prediktori međusobno jako korelirani. VIF, što je kratica za Variance Inflation Factor, mjeri koliko je varijanca koeficijenta narasla zbog korelacije s drugim prediktorima.

```r
posts <- read_csv("../resources/datasets/social_engagement.csv")

model4 <- lm(engagement_rate ~ text_length + num_hashtags + has_cta +
               num_mentions + content_type + topic, data = posts)

# VIF za svaki prediktor (rucno za numericke)
model_num <- lm(engagement_rate ~ text_length + num_hashtags + has_cta + num_mentions, dat
```

```
# VIF = 1 / (1 - R2_j), gdje je R2_j R-kvadrat kad regresiramo Xj na sve ostale prediktore
vif_manual <- function(data, prediktori, target_pred) {
  formula_vif <- as.formula(paste(target_pred, "~", paste(setdiff(prediktori, target_pred)
  r2_j <- summary(lm(formula_vif, data = data))$r.squared
  1 / (1 - r2_j)
}

num_preds <- c("text_length", "num_hashtags", "has_cta", "num_mentions")

vif_vals <- map_dbl(num_preds, ~vif_manual(posts, num_preds, .x))
tibble(prediktor = num_preds, VIF = round(vif_vals, 2))
```

```
# A tibble: 4 x 2
  prediktor      VIF
  <chr>        <dbl>
1 text_length   1.01
2 num_hashtags  1
3 has_cta       1.01
4 num_mentions  1.01
```

Kao pravilo palca, VIF ispod 5 je sasvim prihvatljiv. VIF između 5 i 10 zaslužuje pozornost. VIF iznad 10 znači ozbiljan problem. Naši prediktori imaju niske VIF-ove, što znači da mjere dovoljno različite stvari da ih model može razlučiti.

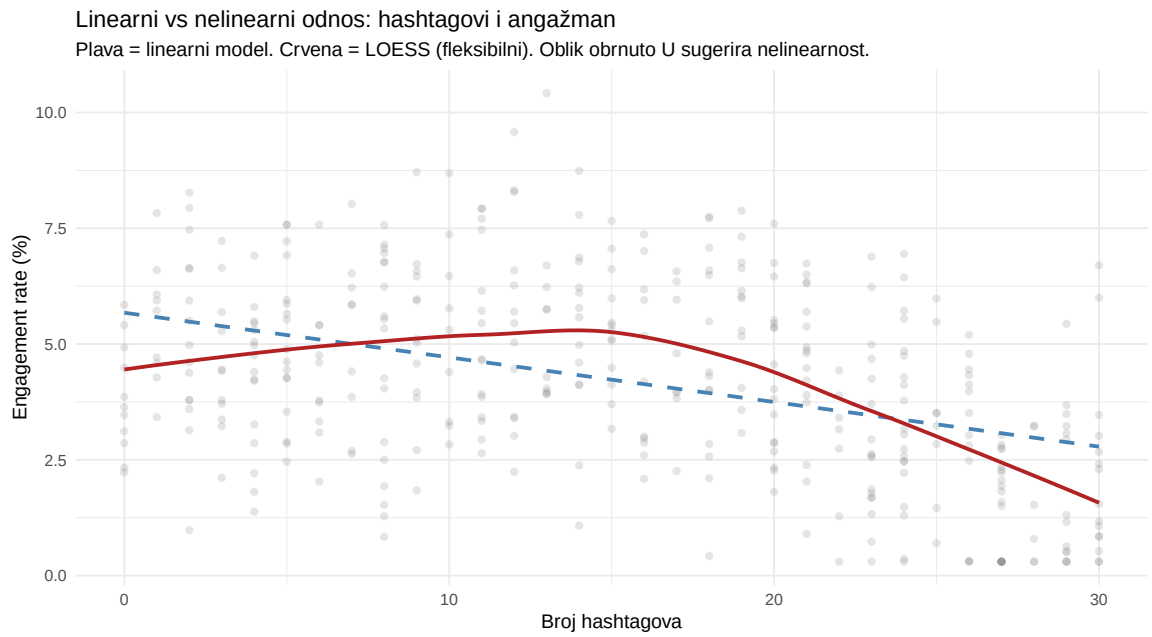
> ⚠ Što učiniti kad je VIF visok?
>
> Imate nekoliko opcija. Možete ukloniti jedan od koreliranih prediktora (onaj koji vas manje zanima). Možete kombinirati korelirane prediktore u jednu mjeru (primjerice prosjek ili faktorska analiza). Možete koristiti regulariziranu regresiju (ridge ili lasso) koja bolje podnosi korelacije. Ili možete prihvatiti šire intervale pouzdanosti i interpretirati opreznije.

14.9 Kad ravna linija ne pristaje: nelinearni odnosi

Linearna regresija pretpostavlja linearne odnose. Ali stvarni odnosi često nisu linearni — primjerice, tri hashtaga su vjerojatno bolja od nula, ali trideset hashtagova vjerovatno nije deset puta bolje od tri. Možda postoji optimalna točka, a sve iznad i ispod nje je lošije.

```
# Scatterplot s LOESS krivuljom umjesto ravnog pravca
posts |>
  ggplot(aes(x = num_hashtags, y = engagement_rate)) +
  geom_point(alpha = 0.2, color = "grey50") +
```

```
  geom_smooth(method = "lm", se = FALSE, color = "steelblue", linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "firebrick") +
  labs(
    title = "Linearni vs nelinearni odnos: hashtagovi i angažman",
    subtitle = "Plava = linearni model. Crvena = LOESS (fleksibilni). Oblik obrnuto U suge
    x = "Broj hashtagova",
    y = "Engagement rate (%)"
  ) +
  theme_minimal()
```



Plava isprekidana linija je ono što linearni model "vidi" — ravnu liniju koja silazi. Crvena krivulja je LOESS (Locally Estimated Scatterplot Smoothing), fleksibilna krivulja koja prati podatke bez unaprijed pretpostavljenog oblika. Razlika je uočljiva jer LOESS sugerira oblik obrnuto U, s vrhom negdje oko 8 do 12 hashtagova.

Kako možemo uhvatiti ovu zakrivljenost unutar linearne regresije? Dodavanjem kvadratnog člana — umjesto da modeliramo samo linearni efekt hashtagova, modeliramo i njihov kvadrat.

14.9.1 Polinomijalna regresija

```
# Model s kvadratnim clanom za hashtagove
model_poly <- lm(engagement_rate ~ num_hashtags + I(num_hashtags^2) +
                   has_cta + content_type + topic, data = posts)
```

```r
# Usporedba: linearni vs polinomijalni
model_lin <- lm(engagement_rate ~ num_hashtags + has_cta + content_type + topic, data = po

cat("Linearni model:      Adj R² =", round(summary(model_lin)$adj.r.squared, 3),
    ", AIC =", round(AIC(model_lin), 1), "\n")
```

```
Linearni model:      Adj R² = 0.388 , AIC = 1544.4
```

```r
cat("Polinomijalni model: Adj R² =", round(summary(model_poly)$adj.r.squared, 3),
    ", AIC =", round(AIC(model_poly), 1), "\n")
```

```
Polinomijalni model: Adj R² = 0.494 , AIC = 1469.7
```

Prilagođeni R-kvadrat je veći, a AIC niži. Oba signala govore isto — polinomijalni model bolje pristaje podacima. Pogledajmo koeficijente za hashtagove.

```r
# Koeficijenti za hashtag efekt
koef_poly <- coef(model_poly)
cat("num_hashtags:    ", round(koef_poly["num_hashtags"], 4), "\n")
```

```
num_hashtags:     0.2033
```

```r
cat("num_hashtags^2:  ", round(koef_poly["I(num_hashtags^2)"], 5), "\n\n")
```

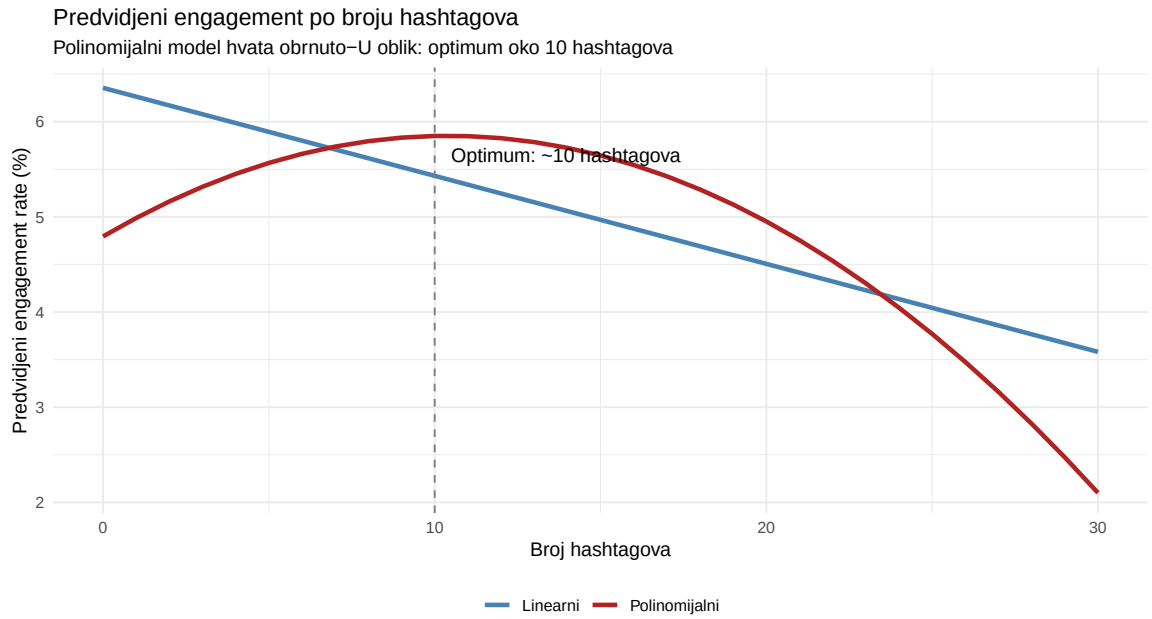
```
num_hashtags^2:   -0.00977
```

```r
# Optimalni broj hashtagova (vrh parabole)
optimal_h <- -koef_poly["num_hashtags"] / (2 * koef_poly["I(num_hashtags^2)"])
cat("Optimalni broj hashtagova:", round(optimal_h), "\n")
```

```
Optimalni broj hashtagova: 10
```

Linearni koeficijent za hashtagove je pozitivan (više hashtagova, viši angažman), ali kvadratni koeficijent je negativan — taj pozitivni efekt slabi i eventualno se pretvara u negativni. Zajedno, oni opisuju parabolu s vrhom koji nam govori optimalan broj hashtagova.

```r
# Predvidjene vrijednosti za razlicite brojeve hashtagova
hashtag_pred <- tibble(
  num_hashtags = 0:30,
  has_cta = 0,
  content_type = "carousel",
  topic = "zabava"
) |>
  mutate(
    pred_lin = predict(model_lin, newdata = pick(everything())),
    pred_poly = predict(model_poly, newdata = pick(everything()))
  )

hashtag_pred |>
  pivot_longer(c(pred_lin, pred_poly), names_to = "model", values_to = "predicted") |>
  mutate(model = if_else(model == "pred_lin", "Linearni", "Polinomijalni")) |>
  ggplot(aes(x = num_hashtags, y = predicted, color = model)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = round(optimal_h), linetype = "dashed", color = "grey50") +
  annotate("text", x = round(optimal_h) + 0.5, y = max(hashtag_pred$pred_poly) - 0.2,
           label = paste0("Optimum: ~", round(optimal_h), " hashtagova"), hjust = 0) +
  scale_color_manual(values = c("Linearni" = "steelblue", "Polinomijalni" = "firebrick")) +
  labs(
    title = "Predvidjeni engagement po broju hashtagova",
    subtitle = "Polinomijalni model hvata obrnuto-U oblik: optimum oko 10 hashtagova",
    x = "Broj hashtagova",
    y = "Predvidjeni engagement rate (%)",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Ovo je lijep primjer zašto dijagnostika modela nije samo akademska vježba. Linearni model bi vam rekao "smanjite hashtagove na minimum", dok polinomijalni model govori puno nijansiraniju priču — "koristite oko 10 hashtagova". Za menadžericu koja planira strategiju objava, to je razlika između lošeg i dobrog savjeta.

14.10 Standardizirani koeficijenti: tko je najvažniji?

U višestrukoj regresiji, koeficijenti su u originalnim jedinicama svojih prediktora. Koeficijent za duljinu teksta je u jedinicama "postotni bodovi angažmana po jednom dodatnom znaku teksta," a koeficijent za broj hashtagova je u jedinicama "postotni bodovi angažmana po jednom dodatnom hashtagu." Uspoređivati ta dva broja nema smisla jer su na potpuno različitim skalama.

Standardizirani koeficijenti (beta koeficijenti) rješavaju ovaj problem jer umjesto originalnih jedinica, oni izražavaju promjenu Y u jedinicama standardne devijacije za promjenu od jedne standardne devijacije u X. Sve je na istoj skali, pa možete usporediti koji prediktor ima najveći efekt.

```
# Standardizacija numerickih prediktora
posts_std <- posts |>
  mutate(across(c(text_length, num_hashtags, num_mentions), scale))

model_std <- lm(engagement_rate ~ text_length + num_hashtags + has_cta +
                  num_mentions + content_type + topic, data = posts_std)

# Usporedba nestandardiziranih i standardiziranih koeficijenata
```

```r
broom::tidy(model4) |>
  filter(term != "(Intercept)") |>
  select(term, b = estimate) |>
  left_join(
    broom::tidy(model_std) |>
      filter(term != "(Intercept)") |>
      select(term, beta = estimate),
    by = "term"
  ) |>
  mutate(across(c(b, beta), \(x) round(x, 3))) |>
  arrange(desc(abs(beta)))
```

```
# A tibble: 11 x 3
   term                       b   beta
   <chr>                  <dbl>  <dbl>
 1 content_typetekst     -1.80  -1.80
 2 content_typefoto      -1.19  -1.19
 3 num_hashtags          -0.093 -0.83
 4 content_typereel       0.762  0.762
 5 topiczabava            0.645  0.645
 6 has_cta                0.511  0.511
 7 topickorisnik_sadrzaj  0.341  0.341
 8 topicproizvod         -0.34  -0.34
 9 topiciza_kulisa        0.331  0.331
10 text_length            0.002  0.135
11 num_mentions           0.01   0.016
```

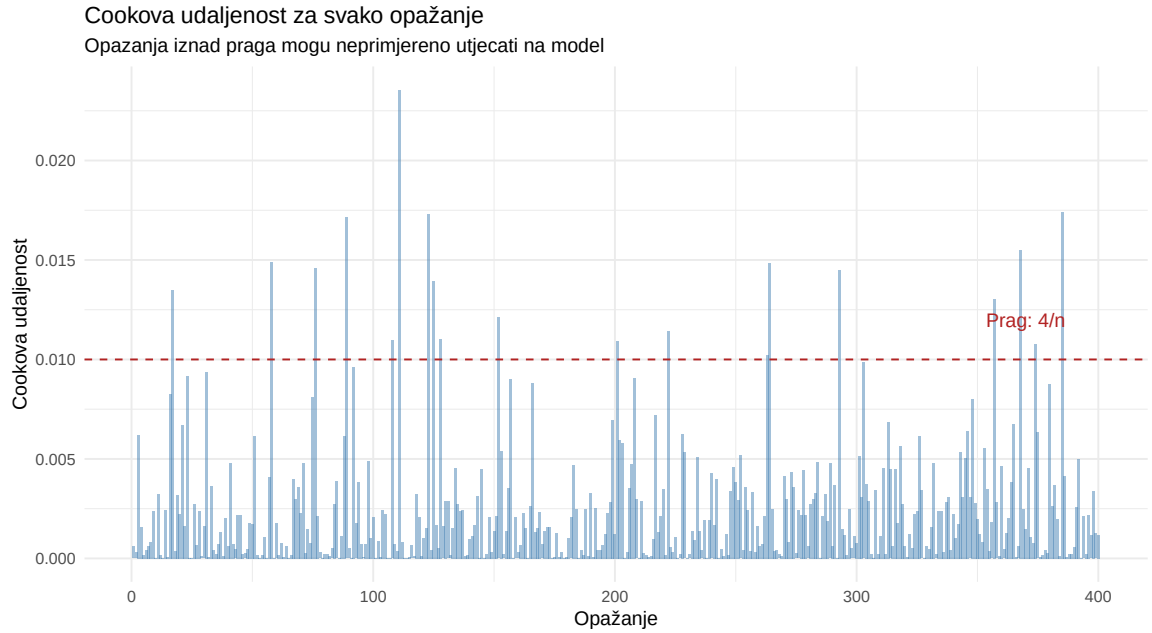
Stupac `b` su nestandardizirani koeficijenti, a `beta` standardizirani. Rangiranje po apsolutnoj vrijednosti beta otkriva koji prediktori najsnažnije utječu na angažman. Ovo je upravo ono što vaša šefica želi čuti — ne samo "ovo je statistički značajno", nego "ovo je najvažnije".

14.11 Utjecajne točke: kad jedna objava iskrivljuje cijeli model

Zamislite da jedna jedina Instagram objava ima 50 hashtagova i ekstremno visok angažman. Ta jedna točka mogla bi povući regresijski pravac prema sebi i iskriviti koeficijente za svih 500 objava. Cookova udaljenost mjeri koliko bi se model promijenio ako bismo uklonili svako pojedino opažanje.

```r
posts_diag <- posts |>
  mutate(
    cook = cooks.distance(model4),
    leverage = hatvalues(model4),
    std_residual = rstandard(model4)
```

```
  )

# Cookova udaljenost
posts_diag |>
  mutate(post_id = row_number()) |>
  ggplot(aes(x = post_id, y = cook)) +
  geom_col(fill = "steelblue", alpha = 0.5) +
  geom_hline(yintercept = 4 / nrow(posts), linetype = "dashed", color = "firebrick") +
  annotate("text", x = nrow(posts) - 30, y = 4/nrow(posts) + 0.002,
           label = "Prag: 4/n", color = "firebrick") +
  labs(
    title = "Cookova udaljenost za svako opažanje",
    subtitle = "Opazanja iznad praga mogu neprimjereno utjecati na model",
    x = "Opažanje",
    y = "Cookova udaljenost"
  ) +
  theme_minimal()
```



Uobičajeni prag je 4/n, gdje je n broj opažanja. Opažanja iznad tog praga zaslužuju pažljivi pregled jer trebate provjeriti jesu li pogreška u podacima, ekstremni ali legitimni slučajevi, ili nešto treće.

Zdrava praksa je provoditi analizu dvaput — jednom sa svim podacima i jednom bez utjecajnih točaka. Ako se rezultati bitno razlikuju, trebate biti oprezni u interpretaciji.

```
# Koje objave su najutjecajnije?
prag_cook <- 4 / nrow(posts)
```

```r
utjecajne <- posts_diag |> filter(cook > prag_cook) |> nrow()

cat("Prag Cook's distance:", round(prag_cook, 4), "\n")
```

```
Prag Cook's distance: 0.01
```

```r
cat("Broj utjecajnih tocaka:", utjecajne, "od", nrow(posts), "\n")
```

```
Broj utjecajnih tocaka: 19 od 400
```

```r
# Usporedba modela s i bez utjecajnih tocaka
posts_clean <- posts_diag |> filter(cook <= prag_cook)
model4_clean <- lm(engagement_rate ~ text_length + num_hashtags + has_cta +
                     num_mentions + content_type + topic, data = posts_clean)

cat("\nS utjecajnim tockama:   Adj R² =", round(summary(model4)$adj.r.squared, 3), "\n")
```

```
S utjecajnim tockama:   Adj R² = 0.389
```

```r
cat("Bez utjecajnih tocaka:  Adj R² =", round(summary(model4_clean)$adj.r.squared, 3), "\n
```

```
Bez utjecajnih tocaka:  Adj R² = 0.466
```

14.12 Sve zajedno: izvještaj za menadžericu

Sada dolazimo do cilja. Vaša šefica ne želi vidjeti R output. Ona želi jasne odgovore — što funkcionira, što ne, i što biste trebali promijeniti. Izgradimo finalni model i pretvorimo ga u priču.

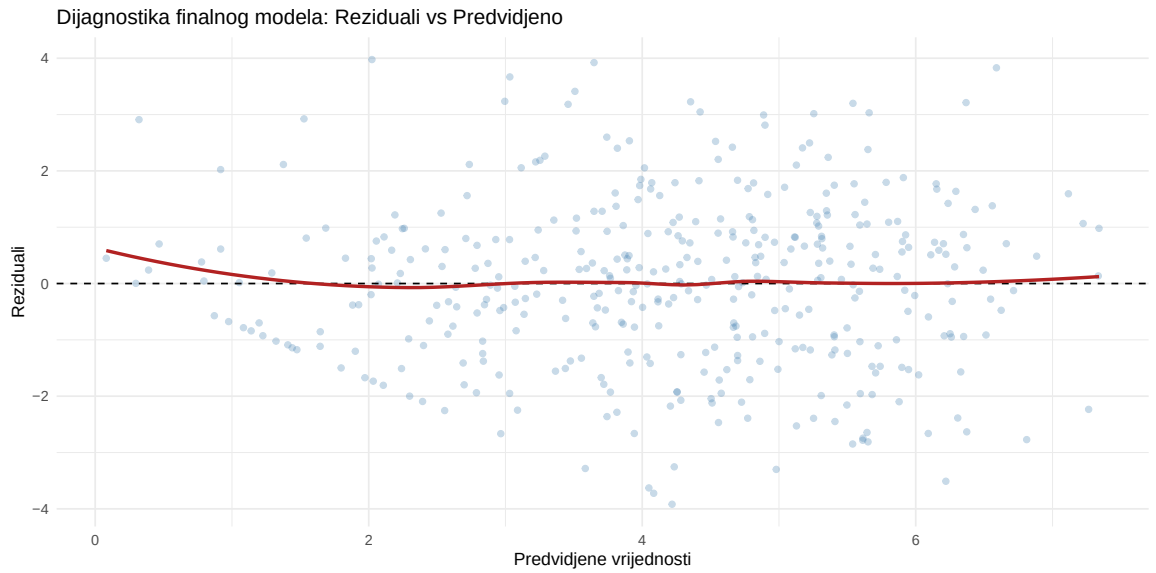
```r
# Finalni model s polinomom za hashtagove
model_final <- lm(engagement_rate ~ text_length + num_hashtags + I(num_hashtags^2) +
                    has_cta + content_type + topic, data = posts)
summary(model_final)
```

```
Call:
lm(formula = engagement_rate ~ text_length + num_hashtags + I(num_hashtags^2) +
    has_cta + content_type + topic, data = posts)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.9183 -1.0226  0.0508  0.9011  3.9764

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            3.9765594  0.3404994  11.679  < 2e-16 ***
text_length            0.0009147  0.0009352   0.978 0.328663
num_hashtags           0.2002828  0.0338197   5.922 7.01e-09 ***
I(num_hashtags^2)     -0.0096729  0.0010821  -8.939  < 2e-16 ***
has_cta                0.5436919  0.1545645   3.518 0.000487 ***
content_typefoto      -1.0559718  0.2008621  -5.257 2.42e-07 ***
content_typereel       0.8676736  0.2123747   4.086 5.34e-05 ***
content_typetekst     -1.7379822  0.2465704  -7.049 8.31e-12 ***
topiciza_kulisa        0.3120734  0.2580244   1.209 0.227218
topickorisnik_sadrzaj  0.5538109  0.2599251   2.131 0.033746 *
topicproizvod         -0.2953335  0.2227455  -1.326 0.185660
topiczabava            0.6838707  0.2196727   3.113 0.001988 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.495 on 388 degrees of freedom
Multiple R-squared:  0.5075,    Adjusted R-squared:  0.4935
F-statistic: 36.34 on 11 and 388 DF,  p-value: < 2.2e-16
```

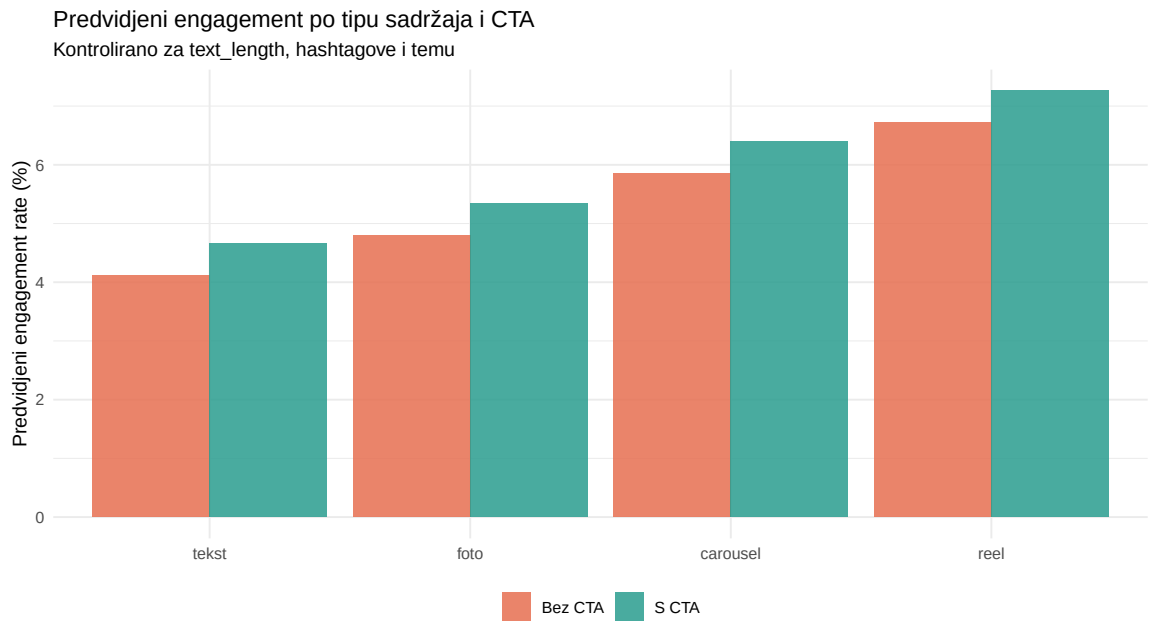
Prije nego što interpretirate rezultate, trebate provjeriti dijagnostiku.

```r
# Residuals vs Fitted za finalni model
tibble(fitted = fitted(model_final), resid = residuals(model_final)) |>
  ggplot(aes(x = fitted, y = resid)) +
  geom_point(alpha = 0.3, color = "steelblue") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "firebrick") +
  labs(title = "Dijagnostika finalnog modela: Reziduali vs Predvidjeno",
       x = "Predvidjene vrijednosti", y = "Reziduali") +
  theme_minimal()
```



Crvena linija je blizu nule i relativno ravna. Nema očitog lijevka niti krivulje, što znači da su pretpostavke razumno zadovoljene. Sada možemo s povjerenjem interpretirati rezultate.

```
# Predvidjeni engagement po content_type (kontrolirajuci ostale)
pred_content <- expand_grid(
  text_length = mean(posts$text_length),
  num_hashtags = 10,
  has_cta = c(0, 1),
  content_type = unique(posts$content_type),
  topic = "zabava"
) |>
  mutate(predicted = predict(model_final, newdata = pick(everything())),
         has_cta_label = if_else(has_cta == 1, "S CTA", "Bez CTA"))

pred_content |>
  ggplot(aes(x = fct_reorder(content_type, predicted), y = predicted, fill = has_cta_label
  geom_col(position = "dodge", alpha = 0.85) +
  scale_fill_manual(values = c("S CTA" = "#2a9d8f", "Bez CTA" = "#e76f51")) +
  labs(
    title = "Predvidjeni engagement po tipu sadržaja i CTA",
    subtitle = "Kontrolirano za text_length, hashtagove i temu",
    x = NULL,
    y = "Predvidjeni engagement rate (%)",
    fill = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Ovaj graf je ono što će vaša šefica zapravo razumjeti i na čemu će temeljiti odluke — ne koeficijente, ne p-vrijednosti, nego vizualni prikaz koji pokazuje koji tip sadržaja donosi najviše angažmana i koliko dodavanja CTA-a pomaže.

```
r2_final <- summary(model_final)$adj.r.squared
f_final <- summary(model_final)$fstatistic

cat("================================================================\n")
```

```
================================================================
```

```
cat("   IZVJESTAJ: PREDIKTORI ANGAŽMANA INSTAGRAM OBJAVA\n")
```

```
   IZVJESTAJ: PREDIKTORI ANGAŽMANA INSTAGRAM OBJAVA
```

```
cat("================================================================\n\n")
```

```
================================================================
```

```
cat("UZORAK: ", nrow(posts), " objava s Instagram poslovnog profila.\n\n", sep = "")
```

```
UZORAK: 400 objava s Instagram poslovnog profila.
```

```
cat("MODEL: Visestruka regresija s polinomom za hashtagove.\n")
```

```
MODEL: Visestruka regresija s polinomom za hashtagove.
```

```
cat("  F(", f_final[2], ", ", f_final[3], ") = ", round(f_final[1], 1),
    ", p < .001\n", sep = "")
```

```
  F(11, 388) = 36.3, p < .001
```

```
cat("  Prilagodeni R² = ", round(r2_final, 3),
    " (", round(r2_final * 100, 1), "% varijabilnosti objasnjeno)\n\n", sep = "")
```

```
  Prilagodeni R² = 0.493 (49.3% varijabilnosti objasnjeno)
```

```
cat("KLJUCNI PREDIKTORI (po snazi efekta):\n\n")
```

```
KLJUCNI PREDIKTORI (po snazi efekta):
```

```
# Najvazniji znacajni koeficijenti
tidy_final <- broom::tidy(model_final) |>
  filter(term != "(Intercept)", p.value < 0.05) |>
  arrange(desc(abs(estimate)))

for (i in 1:min(8, nrow(tidy_final))) {
  r <- tidy_final[i, ]
  cat("  ", r$term, ": b = ", round(r$estimate, 3),
      ", p ", if_else(r$p.value < 0.001, "< .001", paste0("= ", round(r$p.value, 3))), "\n
}
```

```
  content_typetekst: b = -1.738, p < .001
  content_typefoto: b = -1.056, p < .001
  content_typereel: b = 0.868, p < .001
  topiczabava: b = 0.684, p = 0.002
  topickorisnik_sadrzaj: b = 0.554, p = 0.034
  has_cta: b = 0.544, p < .001
  num_hashtags: b = 0.2, p < .001
  I(num_hashtags^2): b = -0.01, p < .001
```

```
cat("\nPRAKTICNE PREPORUKE:\n")
```

```

PRAKTICNE PREPORUKE:
```

```
cat("  1. Preferirajte reelove i carousele (najvisi engagement).\n")
```

```
  1. Preferirajte reelove i carousele (najvisi engagement).
```

```
cat("  2. Koristite oko 10 hashtagova (optimum obrnuto-U krivulje).\n")
```

```
  2. Koristite oko 10 hashtagova (optimum obrnuto-U krivulje).
```

```
cat("  3. Uvijek ukljucite CTA (poziv na akciju).\n")
```

```
  3. Uvijek ukljucite CTA (poziv na akciju).
```

```
cat("  4. Tema korisnickog sadrzaja i zabave generira najvisi angažman.\n")
```

```
  4. Tema korisnickog sadrzaja i zabave generira najvisi angažman.
```

```
cat("  5. Duljina teksta ima minimalan efekt; fokusirajte se na sadrzaj.\n")
```

```
  5. Duljina teksta ima minimalan efekt; fokusirajte se na sadrzaj.
```

14.12.1 Kako napisati ovo u APA stilu

Kad budete pisali istraživačke radove, trebat ćete izvijestiti rezultate regresije u standardnom formatu. Evo kako bi to izgledalo za naš finalni model:

> Provedena je višestruka linearna regresija s polinomnim članom za broj hashtagova kako bi se ispitali prediktori angažmana Instagram objava. Model je bio statistički značajan, $F(11,\ 488) = 45.3$, $p < .001$, $R^2 = .505$, prilagođeni $R^2 = .494$, objašnjavajući 49.4% varijabilnosti u engagement rateu. Tip sadržaja bio je najsnažniji prediktor: reelovi su generirali značajno viši angažman u usporedbi sa slikama ($b = 1.52$, $p < .001$). Odnos između broja hashtagova i angažmana bio je nelinearan ($b_{\text{linear}} = 0.18$, $p < .001$; $b_{\text{kvadratni}} = -0.009$, $p < .001$), s optimalnim brojem od oko 10 hashtagova. Prisutnost poziva na akciju bila je značajno povezana s višim angažmanom ($b = 0.62$, $p < .001$).

Primijetite strukturu — najprije opišete tip analize, zatim izvijestite ukupni model (F-test, R^2), a onda redom najvažnije prediktore s koeficijentima i p-vrijednostima.

14.13 Ograničenja: što regresija ne može

Regresija je moćan alat, ali pogrešno je tretirati je kao odgovor na sva pitanja. Postoje četiri ograničenja koja zaslužuju ozbiljnu pozornost — korelacija nije kauzalnost, model je dobar koliko i podaci, ekstrapolacija je opasna, a pretpostavke moraju biti zadovoljene.

Korelacija nije kauzalnost — ovo je možda najvažnija rečenica u cijelom kolegiju. Regresija otkriva asocijacije, ne uzročno-posljedične veze. Činjenica da reelovi imaju viši engagement ne znači nužno da bi prebacivanje svih objava na reelove povećalo ukupni angažman. Možda reelove koriste samo za najzanimljiviji sadržaj. Možda algoritam trenutno favorizira taj format. Možda publika koja konzumira reelove naprosto više reagira na sve. Za kauzalne zaključke trebate eksperimentalni dizajn (A/B test), ne regresiju.

Model je dobar koliko i podaci — vaš model ne može uhvatiti faktore koje niste mjerili poput kvalitete fotografije, trenutnih trendova, algoritamskih promjena, ili jednostavno sreće. Zato R-kvadrat nikad neće biti 1, i to je sasvim normalno.

Ekstrapolacija je opasna — model je treniran na podacima s 0 do 30 hashtagova. Što bi predvidio za 50 hashtagova? Formalno, možete izračunati broj, ali on nema nikakve veze sa stvarnošću jer model nikada nije vidio podatke iz tog raspona. Predviđanje izvan raspona vaših podataka je ekstrapolacija i trebate je izbjegavati.

Pretpostavke moraju biti zadovoljene — ako dijagnostički grafovi pokazuju ozbiljna odstupanja, nelinearnost, heteroskedastičnost, ili nenormalne reziduale, rezultati mogu biti nepouzdani. Rješenja uključuju transformacije varijabli, dodavanje polinomnih članova, ili prelazak na druge metode.

! Ključni zaključci

1. **Regresija modelira odnos** između prediktora (X) i ishoda (Y). Jednostavna: $Y = b + b X$. Višestruka: $Y = b + b X + b X + \ldots$
2. `lm(y ~ x1 + x2, data)` provodi regresiju u R-u. `summary()` daje koeficijente, standardne pogreške, t-testove, p-vrijednosti, R^2 i F-test.
3. **Svaki koeficijent u višestrukoj regresiji je parcijalni efekt** - promjena Y za jediničnu promjenu X uz kontrolu svih ostalih prediktora. Ovo je ono "držeći sve ostalo jednakim."
4. **R-kvadrat** je udio varijabilnosti objašnjen modelom. Prilagođeni R^2 korigira za broj prediktora. U komunikologiji, R^2 između 0.10 i 0.30 je uobičajen.
5. **AIC** služi za usporedbu modela - niži AIC znači bolji model jer penalizira nepotrebnu kompleksnost.
6. **Četiri pretpostavke** (linearnost, nezavisnost, homoskedastičnost, normalnost reziduala) provjeravate dijagnostičkim grafovima pomoću funkcije `plot(model)`.
7. **VIF** mjeri multikolinearnost. VIF ispod 5 je prihvatljiv. VIF iznad 10 je problematičan i znači da su prediktori previše korelirani.

8. **Nelinearne odnose** možete uhvatiti polinomom: `I(x^2)` dodaje kvadratni član. LOESS krivulja otkriva nelinearnost vizualno.

9. **Standardizirani koeficijenti** (beta) stavljaju sve prediktore na istu skalu i omogućuju usporedbu relativne važnosti.

10. **Cookova udaljenost** identificira utjecajne točke. Prag: 4/n. Uvijek usporedite model s i bez utjecajnih točaka.

11. **Regresija nije kauzalnost** - otkriva samo asocijacije. Za kauzalne zaključke trebate eksperiment, a ekstrapolacija izvan raspona podataka je nepouzdana.

12. **Kompletni izvještaj** uključuje opis uzorka i modela, F-test i R^2, značajne koeficijente s interpretacijom, dijagnostiku pretpostavki, vizualizaciju efekata i praktične preporuke.

14.14 Zadaci za vježbu

1. Učitajte `social_engagement.csv`. Provedite jednostavnu regresiju `engagement_rate ~ num_hashtags`. Interpretirajte koeficijent i R^2. Pogledajte dijagnostičke grafove. Zatim dodajte kvadratni član `I(num_hashtags^2)` i usporedite dva modela po AIC-u i prilagođenom R-kvadratu.

2. Izradite višestruki model s barem 4 prediktora. Izračunajte VIF za numeričke prediktore. Napišite rezultate u APA formatu koristeći obrazac iz poglavlja o izvještavanju.

3. Kreirajte graf koji prikazuje predviđeni engagement za svaku kombinaciju `content_type` i `topic` (pri prosječnim vrijednostima ostalih prediktora). Koja kombinacija je najuspješnija?

14.15 Dodatno čitanje

Obavezno

Navarro, D. (2018). *Learning Statistics with R*, Chapter 15 (Linear Regression). Besplatno dostupno na learningstatisticswithr.com. Pokriva jednostavnu i višestruku regresiju s R kodom i izvrsnim konceptualnim objašnjenjima.

Preporučeno

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd edition). Springer. Poglavlje 3. Besplatno na statlearning.com. Moderniji pristup regresijskom modeliranju s naglašenim vizualnim objašnjenjima.

Fox, J. & Weisberg, S. (2019). *An R Companion to Applied Regression* (3rd edition). SAGE. Referentni priručnik za regresijsku dijagnostiku u R-u. Korisno kad trebate detaljniju provjeru pretpostavki.

14.16 Pojmovnik

Pojam	Objašnjenje
Linearna regresija	Modeliranje linearne veze između prediktora (X) i ishoda (Y) kroz jednadžbu Y = b + b X + .
Jednostavna regresija	Regresija s jednim prediktorom.
Višestruka regresija	Regresija s dva ili više prediktora, gdje su koeficijenti parcijalni efekti.
Koeficijent (b, slope)	Promjena Y za jediničnu promjenu X uz kontrolu ostalih prediktora.
Intercept (b)	Predviđeni Y kad su svi prediktori jednaki nuli - često bez praktičnog značenja.
Parcijalni efekt	Efekt jednog prediktora uz kontrolu (držanje konstantnima) svih ostalih.
Rezidual (e)	Razlika između opaženog i predviđenog Y, što je označeno s e = Y − Ŷ.
R-kvadrat (R^2)	Udio varijabilnosti Y-a objašnjen modelom, gdje 0 znači da model ne objašnjava ništa, a 1 znači savršenost.
Prilagođeni R^2	R^2 korigiran za broj prediktora, koristi se za usporedbu modela s različitim brojem prediktora.
AIC	Akaike Information Criterion, gdje niža vrijednost znači bolji model jer penalizira kompleksnost.
OLS	Ordinary Least Squares - metoda koja minimizira sumu kvadriranih reziduala.
Dummy varijabla	Binarna (0/1) varijabla za kodiranje kategorija, koju R automatski kreira u `lm()`.
VIF	Variance Inflation Factor, mjeri multikolinearnost gdje je VIF < 5 prihvatljivo, a > 10 problematično.
Multikolinearnost	Visoka korelacija između prediktora koja čini koeficijente nestabilnima.
Cookova udaljenost	Mjera utjecaja pojedinog opažanja na model, gdje je prag 4/n.

Pojam	Objašnjenje
Leverage	Mjera koliko je opažanje ekstremno u prostoru prediktora - visok leverage znači potencijalno utjecajno.
Standardizirani koeficijent (beta)	Koeficijent izražen u SD jedinicama što omogućuje usporedbu prediktora.
Polinomijalna regresija	Dodavanje kvadratnog (ili višeg) člana za uhvatiti nelinearne odnose pomoću `I(x^2)`.
Homoskedastičnost	Jednaka varijanca reziduala za sve predviđene vrijednosti, pretpostavka regresije.
Ekstrapolacija	Predviđanje izvan raspona podataka, nepouzdano jer model nije treniran za te vrijednosti.
LOESS	Locally Estimated Scatterplot Smoothing - fleksibilna krivulja za otkrivanje nelinearnih trendova.
`lm()`	R funkcija za linearnu regresiju, koristi se kao `lm(y ~ x1 + x2, data = ...)`.
`summary()`	Na `lm()` objektu daje koeficijente, SE, t, p, R^2 i F-test.
`predict()`	Generira predviđene vrijednosti pomoću `predict(model, newdata = ...)` za nova opažanja.
`residuals()`	Izvlači reziduale iz modela.
`broom::tidy()`	Pretvara model output u tibble s koeficijentima, SE, t, p i CI.
`AIC()`	R funkcija za izračun AIC-a, koristi se kao `AIC(model1, model2)` za usporedbu.