

Osnove statistike

Kolegij za studente komunikologije

doc. dr. sc. Luka Šikić

2026-02-27

Table of contents

1 Osnove statistike	3
1.1 Struktura kolegija	3
1.2 Brzi pristup	3
1.3 Obavijesti	4
I Uvod i osnove programiranja	5
2 Tjedan 1: Zašto statistika? Uvod u istraživački dizajn	6
2.1 Imate li uopće izbora?	6
2.2 Zašto komunikolog treba statistiku	7
2.3 Kad nas intuicija iznevjeri	8
2.4 Što je uopće mjerenje?	9
2.5 Razine mjerenja	11
2.6 Pouzdanost mjerenja	13
2.7 Valjanost mjerenja	14
2.8 Varijable u istraživanju: nezavisne i zavisne	15
2.9 Eksperimentalni istraživački dizajn	16
2.10 Neeksperimentalni istraživački dizajn	17
2.11 Eksterna valjanost: možemo li generalizirati?	19
2.12 Pregled istraživačkih dizajna u komunikologiji	19
3 Tjedan 6: Programiranje u R-u	21
3.1 Koliko programiranja treba komunikolog?	21
3.2 Naši podaci: newsletter kampanje	22
3.3 Zašto funkcije? Problem kopiranja koda	23
3.4 Pisanje vlastite funkcije	24
3.5 Uvjetne naredbe: if i else	29
3.6 For petlje: ponavljanje operacija	32
3.7 map(): moderna alternativa petljama	35
3.8 DRY princip i organizacija skripte	38
3.9 Praktični primjer: automatizirana analiza po kampanjama	42
3.10 Rad s više datoteka	44
3.11 Debugging: pronalaženje i ispravljanje grešaka	46
3.12 Quarto: integracija koda, teksta i rezultata	50
3.13 Funkcionalni za složenije radne tokove	52
3.14 Kompletna analiza: automatizirani izvještaj o kampanjama	54
3.15 Dodatno čitanje	61

3.16 Pojmovnik	61
4 Tjedan 2: Uvod u R i tidyverse	64
4.1 Zašto R, a ne nešto drugo?	64
4.2 R i Positron: vaš radni prostor	66
4.3 Prve naredbe: R kao kalkulator	66
4.4 Objekti: pohranjivanje vrijednosti	69
4.5 Vektori: rad s više vrijednosti odjednom	71
4.6 Tipovi podataka	74
4.7 Tibble: moderna tablica podataka	78
4.8 Pipe operator: čitljivo ulančavanje	81
4.9 Paketi: proširivanje R-a	82
4.10 Učitavanje podataka: read_csv()	84
4.11 Istraživanje podataka: prvi uvidi	86
4.12 Logički operatori: kombiniranje uvjeta	88
4.13 Nedostajuće vrijednosti: NA	91
4.14 Korisne funkcije za vektore	93
4.15 Pisanje čistih R skripti	97
4.16 Spremanje podataka: write_csv()	98
4.17 Traženje pomoći	99
4.18 Česte greške i kako ih popraviti	100
4.19 Sve zajedno: kompletna mini analiza	102
4.20 Dodatno čitanje	105
4.21 Pojmovnik	106
5 Tjedan 3: Rad s podacima u tidyverse	109
5.1 Prljava tajna analize podataka	109
5.2 Naši podaci: anketa o medijskim navikama studenata	110
5.3 Korak nula: čišćenje imena stupaca	112
5.4 filter(): odabir redova po uvjetu	113
5.5 select(): odabir i preimenovanje stupaca	118
5.6 mutate(): kreiranje i transformacija varijabli	123
5.7 arrange(): sortiranje podataka	131
5.8 Kombiniranje glagola u pipeline	133
5.9 Brzi pregled očišćenog dataseta	136
5.10 group_by() i summarise(): statistike po grupama	139
5.11 across(): ista operacija na više stupaca	142
5.12 pivot_longer() i pivot_wider(): preoblikovanje podataka	145
5.13 Spajanje tablica: left_join()	149
5.14 Stringovi: osnove rada s tekstrom	152
5.15 Sve zajedno: kompletna analiza od sirovih do gotovih podataka	154
5.16 Dodatno čitanje	159
5.17 Pojmovnik	159

II Deskriptivna statistika i vizualizacija	163
6 Tjedan 4: Deskriptivna statistika	164
6.1 Zašto su brojke same po sebi beskorisne	164
6.2 Naši podaci: anketa o korištenju TikToka	165
6.3 Mjere centralne tendencije	166
6.4 Mjere varijabilnosti	172
6.5 Ukupni sažetak varijable	178
6.6 Deskriptivne statistike po grupama	180
6.7 Oblik distribucije: asimetrija i zaobljenost	182
6.8 Standardni rezultati (z-scores)	185
6.9 Korelacije	187
6.10 Rad s nedostajućim vrijednostima	191
6.11 Sve zajedno: kompletna deskriptivna analiza	194
6.12 Dodatno čitanje	196
6.13 Pojmovnik	197
7 Tjedan 5: Vizualizacija podataka s ggplot2	199
7.1 Zašto je vizualizacija važna	199
7.2 Naši podaci: angažman čitatelja na portalima	200
7.3 Gramatika grafike: kako ggplot2 razmišlja	202
7.4 Histogrami: distribucija jedne varijable	203
7.5 Stupčasti grafovi: kategoričke varijable	208
7.6 Boxplot: usporedba distribucija između grupa	214
7.7 Točkasti grafovi (scatterplots): odnos dviju varijabli	217
7.8 Estetike unutar i izvan aes()	222
7.9 labs(): naslovi, oznake i natpisi	225
7.10 Brzi pregled: koji graf za koji podatak?	226
7.11 Facetiranje: mali višestruki grafovi	227
7.12 Teme: vizualni izgled grafa	232
7.13 Skale boja	236
7.14 Formatiranje osi	239
7.15 Linijski grafovi: trendovi i serije	242
7.16 Kombiniranje grafova s patchwork	243
7.17 Spremanje grafova: ggsave()	246
7.18 Česte greške i kako ih izbjegći	247
7.19 Kompletna analiza: od pitanja do gotovog grafa	249
7.20 Dodatno čitanje	254
7.21 Pojmovnik	254
III Statistička teorija	257
8 Tjedan 7: Uvod u vjerojatnost	258
8.1 Zašto vjerojatnost?	258
8.2 Naši podaci: objave na društvenim mrežama	259

8.3	Što je vjerojatnost?	260
8.4	Osnovna pravila vjerojatnosti	263
8.5	Distribucije vjerojatnosti: od podataka do modela	267
8.6	Binomna distribucija	267
8.7	Distribucija u stvarnim podacima	274
8.8	Normalna distribucija	277
8.9	Z-score: standardizacija	282
8.10	R funkcije za normalnu distribuciju	284
8.11	QQ-plot: je li moja varijabla normalno distribuirana?	287
8.12	Praktična primjena: postavljanje pragova i identifikacija outliera	291
8.13	Od vjerojatnosti do statističkog zaključivanja	294
8.14	Dodatno čitanje	298
8.15	Pojmovnik	298
9	Tjedan 9: Uzorkovanje, procjena i intervali pouzdanosti	301
9.1	Temeljni problem statistike	301
9.2	Naši podaci: populacija i uzorci	302
9.3	Populacija vs uzorak: terminologija	303
9.4	Što se događa kad ponovimo uzorkovanje?	304
9.5	Distribucija uzorkovanja	305
9.6	Standardna pogreška	307
9.7	Centralni granični teorem	310
9.8	Pristranosti u uzorkovanju	313
9.9	Procjena proporcija	315
9.10	Interval pouzdanosti: osnovna ideja	317
9.11	Od z do t: mali uzorci	321
9.12	<code>t.test()</code> : sve u jednoj funkciji	323
9.13	Interval pouzdanosti za proporcije	327
9.14	Margina pogreške i planiranje uzorka	330
9.15	Čitanje medijskih anketa kritički	333
9.16	Bootstrapping: alternativni pristup	335
9.17	Potpuna analiza: povjerenje u medije po demografskim skupinama	337
9.18	Uobičajene pogreške pri interpretaciji CI	342
9.19	Zadaci za pripremu	345
9.20	Dodatno čitanje	346
9.21	Pojmovnik	346
10	Tjedan 10: Testiranje hipoteza	348
10.1	Motivacija: A/B test na Instagramu	348
10.2	Logika testiranja hipoteza	349
10.3	Od hipoteze do odluke: korak po korak	350
10.4	Jednouzorački t-test	351
10.5	Dvosmjerni vs jednosmjerni test	355
10.6	Dvouzorački t-test: Instagram A/B test	356
10.7	Simulacija: što znači p-vrijednost?	359
10.8	Greška tipa I i greška tipa II	362

10.9 P-vrijednost: što jest, a što nije	364
10.10 Veličina učinka: Cohenov d	366
10.11 Statistička snaga	369
10.12 Upareni t-test	372
10.13 Statistička vs praktična značajnost	375
10.14 Potpuna analiza: Instagram A/B test izvještaj	377
10.15 Još o p-vrijednostima: ASA izjava i višestruko testiranje	386
10.16 Pregled svih t-testova	388
10.17 Zadaci za pripremu	389
10.18 Dodatno čitanje	389
10.19 Pojmovnik	390

IV Inferencijalna statistika **392**

11 Tjedan 11: Kategorički podaci i hi-kvadrat testovi	393
11.1 Motivacija: pitanja na koja t-test ne može odgovoriti	393
11.2 Naši podaci	394
11.3 Kontingencijska tablica	395
11.4 Hi-kvadrat test za dobrotu prilagodbe (goodness-of-fit)	397
11.5 Hi-kvadrat test nezavisnosti	400
11.6 Standardizirani reziduali: gdje je veza najjača?	403
11.7 Mjera veličine učinka: Cramérovo V	404
11.8 Vizualizacija kategoričkih podataka	406
11.9 Hi-kvadrat test u praksi: korak po korak	408
11.10 Fisherov egzaktni test	410
11.11 Spajanje kategorija	413
11.12 Stratificirana analiza	415
11.13 Simpsonov paradoks	416
11.14 McNemarov test za uparene proporcije	418
11.15 Potpuna analiza: medijske navike i demografija	419
11.16 Uobičajene pogreške i ograničenja	424
11.17 Pomoćna funkcija za kompletни hi-kvadrat izvještaj	426
11.18 Pregled svih testova za kategoričke podatke	428
11.19 Zadaci za pripremu	430
11.20 Dodatno čitanje	430
11.21 Pojmovnik	430
12 Tjedan 12: Usporedba prosjeka t-testovima	432
12.1 Motivacija: vizuali u člancima	432
12.2 Podsjetnik: tri vrste t-testa	433
12.3 Pretpostavke t-testa	434
12.4 Provjera normalnosti	435
12.5 Provjera homogenosti varijance	439
12.6 Upareni t-test na podacima o vizualima	440
12.7 Sva četiri ishoda: sustavni pristup	444

12.8 Neparametrijske alternative: Wilcoxonov test	446
12.9 Efekt vizuala po kategoriji članka	448
12.10 Izvještavanje u APA formatu	450
12.11 Nezavisni t-test: kratki vs dugi članci	452
12.12 Utjecaj outliera	456
12.13 Formula pristup: t-test s formulom	459
12.14 Potpuna analiza: izvještaj za uredništvo	460
12.15 Dijagram odlučivanja: koji test odabrati?	468
12.16 Usporedba svih t-testova na jednom mjestu	469
12.17 Zadaci za pripremu	470
12.18 Dodatno čitanje	470
12.19 Pojmovnik	471
13 Tjedan 13: Usporedba više grupa ANOVA-om	473
13.1 Motivacija: vjerodostojnost vijesti po izvoru	473
13.2 Naši podaci	475
13.3 Logika ANOVA-e	477
13.4 ANOVA u R-u	480
13.5 Pretpostavke ANOVA-e	481
13.6 Vizualizacija ANOVA rezultata	485
13.7 Post-hoc testovi: Tukey HSD	487
13.8 Veličina učinka: eta-kvadrat	490
13.9 Planirane usporedbe	493
13.10 Kruskal-Wallisov test	494
13.11 Potpuna analiza: izvještaj	495
13.12 Dijagram odlučivanja: ANOVA ili nešto drugo?	499
13.13 Zadaci za pripremu	501
13.14 Dodatno čitanje	501
13.15 Pojmovnik	501
14 Tjedan 14: Linearna regresija	503
14.1 Motivacija: što pokreće angažman?	503
14.2 Od korelacije do regresije	504
14.3 Jednostavna linearna regresija	505
14.4 Što su reziduali?	508
14.5 Pretpostavke linearne regresije	509
14.6 Višestruka regresija	511
14.7 R-kvadrat i prilagodeni R-kvadrat	516
14.8 Multikolinearnost i VIF	518
14.9 Nelinearni odnosi	519
14.10 Standardizirani koeficijenti	522
14.11 Utjecajne točke: Cookova udaljenost	523
14.12 Potpuna analiza: izvještaj za menadžericu	525
14.13 Ograničenja regresije	530
14.14 Zadaci za pripremu	531
14.15 Dodatno čitanje	531

14.16Pojmovnik	532
--------------------------	-----

1 Osnove statistike

Dobrodošli na stranicu kolegija **Osnove statistike!** Ovaj kolegij pruža temeljna znanja iz statistike prilagođena studentima komunikologije, s naglaskom na praktičnu primjenu u analizi medijskih podataka.

Kroz 15 tijedana naučit ćete kako prikupljati, analizirati i interpretirati podatke koristeći programski jezik R i tidyverse ekosustav. Svi primjeri dolaze iz svijeta medija i komunikacija — od analize angažmana na društvenim mrežama do istraživanja povjerenja u medije.

i Osnovni podaci o kolegiju

Studij	Komunikologija, HKS
Semestar	Ljetni semestar 2024/2025
ECTS	6
Opterećenje	30P + 30S
Nositelj	doc. dr. sc. Luka Šikić (luka.sikic@unicath.hr)
Konzultacije	Prema dogovoru putem emaila

1.1 Struktura kolegija

Kolegij je organiziran u tri cjeline:

Cjelina 1: Deskriptivna statistika (tjedni 1–5). Upoznajete se sa statističkim razmišljanjem, naučite koristiti R i tidyverse za rad s podacima, ovlastate deskriptivnom statistikom i vizualizacijom.

Cjelina 2: Statistička teorija (tjedni 6–10). Učite programirati u R-u, upoznajete se s teorijom vjerojatnosti, distribucijama, uzorkovanjem i temeljima testiranja hipoteza.

Cjelina 3: Inferencijalna statistika (tjedni 11–15). Primjenjujete inferencijalne metode — hi-kvadrat testove, t-testove, ANOVA-u i linearnu regresiju — na primjerima iz komunikologije, te prezentirate vlastite projekte.

1.2 Brzi pristup

Resurs	Opis
Silabus	Cjelokupni plan kolegija s opisima tjedana
Raspored	Tjedni raspored s datumima i statusom
Popis literature	Obavezna i preporučena literatura
Alati	Softver i platforme za kolegij
Praktični projekt	Upute i rubrika za završni projekt

1.3 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 će te

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 zdravorazumno zaključivanje vodi na krivi trag i objasniti Simpsonov paradoks na primjeru medijskih podataka.
3. Razlikovati četiri razine mjerena (nominalna, ordinalna, intervalna, omjerna) i prepoznati kojoj razini pripadaju tipične varijable u komunikološkim istraživanjima.
4. Objasniti pojmove pouzdanosti i valjanosti mjerena 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: varijabla, nezavisna i zavisna varijabla, operacionalizacija, konfundirajuća varijabla, interna i eksterna valjanost.
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: naš portal gubi čitatelje, 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: vrijeme objave, duljina članaka, tematika, konkurenčni portali koji su lansirali novu aplikaciju, ili činjenica

da je ljeto i ljudi su na moru umjesto za računalom. Bez podataka, ova rasprava je čisto nagadanje. 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 komplikirati 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 nagadanje 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. Oглаšivačka 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: 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 uzorka, ali taj isti mehanizam nas navodi da vidimo uzorce 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 grub 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 kolega 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 objavljaju 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 objavljaju 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 mjerjenje?

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: klikom, komentarom, dijeljenjem, vremenom provedenim 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 ljudi 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. Tvrđnje 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 **varijablim**. 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 očekivanja odražavaju konstrukt koji vas zapravo zanima.

2.5 Razine mjerena

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 mjerena koja se koristi do danas. Postoje četiri razine, i svaka dopušta drugačije statističke operacije.

2.5.1 Nominalna razina

Na nominalnoj razini, brojevi (ili označke) služe samo za imenovanje kategorija. Nema nikakve prirodne hijerarhije ili redoslijeda između kategorija. U komunikološkim istraživanjima tipične nominalne varijable su: spol (muški, ženski, ostalo), tip medija (televizija, radio, web portal, društvena mreža, podcast), politička opcija, država prebivališta, ime platforme.

Jedino što možemo smisleno napraviti s nominalnim podacima jest prebrojati koliko očekivanja 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 jednakim. Klasičan primjer su odgovori na Likertovoj skali: “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: rangiranje omiljenih medija (prvi, drugi, treći izbor), obrazovna razina (osnovna škola, srednja škola, prvostupnik, magistar, doktor), 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 jednak razmak. 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: razlika između 20 i 30 stupnjeva jednak 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 jednak), 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 absolutnu nullu točku koja znači potpuno odsustvo mjerne veličine. Vrijeme provedeno na platformi je omjerna varijabla: 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 mjerjenja varijabli nije nešto što se naknadno određuje. Morate unaprijed razmisliti o tome kakve podatke želite i kakve statističke analize planirate provoditi. Ako želite izračunavati prosjekte 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 mjerena

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 mjerena nisu **pouzdana** (reliable). Pouzdanost mjerena odnosi se na stupanj u kojem dobivamo konzistentne rezultate kad ponovimo isto mjerjenje 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 tјedan 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 mjerena (test-retest reliability) mjeri konzistentnost rezultata kroz vrijeme. Ako danas mjerite stavove studenata o pouzdanosti televizijskih vijesti i ponovite isto mjerjenje 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 mjerena 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 mjerena

Pouzdanost je nužan, ali ne i dovoljan uvjet za dobro mjerjenje. Mjerjenje 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) mjerena 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 zahtjeva dokaze iz više izvora. Jedanaest mjeru medijskog angažmana (klikovi, dijeljenja, komentari, vrijeme na stranici) moglo bi se sve nazvati mjerama angažmana, ali zapravo mjeru različite stvari. Klikovi mjeru početni interes, vrijeme na stranici mjeri dubinu čitanja, komentari mjeru 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 (laboratorijskih, 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 konkurenčkih 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 prepostavljamo (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 neekperimentalnim istraživanjima nema prave manipulacije. Kad proučavate vezu između dobi i korištenja društvenih mreža, ne manipulirate ničjom 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: 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 važe 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 kulturnim 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 ispitanje 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 nadete 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 Kvaziekspimenti

Kvaziekspiment zauzima srednje mjesto između eksperimenta i opažajne studije. U kvaziekspertu, 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.

Kvaziekspimenti 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 posjeće 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, kulturne 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 intervjuiima 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 mjerjenja (nominalna, ordinalna, intervalna, omjerna) određuju koje statističke operacije smijemo primjenjivati na podatke.
5. Pouzdanost mjerjenja (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 pružanje 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 6: 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: učitavanje, čišćenje, analiza, vizualizacija, izvoz.
8. Prepoznati kada je pisanje vlastite funkcije isplativije od kopiranja koda.

3.1 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.

3.2 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, FALS~
$ revenue          <dbl> 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.

3.3 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: `group_by()`.

```
n1 |>
  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.

3.4 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: `mean()`, `filter()`, `ggplot()` su sve funkcije koje je netko napisao. Sad ćete naučiti pisati vlastite.

3.4.1 Anatomija funkcije

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

postotak_u_razlomak(25)
```

```
[1] 0.25
```

```
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).

3.4.2 Funkcija s više argumenata

```
# 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
```

```
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.

3.4.3 Default vrijednosti argumenata

Ponekad želite da argument ima podrazumijevanu (default) vrijednost koju korisnik može promjeniti 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 decimalne
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 eksplisitno navedete, koristi se vaša vrijednost. Ovo čini funkciju fleksibilnom bez opterećivanja korisnika nepotrebnim odlukama.

3.4.4 Funkcija koja radi s tibbleom

Funkcije koje primaju cijeli tibble i koriste dplyr glagole unutar sebe su izuzetno korisne u praksi.

```

# 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

```

```
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.

3.4.5 Funkcija koja vraća graf

Funkcije mogu vraćati bilo koji R objekt, uključujući ggplot grafove.

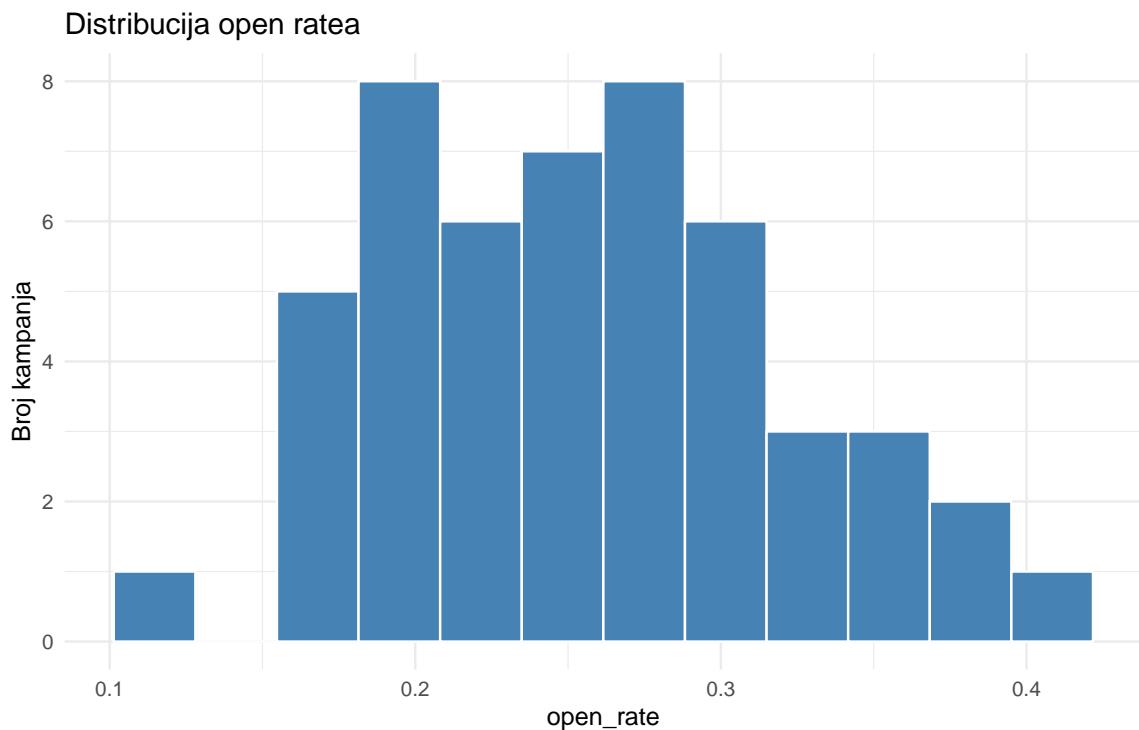
```

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()

```

```
}
```

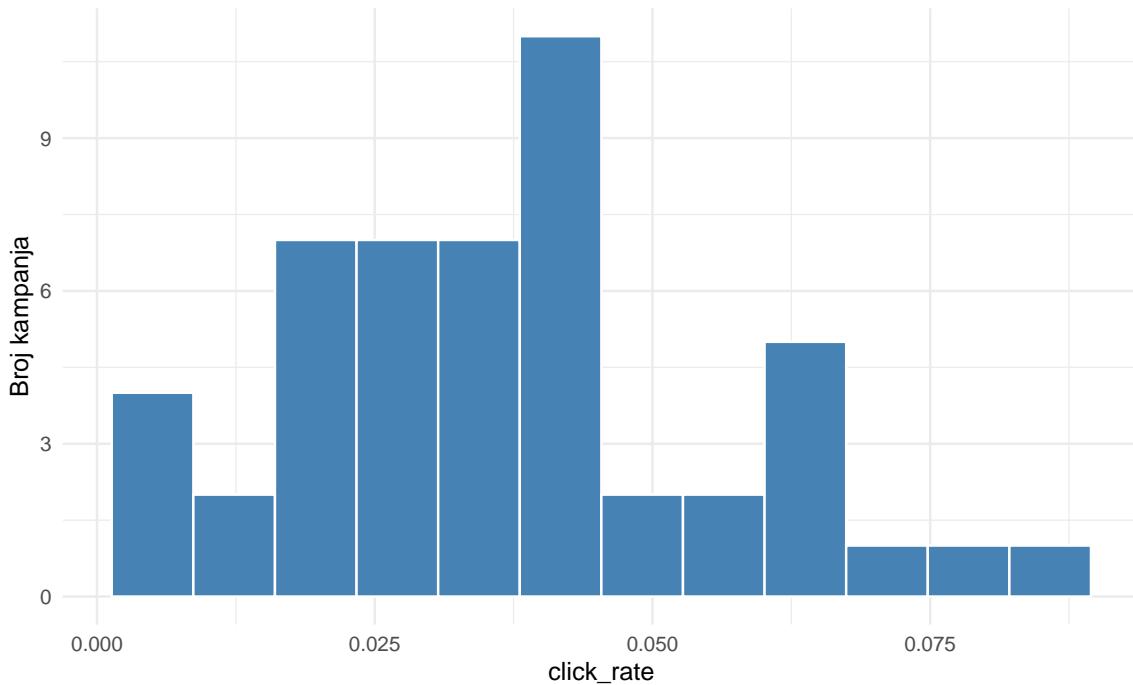
```
graf_distribucije(nl, "open_rate", "Distribucija open ratea")
```



Konstrukcija `.data[[varijabla]]` omogućuje proslijedivanje 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")
```

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

Pravilo 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.

3.5 Uvjetne naredbe: if i else

Uvjetne naredbe omogućuju R-u da doneše 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.

3.5.1 Osnovna sintaksa

```

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).

3.5.2 if, else if, else

Za više od dva ishoda, koristite `else if`.

```

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"

```
ocijeni_kampanju(0.22)
```

[1] "dobra"

```
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.

3.5.3 Razlika između if/else i if_else()/case_when()

Ovo je česta točka zbumjenosti. 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 obradivati 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 ocjeni_kampanju)
```

Pravilo je jednostavno. Unutar mutate() koristite if_else() ili case_when(). Izvan mutate(), u funkcijama i skriptama, koristite klasični if/else.

3.5.4 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()` eksplisitno 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 krhkikh. 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.

3.6 For petlje: ponavljanje operacija

Petlja je naredba koja ponavlja blok koda za svaki element u skupu. `for` petlja u R-u ima jednostavnu sintaksu.

3.6.1 Osnovna for petlja

```

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.

3.6.2 For petlja za generiranje rezultata

Čest obrazac je korištenje petlje za prikupljanje rezultata u listu ili tibble.

```

# 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

```

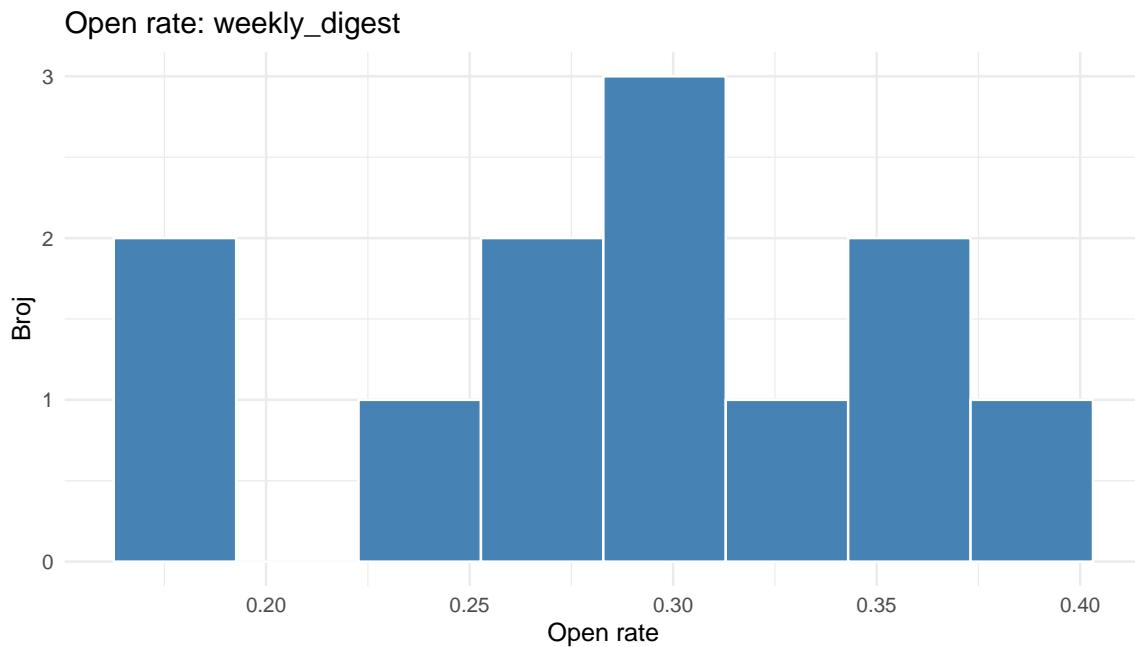
2	weekly_digest	12	0.289	0.029
3	breaking_news	7	0.198	0.0382
4	sponsored	8	0.248	0.0223
5	event_promo	6	0.25	0.0294

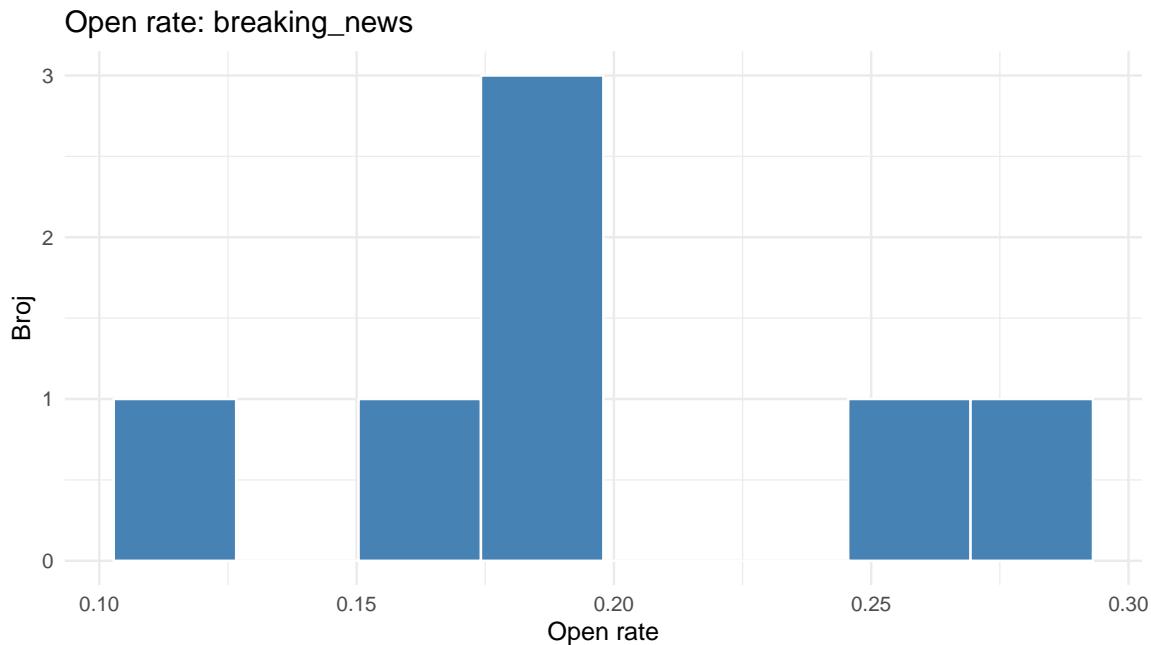
Kreiramo praznu listu `rezultati`, u svakoj iteraciji računamo sažetak i pohranjujemo ga u listu pod imenom tipa kampanje, a na kraju sve spajamo u jedan tibble s `bind_rows()`.

3.6.3 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 eksplisitno ispisati s `print()`. Izvan petlje, R automatski ispisuje zadnji objekt, ali unutar petlje to ne radi. Ovo je čest izvor frustracija 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).

3.7 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.

3.7.1 Osnovni `map()`

```

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.

3.7.2 Skraćena lambda sintaksa

Umjesto `function(tip) { ... }` možete koristiti skraćenu lambda sintaksu s tildom.

```

# 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

```

```

<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.

3.7.3 Varijante map-a

`map()` uvijek vraća listu. Kad znate kakav tip rezultata očekujete, koristite specifičniju varijantu.

```

# map_dbl() vraća numerički vektor
projeci <- map_dbl(tipovi, \(tip) {
  nl |>
    filter(campaign_type == tip) |>
    pull(open_rate) |>
    mean()
}

tibble(tip = tipovi, prosjek_or = round(projeci, 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.

3.7.4 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).

3.8 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.

3.8.1 Primjer: parametri na jednom mjestu

```

# 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.

3.8.2 Struktura analitičke skripte

Dobro organizirana skripta ima jasne sekcije. Svaka sekcija radi jednu stvar i jasno je označena.

```

# =====
# Analiza newsletter kampanja
# Autor: Ime Prezime
# Datum: 2025-03-29
# Opis: Sažetak performansi email kampanja
# =====

```

```

# 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.

3.8.3 Pomoćne funkcije na vrhu skripte

Kad imate funkcije koje koristite na više mesta 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

```

```

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.

3.9 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.

```

# Funkcija za kompletну 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.

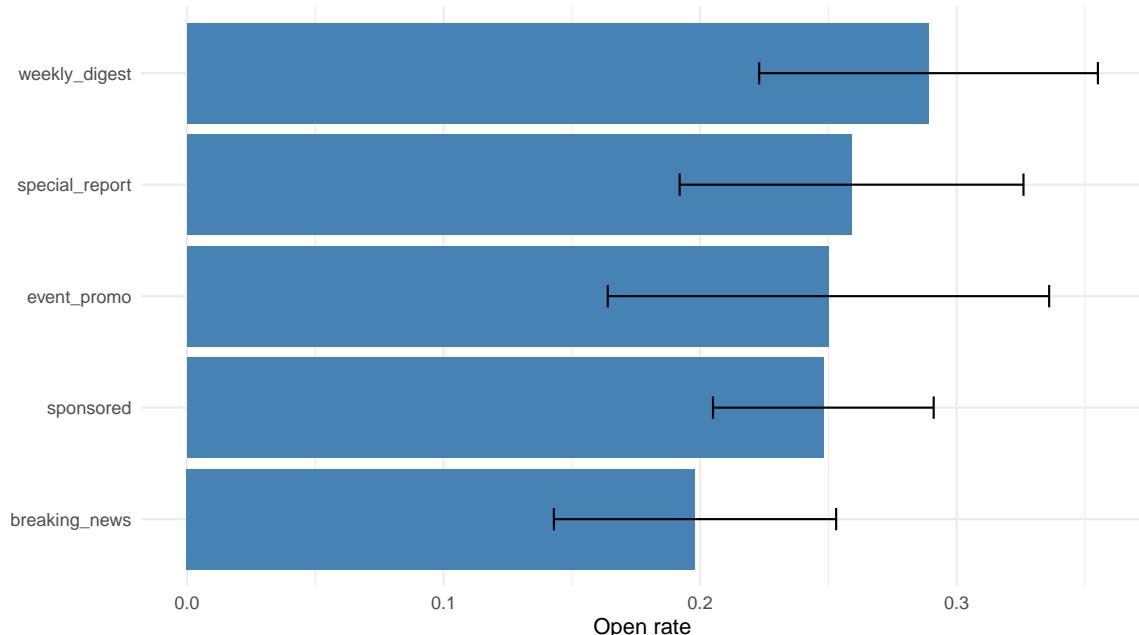
```

# 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 = projek, error bar = +/- 1 SD",
    x = NULL,
    y = "Open rate"
  ) +
  theme_minimal()

```

Prosječni open rate po tipu newsletter kampanje

Stupac = prosjek, error bar = ± 1 SD



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.

i 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: rad s više datoteka, debugging, Quarto izvještaji i kompletirana automatizirana analiza.

3.10 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.

3.10.1 Pronalaženje datoteka

Funkcija `list.files()` pronalazi datoteke u direktoriju po zadatom 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/instagram_ab_test.csv"
[4] "../resources/datasets/media_habits_raw.csv"
[5] "../resources/datasets/media_population.csv"
[6] "../resources/datasets/media_trust.csv"
[7] "../resources/datasets/newsletter_campaign.csv"
[8] "../resources/datasets/newsletter_campaigns.csv"
[9] "../resources/datasets/social_media_survey.csv"
[10] "../resources/datasets/social_posts.csv"
[11] "../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.

3.10.2 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.

3.10.3 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).

3.11 Debugging: pronalaženje i ispravljanje grešaka

Greške su neizbjeglan 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.

3.11.1 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: pogrešno ime (tipfeler), 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).

3.11.2 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
```

```
# 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.

3.11.3 print() i glimpse() kao dijagnostika

Unutar funkcija i petlji, dodajte privremene print() naredbe da vidite što se dogada.

```

# 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 zatrپava konzolu nepotrebnim ispisom.

3.11.4 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.

3.12 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.

3.12.1 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 tekstrom. Rezultat je profesionalan izvještaj u kojem su analiza i prezentacija neodvojivi.

3.12.2 Chunk opcije za kontrolu ispisu

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.

3.12.3 Inline R kod

Osim code chunkova, R vrijednosti možete umetnuti direktno u tekst.

```

n_kampanja <- nrow(nl)
projek_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.

3.12.4 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: skripta za teški posao (čišćenje, modeliranje), Quarto za prezentaciju rezultata. Skripta generira čiste podatke i grafove, Quarto ih ugrađuje u priču.

3.13 Funkcionali za složenije radne tokove

Vratimo se purrr paketu i pogledajmo naprednije obrasce koji su korisni u praksi.

3.13.1 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 tihom 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 zatrپava konzolu listom NULL vrijednosti.

3.13.2 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.

3.13.3 `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 proslijedi i element i njegovo ime (ili indeks). Korisna je kad trebate znati koji element trenutno obrađujete, na primjer za imenovanje rezultata ili za dijagnostiku.

3.13.4 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.

3.14 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
  )

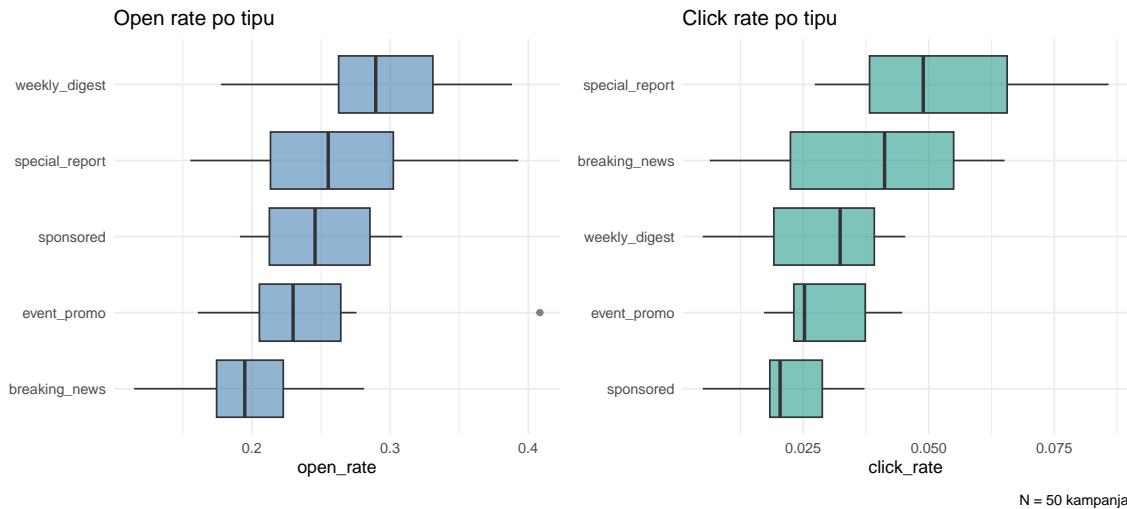
# A tibble: 12 x 7
  campaign_type      n informativni personalizirani upitni brojke hitno
  <chr>        <int>      <dbl>            <dbl>      <dbl>      <dbl>      <dbl>
1 breaking_news      5       0.182        NA       NA       NA       NA
2 event_promo         2        NA        0.285     NA       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")
  )

```

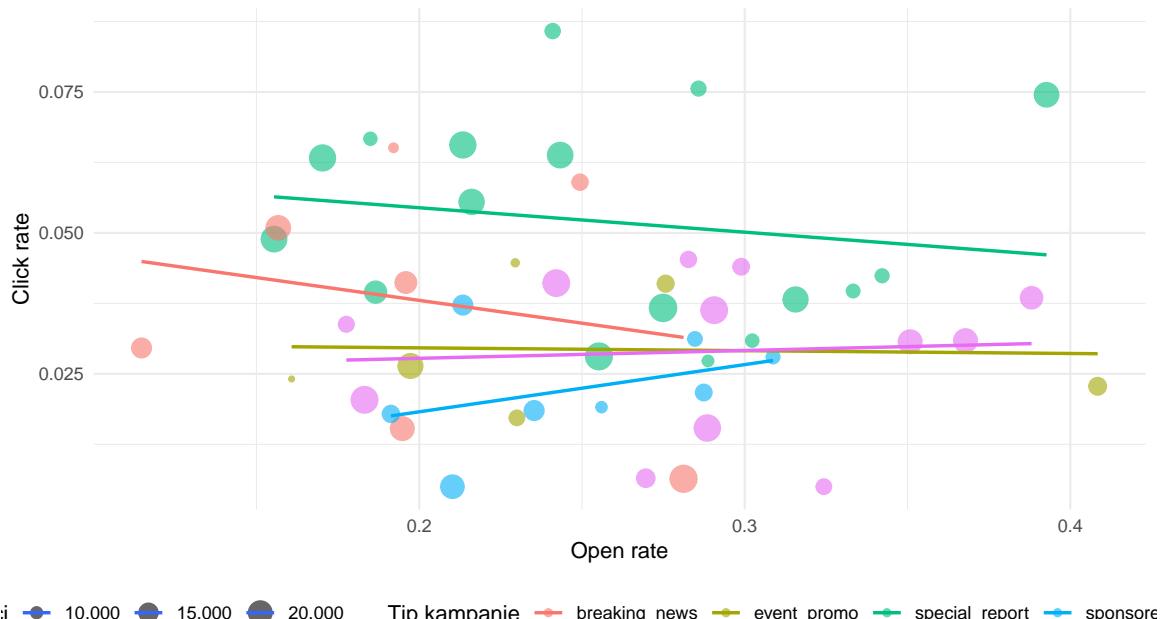
Performanse newsletter kampanja
Usporedba open rate i click rate po tipu kampanje



```
# 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")
```

Kampanje s višim open rateom tendiraju imati viši click rate

Velicina tocke proporcionalna broju pretplatnika



```
# 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

- Funkcije su alat za izbjegavanje ponavljanja koda. Pravilo tri: ako ste kopirali isti kod tri puta, pretvorite ga u funkciju. Default vrijednosti argumenata čine funkcije fleksibilnima.
- Klasični `if/else` radi s jednom vrijednošću (za skripte i funkcije). `if_else()` i `case_when()` su vektorizirani (za `mutate()`). Ne miješajte ih.
- 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: čitanje poruka, izolacija problema (korak po korak), privremeni `cat()/print()` ispisi.
9. DRY princip: parametri na jednom mjestu, logika u funkcijama, struktura skripte u jasnim sekcijama.
10. Quarto dokumenti integriraju tekst, kod i rezultate. Koristite ih za izvještaje, radevine i prezentacije. R skripte su za teški izračun, Quarto za komunikaciju.
11. Chunk opcije (`echo`, `eval`, `message`, `warning`, `fig-width`) kontroliraju što se prikazuje u renderiranom dokumentu. `echo: false` sakriva kod za klijente.
12. Cilj ponovljive analize: 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**: što je vjerojatnost, kako ju računamo, 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.

3.15 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, poglavlj 27 iteraciju s purrr, poglavlj 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, poglavlj 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.

3.16 Pojmovnik

Pojam	Objašnjenje
Funkcija	Objekt koji prima argumente, izvršava operacije i vraća rezultat. Definira se s <code>function()</code> .
Argument	Ulagani podatak funkcije. Navodi se unutar zagrade 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 <code>return()</code> .
<code>return()</code>	Eksplicitno vraća vrijednost i izlazi iz funkcije. Korisno za ranu validaciju.
<code>if/else</code>	Uvjetna naredba za kontrolu toka. Radi s jednom vrijednošću (nije vektorizirana).

Pojam	Objašnjenje
<code>if_else()</code>	Vektorizirana uvjetna funkcija za <code>mutate()</code> . Radi na cijelom stupcu.
<code>case_when()</code>	Vektorizirana funkcija za složeno rekodiranje s više uvjeta.
<code>for</code> petlja	Ponavlja blok koda za svaki element u skupu. Sintaksa: <code>for (x in skup) { ... }</code> .
<code>map()</code>	purrr funkcija koja primjenjuje funkciju na svaki element i vraća listu.
<code>map_dbl()</code>	Varijanta <code>map()</code> koja vraća numerički vektor.
<code>map_chr()</code>	Varijanta <code>map()</code> koja vraća tekstualni vektor.
<code>map_lgl()</code> <code>map2()</code>	Varijanta <code>map()</code> koja vraća logički vektor. purrr funkcija za paralelnu iteraciju preko dva vektora.
<code>imap()</code>	purrr funkcija koja proslijediće i element i njegovo ime/indeks.
<code>walk()</code>	Varijanta <code>map()</code> za popratne efekte (spremanje datoteka). Ne vraća rezultat.
<code>possibly()</code>	purrr funkcija koja omotava funkciju u zaštitni sloj. Greška vraća default vrijednost umjesto prekida.
<code>purrr</code>	Paket iz tidyverse za funkcionsko programiranje.
<code>nest()</code>	tidy funkacija koja pakira podatke grupe u ugniježđeni tibble.
<code>bind_rows()</code> <code>list.files()</code>	Vertikalno spaja listu tibbleova u jedan. Base R funkcija za pronalaženje datoteka u direktoriju po uzorku.
<code>set_names()</code>	Dodjeljuje imena elementima vektora ili liste.
<code>DRY</code>	Don't Repeat Yourself. Princip da informacija postoji na jednom mjestu u kodu.
Lambda funkcija	Anonimna funkcija. Piše se kao <code>\(x) x + 1</code> ili <code>function(x) x + 1</code> .
<code>.data[[var]]</code>	Pristup stupcu po imenu pohranjenom u varijabli. Za tidyverse funkcije.
<code>cat()</code>	Ispis teksta u konzolu. Bez navodnih oznaka i indeksa.
<code>warning()</code> <code>stop()</code>	Ispis upozorenja. Ne zaustavlja program. Ispis greške i zaustavljanje programa. Za kritične probleme.

Pojam	Objašnjenje
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 gresaka 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.

4 Tjedan 2: Uvod u R i tidyverse

Vaš novi najdraži alat za rad s podacima

`library(tidyverse)`

i 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.

4.1 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 gumbе 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. 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 hoćeće li imati pristup nakon diplome. SPSS licenca košta stotine eura godišnje. Osim toga, R ima ogromnu zajednicu korisnika i tisuće paketa za svaku zamislu 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 traženje 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: ići polako, objašnjavati svaki korak i ne prepostavljati nikakvo prethodno iskustvo 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 zbuњujući. 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.

4.2 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: pisanje koda u skripti (gornji lijevi panel), izvršavanje koda u konzoli (donji panel) i pregled rezultata. Sve ostalo ćete naučiti usput, kad bude potrebno.

4.2.1 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.

4.3 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.

```
# Zbrajanje  
10 + 25
```

```
[1] 35
```

```
# Oduzimanje  
100 - 37
```

```
[1] 63
```

```
# Množenje  
12 * 8
```

```
[1] 96
```

```
# Dijeljenje  
144 / 12
```

```
[1] 12
```

```
# 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.

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

```
[1] 11
```

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

```
[1] 16
```

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

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

```
[1] 12
```

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

```
[1] 42
```

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

```
[1] 3.14
```

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

```
[1] 4.60517
```

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

```
[1] 2
```

Funkcija `sqrt()` računa kvadratni korijen, `abs()` vraća absolutnu 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.

4.4 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_ispitnika <- 500  
  
# Pohranjujemo prosječno dnevno korištenje (u minutama)  
prosjek_minuta <- 87.3  
  
# Sada možemo koristiti te objekte  
n_ispitnika
```

```
[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_ispitnika * prosjek_minuta  
ukupno_minuta
```

```
[1] 43650
```

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

```
[1] 727.5
```

```
# Prosječno korištenje u satima  
projek_sati <- projek_minuta / 60  
round(projek_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.

4.4.1 Pravila imenovanja objekata

R ima nekoliko pravila i mnogo dobrih praksi za imenovanje objekata. Pravila su stroga: ime mora počinjati slovom (ne brojem), ne smije sadržavati razmake ni specijalne znakove osim točke i podvlake, i R razlikuje velika i mala slova (`projek` i `Projek` 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: `projek_minuta`, `n_ispitani`, `dnevno_koristenje`. Ova konvencija je standard u tidyverse zajednici i čini kod čitljivijim od alternativa poput `projekMinuta` (camelCase) ili `projek.minuta` (dot notation).

```
# Dobra imena (snake_case, opisna)  
broj_portala <- 15  
projecni_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 `projek_dobi` umjesto generičnog `mean`.

4.4.2 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`, nemojte koristiti isti objekt `temp` za oboje.

4.5 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
```

```
# 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 stavlaju 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.

```
# 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
```

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

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

```
# 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.

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

```
[1] 8
```

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

```
[1] 68.75
```

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

```
[1] 66.5
```

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

```
[1] 44.18064
```

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

```
[1] 8
```

```
max(dnevno_tiktok)
```

```
[1] 135
```

```
# 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.

4.5.1 Indeksiranje vektora

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

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

```
[1] 112
```

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

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

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

```
[1] 112 135
```

```
# 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()` primjenjena 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.

4.6 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.

4.6.1 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
```

4.6.2 Tekstualni tip (character)

Tekst u R-u se označava navodnicima, bilo jednostrukim ili dvostrukim. U ovom kolegiju koristimo dvostrukе 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()`.

```
ime <- "Portal"  
broj <- "Index.hr"  
  
# paste() spaja s razmakom (po defaultu)  
paste(ime, broj)
```

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

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

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

4.6.3 Logički tip (logical)

Logičke vrijednosti su samo dvije: TRUE i FALSE. Nastaju kad R evaluira uvjete.

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

```
[1] TRUE
```

```
10 < 5
```

```
[1] FALSE
```

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

```
[1] TRUE
```

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

```
[1] TRUE
```

```
# 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.

4.6.4 Faktorski tip (factor)

Faktori su poseban tip za kategorijalene podatke. Iznad smo vidjeli da razine mjerjenja 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.

4.6.5 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.

4.7 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", "TikTok", "Instagram"),  
  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 popunjena 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.

4.7.1 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 dohvate jedan stupac.

4.8 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: 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 nelegantna.

```
# 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 jer morate čitati iznutra prema van: najprije vidite `mean()`, pa se morate 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 zatrپava 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.

```
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. Zamislite da ovo morate napisati bez pipea: `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.

4.9 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, jedan paket (zapravo kolekcija paketa) dominira nad svima ostalima: **tidyverse**.

4.9.1 Što je tidyverse?

Tidyverse nije jedan paket nego skup od osam paketa koji dijele zajedničku filozofiju dizajna i besprijeckorno 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. `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.

4.9.2 Instalacija i učitavanje paketa

Paketi se instaliraju jednom, a učitavaju svaki put kad pokrenete R sesiju. Analogija: instalacija je poput kupnje knjige (radite to jednom), a učitavanje je 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.

4.10 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 pogoda 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 pogoda tipove stupaca i daje informativnije poruke.

4.10.1 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: stupci, tipovi, prvih nekoliko vrijednosti
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", "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, 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", "port~
$ weekly_posts         <dbl> 2, 9, 6, 14, 6, 0, 0, 0, 3, 3, 19, 0, 2, 0, 13, ~
$ privacy_concern     <dbl> 7, 6, 8, 7, 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.

```
# 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.

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

```
[1] 500
```

```
ncol(social)
```

```
[1] 12
```

```
# 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.

4.10.2 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 odabralih 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.

4.11 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: minimum, prvi kvartil, medijan, prosjek, treći kvartil i maksimum. Detaljno ćemo objasniti sve ove mjere u tјednu 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: 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.

4.12 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: 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.

4.13 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 prepostavlja 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
```

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

```
[1] 16
```

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

```
[1] 4
```

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

```
[1] 0.8164966
```

4.13.1 Provjera i prepoznavanje NA

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

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

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

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

```
[1] 1
```

```
# 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.

4.13.2 NA u tibbleovima

Kad učitavate podatke, prazne čelije 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 čelija, razmak umjesto broja, tekst u numeričkom stupcu). Pravilo je jednostavno: uvijek provjerite, nikad ne pretpostavljajte.

4.14 Korisne funkcije za vektore

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

4.14.1 Generiranje nizova

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

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

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

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

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

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

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

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

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

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

```
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).

4.14.2 Sortiranje i redoslijed

```
minute <- c(95, 22, 112, 45, 78, 8, 135, 55)  
  
# Sortirano uzlazno  
sort(minute)
```

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

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

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

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

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

```
# 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.

4.14.3 Jedinstvene vrijednosti i tablice frekvencija

```
platforme <- c("TikTok", "Instagram", "TikTok", "YouTube", "Instagram", "TikTok", "Facebook")  
  
# Jedinstvene vrijednosti  
unique(platforme)
```

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

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

```
[1] 4
```

```
# 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.

4.14.4 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 prosjekte i standardne devijacije. Na ovom kolegiju ćemo se držati tih konvencija.

4.15 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.

4.15.1 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/0snove 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/analyse/` 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.

4.16 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()` spremi 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")`.

4.17 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.

4.17.1 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".

4.18 Česte greške i kako ih popraviti

Greške su normalan i neizbjeglan 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.

4.18.1 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), zaboravljeni pokretanje koda koji kreira objekt, ili pokretanje koda izvan redoslijeda (pokušavate koristiti objekt prije nego ste ga kreirali). Rješenje: provjerite ime, provjerite jeste li pokrenuli sve prethodne retke koda.

4.18.2 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.

4.18.3 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.

4.18.4 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.

4.19 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 je: pisanje koda u editoru, izvršavanje u konzoli, 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: zagлавje, učitavanje paketa, učitavanje podataka, pregled podataka, analiza. 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 i `pivot_longer()` / `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. Eksperimentirajte: promjenite brojeve, dodajte nove vektore, isprobajte 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`.

4.20 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, poglavljje 3 pokriva transformaciju podataka, a poglavljje 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.

4.21 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. Klasifikacija vrijednosti koja određuje moguće operacije. Osnovni tipovi: numeric, character, logical, factor.
Tip podataka	

Pojam	Objašnjenje
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 (<code> ></code>)	Operator koji proslijedi 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 <code>install.packages()</code> , učitava s <code>library()</code> .
<code>read_csv()</code>	Funkcija iz paketa readr za učitavanje CSV datoteke. Vraća tibble i automatski pogoda tipove stupaca.
<code>write_csv()</code>	Funkcija iz paketa readr za spremanje tibble u CSV datoteku. Korisna za izvoz obrađenih podataka.
<code>glimpse()</code>	Funkcija iz tidyverse koja prikazuje strukturu dataseta: stupce, tipove i prvih nekoliko vrijednosti.
<code>count()</code>	Funkcija iz dplyr koja prebrojava opažanja po kategorijama.
<code>group_by()</code>	Funkcija iz dplyr koja dijeli podatke u grupe po jednoj ili više varijabli. Koristi se u kombinaciji sa <code>summarise()</code> .
<code>summarise()</code>	Funkcija iz dplyr koja izračunava sažetke (prosjek, medijan, SD i sl.) za svaku grupu ili za cijeli dataset.
<code>filter()</code>	Funkcija iz dplyr koja odabire retke koji zadovoljavaju zadani uvjet.
<code>mutate()</code>	Funkcija iz dplyr koja kreira nove stupce ili mijenja postojeće.
<code>arrange()</code>	Funkcija iz dplyr koja sortira retke po vrijednostima jednog ili više stupaca.
<code>select()</code>	Funkcija iz dplyr koja odabire stupce po imenu.

Pojam	Objašnjenje
NA (not available)	Oznaka za nedostajuću vrijednost u R-u. Svaka operacija s NA vraća NA osim ako se eksplicitno kaže <code>na.rm = TRUE</code> .
<code>is.na()</code>	Funkcija koja provjerava jesu li vrijednosti NA. Jedini ispravan način provjere (nikad koristiti <code>== NA</code>).
Logički operatori	Operatori za kombiniranje uvjeta: <code>&</code> (i), <code> </code> (ili), <code>!</code> (ne), <code>%in%</code> (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 <code>getwd()</code> .
snake_case	Konvencija imenovanja: riječi odvojene podvlakom, sve malim slovima (npr. <code>dnevno_koristenje</code>). Standard u tidyverse zajednicici.
CSV (comma-separated values)	Tekstualni format za pohranu tabličnih podataka u kojem su vrijednosti odvojene zarezima.
Komentar	Tekst u kodu koji počinje s <code>#</code> i koji R ignorira. Služi za objašnjavanje koda.

5 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 rekurzivno vrednosti 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.

5.1 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: na 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 čelije. 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 tidyverse 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.

5.2 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 respondent`                <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, zgrade 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 respondent` `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.

5.3 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, zgrade i specijalne znakove. Kad ih želite koristiti u kodu, morate ih stavljati u obrnute navodnike: `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: mala slova, razmaci zamijenjeni podvlakama, specijalni znakovi uklonjeni.

```
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~
$ portalni_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~

```

5.4 filter(): 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.

5.4.1 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 2 2025-03-28 12:54:00 18 fema~ Split 1 <NA>
3 3 2025-03-25 15:16:00 20 muški Zagr~ 1 0
4 4 2025-04-20 11:09:00 20 Žens~ Zagr~ 2 0
5 5 2025-04-07 18:06:00 19 muški Split 1 0
# i 10 more variables: portalni_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.

5.4.2 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 eksplisitniji. Koristite što vam je čitljivije.

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

```

# 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

# 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%`.

5.4.3 Filtriranje numeričkih raspona

```

# 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

# 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.

5.4.4 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”.

5.4.5 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
```

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

```
[1] 32
```

```
# 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 eksplisitno zadržati retke s NA, morate to navesti.

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

```
[1] 124
```

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

```
[1] 32
```

```
# 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.

5.5 `select()`: odabir i preimenovanje stupaca

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

5.5.1 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

```

Stupe 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.

5.5.2 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_platfrome_koristi"	"koliko_cesto_prati_vijesti"

5.5.3 Pomoćne funkcije za odabir

Kad imate mnogo stupaca, ručno nabranjanje 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>

```

# 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>

# 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 tekstrom. `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).

5.5.4 Preimenovanje stupaca

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

```

# 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.

5.5.5 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 Zagreb 2025-03-28 17:05:00 20 ženski 2 0
2 2 Zadar 2025-04-20 21:11:00 27 Muški 3 0
3 3 Zagreb 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.

5.6 mutate(): kreiranje i transformacija 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.

5.6.1 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.

5.6.2 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)
```

	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.

5.6.3 Čišćenje stupca spol: `str_to_lower()` i `case_when()`

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

```
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.

```
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 `str_to_lower()`, pa onda provjere.

5.6.4 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.

5.6.5 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/umjereni"),
    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/umjereni          117
2 visoko                  133
```

Funkcija `if_else()` prima tri argumenta: 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.

5.6.6 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 nagadati.

5.6.7 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 celije. Moramo ga pretvoriti u broj.

```

# 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

# "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”.

5.7 `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
```

```
# 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
```

5.7.1 Sortiranje po više stupaca

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

```
# 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.

5.7.2 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.

5.8 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: 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. Odaberis 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 kompletну analizu. Ovo je radni obrazac koji ćete koristiti stotine puta.

Pogledajmo drugi primjer. Želimo 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.

5.8.1 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", "~"
$ grad                    <chr> "Zagreb", "Zadar", "Zagreb", "Split", "Zagreb", "~"
$ godina                  <dbl> 2, 3, 1, 1, 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.

5.9 Brzi pregled očišćenog dataseta

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

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

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

```
# 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
```

```
# 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. Najmladi (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.

5.10 group_by() i summarise(): 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?

5.10.1 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"`.

5.10.2 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.

5.10.3 `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

```

```
# 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()`.

5.10.4 `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.

```
# 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 iznenadjenja.

5.11 `across()`: ista operacija 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”.

5.11.1 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 imenovanu 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.

5.11.2 `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.

5.11.3 `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.

5.12 pivot_longer() i pivot_wider(): 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.

5.12.1 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.

5.12.2 pivot_longer(): od širokog prema dugačkom

```
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. Argumenti su:

`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 dobroj 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 prirodnog s dugačkim podacima.

5.12.3 pivot_wider(): od dugačkog prema širokom

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.

5.12.4 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 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 je: učitajte podatke, pretvorite u dugački format, analizirajte, pretvorite rezultate u široki format za prezentaciju.

5.13 Spajanje tablica: `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.

5.13.1 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.

5.14 Stringovi: osnove rada s tekstrom

U komunikologiji se često radi s tekstualnim podacima: imena platformi, naslovi članaka, 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
```

5.14.1 Brojanje i izdvajanje uzorka

```
# 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.

5.14.2 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 razmaka s početka i kraja teksta, što je korisno kad ispitanici slučajno unesu razmak.

5.15 Sve zajedno: kompletna analiza od sirovih do gotovih podataka

Zaokružimo ovo predavanje tako da napišemo kompletну analizu koja prolazi kroz sve faze: učitavanje, čišćenje, transformaciju, analizu i prezentaciju 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

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

```

# FAZA 2: Provjera
# Koliko NA po stupcu?
anketa_clean |>
  summarise(across(everything(), ~sum(is.na(.x)))) |>
  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 TV              42.3  33.0   33.0
2 Web portali    43.6  34.4   34.4
3 Društvene mreže 42.9  35.6   35.6

```

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

```
# 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 `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: učitavanje, čišćenje, provjera, analiza, prezentacija. Dokumentirajte svaki korak i svaki izbor (osobito koliko redova gubite filtranjem).

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 kompletne 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).

5.16 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.

5.17 Pojmovnik

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

Pojam	Objašnjenje
<code>if_else()</code>	Funkcija za binarno rekodiranje. Prima uvjet, vrijednost za TRUE i vrijednost za FALSE.
<code>between()</code>	Pomoćna funkcija: provjera je li vrijednost unutar raspona. Kratica za <code>x >= left & x <= right</code> .
<code>str_detect()</code>	stringr funkcija: provjerava sadrži li tekst zadani uzorak. Vraća TRUE/FALSE.
<code>str_to_lower()</code>	stringr funkcija: pretvara tekst u mala slova. Korisna za standardizaciju.
<code>str_replace()</code>	stringr funkcija: zamjenjuje prvo pojavljivanje uzorka u tekstu. <code>str_replace_all()</code> zamjenjuje sva.
<code>str_count()</code>	stringr funkcija: broji pojavljivanja uzorka u tekstu.
<code>pivot_longer()</code>	tidy funkacija za pretvaranje stupaca u redove (široki u dugački format). Ključni argumenti: <code>cols</code> , <code>names_to</code> , <code>values_to</code> .
<code>pivot_wider()</code>	tidy funkacija za pretvaranje redova u stupce (dugački u široki format). Ključni argumenti: <code>names_from</code> , <code>values_from</code> .
<code>left_join()</code>	dplyr funkcija za spajanje dviju tablica po zajedničkom stupcu. Zadržava sve retke iz lijeve tablice.
<code>inner_join()</code>	Spajanje koje zadržava samo retke koji postoje u obje tablice.
<code>anti_join()</code>	Spajanje koje zadržava retke iz lijeve tablice koji nemaju podudaranje u desnoj.
<code>clean_names()</code>	Dijagnostički alat. janitor funkcija: pretvara imena stupaca u snake_case. Uklanja razmake, zagrade i specijalne znakove.
<code>drop_na()</code>	tidy 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 celija vrijednost.
Široki format	Organizacija podataka u kojoj su različita mjerena iste varijable raspoređena u zasebne stupce. Čitljiv za ljudi.

Pojam	Objašnjenje
Dugački format	Organizacija podataka u kojoj su različita mjerena u zasebnim redovima s identifikacijskim stupcem. Pogodan za analizu i vizualizaciju.
Pipeline	Niz operacija spojenih pipe operatorom (<code> ></code>) 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”).

Dio II

Deskriptivna statistika i vizualizacija

6 Tjedan 4: 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 огромnom 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: 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)
```

	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 mladi 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 intuitivna, ali čim pokušate biti precizni, stvari

postaju komplikiranije 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: aritmetičku sredinu, skraćenu sredinu, 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 apstraktно, 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

projek 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 projek 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. Projek je 30 minuta. Sada zamislite da šesti korisnik provodi 480 minuta (8 sati!) dnevno. Projek 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"),
  projek = c(mean(bez_outliera), mean(s_outlierom)),
  medijan = c(median(bez_outliera), median(s_outlierom))
)

# A tibble: 2 x 3
#>   skup           projek    medijan
#>   <chr>        <dbl>     <dbl>
#> 1 Bez ekstremne vrijednosti      30     30
#> 2 S ekstremnom vrijednošću     105    32.5
```

Primjetite kako projek 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 projek 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 projek, izbacimo određeni postotak najmanjih i najvećih vrijednosti. Najčešće se koristi 5% skraćivanje, što znači da izbacimo 5% najmanjih i 5% najvećih vrijednosti, pa izračunamo projek 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 funkcioniра. 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 jednakom, 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 poretku. Vidjeli smo to u primjeru iznad. Zato kažemo da je medijan **robustna** 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 mlađih 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 za mlade korisnike (oko 100 minuta) i jedan 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 izvjestiti 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 orientir, ali za ozbiljnu analizu trebamo nešto bolje.

6.4.2 Prosječno absolutno odstupanje

Prije nego prijeđemo na varijancu, vrijedi se nakratko zadržati na jednoj mjeri koja je konceptualno jednostavnija, a to je **prosječno absolutno odstupanje** (average absolute deviation, AAD). Ideja je sljedeća: za svako opažanje izračunamo koliko se razlikuje od aritmetičke sredine (to je odstupanje ili devijacija), uzmememo apsolutnu vrijednost tog odstupanja (jer nas zanima veličina odstupanja, ne smjer), i izračunamo prosjek svih apsolutnih odstupanja.

$$AAD = \frac{1}{N} \sum_{i=1}^N |X_i - \bar{X}|$$

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 davnio 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:

Otprikljike 68% podataka nalazi se unutar jedne standardne devijacije od sredine. Otprikljike 95% podataka nalazi se unutar dvije standardne devijacije. Otprikljike 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 otprikljike 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 komplikirano, 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 dobroj 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     6.0
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 eksplisitno 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       0.0          3
4 45+          50         2.0       0.0          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 oštiri 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 prepostavljaju normalnost.

Praktični savjet

Asimetrija i zaobljenost postaju osobito važne kad počnemo raditi inferencijsku statistiku (t-testove, ANOVA-u, regresiju) jer mnogi od tih testova prepostavljaju normalnost distribucije. U tom kontekstu, asimetrija i zaobljenost služe kao dijagnostički alati za provjeru prepostavki. O tome ćemo detaljno govoriti u kasnijim tjednima.

6.8 Standardni rezultati (z-scores)

Ponekad trebamo usporediti vrijednosti na potpuno različitim skalamama. 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 Korelacija

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: **kovarijancu**. 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

$$\text{Cov}(X, Y) = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})$$

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: kovarijancu podijelimo s produkтом standardnih devijacija obiju varijabli.

$$r_{XY} = \frac{\text{Cov}(X, Y)}{s_X \cdot s_Y} = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{X_i - \bar{X}}{s_X} \right) \left(\frac{Y_i - \bar{Y}}{s_Y} \right)$$

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 korelacijske matrice kreće se od minus 1 do plus 1. Korelacija blizu plus 1 znači snažnu pozitivnu linearu vezu (kad jedna varijabla raste, raste i druga). Korelacija blizu minus 1 znači snažnu negativnu linearu 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 mjerljim 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 orientiri 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 uвijek 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 korelacijski 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 korelaciјe

Pearsonov koeficijent mjeri linearu 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 korelaciјe** (ρ ili r_s). Spearmanova korelacija funkcioniра 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: 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 korelaciјe

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: 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 touchscreensu. 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 frustracija.

💡 Praktični savjet

Ako neka varijabla ima velik postotak nedostajućih vrijednosti (recimo više od 20%), razmislite li uopće koristiti tu varijablu u analizi. Također, umjesto `na.rm = TRUE`, ponekad je bolje koristiti `tidy::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 korelacijske 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 (mladi 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: sredina koristi sve podatke ali je osjetljiva na outliere, medijan je robustan ali ignorira većinu podataka, skraćena sredina je 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. Medejske 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 mernim jedinicama.
 6. Kombinacija `group_by()` i `summarise()` je temeljni alat za izračunavanje deskriptivnih statistika po grupama u R-u.
 7. Pearsonov koeficijent korelacije mjeri linearu 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: `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 predavaanje, 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 5: 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_*`).
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 jednakno.

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 grafičke) 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, FALSE, ~
$ has_video       <lgl> FALSE, FALSE, TRUE, 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, 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, TRUE, FALSE, 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.

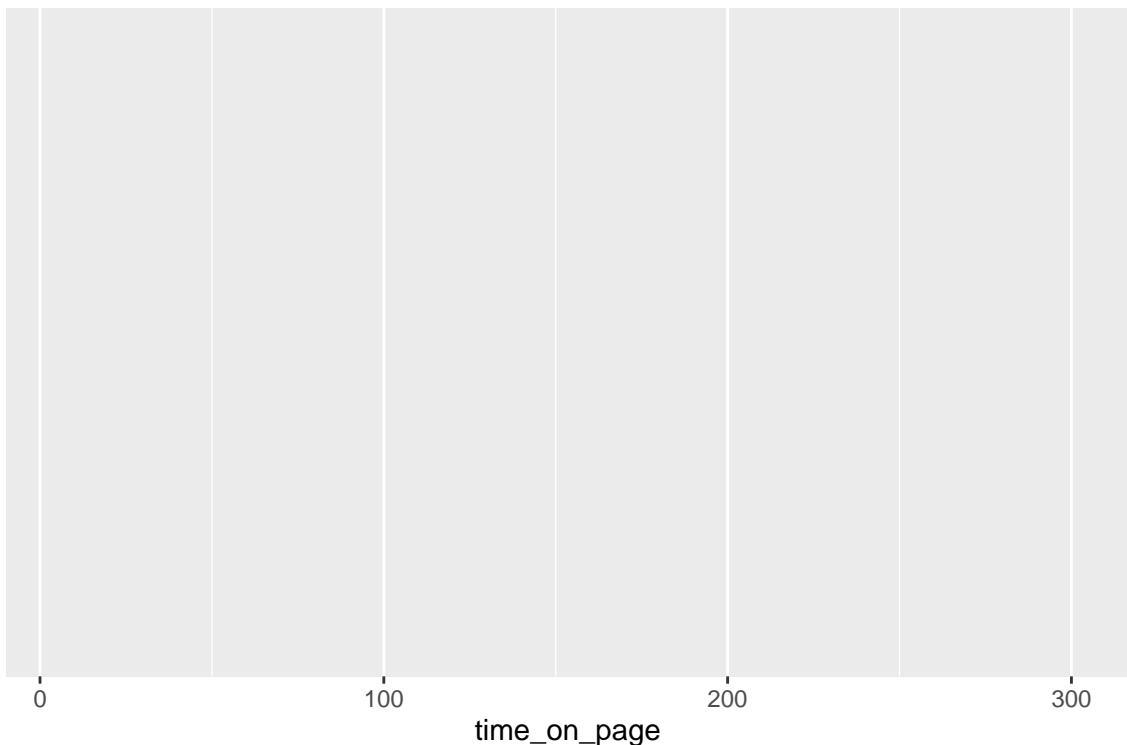
Estetike (aesthetics, `aes()`): mapiranja varijabli na vizualne dimenzije grafa. Na primjer, varijabla `time_on_page` ide na x os, `shares` na y os, `category` određuje boju.

Geometrija (geometry, `geom_*`): oblik kojim se podaci prikazuju. Točke za scatterplot (`geom_point()`), stupci za bar chart (`geom_bar()`), linije za linijski graf (`geom_line()`).

Osim ova tri, graf može imati i dodatne slojeve: statističke transformacije (stat), prilagodbe skala (scale), podjelu u panele (facet), koordinatni sustav (coord) i vizualnu temu (theme). Svaki od ovih elemenata se dodaje operatorom `+`.

Pogledajmo najjednostavniji mogući ggplot2 kod.

```
ggplot(data = clanci, mapping = aes(x = time_on_page))
```

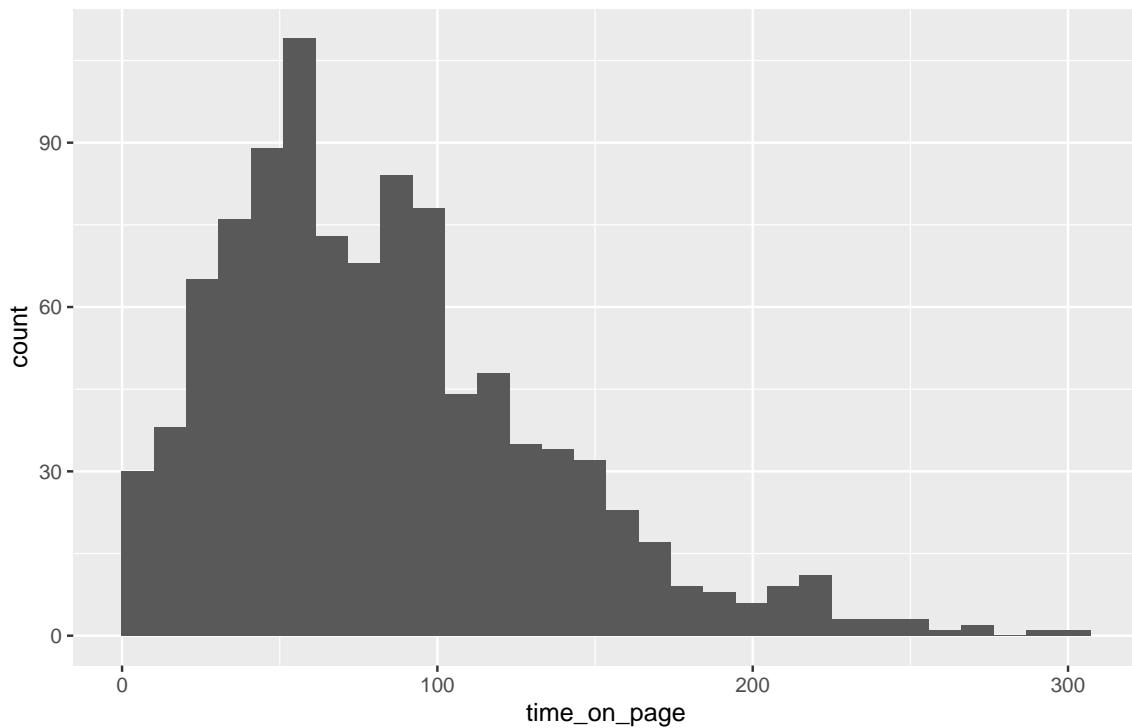


Ovaj kod kreira prazan graf. Definirali smo podatke (`clanci`) i jednu estetiku (varijabla `time_on_page` na x osi), ali nismo rekli ggplotu KAKO da prikaže te podatke (nismo dodali geometriju). Rezultat je prazan koordinatni sustav s ispravno postavljenom x osi. Ggplot zna raspon varijable i pripremio je platno, ali čeka da mu kažemo što da nacrta.

7.4 Histogrami: distribucija jedne varijable

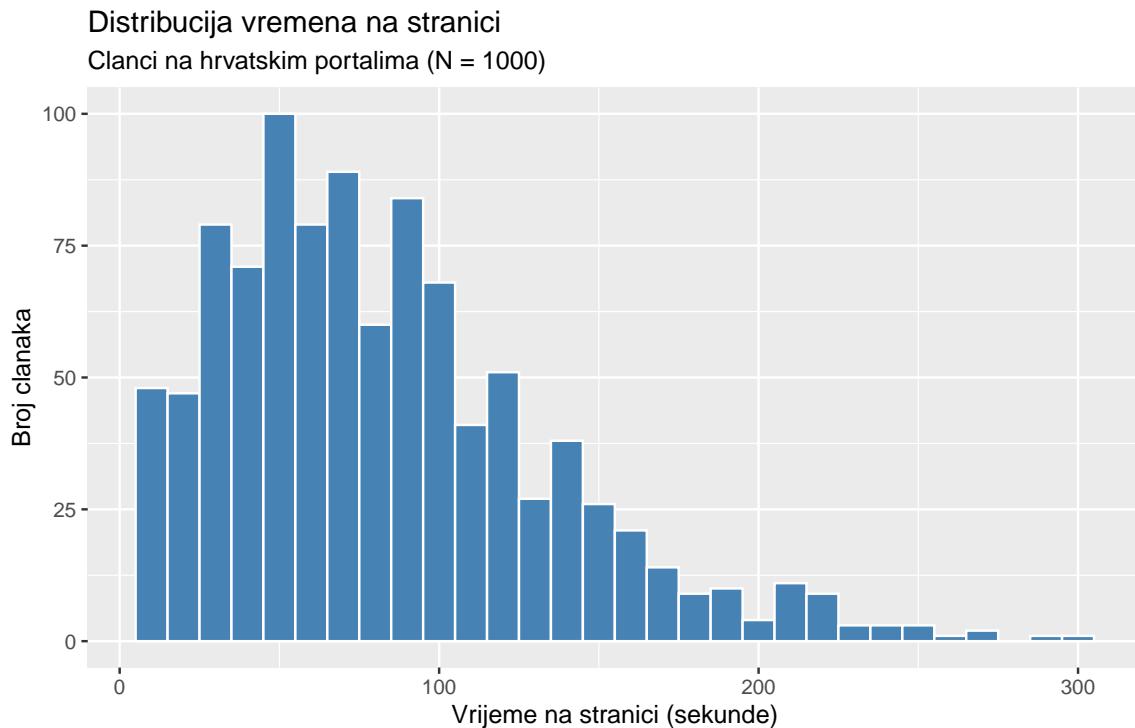
Histogram je najvažniji graf za razumijevanje distribucije jedne kontinuirane varijable. Dijeli raspon vrijednosti u jednake intervale (binove) i prikazuje koliko opažanja pada u svaki interval.

```
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"
  )
```



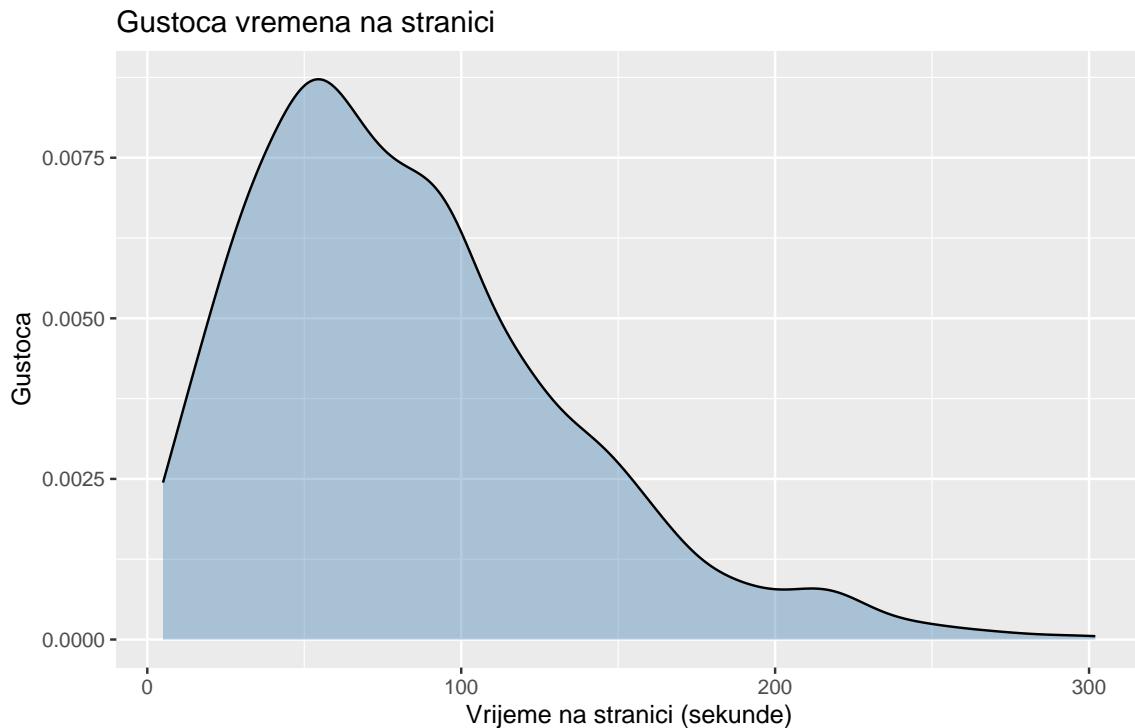
Ovdje smo napravili nekoliko poboljšanja. Argument `binwidth = 10` postavlja širinu svakog bina na 10 sekundi, što daje jasniju sliku od defaultnih 30 binova. `fill = "steelblue"` boja ispunu stupaca, a `color = "white"` crta bijeli rub između stupaca za bolju čitljivost. Funkcija `labs()` dodaje naslov, 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"
  )
```



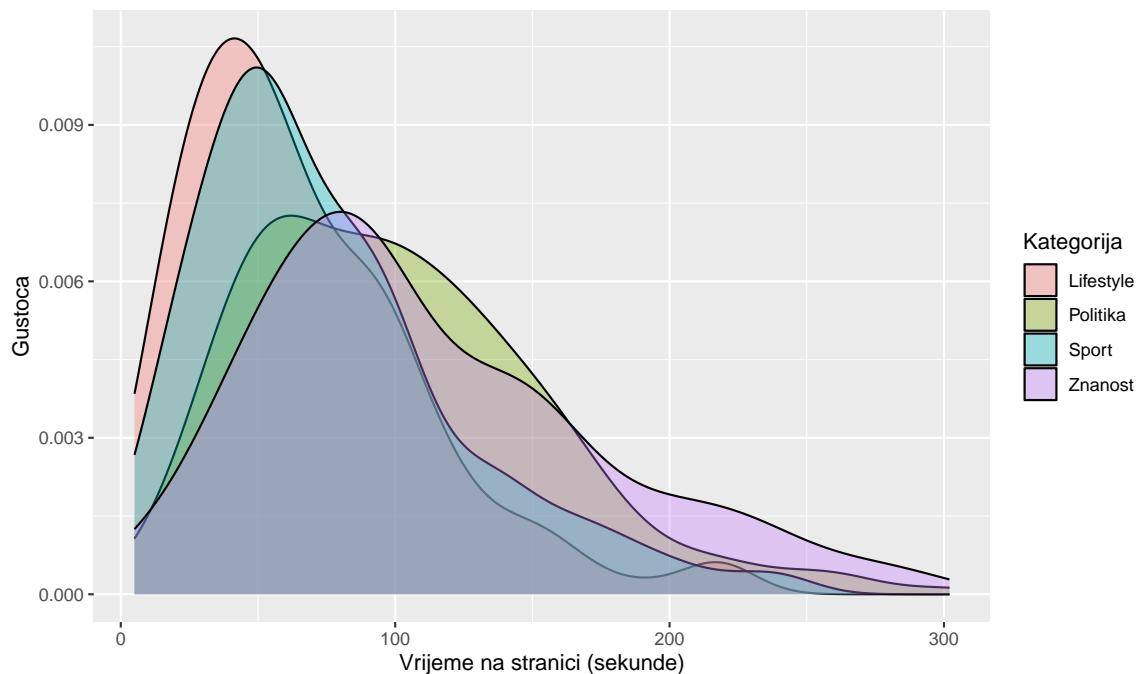
Argument `alpha = 0.4` kontrolira transparentnost ispune (0 je potpuno prozirno, 1 potpuno neprozirno). Transparentnost je osobito korisna kad preklapate više distribucija.

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 preklopljениh 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"
  )
```

Distribucija vremena čitanja po kategoriji



Primijetite novu estetiku: `fill = category` unutar `aes()`. Ovo 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 prekloppljenih 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

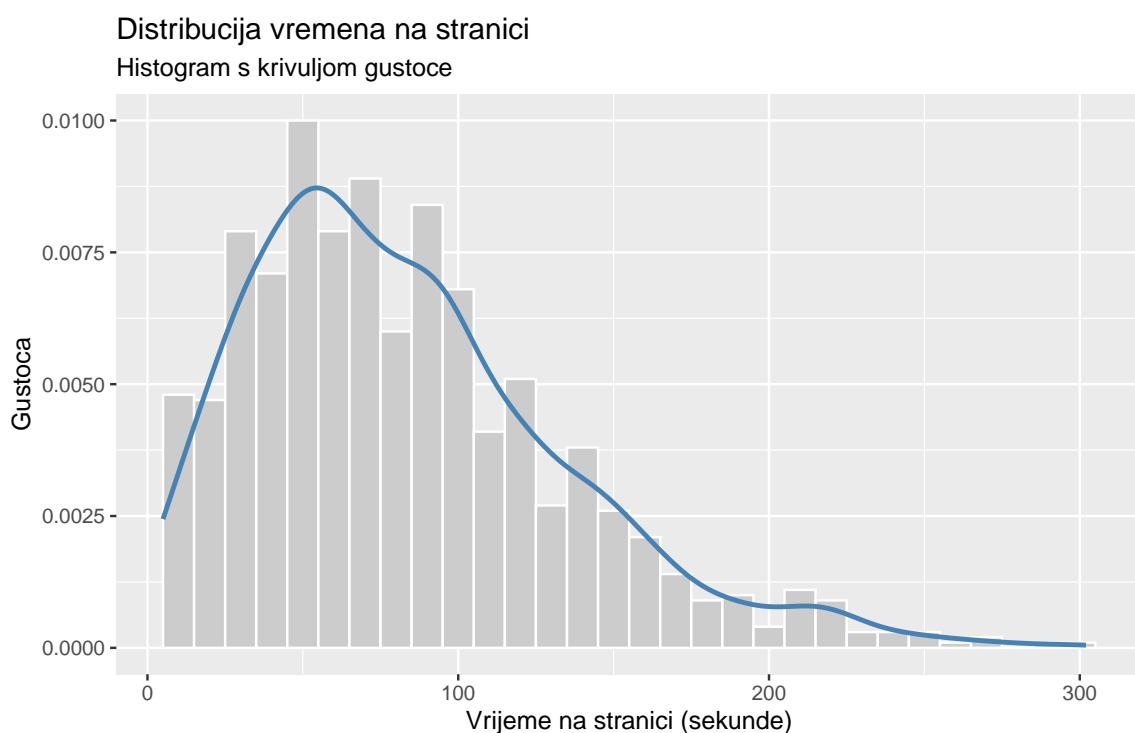
Ponekad je korisno nacrtati oboje na istom grafu. Za to moramo histogramu reći da na y osi prikaže gustoću umjesto broja opažanja, kako bi skale bile usporedive.

```
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"
)

```



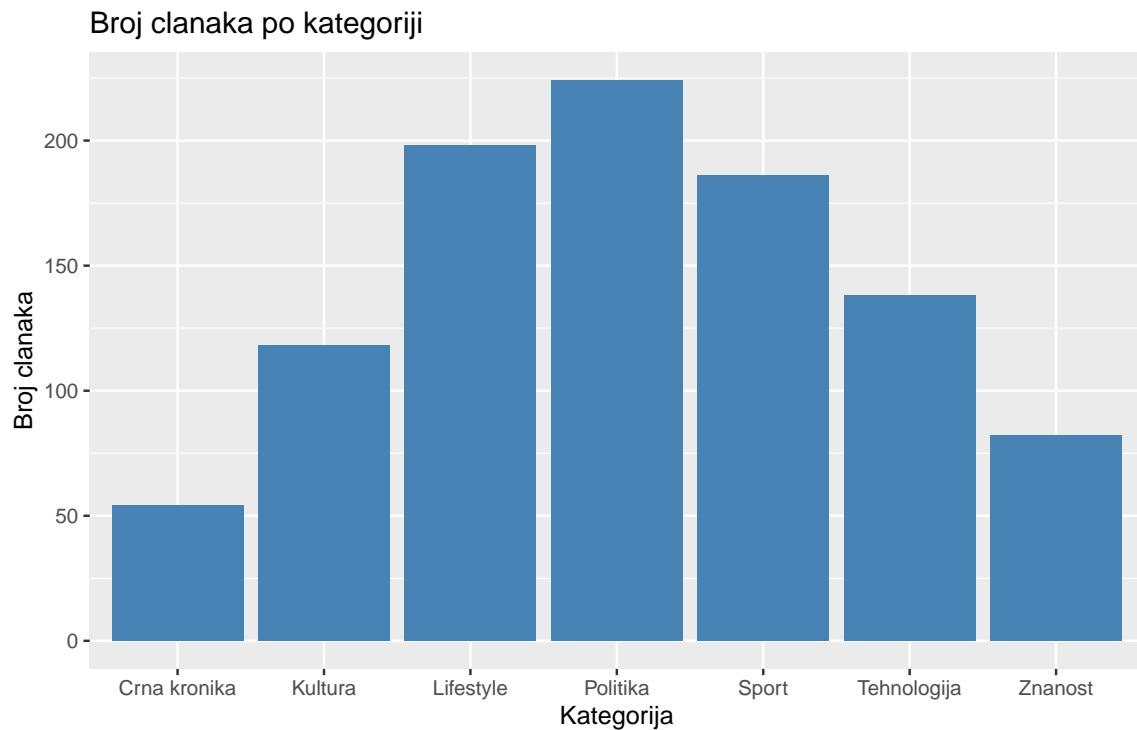
Ključan detalj je `aes(y = after_stat(density))` unutar `geom_histogram()`. Ovo govori ggplotu da na y osi prikaže gustoću (proporciju) umjesto apsolutnog broja, čime histogram i krivulja gustoće postaju usporedivi.

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

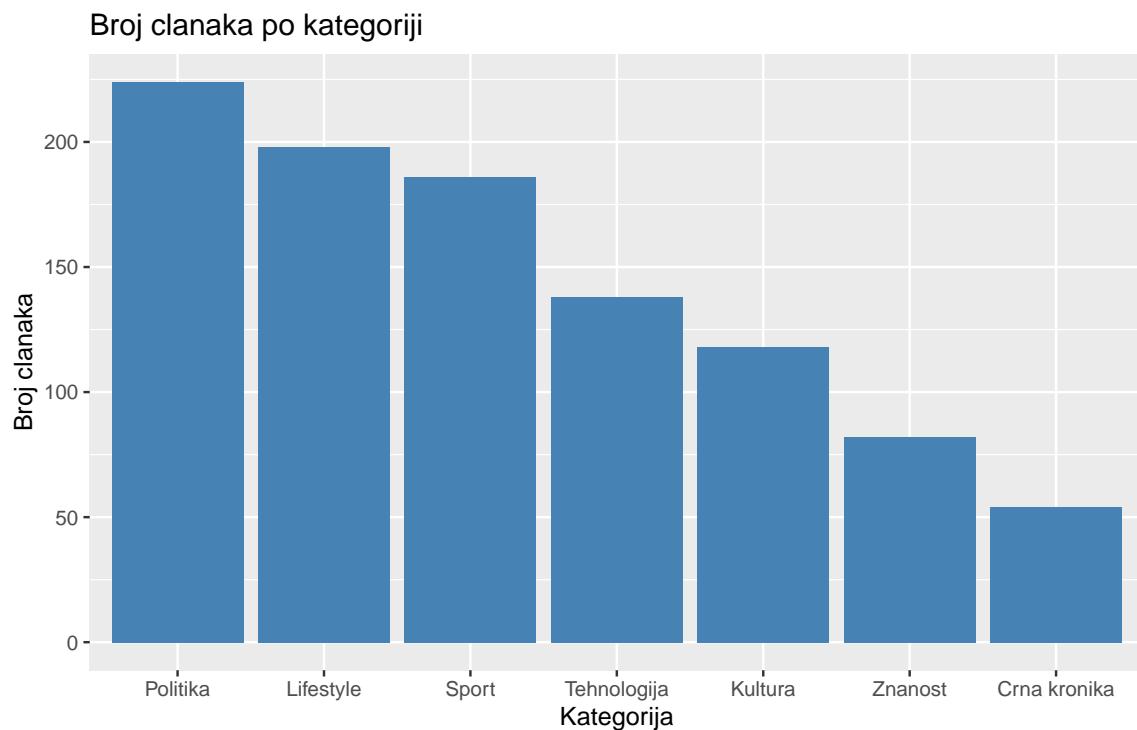
Abecedni redoslijed kategorija rijetko je informativan. Bolje je sortirati stupce po veličini pomoću `fct_infreq()` iz paketaforcats (dio tidyverse).

```
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))`.

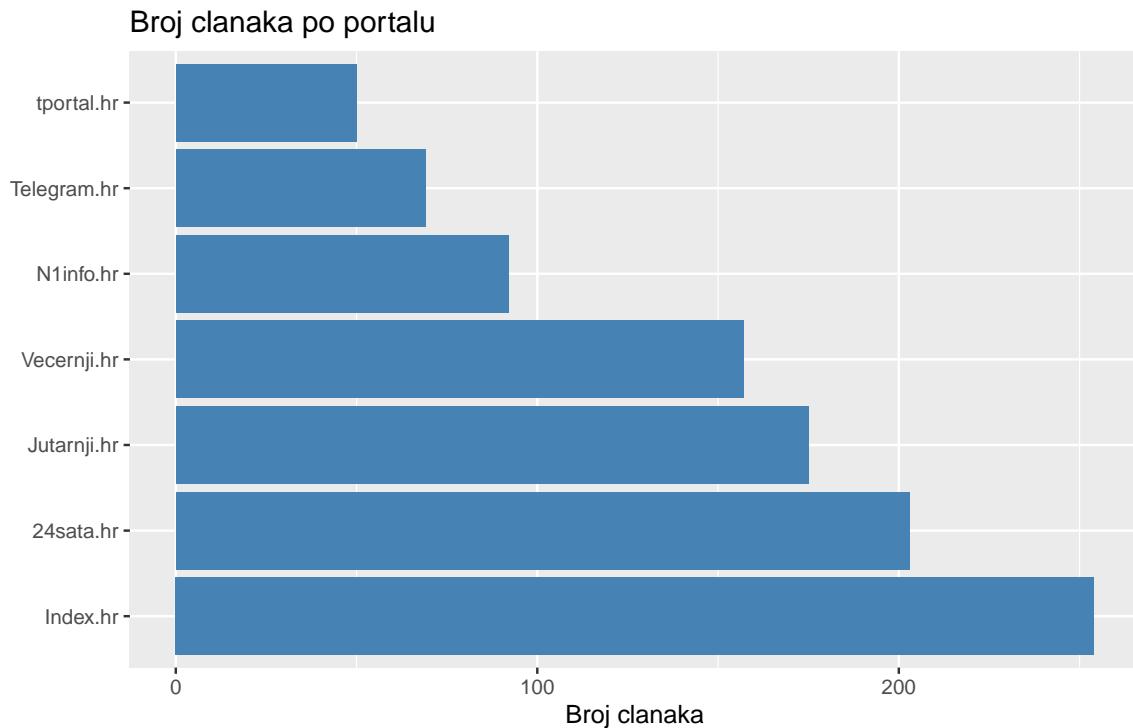
7.5.3 Horizontalni stupčasti graf

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
)

```



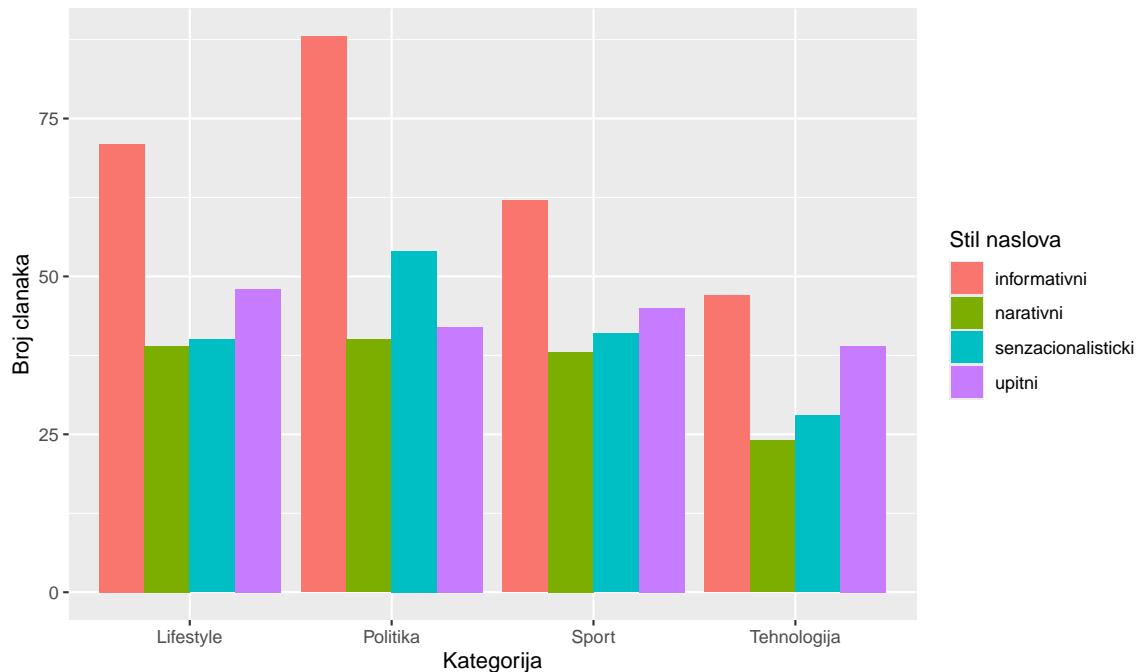
Trik je jednostavan: umjesto x koristite y u `aes()`, i ggplot automatski crta horizontalne stupce. Postavili smo y = NULL u `labs()` da uklonimo nepotrebnu oznaku osi jer su imena portala samorazumljiva.

7.5.4 Grupirani i složeni stupčasti grafovi

Kad želimo prikazati odnos između dviju kategoričkih varijabli, koristimo boju za drugu varijablu.

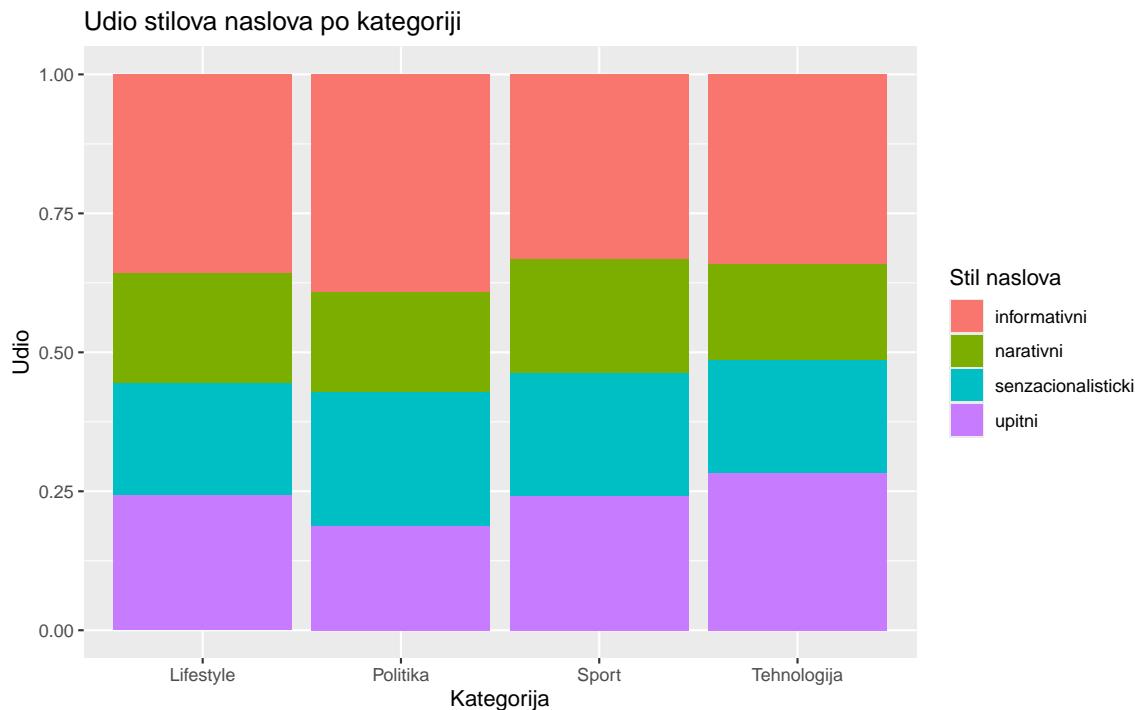
```
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija", "Lifestyle")) |>
  ggplot(aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge") +
  labs(
    title = "Stil naslova po kategoriji članaka",
    x = "Kategorija",
    y = "Broj članaka",
    fill = "Stil naslova"
  )
```

Stil naslova po kategoriji clanaka



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"
  )
```



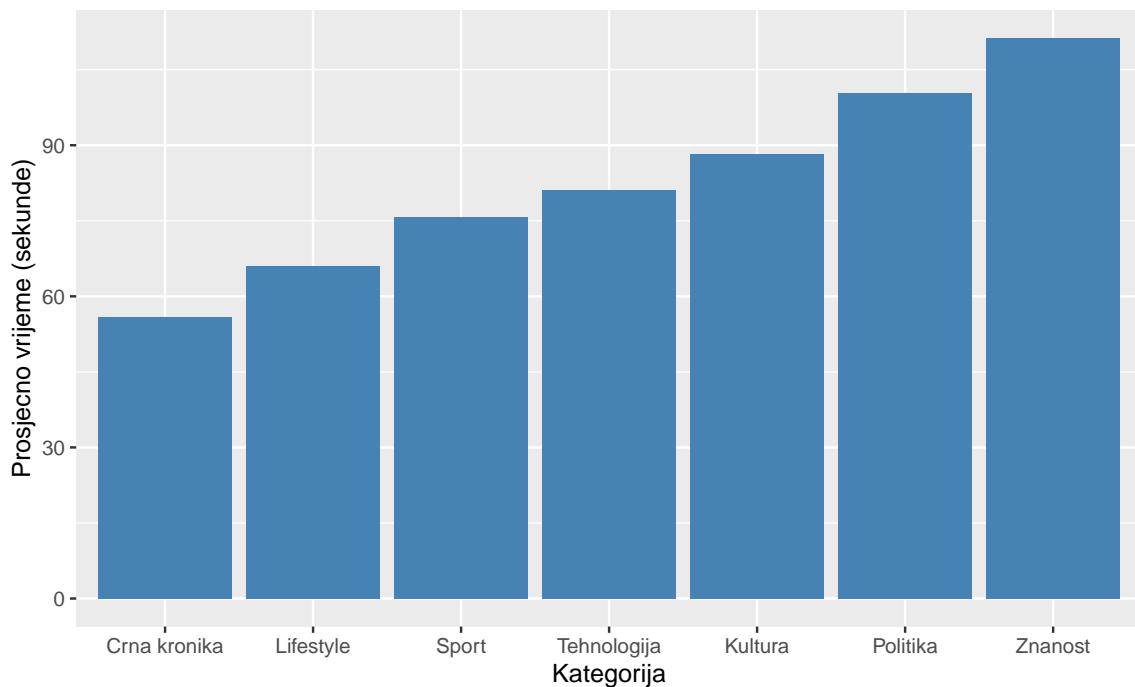
Ovaj graf otkriva zanimljive obrasce. Proporcija senzacionalističkih naslova se razlikuje po kategorijama. Ovo je vizualna verzija tablice unakrsnih frekvencija (contingency table) i koristit ćete ju kad budemo radili hi-kvadrat testove u tjednu 11.

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)"
  )
```

Prosječno vrijeme citanja po kategoriji



Ovdje smo najprije izračunali prosjek, zatim koristili `fct_reorder()` da sortiramo kategorije po prosječnom vremenu (ne po frekvenciji kao `fct_infreq()`), i onda prikazali te prosjekе s `geom_col()`.

Razlika između `geom_bar()` i `geom_col()` je ključna. `geom_bar()` sam broji retke i ne treba je estetiku. `geom_col()` prikazuje y vrijednosti koje ste vi pripremili. Koristite `geom_bar()` za frekvencije, `geom_col()` za sve ostalo (prosjekе, medijane, postotke, bilo kakve prethodno izračunate sažetke).

! Važna napomena

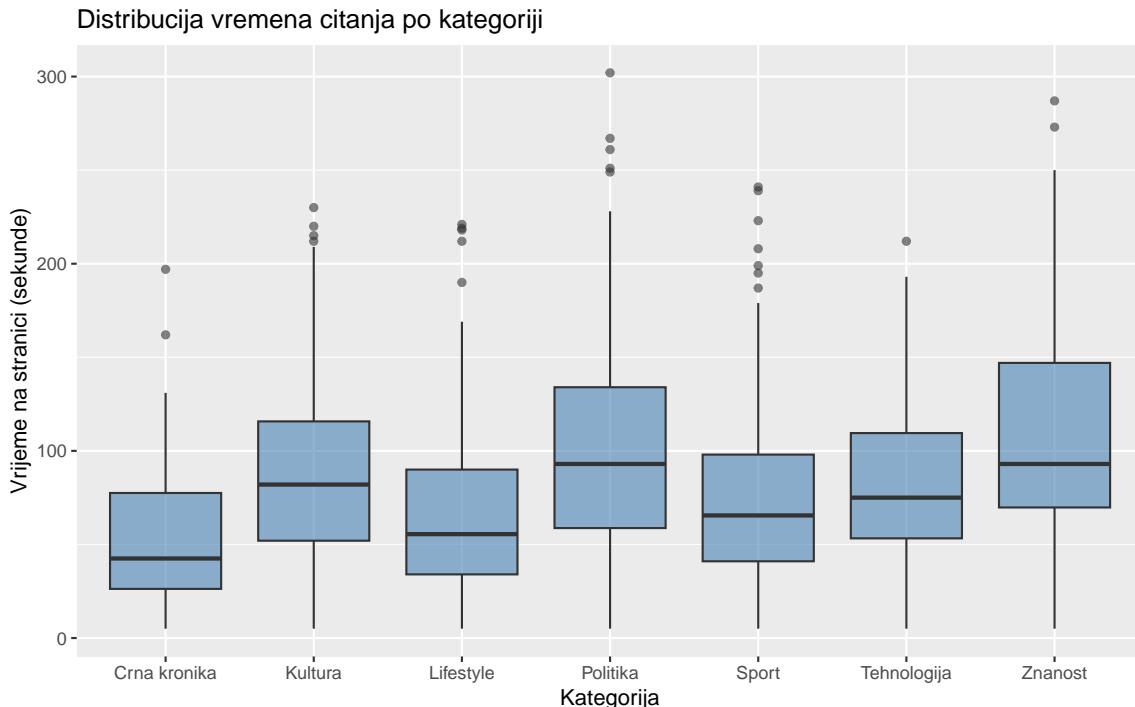
Stupčasti graf prosjeka skriva distribuciju podataka. Kad vidite samo stupac visine 83, ne znate je li to zato što su svi oko 83 ili zato što su pola ljudi na 10 i pola na 156. Za usporedbu distribucija između grupa, boxplot ili violin plot su gotovo uvijek bolji izbor. Stupčaste grafove prosjeka koristite 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, plus 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 ovako. Deblja crta unutar pravokutnika je medijan. Donji rub pravokutnika je Q1 (25. percentil), 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 $1.5 \times IQR$ od ruba pravokutnika. Točke izvan toga su 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.

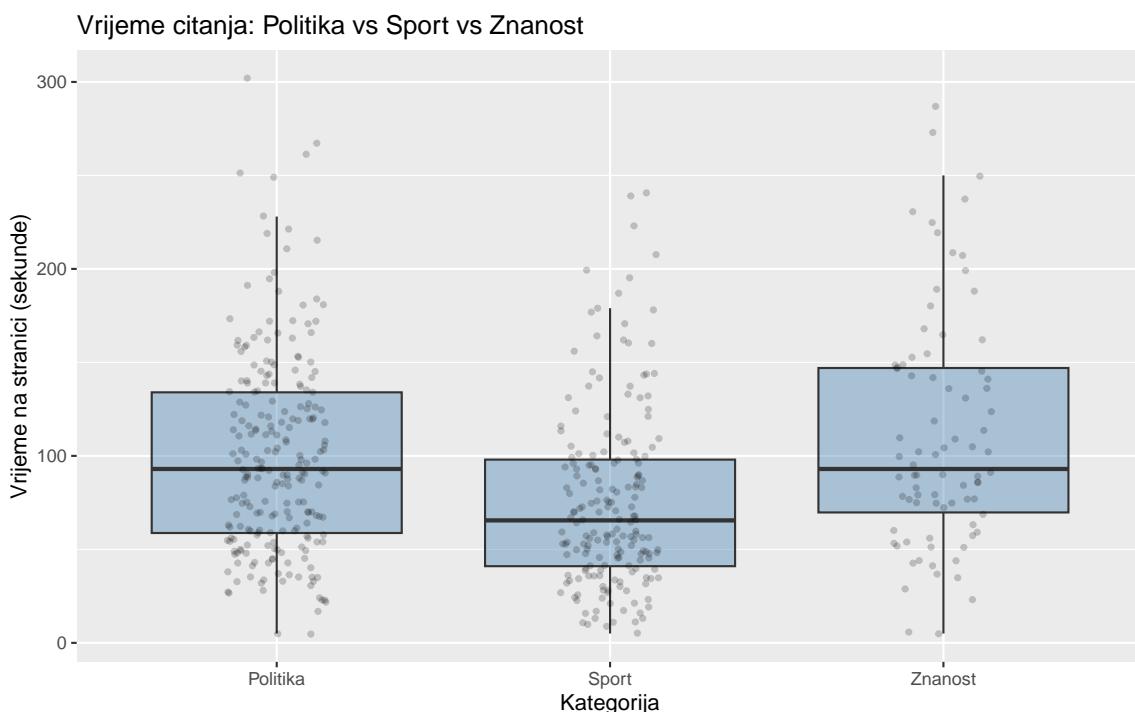
7.6.1 Boxplot s točkama

Boxplot sažima distribuciju u pet brojeva, pa neke informacije gubi. Dodavanje pojedinačnih točaka vraća taj kontekst.

```

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. `alpha = 0.2` čini točke poluprozirnim pa vidimo gustoću. `outlier.shape = NA` u boxplotu isključuje prikaz outliera jer bi se inače udvostručili s jitter točkama.

Ovaj kombinirani prikaz daje kompletну sliku: boxplot za sažetak distribucije i točke za uvid u stvarne podatke.

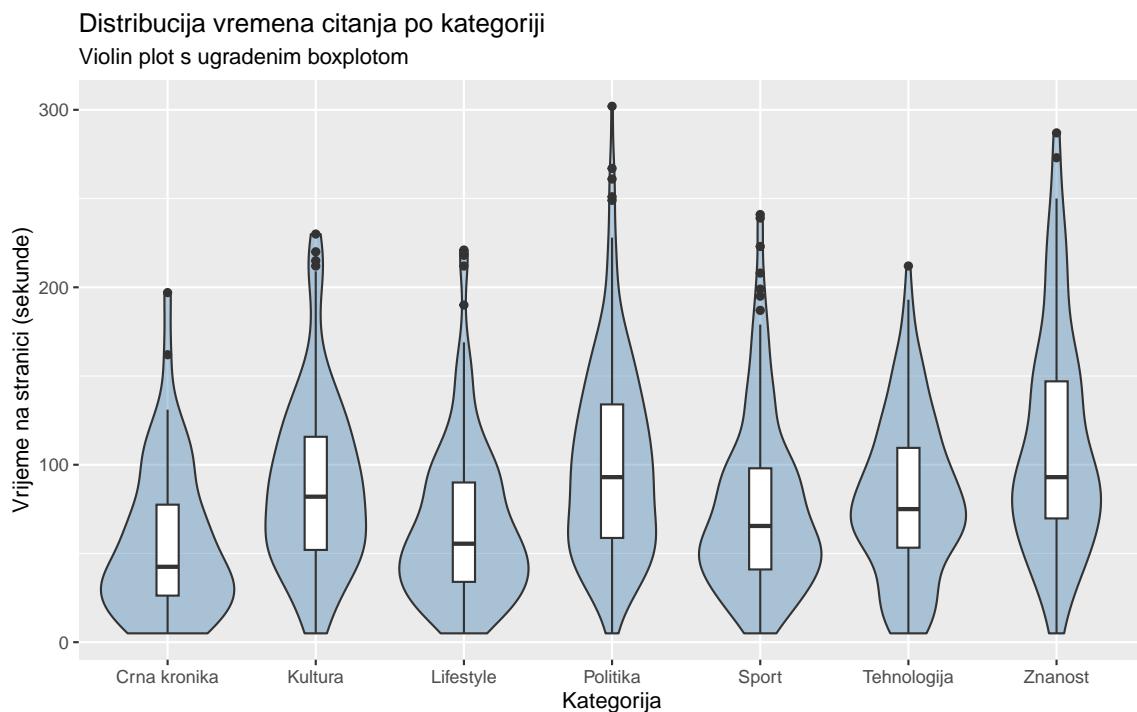
7.6.2 Violin plot: oblik distribucije

Violin plot je varijanta boxplota koja prikazuje oblik distribucije pomoću zrcaljene krivulje gustoće. Tamo gdje je graf širi, ima više podataka.

```

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)"
  )

```

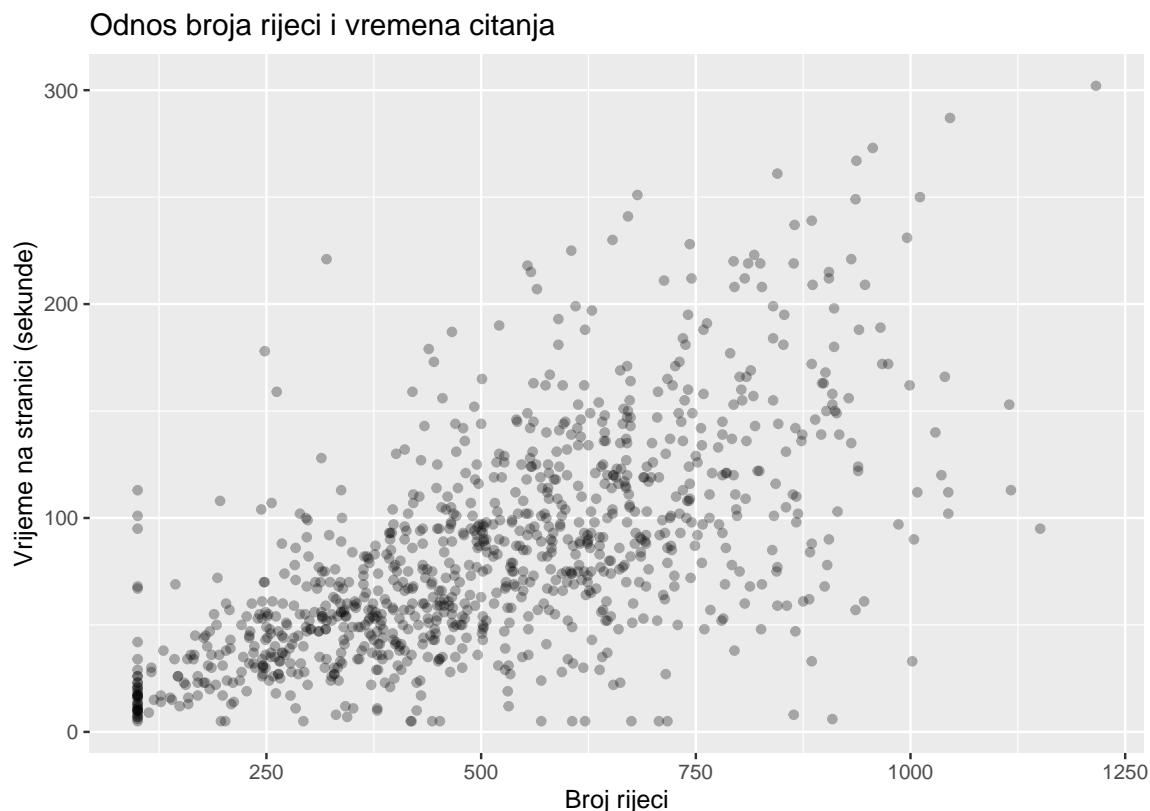


Kombinacija violin plota (za oblik distribucije) i uskog boxplota (za medijan i kvartile) daje bogat prikaz koji je ujedno i informativan i vizualno privlačan.

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.

```
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

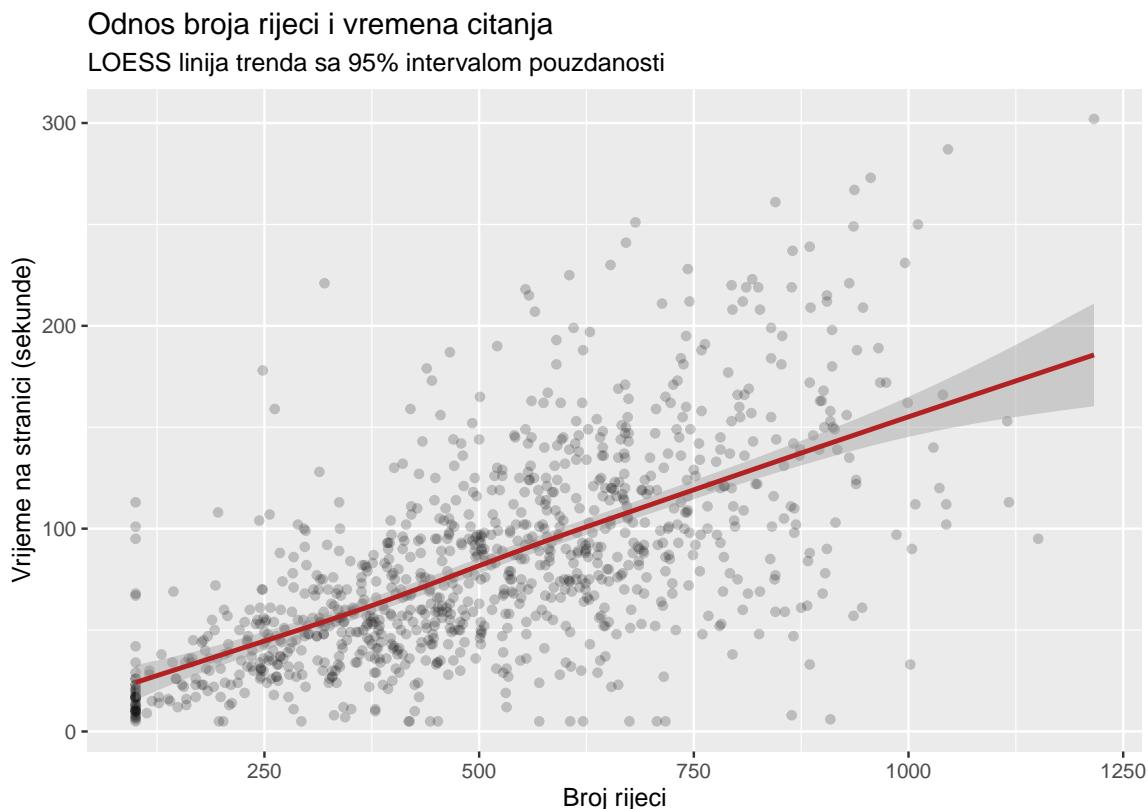
Funkcija `geom_smooth()` dodaje statističku liniju trenda na scatterplot. Po defaultu koristi LOESS (lokalno ponderiranu regresiju) koja je fleksibilna i prati oblik podataka.

```
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)"
)

```



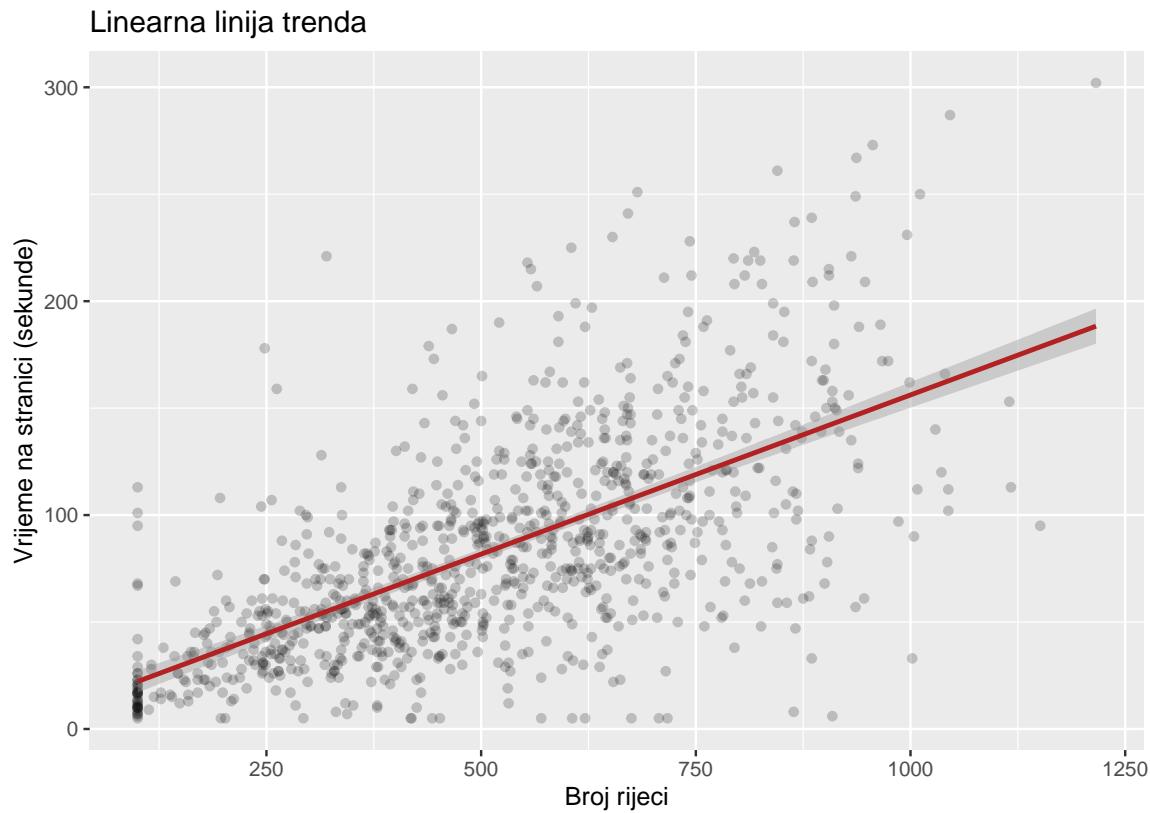
Sivi pojas oko linije je 95% interval pouzdanosti za procjenu trenda. Što je pojas uži, to smo sigurniji u procjeni. Na krajevima distribucije (malo i mnogo riječi) pojas je širi jer imamo manje podataka.

Za linearnu liniju trenda (onu koju smo računali kad smo radili korelaciju 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)"
)

```

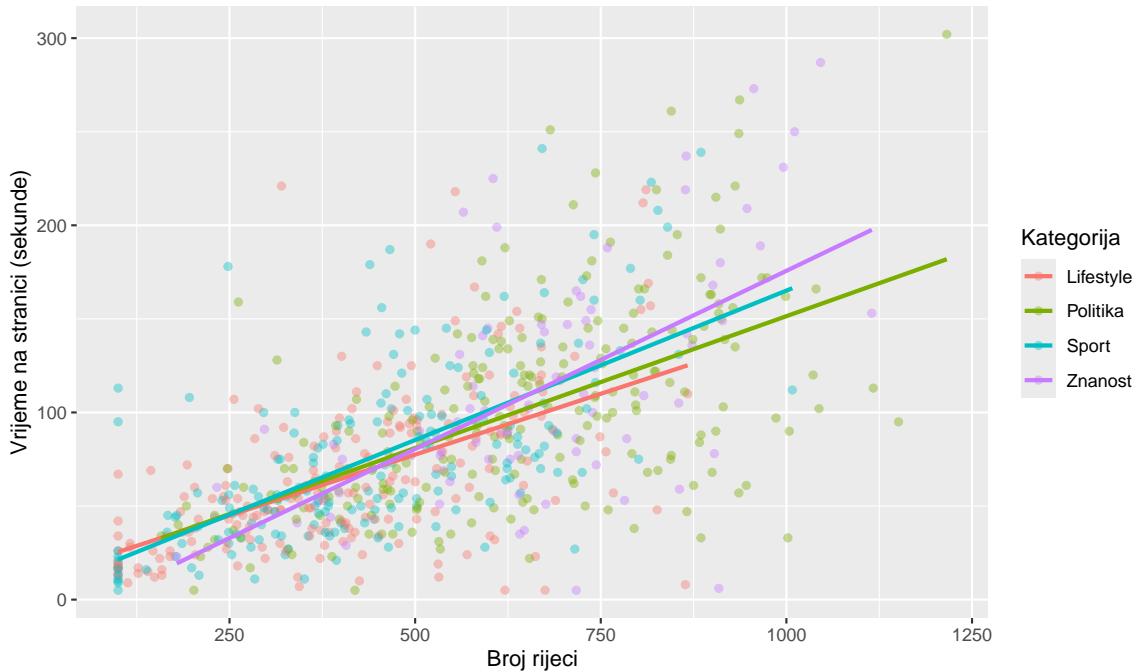


7.7.2 Kodiranje treće varijable bojom

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"
  )
```

Broj riječi i vrijeme citanja po kategoriji



Argument `se = FALSE` uklanja interval pouzdanosti da graf ne bude pretrpan. Sad vidimo da je pozitivan odnos između broja riječi i vremena čitanja prisutan u svim kategorijama, ali se kategorije razlikuju po razini (intercept): za isti broj riječi, znanstveni članci imaju duže prosječno čitanje od lifestyle članaka.

7.7.3 Kodiranje veličine i oblika

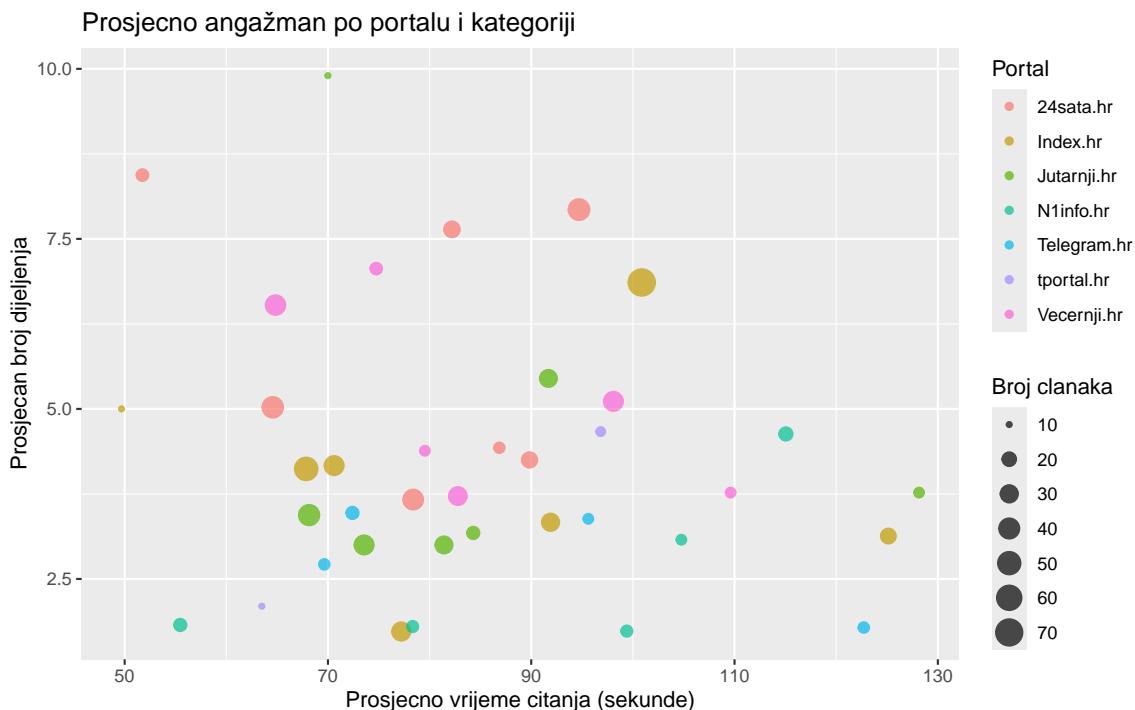
Osim boje, ggplot2 nudi i druge estetike za kodiranje varijabli.

```
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 zbnjenica 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.

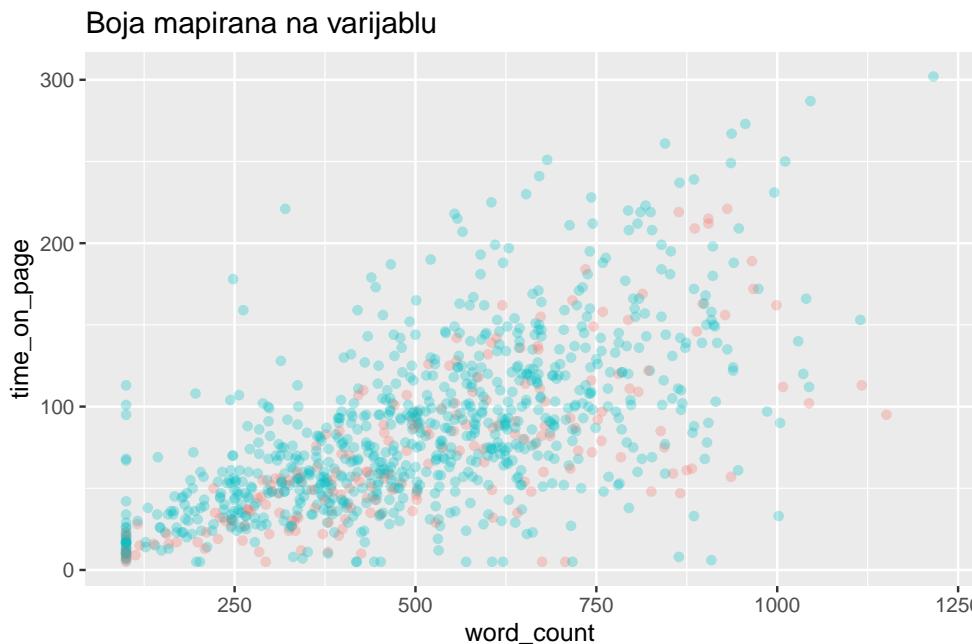
```

# 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")

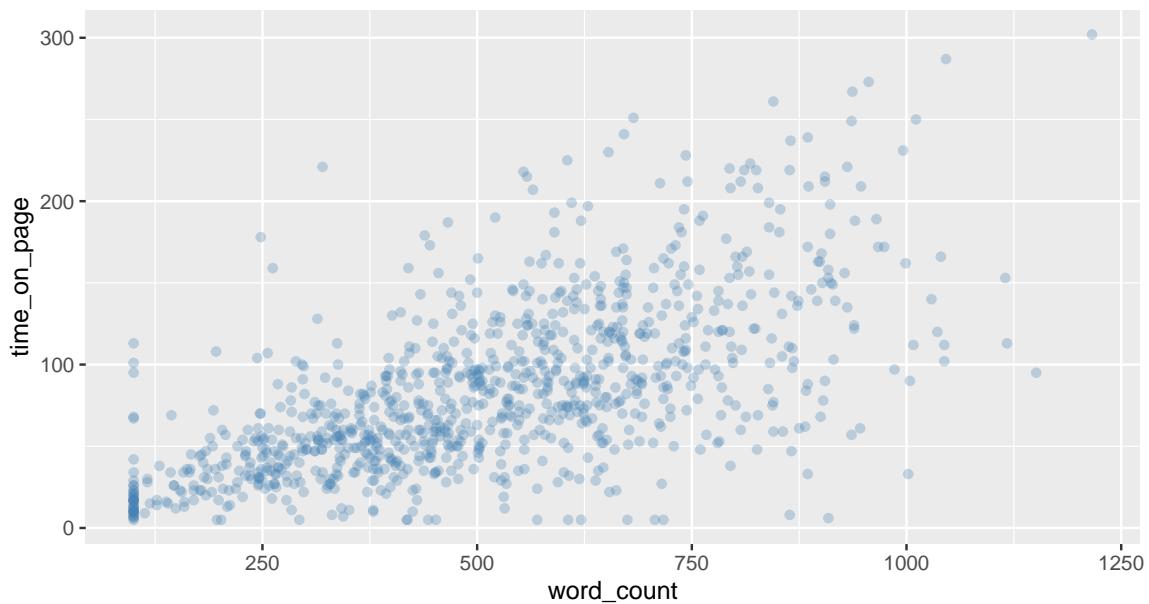
p1

```



p2

Fiksna boja za sve tocke

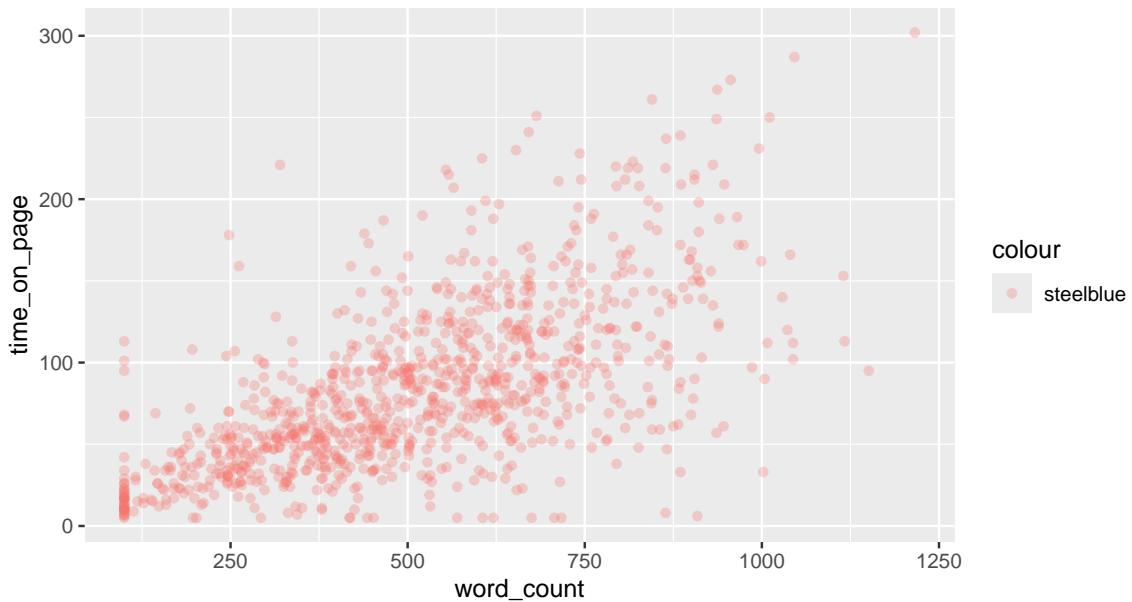


Ova razlika se proteže na sve estetike: `fill`, `color`, `size`, `shape`, `alpha`, `linewidth`. Kad ovisi o varijabli, ide unutar `aes()`. Kad je fiksno, ide izvan.

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()")
```

Cesta greška: ime boje unutar aes()



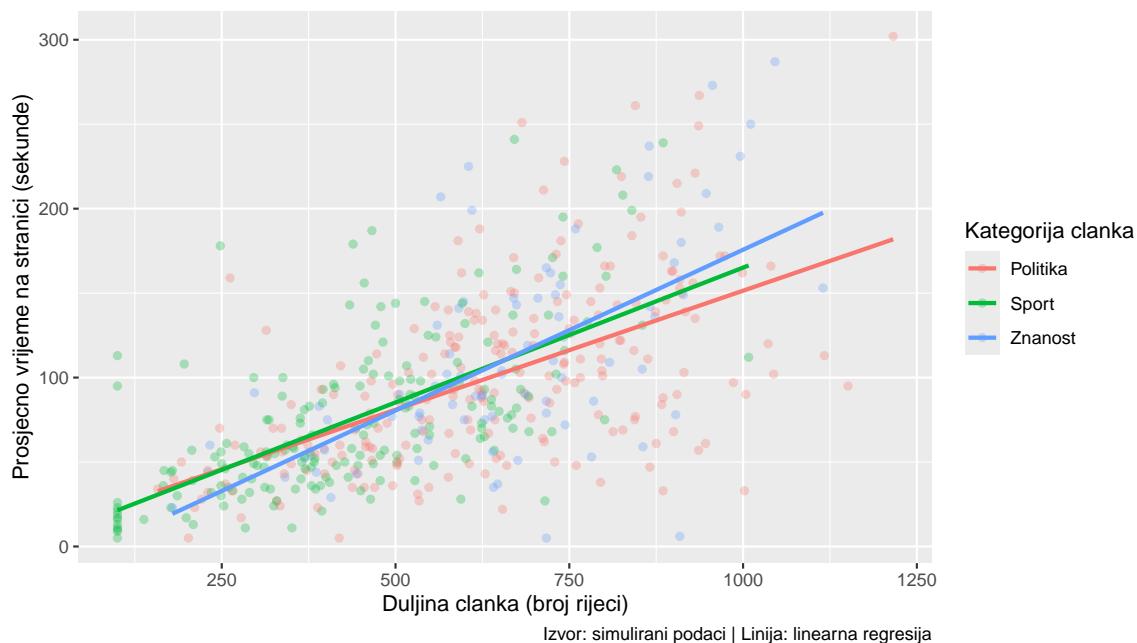
Ggplot interpretira "steelblue" kao tekstualnu varijablu s jednom kategorijom i dodjeljuje joj svoju paletu (obično crvenu). Rezultat je legenda s jednom stavkom "steelblue" koja je obojana u boju koju ggplot sam odabere. Ovo je jedan od najčešćih bugova kod početnaka.

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"
  )
```

Duži clanci zadržavaju citatelje duže
Analiza 1000 clanaka s hrvatskih informativnih portala



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. Primjer: 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. Primjer: 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. Primjer: 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. Primjer: 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'~/fill')
```

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.

i Podsjetnik

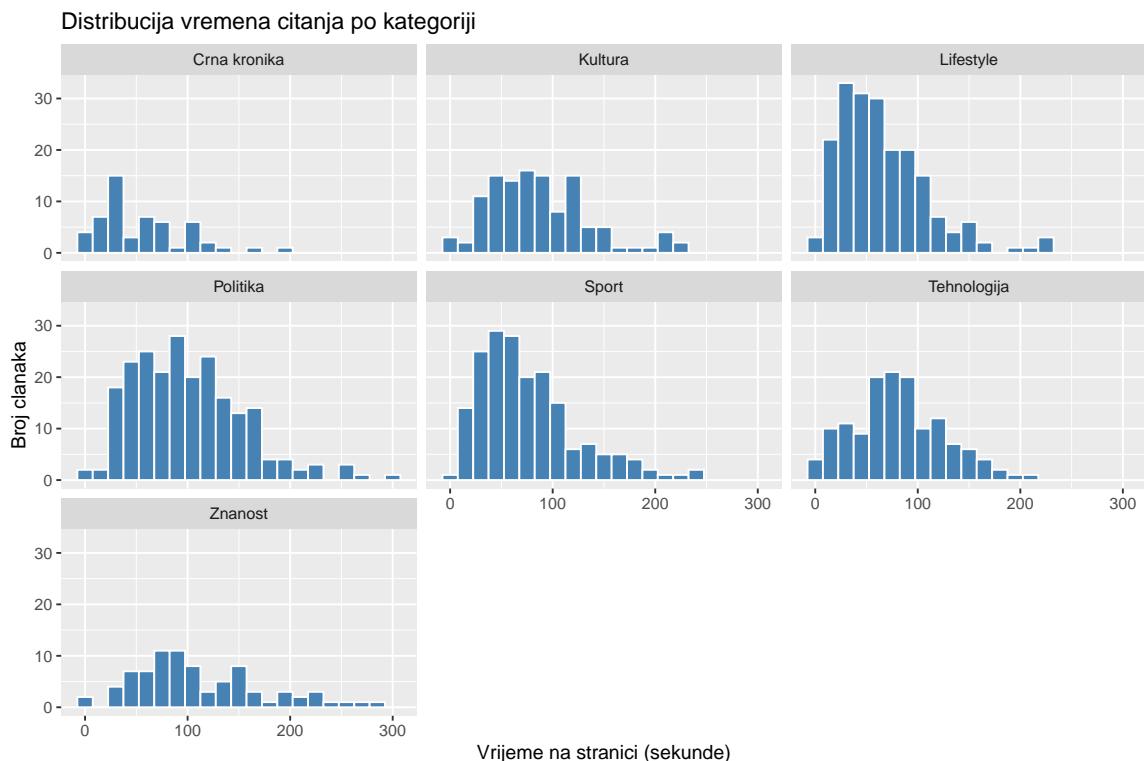
U prvom dijelu naučili smo četiri temeljna tipa grafova: histogram/density za distribucije, bar chart za kategorije, boxplot/violin za usporedbu grupa i scatterplot za odnos dviju varijabli. U ovom dijelu učimo kako grafove učiniti profesionalnima i prezentabilnima.

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 zatpan 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"  
)
```

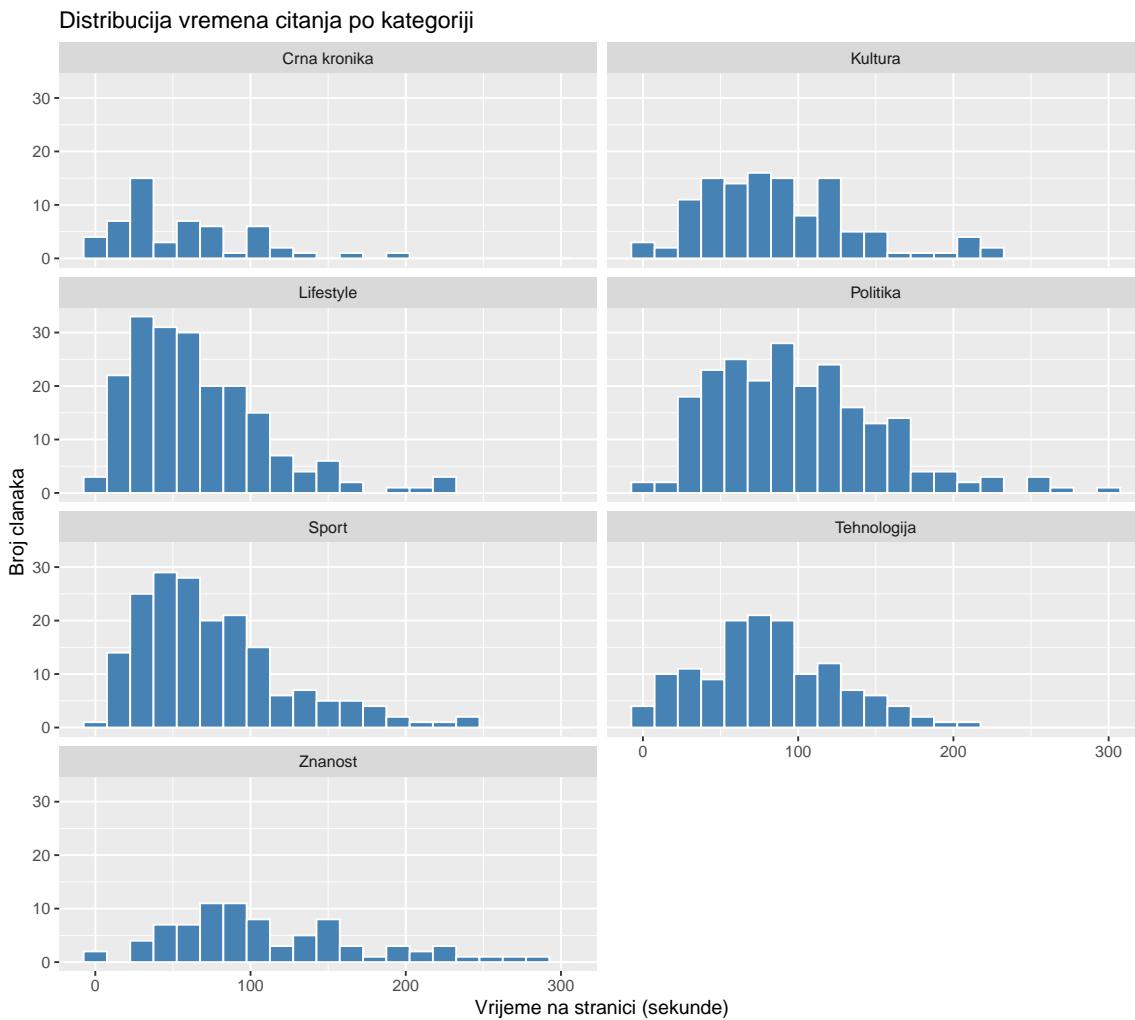


Sintaksa `facet_wrap(~category)` govori ggplotu da napravi zaseban panel za svaku razinu varijable `category`. Tilda (~) je obavezna i čita se kao “po” (podijeli po kategoriji). Paneli se automatski slažu u mrežu.

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

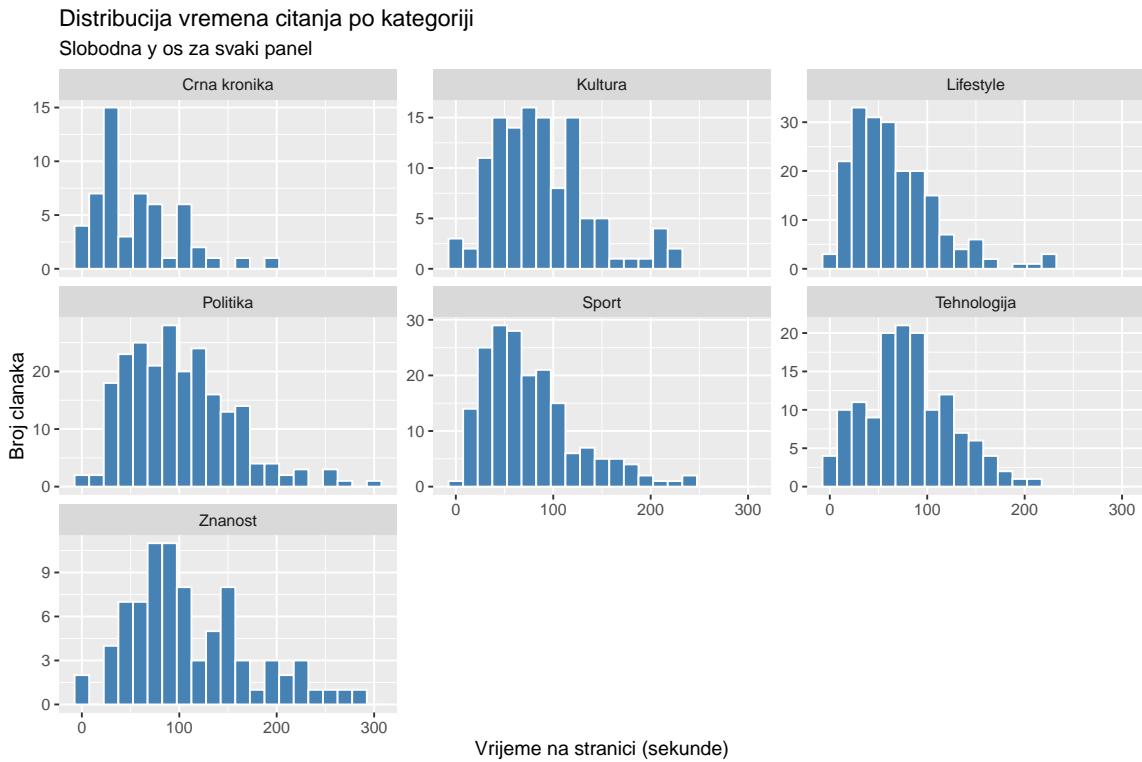
Po defaultu, svi paneli dijele iste osi. Ovo je dobro za usporedbu apsolutnih vrijednosti, ali ponekad želite da svaki panel ima vlastitu skalu (na primjer, kad se grupe drastično razlikuju po broju opažanja).

```
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` su: "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.

```

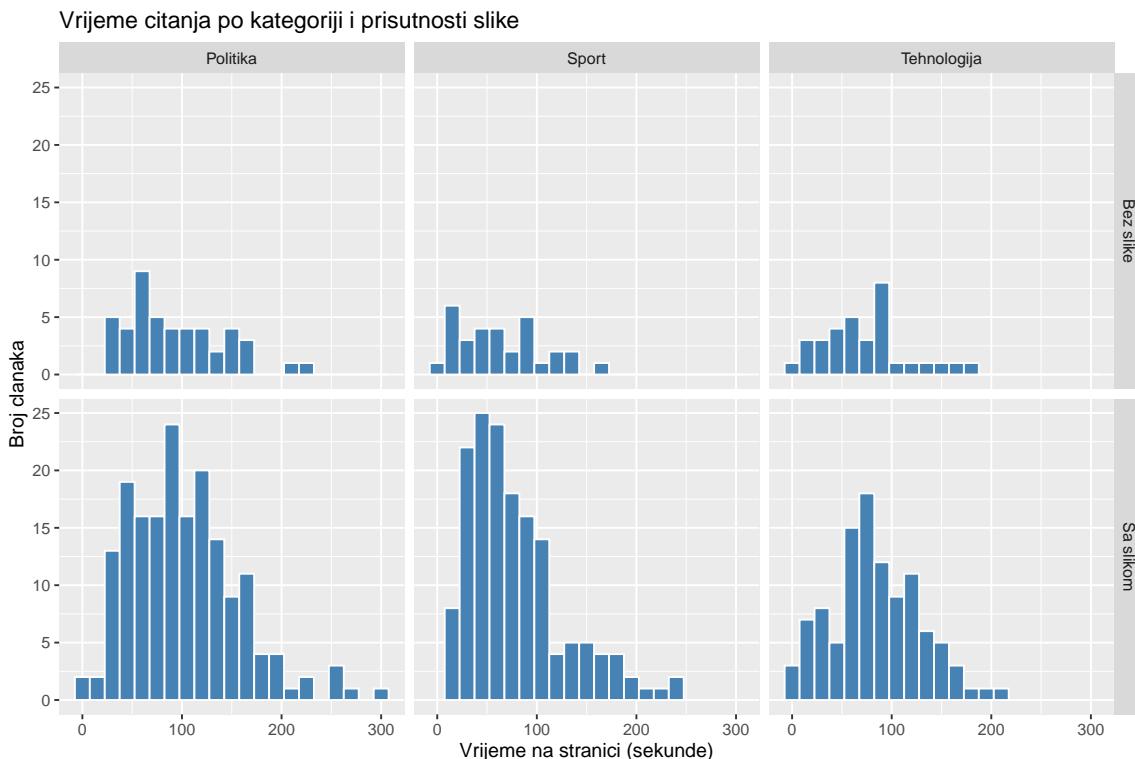
clanci |>
  filter(category %in% c("Politika", "Sport", "Tehnologija")) |>
  mutate(ima_sliku = if_else(has_image, "Sa slikom", "Bez slike")) |>
  ggplot(aes(x = time_on_page)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  facet_grid(ima_sliku ~ category) +

```

```

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 i vertikalno (unutar iste kategorije, sa i bez slike) i horizontalno (između kategorija, za isti format).

7.11.4 Facetiranje scatterplota

Facetiranje radi sa svakim tipom grafa, ne samo s histogramima.

```

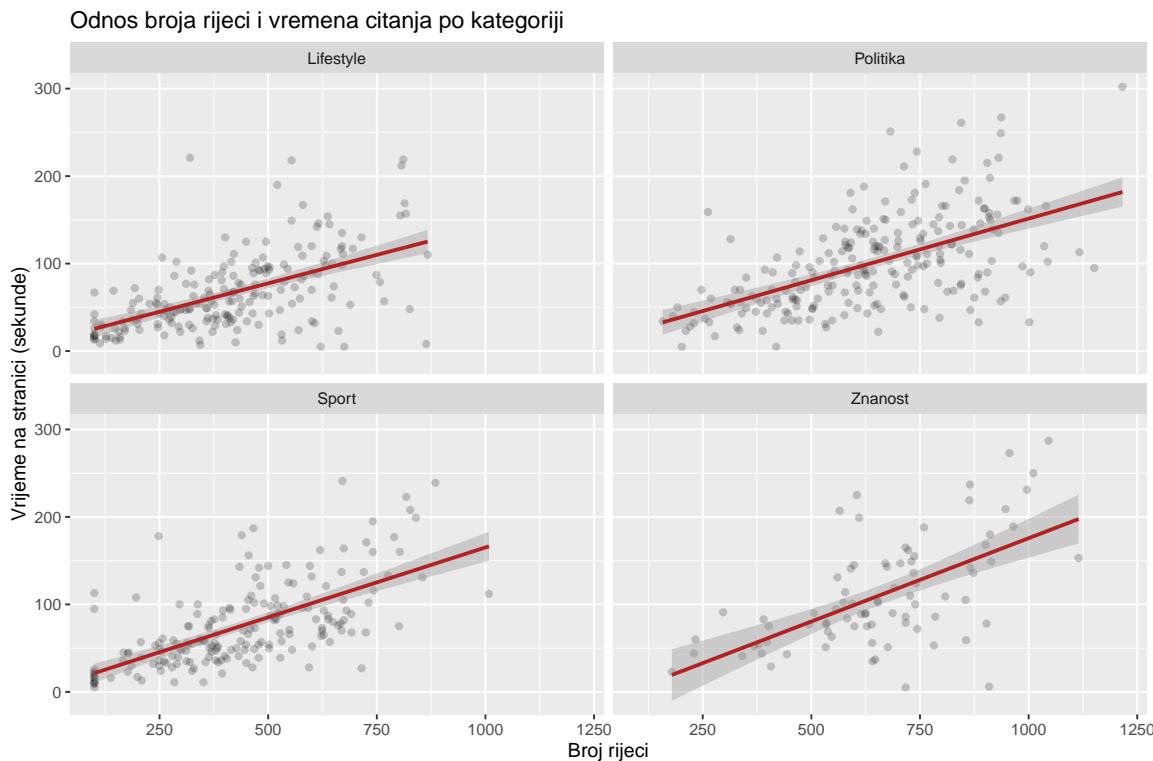
clanci |>
  filter(category %in% c("Politika", "Sport", "Znanost", "Lifestyle")) |>
  ggplot(aes(x = word_count, y = time_on_page)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "firebrick") +
  facet_wrap(~category) +
  labs(
    title = "Odnos broja riječi i vremena čitanja po 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.

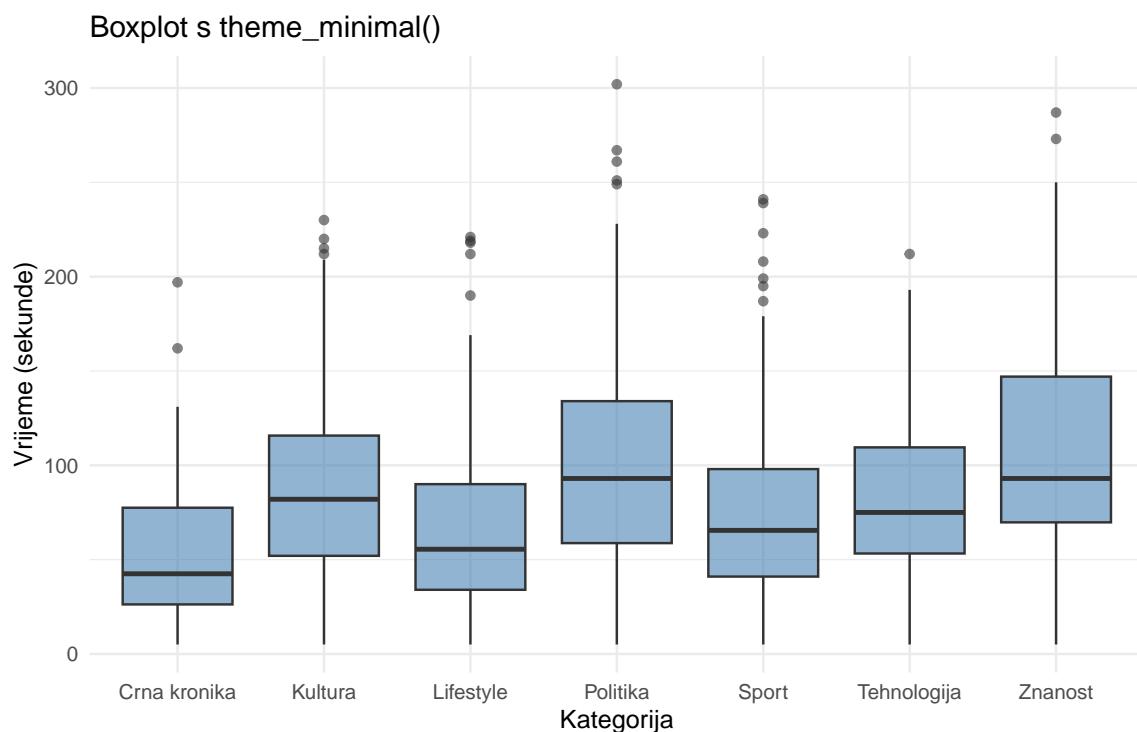
7.12 Teme: vizualni izgled grafa

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.

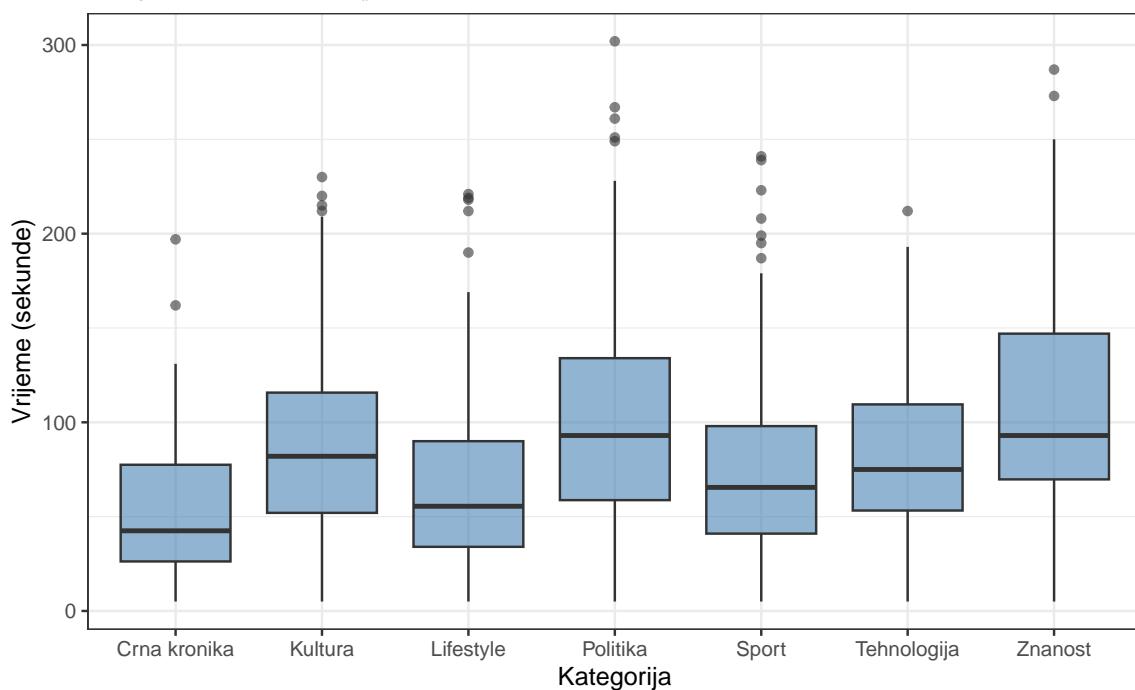
7.12.1 Ugrađene teme

```
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)"  
)
```

Boxplot s theme_bw()



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 ugradene 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()

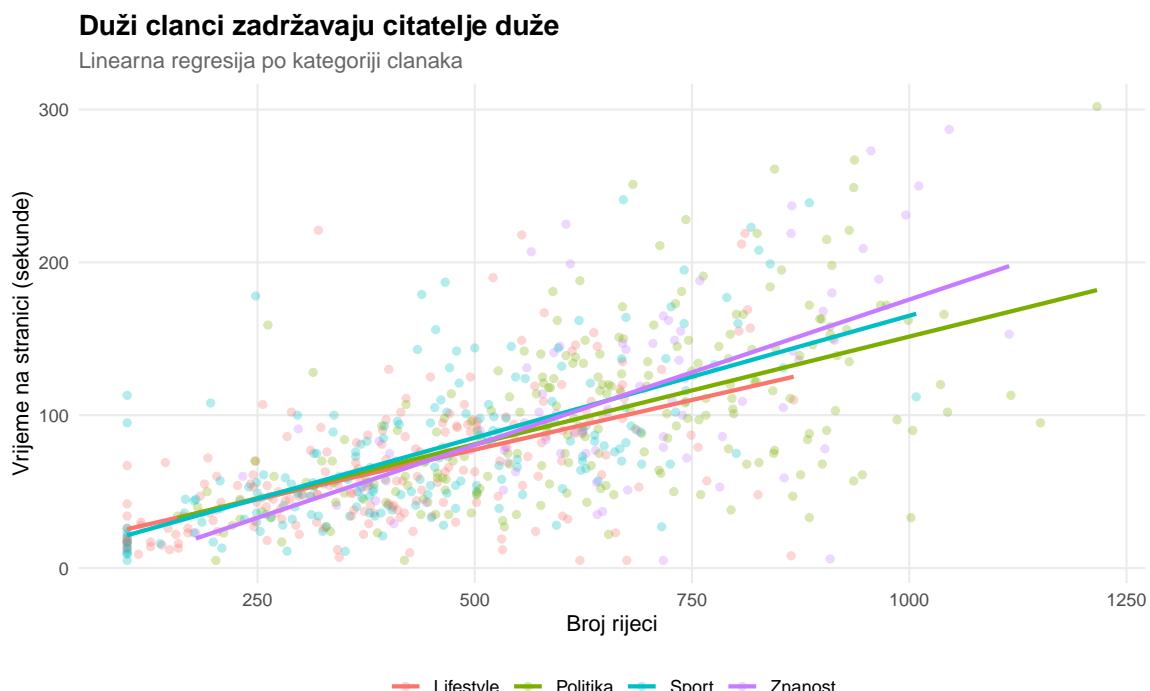
Funkcija theme() omogućuje prilagodbu pojedinačnih vizualnih elemenata. Ovo je detaljni alat za fino podešavanje izgleda.

```
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). `element_blank()` potpuno uklanja element (ovdje 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.

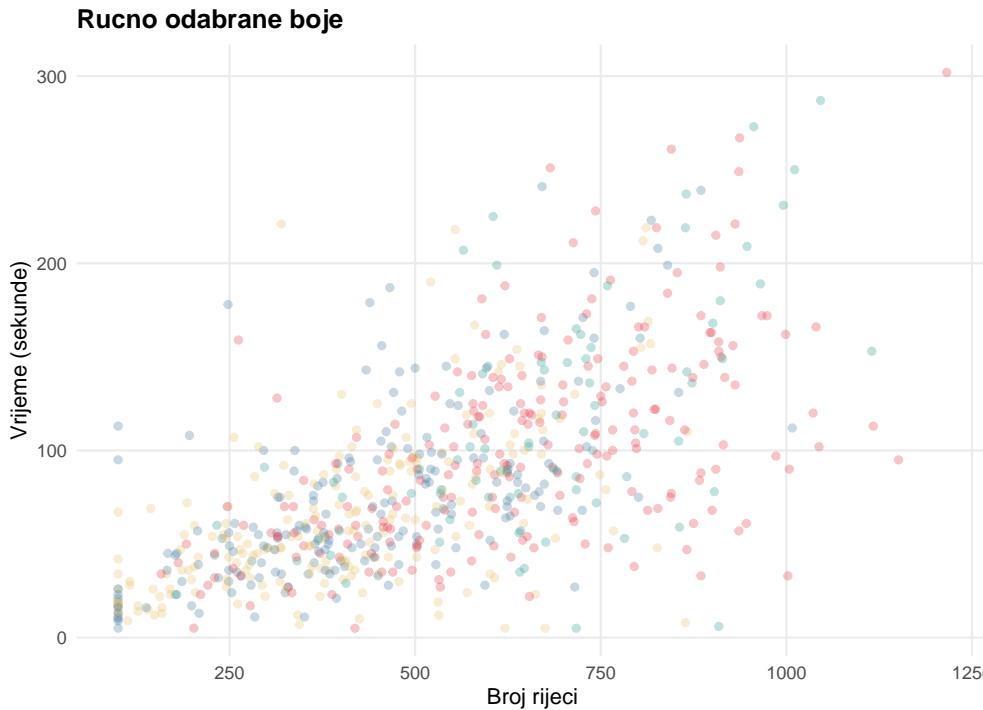
7.13 Skale boja

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"
  )
```

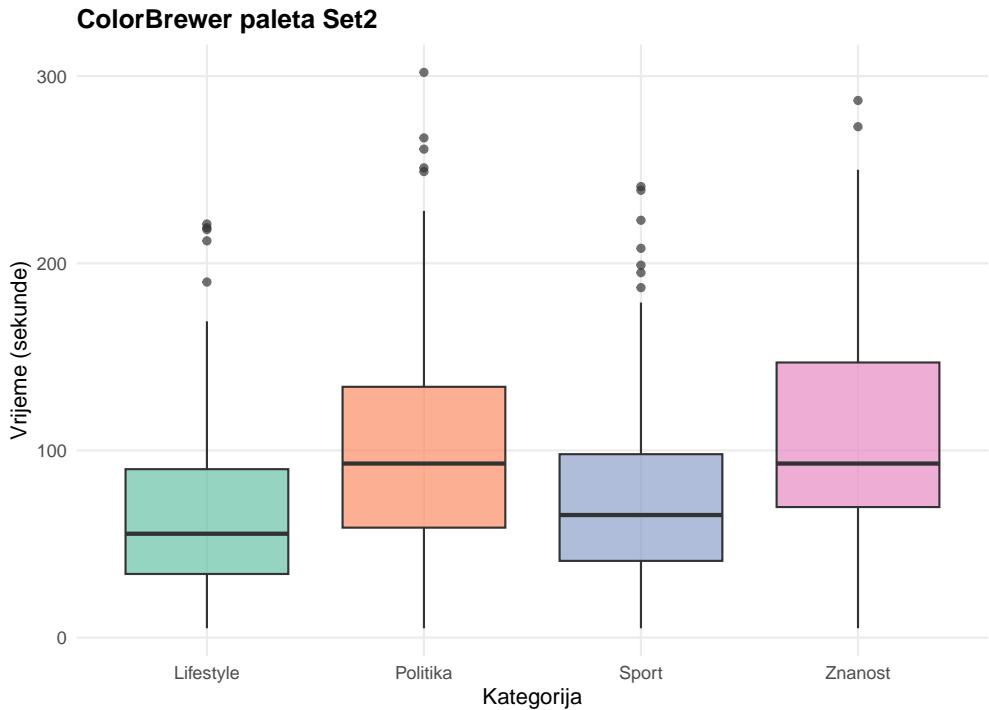


`scale_color_manual()` (za boju linija i točaka) i `scale_fill_manual()` (za boju ispune) primaju imenovani vektor boja. Prednost ručnog odabira je potpuna kontrola, ali zahtijeva poznavanje hex kodova boja ili korištenje alata poput [colors.co](#) za odabir usklađenih paleta.

7.13.2 ColorBrewer paleta

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"
  )
```



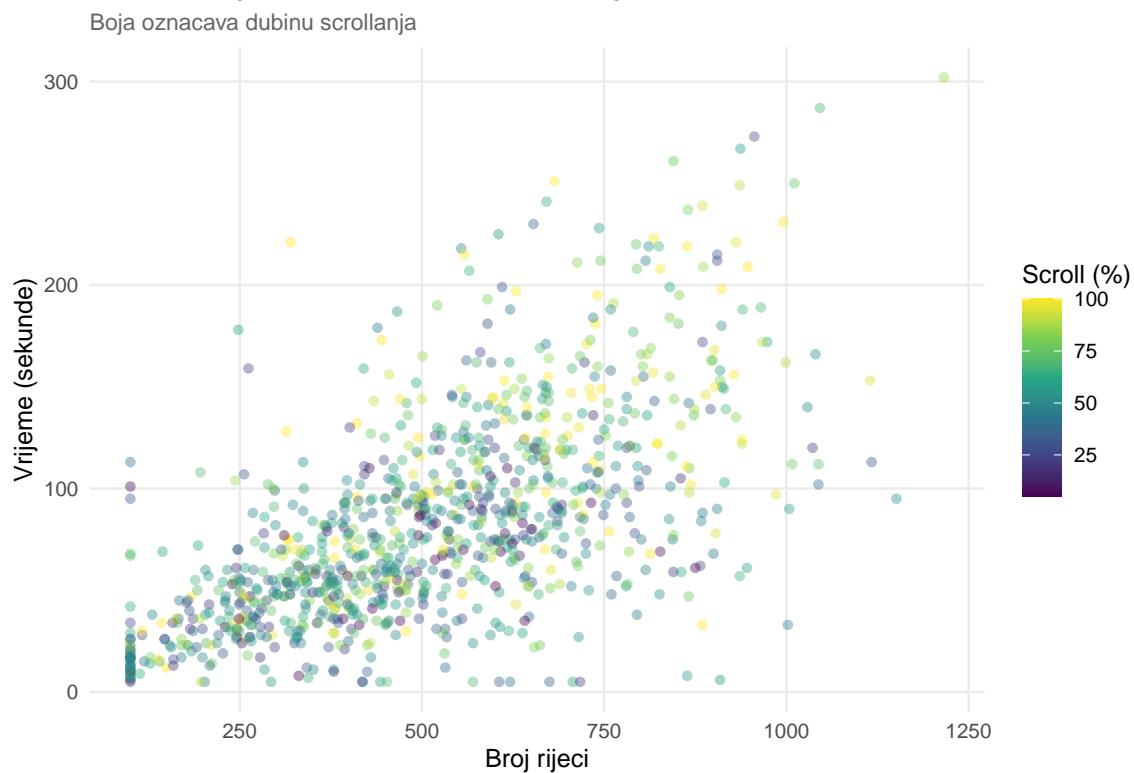
Brewer palete dolaze u tri tipa: **kvalitativne** (za kategorije, npr. "Set1", "Set2", "Dark2", "Pastel1"), **sekvencijalne** (za gradijent od niske do visoke vrijednosti, npr. "Blues", "Reds", "YlOrRd") i **divergentne** (za vrijednosti koje se razilaze od sredine, npr. "RdBu", "PRGn").

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 (%)"
  )
```

Odnos duljine clanca i vremena citanja



`scale_color_viridis_c()` koristi kontinuiranu viridis paletu (za numeričke varijable). `scale_color_viridis_d()` koristi diskretnu verziju (za kategoričke). 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 promjeniti i nećete dobiti grešku, samo prazan vizualni rezultat.

7.14 Formatiranje osi

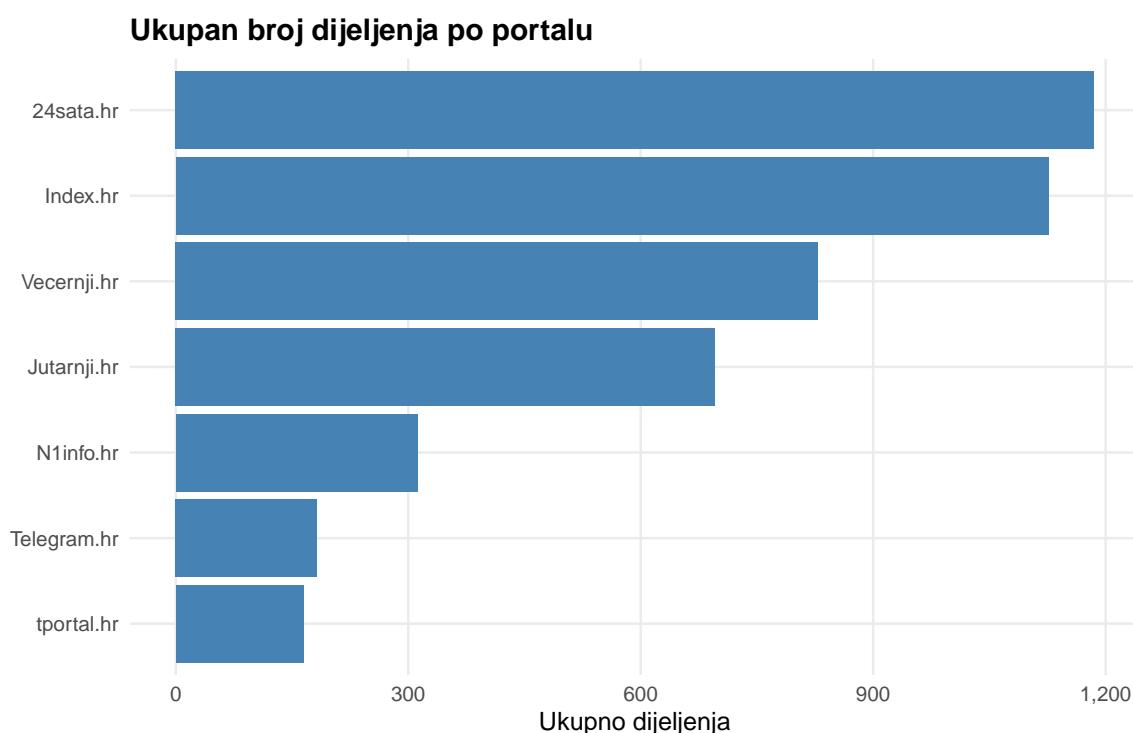
Ponekad defaultne oznake na osima nisu optimalne. Paket scales (automatski učitan s tidyverse) pruža pomoćne funkcije za formatiranje.

```

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"
  )

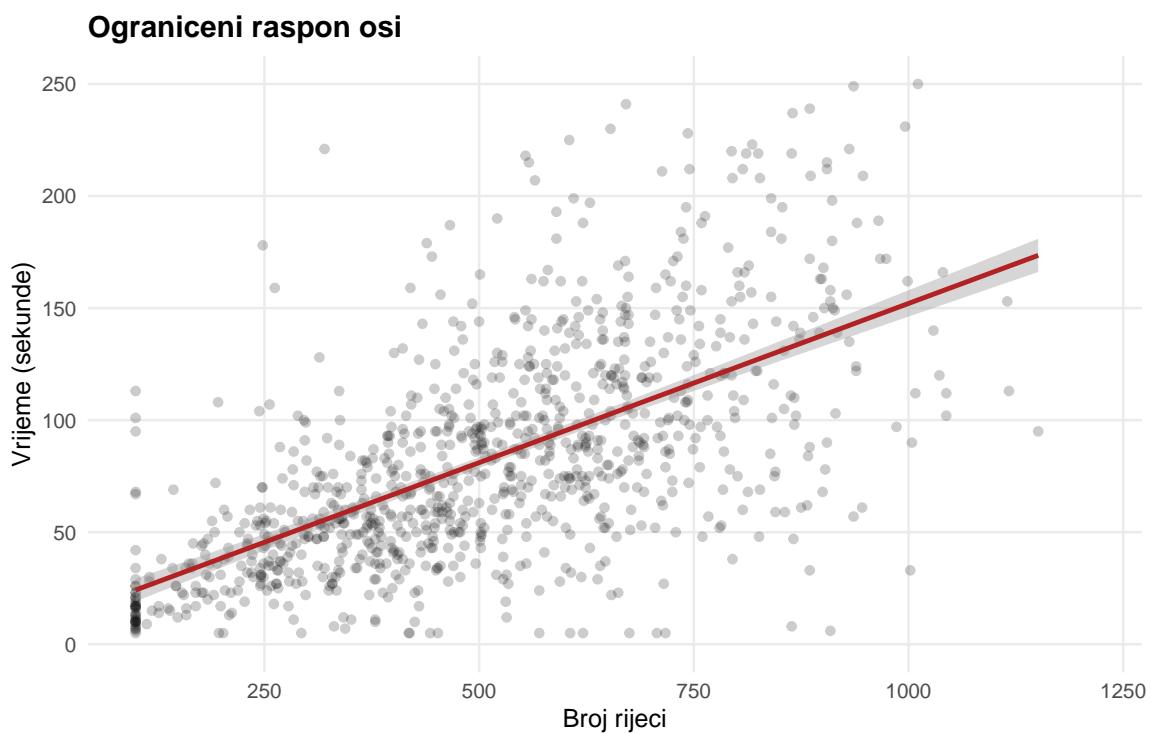
```



Funkcija `label_comma()` formatira brojeve s tisućicama (1,000 umjesto 1000). Druge korisne funkcije iz paketa scales su `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)"  
)
```

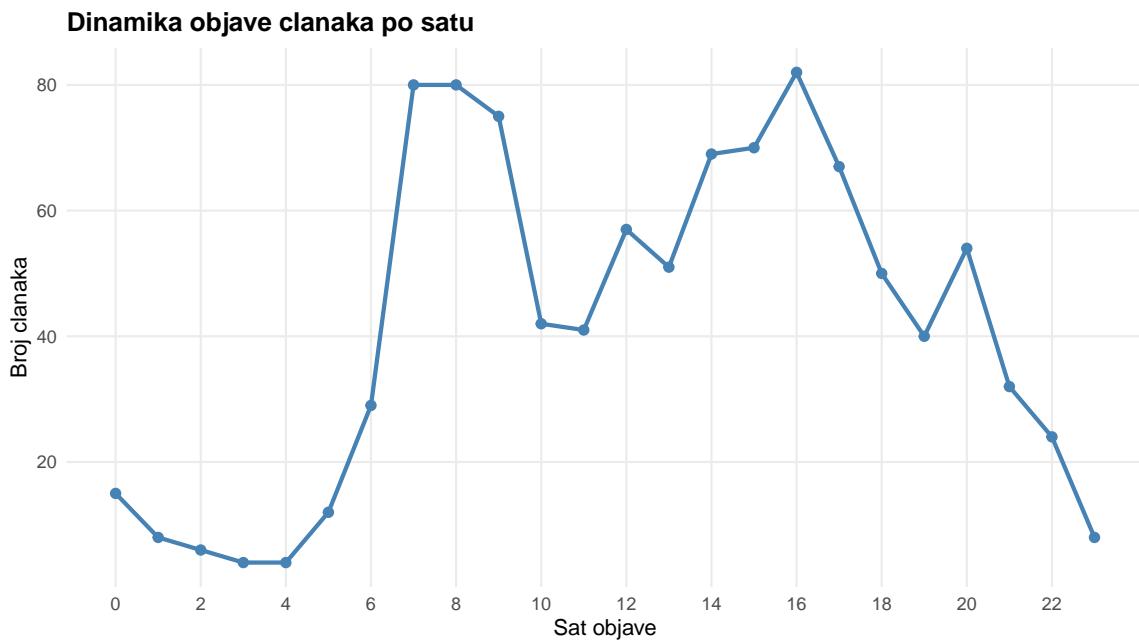


`breaks` kontrolira gdje se pojavljuju oznake na osi. `limits` ograničava raspon osi (točke izvan raspona se uklanaju iz grafa). Koristite `coord_cartesian(ylim = c(0, 250))` umjesto `limits` ako želite “zumirati” bez uklanjanja podataka, jer `coord_cartesian()` samo sužava prikaz dok `limits` zaista filtrira podatke prije nego ih ggplot obradi (što može utjecati na linije trenda).

7.15 Linijski grafovi: trendovi i serije

Linijski grafovi su prirođan 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"
)
```



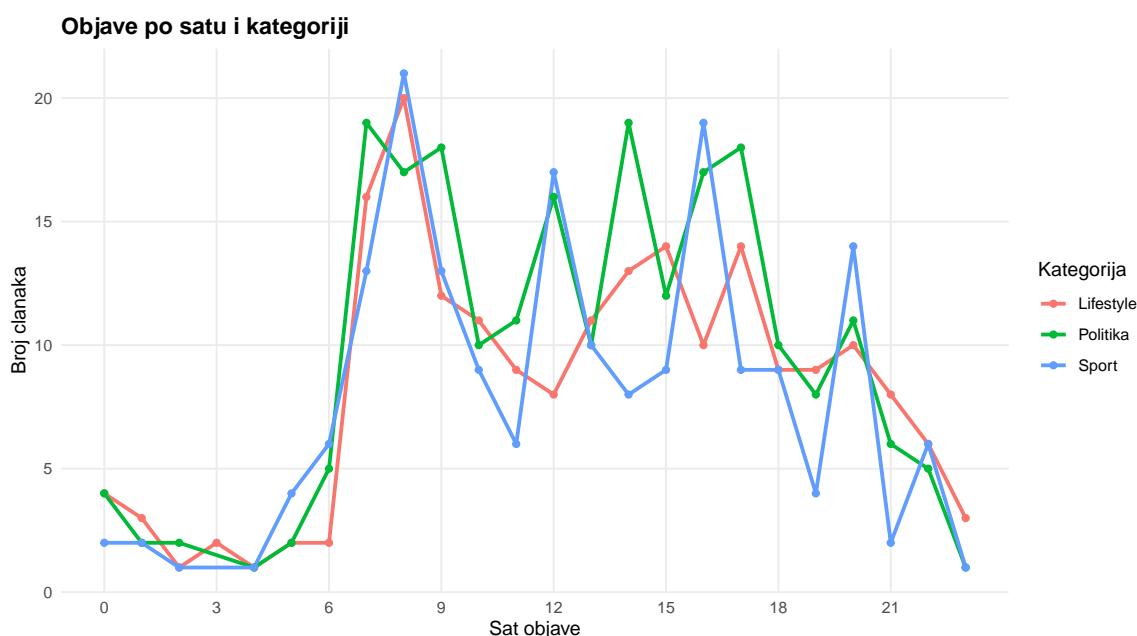
Kombinacija `geom_line()` i `geom_point()` je uobičajena jer linije pokazuju trend, a točke označavaju stvarne podatke. Vidimo jasne vrhunce u jutarnjim satima i kasno popodne, što odgovara redakcijskim ciklusima.

7.15.1 Više linija u jednom grafu

```

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.

```

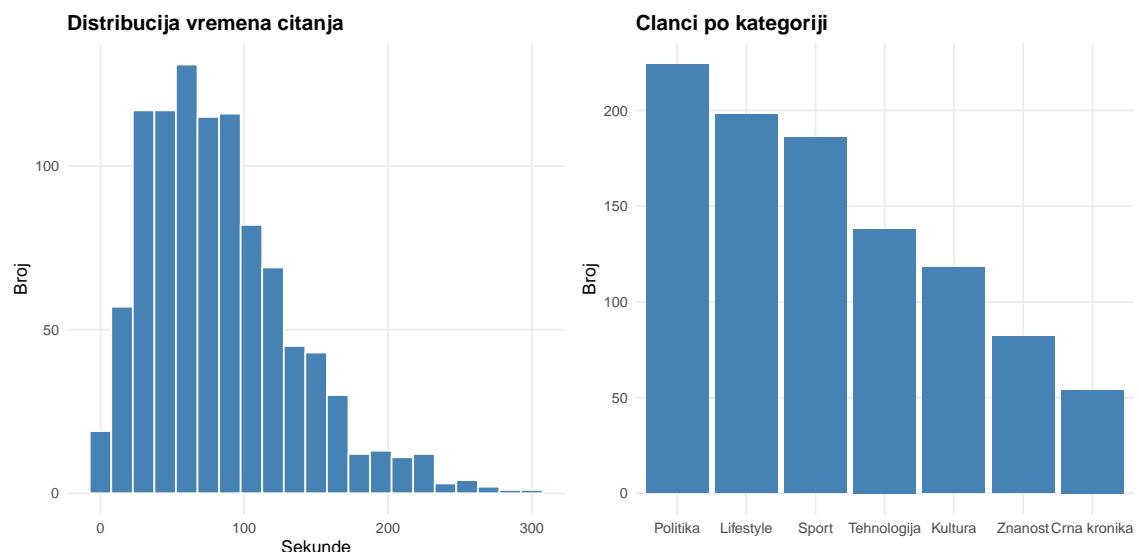
library(patchwork)

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) i | eksplicitno horizontalno.

```

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",

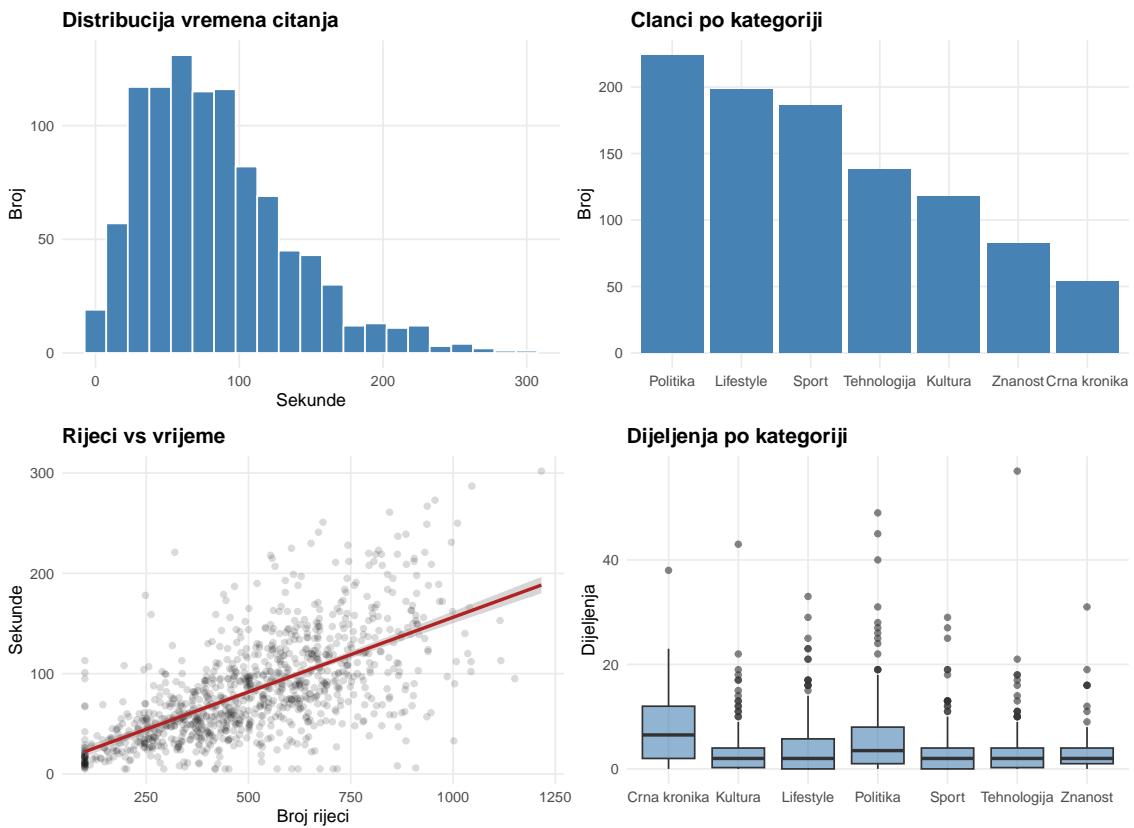
```

```
caption = "Izvor: simulirani podaci, N = 1000 članaka"
```

```
)
```

Analiza angažmana citatelja na portalima

Pregled distribucija, kategorija i odnosa varijabli

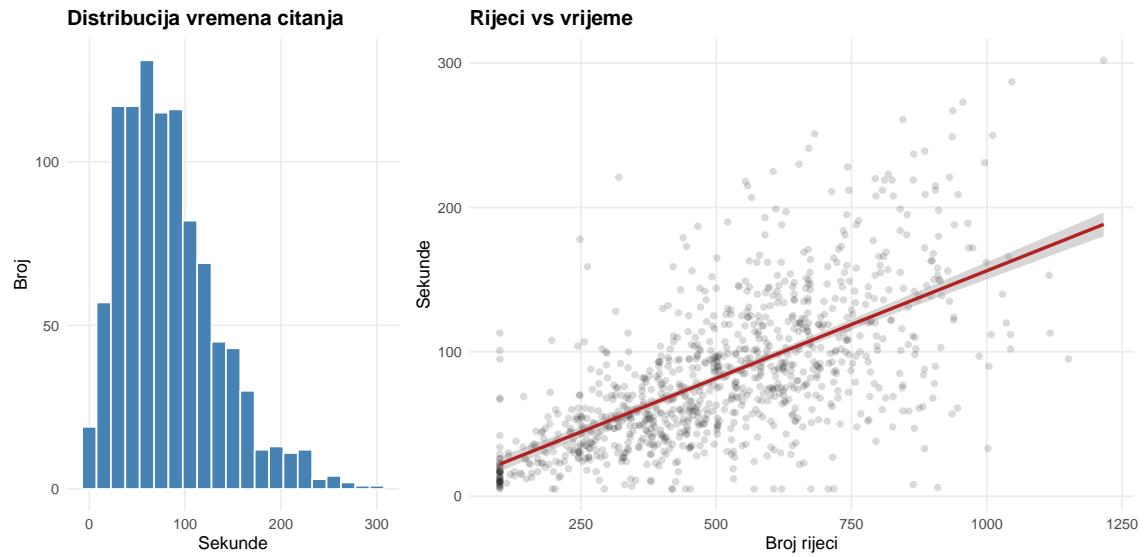


Izvor: simulirani podaci, N = 1000 članaka

Zgrade i operatori kontroliraju raspored: $(p1 \mid p2) / (p3 \mid p4)$ kreira matricu 2×2 . `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.

```
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: PNG, PDF, SVG, JPEG, TIFF.

```
# Spremi zadnji graf kao PNG
ggsave("angažman_portali.png", width = 10, height = 6, dpi = 300)

# Spremi specifični graf kao PDF (vektorski format, idealan za tisk)
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 su `width` i `height` (dimenzije u inčima) i `dpi` (rezolucija za rasterske formate). Za tisk 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 tisk. PNG je rasterski i bolji 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 rade, uže (6x4). Konzistentne dimenzije daju profesionalan izgled cijelom dokumentu.

7.18 Česte greške i kako ih izbjegći

Učenje ggplot2 dolazi s karakterističnim setom grešaka. Prepoznavanje najčešćih štedi sate frustracije.

7.18.1 Greška 1: + umjesto |> (i obrnuto)

```
# KRIVO: pipe unutar ggplot lanza
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()

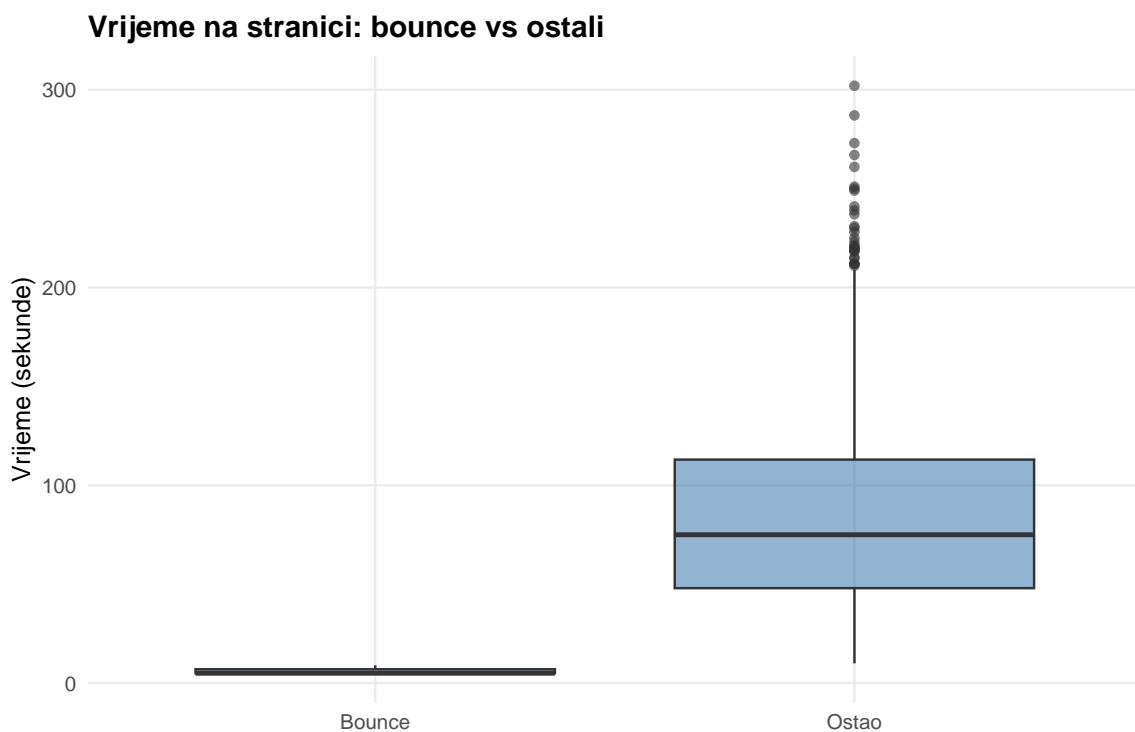
# BOLJE: kategorička varijabla za fill
ggplot(clanci, aes(x = category, fill = headline_style)) +
  geom_bar(position = "dodge")
```

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 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.

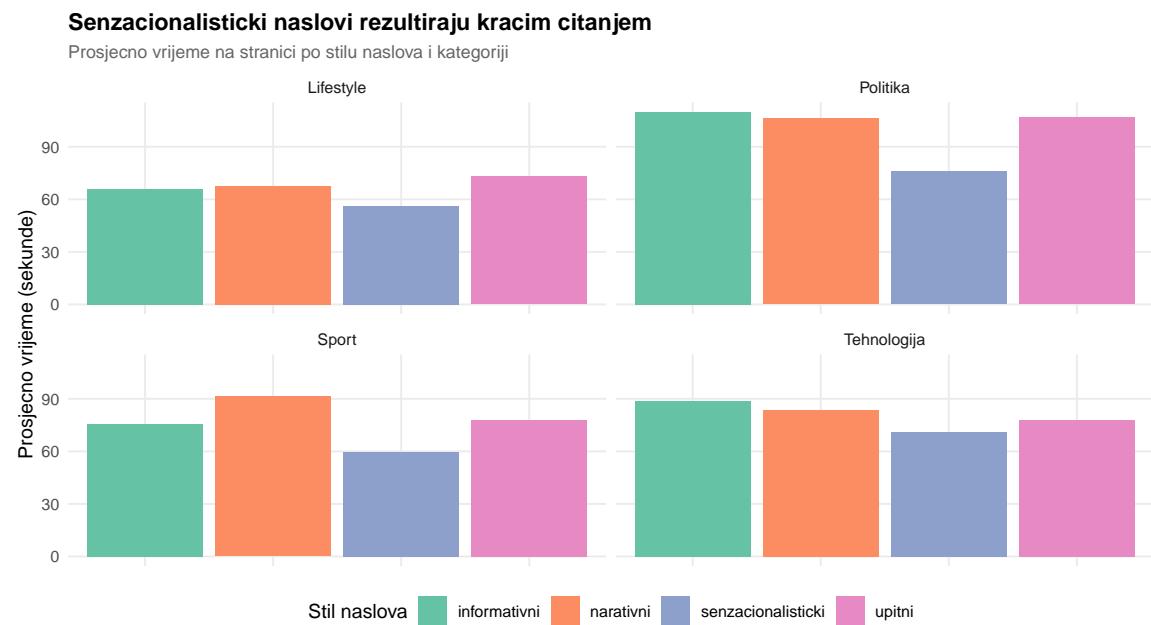
Pitanje: Kako se angažman čitatelja (vrijeme čitanja i dijeljenje) razlikuje ovisno o stilu naslova i kategoriji članka?

```
# 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) +
```

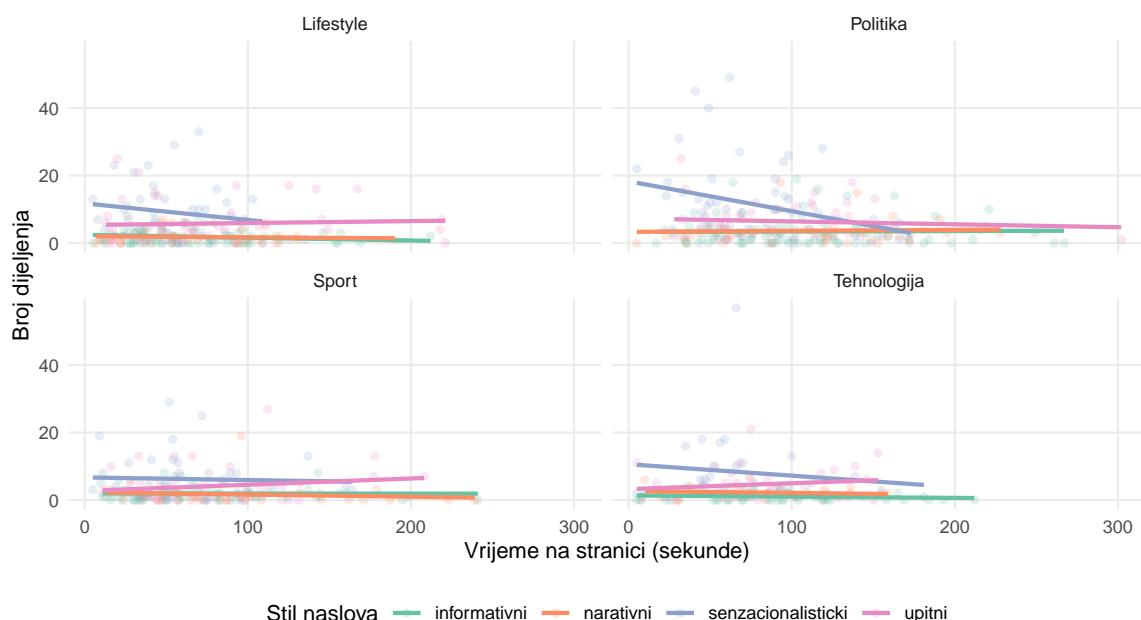
```

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")

```

Više vremena na stranici ne znaci nužno više dijeljenja

Senzacionalistički naslovi imaju više dijeljenja ali kraće čitanje



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")

```

```

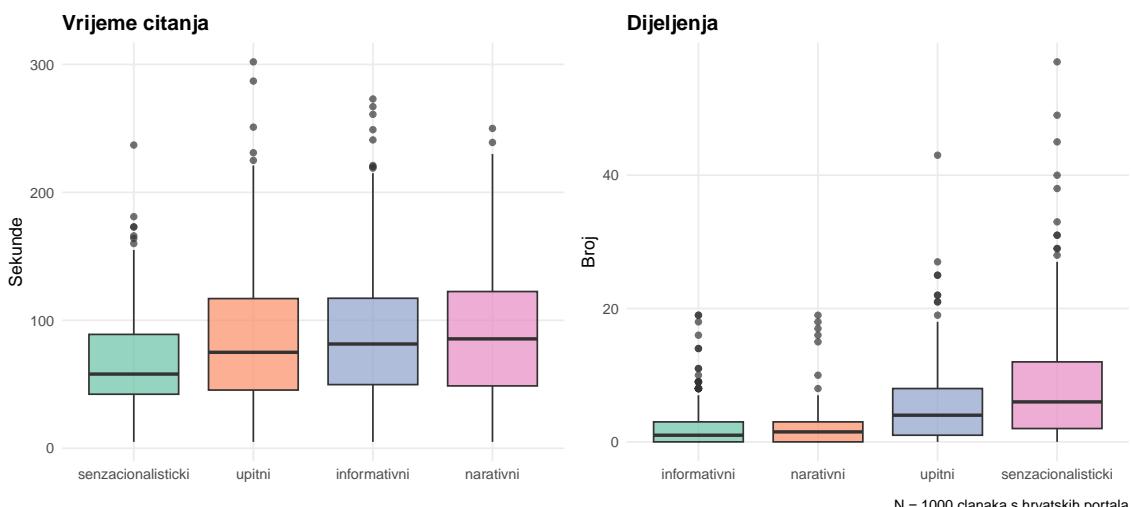
graf_b <- clanci |>
  mutate(headline_style = fct_reorder(headline_style, shares)) |>
  ggplot(aes(x = headline_style, y = shares, fill = headline_style)) +
  geom_boxplot(alpha = 0.7, show.legend = FALSE) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Dijeljenja", x = NULL, y = "Broj")

graf_a + graf_b +
  plot_annotation(
    title = "Paradoks senzacionalizma",
    subtitle = "Senzacionalistički naslovi: manje čitanja, više dijeljenja",
    caption = "N = 1000 članaka s hrvatskih portala"
  )

```

Paradoks senzacionalizma

Senzacionalistički naslovi: manje čitanja, više dijeljenja



Tri grafa zajedno ispričali su kompletну 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 zbumjenosti.

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, sekvensialne (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()` spremi grafove u datoteku. PDF za tisk (vektorski), PNG za web (rasterski). Koristite `dpi = 300` za tisk.
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**: pisanje funkcija, uvjetne naredbe, petlje i organizacija 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

- 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.
<code>aes()</code>	Funkcija za mapiranje varijabli na vizualne dimenziije grafa (x os, y os, boja, veličina, oblik).
Geometrija (<code>geom_*</code>)	Vizualni oblik za prikaz podataka. Svaki tip grafa ima svoju geom funkciju.
<code>geom_histogram()</code>	Geometrija za histogram. Argument <code>binwidth</code> kontrolira širinu bina.
<code>geom_density()</code>	Geometrija za graf gustoće distribucije.
<code>geom_bar()</code>	Geometrija za bar chart koji automatski broji opažanja po kategorijama.
<code>geom_col()</code>	Geometrija za bar chart s prethodno izračunatim y vrijednostima.
<code>geom_boxplot()</code>	Geometrija za boxplot koji prikazuje medijan, kvartile i outliere.
<code>geom_violin()</code>	Geometrija za violin plot koji prikazuje oblik distribucije.
<code>geom_point()</code>	Geometrija za scatterplot.
<code>geom_jitter()</code>	Varijanta <code>geom_point()</code> s nasumičnim pomakom za izbjegavanje preklapanja.
<code>geom_smooth()</code>	Geometrija za liniju trenda. Default LOESS, <code>method = "lm"</code> za linearnu.
<code>geom_line()</code>	Geometrija za linjski graf. Pogodna za vremenske serije i podatke s redoslijedom.
<code>facet_wrap()</code>	Dijeli graf na panele po jednoj varijabli. Argumenti: <code>ncol</code> , <code>scales</code> .
<code>facet_grid()</code>	Dijeli graf na matricu panela po dvjema varijablama. Sintaksa: <code>retci ~ stupci</code> .
<code>labs()</code>	Funkcija za naslove, podnaslove, oznake osi, legende i caption.
<code>theme_minimal()</code>	Ugrađena tema: čista, bez okvira, minimalna mreža. Popularna za profesionalni rad.
<code>theme_bw()</code>	Ugrađena tema: bijela pozadina s crnim okvirom.
<code>theme()</code>	Funkcija za detaljnu prilagodbu vizualnih elemenata (fontovi, margine, legenda, mreža).
<code>theme_set()</code>	Postavlja globalnu temu za sve grafove u dokumentu.
<code>element_text()</code>	Unutar <code>theme()</code> : kontrolira svojstva teksta (veličina, bold, boja, kut).

Pojam	Objašnjenje
<code>element_blank()</code>	Unutar <code>theme()</code> : potpuno uklanja element (mreža, oznake, rubovi).
<code>scale_color_manual()</code>	Ručni odabir boja za <code>color</code> estetiku.
<code>scale_fill_manual()</code>	Ručni odabir boja za <code>fill</code> estetiku.
<code>scale_color_brewer()</code>	ColorBrewer palete za <code>color</code> . Tipovi: kvalitativne, sekvencijalne, divergentne.
<code>scale_fill_brewer()</code>	ColorBrewer palete za <code>fill</code> .
<code>scale_color_viridis_c()</code>	Viridis paleta za kontinuirane varijable. Pristupačna za poremećaj vida boja.
<code>scale_color_viridis_d()</code>	Viridis paleta za diskretne varijable.
<code>alpha</code>	Estetika za transparentnost. Od 0 (prozirno) do 1 (neprozirno).
<code>fill</code>	Estetika za boju ispune (stupci, pravokutnici, područja).
<code>color</code>	Estetika za boju ruba ili linije (točke, linije, rubovi).
<code>position = "dodge"</code>	Stupci jedne do drugih u grupiranom bar chartu.
<code>position = "fill"</code>	Normalizira stupce na proporcije.
<code>fct_infreq()</code>	Sortira faktor po frekvenciji.
<code>fct_reorder()</code>	Sortira faktor po vrijednostima druge varijable.
<code>coord_flip()</code>	Zamjenjuje x i y os za horizontalne grafove.
<code>coord_cartesian()</code>	Zumira graf bez uklanjanja podataka.
<code>ggsave()</code>	Sprema graf u datoteku. Argumenti: širina, visina, dpi, format.
<code>patchwork</code>	Paket za kombiniranje više ggplot2 grafova. Operatori: + (horizontalno), / (vertikalno), (horizontalno).
<code>plot_annotation()</code>	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)`

i 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: 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 datase. 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: t-testova, hi-kvadrat testova, ANOVA-e, regresije. 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, 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.

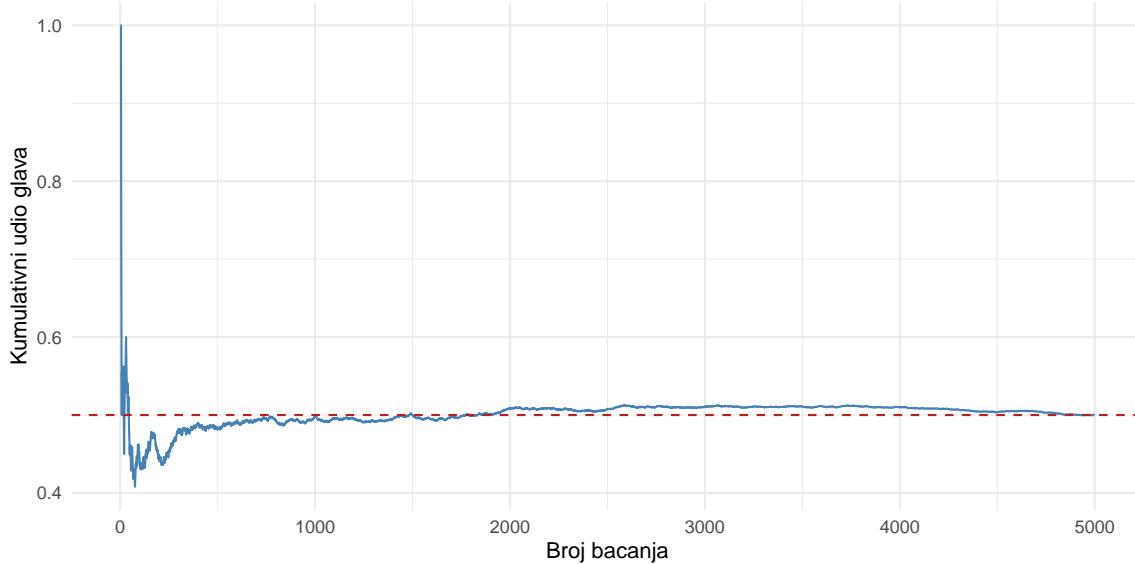
```
set.seed(42)

# 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()
```

Zakon velikih brojeva u akciji
Udio glava konvergira prema 0.5 s više bacanja



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: "na temelju onoga što znam, moja 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 $1 - P(A)$.

$$P(\text{nije } A) = 1 - P(A)$$

```
# Vjerojatnost da je objava viralna
p_viral <- mean(posts$is_viral)
cat("P(viralno) =", round(p_viral, 3), "\n")
```

$P(\text{viralno}) = 0.014$

```
# Vjerojatnost da NIJE viralna (komplement)
p_ne_viral <- 1 - p_viral
cat("P(\text{nije viralno}) =", round(p_ne_viral, 3), "\n")
```

$P(\text{nije viralno}) = 0.986$

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.

$$P(A \text{ ili } B) = P(A) + P(B)$$

```
# 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.

$$P(A \text{ ili } B) = P(A) + P(B) - P(A \text{ i } B)$$

```
# 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
```

```

cat("P(video ILI viralno) =", round(p_video + p_viral - p_video_i_viral, 3), "\n")

P(video ILI viralno) = 0.23

# 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):

$$P(A \text{ i } B) = P(A) \times P(B)$$

```

# 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

```

# 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".

$$P(A|B) = \frac{P(A \text{ i } B)}{P(B)}$$

```
# 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")
```

$P(\text{viralno} | \text{TikTok}) = 0.0356$

```
# Usporedba s ukupnom vjerojatnošću viralnosti
cat("P(viralno) ukupno =", round(mean(posts$is_viral), 4), "\n")
```

$P(\text{viralno}) \text{ ukupno} = 0.014$

```
# 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 $P(\text{viralno} \mid \text{TikTok})$ različit od $P(\text{viralno})$, 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 $X \sim \text{Binomial}(n, p)$ 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: kad je vjerojatnost uspjeha samo 2%, većinu vremena nećete imati nijedan uspjeh u 20 pokušaja.

8.6.3 Vizualizacija binomne distribucije

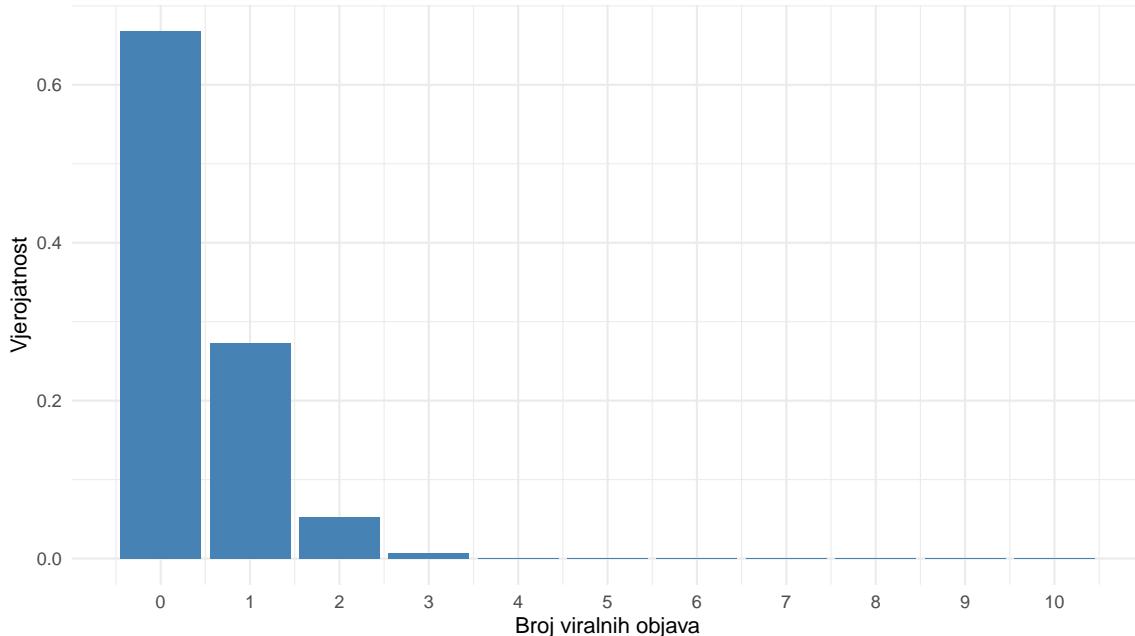
```

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()

```

Binomna distribucija: $B(n = 20, p = 0.02)$

Broj viralnih objava od 20 pokušaja



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 $P(X \leq q)$. To je vjerojatnost da dobijemo q ili manje uspjeha.

```
# P(X <= 1): vjerojatnost 0 ili 1 viralne objave od 20
pbinom(q = 1, size = 20, prob = 0.02)
```

```
[1] 0.940101
```

```
# P(X <= 3): vjerojatnost 3 ili manje
pbinom(q = 3, size = 20, prob = 0.02)
```

```
[1] 0.9994003
```

```
# 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.

```
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.

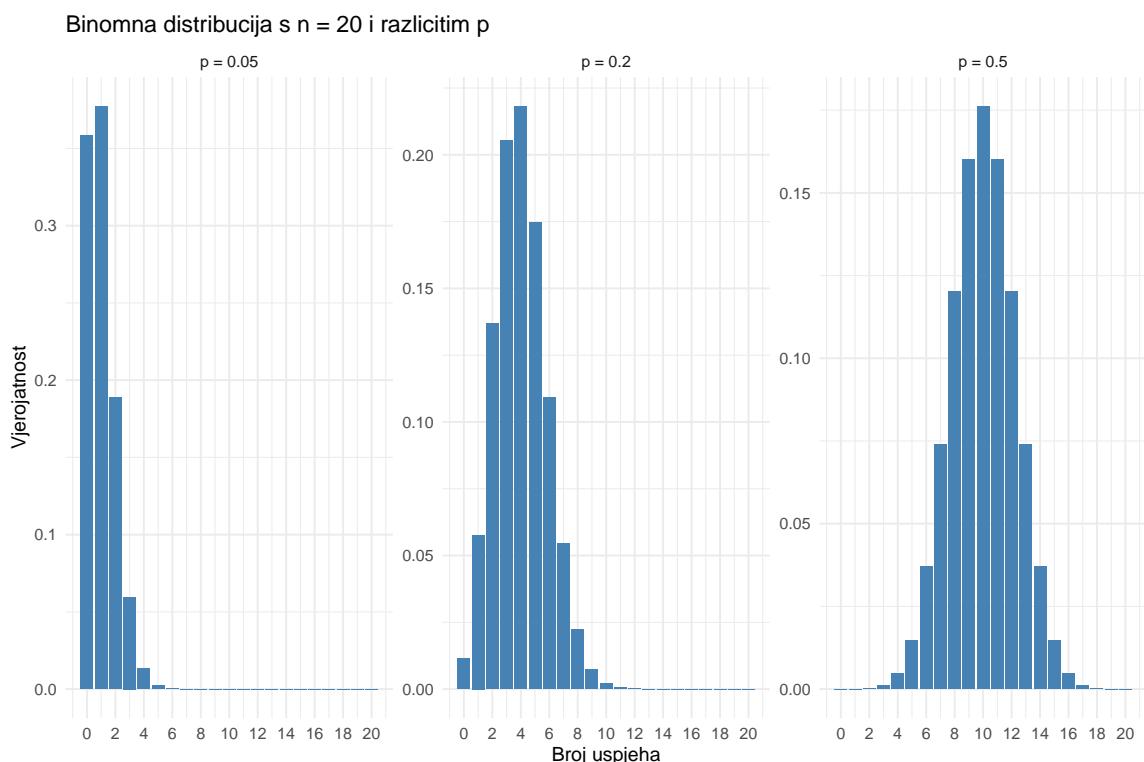
8.6.6 Kako p mijenja distribuciju

Pogledajmo kako se distribucija mijenja s različitim vjerojatnostima uspjeha.

```

# 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 statistici: normalnoj.

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
```

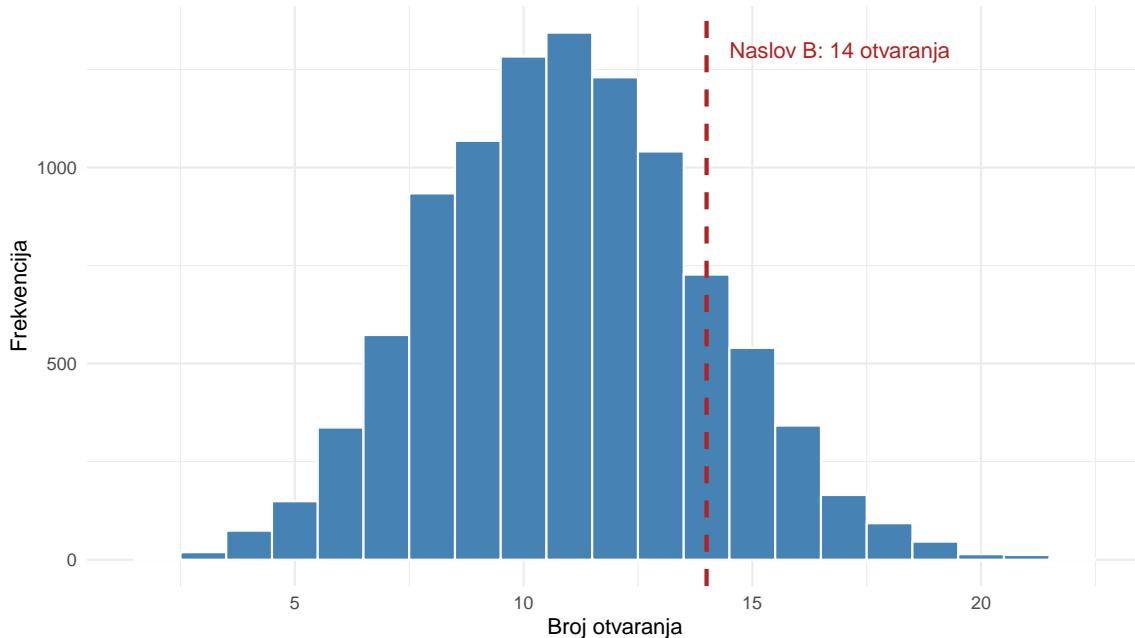
Šansa da vidimo 14 ili više otvaranja ako je pravi open rate samo 22% iznosi oko 19%. To nije zanemarivo. Ne možemo s velikom sigurnošću tvrditi da je B bolji samo na temelju ovog jednog uzorka.

Ovo je srž statističkog razmišljanja i prethodnica formalnih testova hipoteza koje ćemo učiti u tijednu 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()
```

Što bismo očekivali ako je open rate zaista 22%?

Simulacija 10000 uzoraka od 50 emailova



Crvena crta pokazuje rezultat naslova B (14 otvaranja). Vidimo da je to na desnom repu distribucije ali nije ekstremni 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: broj lajkova (svaki pratitelj nezavisno odlučuje), broj otvaranja emaila (svaki pretplatnik nezavisno odlučuje), broj konverzija na landing stranici (svaki posjetitelj nezavisno odlučuje). Prepostavka nezavisnosti je važna: ako jedan lajk poveća vidljivost pa uzrokuje sljedeći lajk (viralnost), stroga binomna prepostavka je narušena.

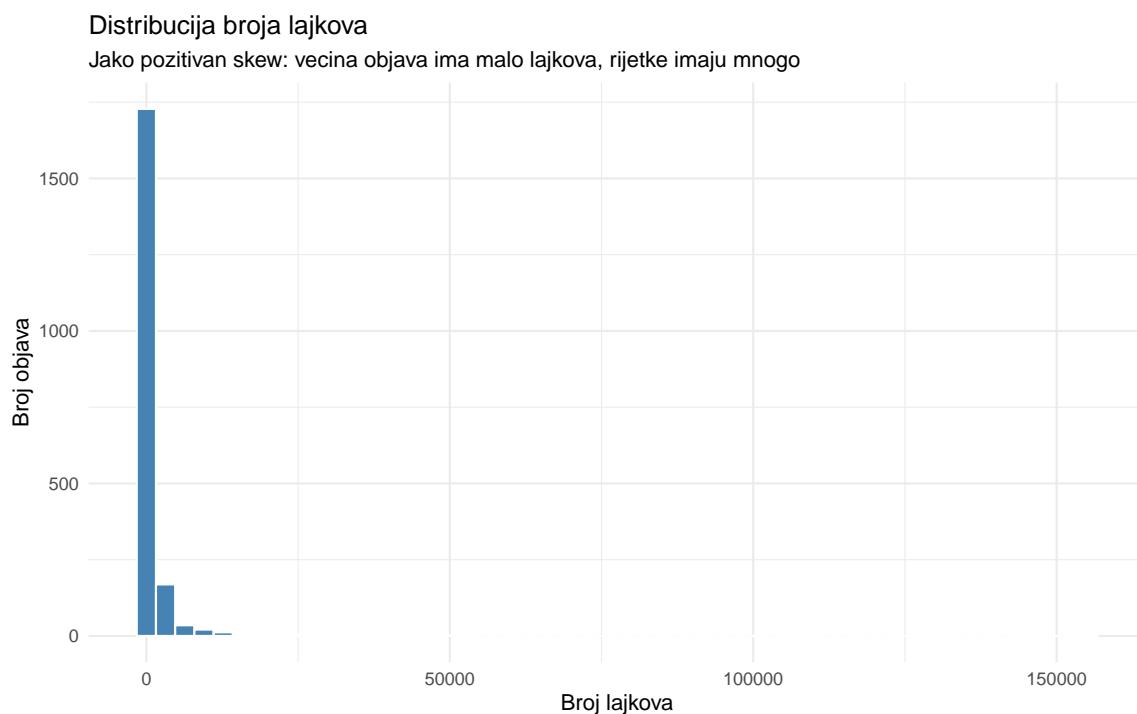
8.7 Distribucija u stvarnim podacima

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)",
  )

```

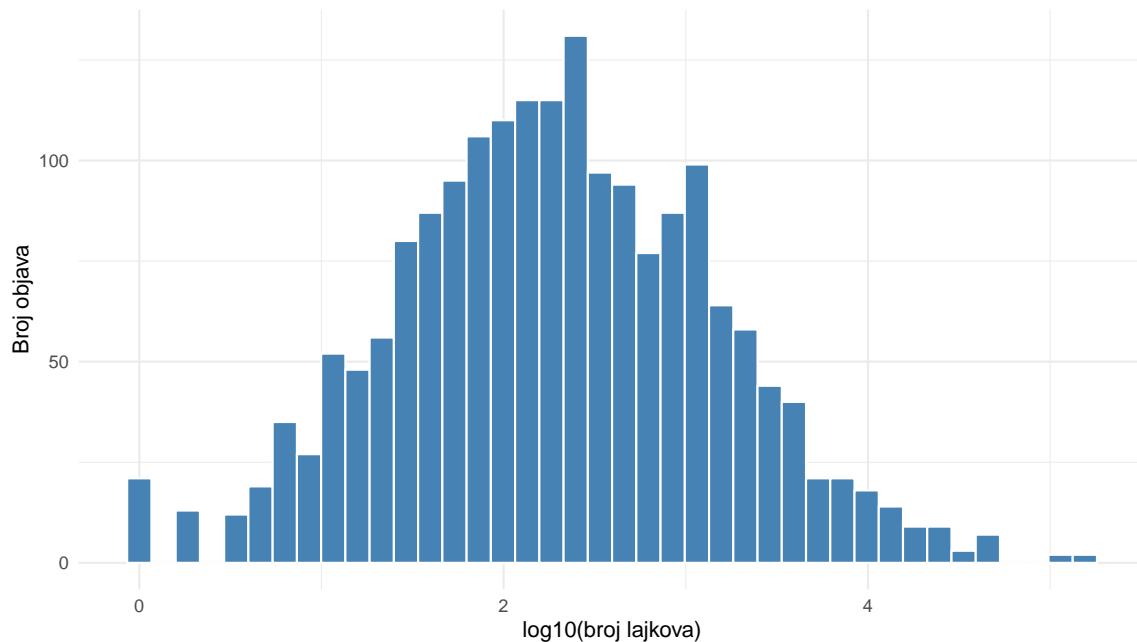
```

y = "Broj objava"
) +
theme_minimal()

```

Distribucija $\log_{10}(\text{lajkova})$

Logaritamska transformacija otkriva normalnu distribuciju ispod površine

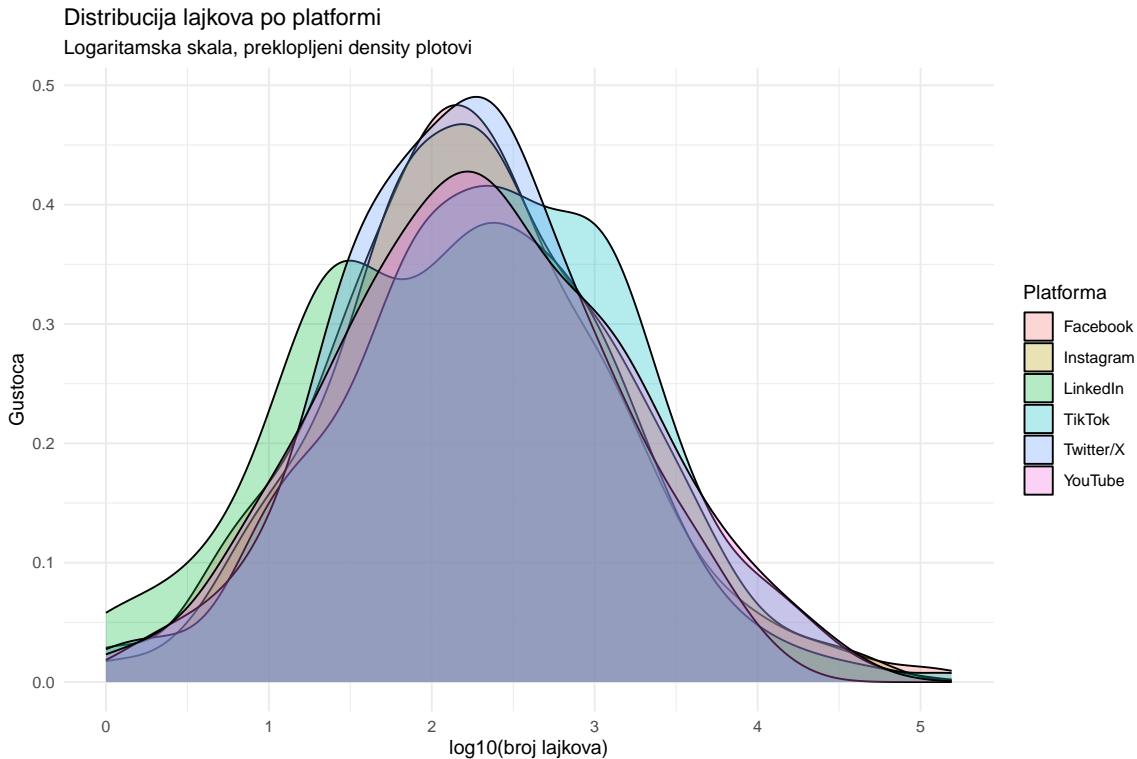


Kad primijenimo logaritamsku transformaciju, distribucija počinje nalikovati na zvonoliku krivulju. Ovo je važno zapažanje: mnoge varijable u komunikologiji koje izgledaju neprirodno iskrivljene zapravo su log-normalno distribuirane. Na logaritamskoj skali, postaju normalne. Normalnu distribuciju ćemo 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 pomaknuto uljevo (manji ali konzistentniji angažman). Ovo odražava fundamentalne razlike u mehanici platformi.

i 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: normalnu.

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 temeljem gotovo svih statističkih testova.

8.8.1 Parametri normalne distribucije

Normalna distribucija ima dva parametra:

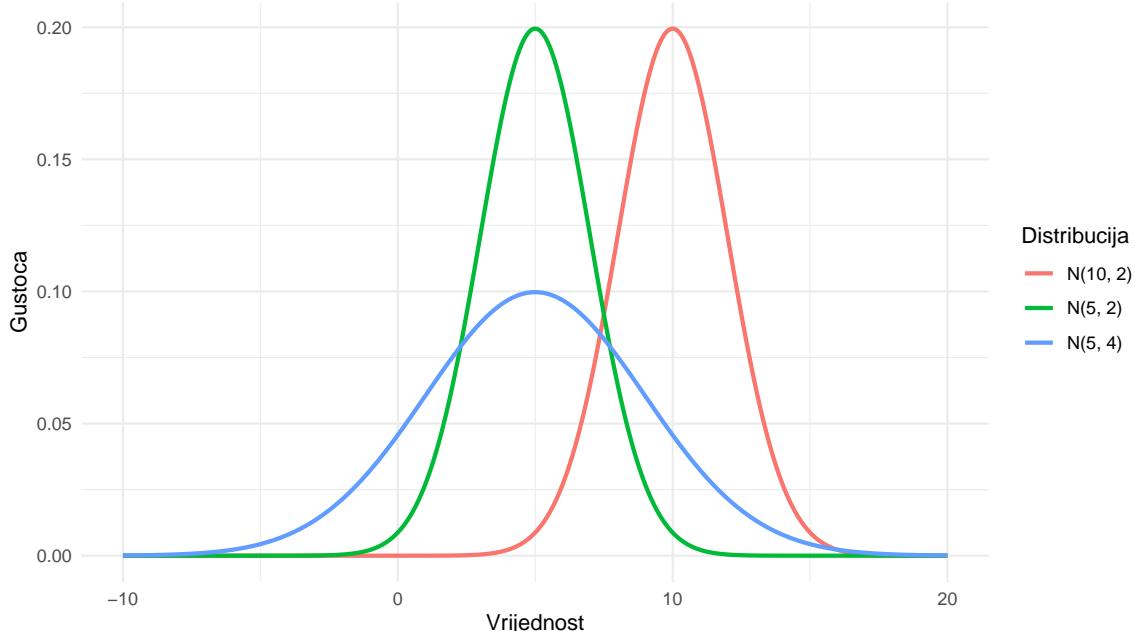
(**mi**) je srednja vrijednost (prosjek), koja određuje centar distribucije.

(**sigma**) je standardna devijacija, koja određuje širinu distribucije.

Piše se $X \sim N(\mu, \sigma)$ i čita "X slijedi normalnu distribuciju s prosjekom μ i standardnom devijacijom σ ".

```
# 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()
```

Normalna distribucija s razlicitim parametrima
 μ pomice centar, s mijenja širinu



$N(5, 2)$ i $N(10, 2)$ imaju istu širinu ($\sigma = 2$) ali različite centre ($\mu = 5$ vs $\mu = 10$). $N(5, 2)$ 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))

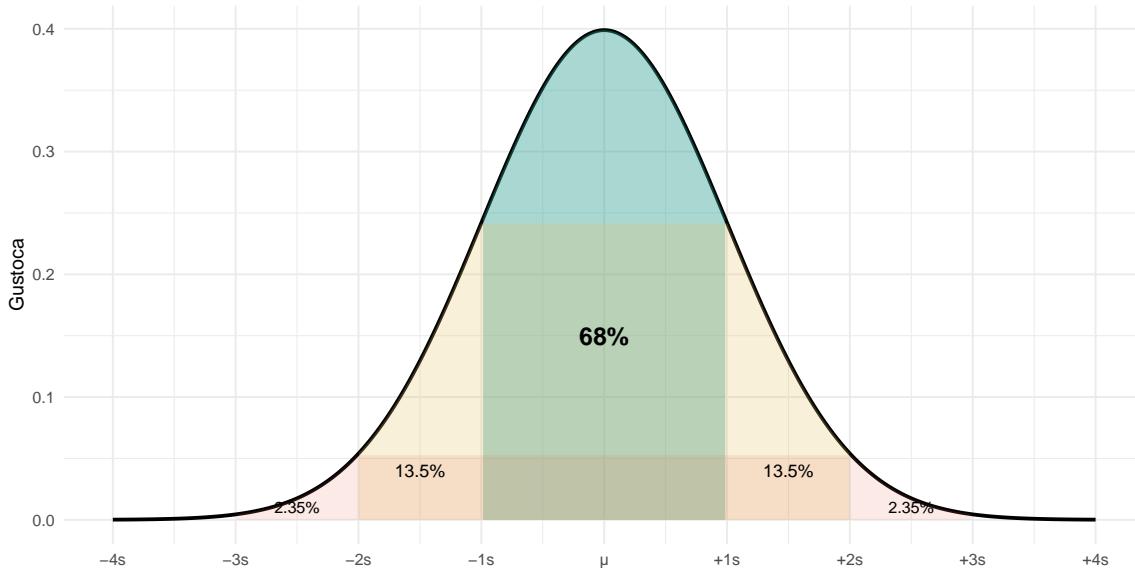
ggplot(norm_data, aes(x = x, y = y)) +
  geom_line(linewidth = 1) +
  geom_area(data = norm_data |> filter(x >= -1, x <= 1), alpha = 0.4, fill = "#2a9d8f") +
  geom_area(data = norm_data |> filter(x >= -2, x <= -1 | x >= 1, x <= 2), alpha = 0.25, fill = "#e6f2ff") +
  geom_area(data = norm_data |> filter(x >= -3, x <= -2 | x >= 2, x <= 3), alpha = 0.15, fill = "#fdd0a2")
```

```

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 ", "+3 ", "+4 "),
  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()

```

Pravilo 68–95–99.7
Postotak podataka unutar 1, 2 i 3 standardne devijacije od prosjeka



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% citatelja unutar raspona 80 ± 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  tekstu     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  tekstu     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 $P(X \leq x)$, odnosno postotak distribucije koji je ispod zadane vrijednosti.

```
# Za standardnu normalnu N(0,1):
pnorm(0)      # 50% distribucije je ispod prosjeka
```

```
[1] 0.5
```

```
pnorm(1)      # ~84% je ispod 1 SD iznad prosjeka
```

```
[1] 0.8413447
```

```
pnorm(-1)    # ~16% je ispod 1 SD ispod prosjeka
```

```
[1] 0.1586553
```

```
# 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.

```
# Postotak čitatelja između 60 i 120 sekundi  
pnorm(120, mean = 80, sd = 25) - pnorm(60, mean = 80, sd = 25)
```

```
[1] 0.7333453
```

```
# Postotak čitatelja iznad 150 sekundi (gornji rep)  
1 - pnorm(150, mean = 80, sd = 25)
```

```
[1] 0.00255513
```

```
# 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.

```
# Ispod koje vrijednosti je 95% distribucije?  
qnorm(0.95, mean = 80, sd = 25)
```

```
[1] 121.1213
```

```
# Ispod koje je 5%? (donji kvintil)  
qnorm(0.05, mean = 80, sd = 25)
```

```
[1] 38.87866
```

```
# Top 1% čitatelja (oni koji čitaju najduže)  
qnorm(0.99, mean = 80, sd = 25)
```

```
[1] 138.1587
```

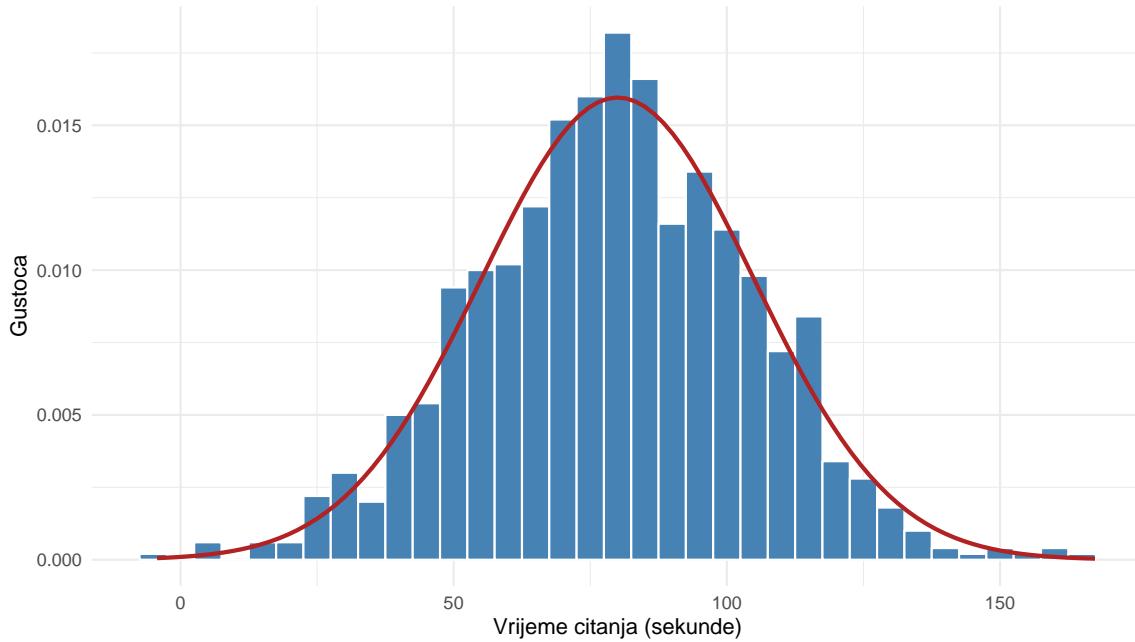
Top 1% najdediciranih čitatelja provodi više od 138 sekundi na članku. Ovo je korisno za postavljanje pragova. Na primjer, možete definirati "super čitatelja" kao onoga koji je u top 5% po vremenu čitanja.

8.10.4 rnorm(): simulacija

```
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()
```

Simulirano vrijeme citanja vs teorijska normalna krivulja

$N(\mu = 80, s = 25)$, 1000 simuliranih članaka



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

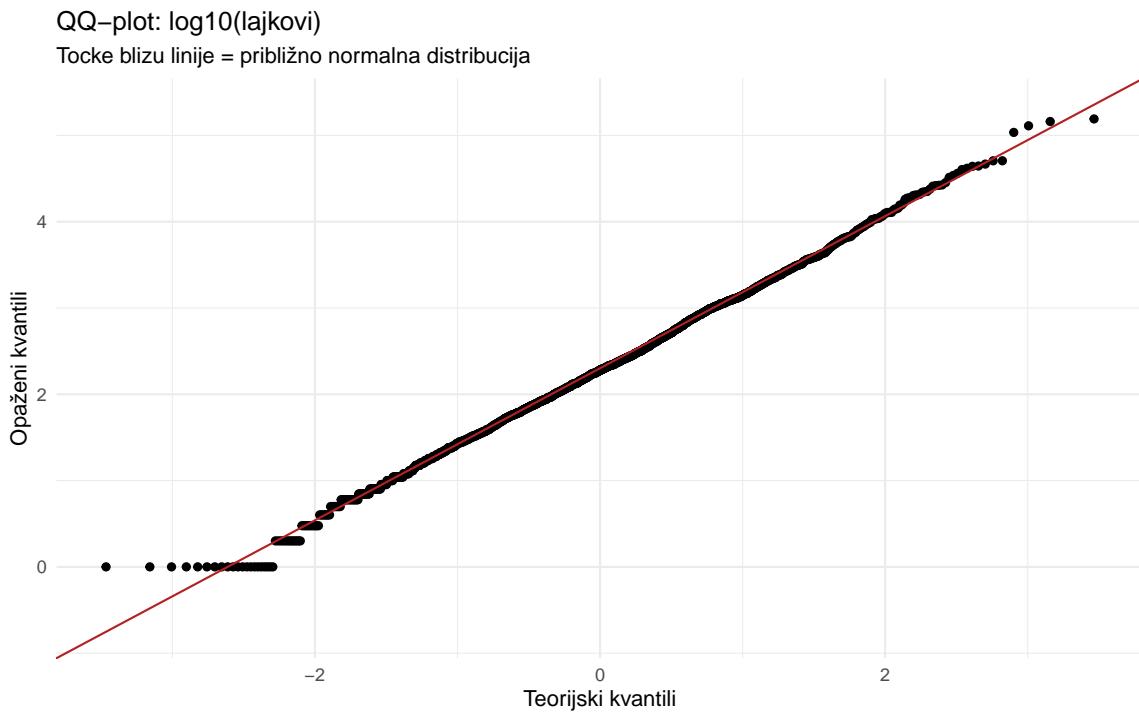
Obrazac d/p/q/r vrijedi za sve distribucije u R-u. `d` daje gustoću (ili vjerojatnost za diskretne), `p` daje kumulativnu vjerojatnost, `q` daje kvantile (obrnuto od p), `r` generira slučajne uzorke. Za binomnu: `dbinom`, `pbinom`, `qbinom`, `rbinom`. Za normalnu: `dnorm`, `pnorm`, `qnorm`, `rnorm`. Za t-distribuciju (tjedan 12): `dt`, `pt`, `qt`, `rt`. Naučite obrazac jednom, primjenite 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.

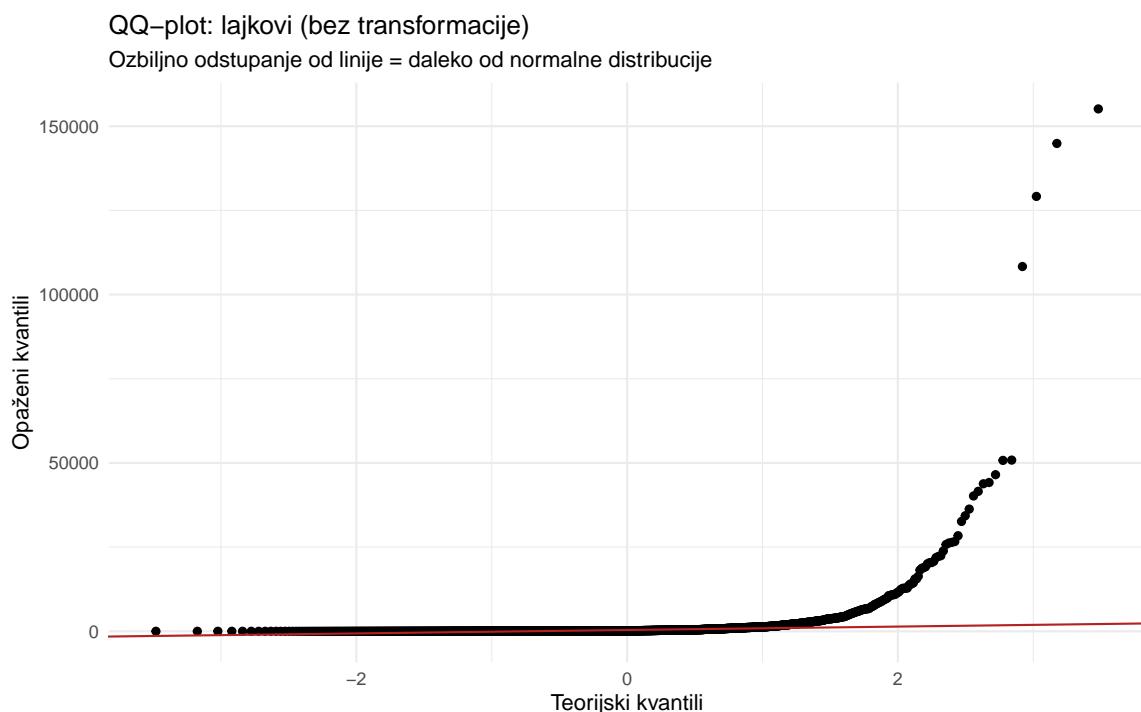
Za usporedbu, pogledajmo QQ-plot za netransformirane lajkove.

```
# 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 predviđela. Ovo je vizualni potpis za pozitivno iskrivljenu (right-skewed) distribuciju.

8.11.1 Čitanje QQ-plota

Različiti obrasci na QQ-platu 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).

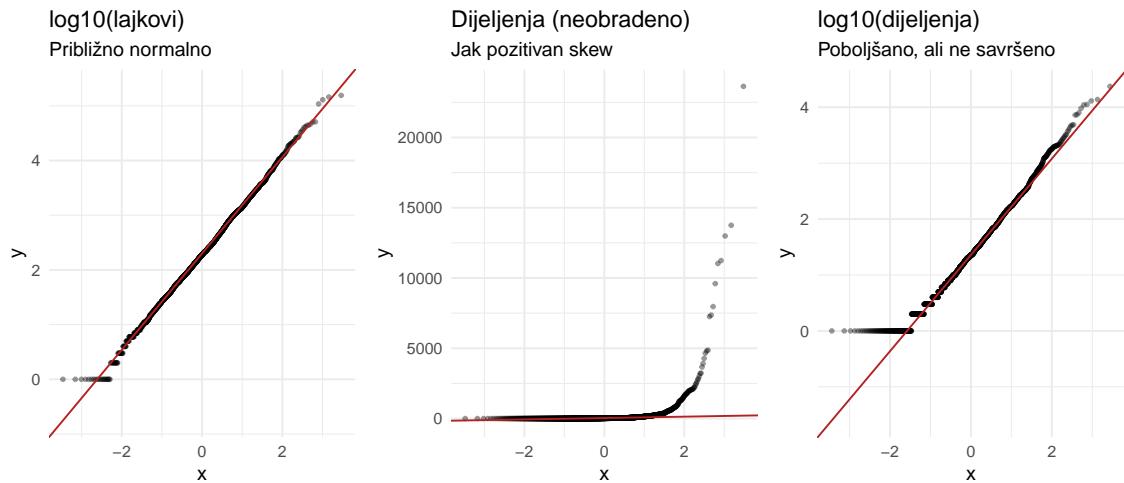
```
library(patchwork)

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"
    )
  )

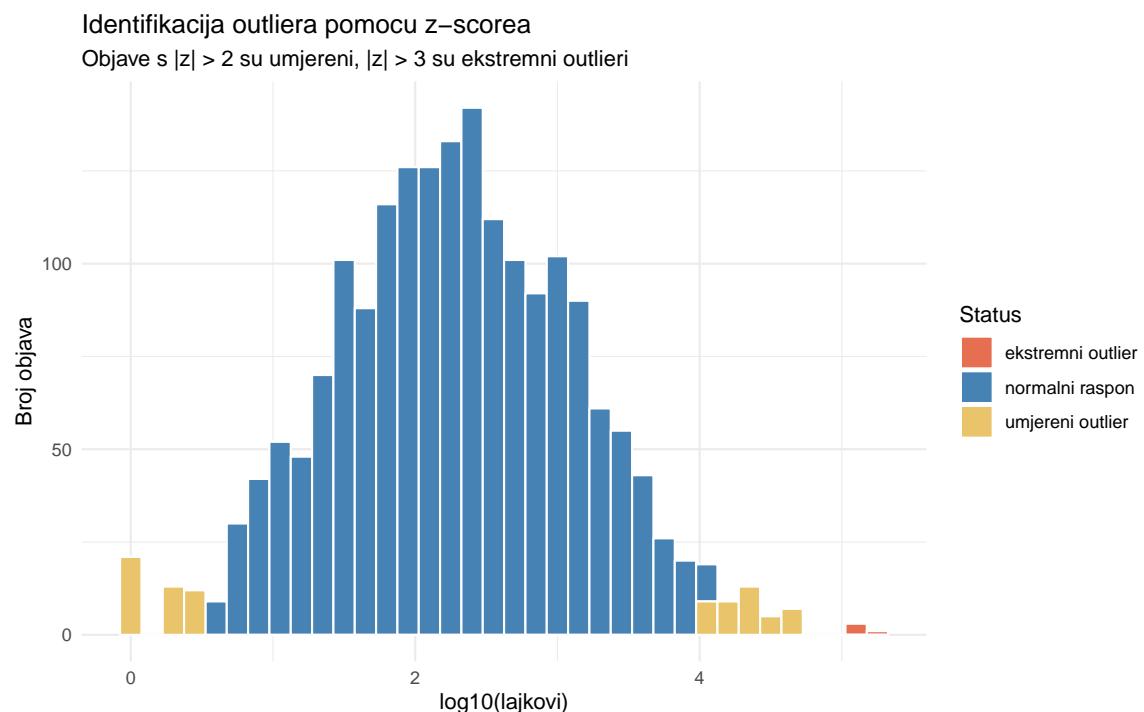
posts_analiza |>
  count(outlier_status) |>
  mutate(udio = round(n / sum(n), 3))

# 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%?
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

```
# 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

```
# 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.

```
# 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 slikा       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 prepostavljaju 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?

```

# 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

```

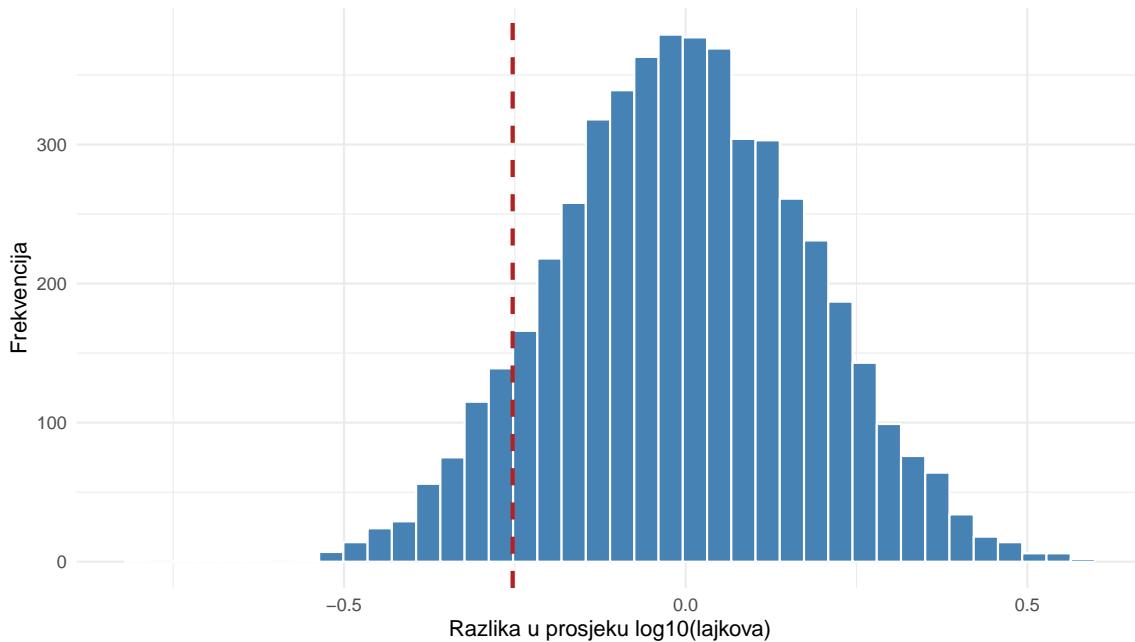
# Koliko je ova razlika neobična? Simulirajmo!
sim_razlike <- replicate(5000, {
  sve <- c(uzorak_tt, uzorak_ig)
  pomjesano <- sample(sve)
  mean(pomjesano[1:50]) - mean(pomjesano[51:100])
})

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()

```

Je li razlika u angažmanu između TikToka i Instagrama slučajnost?

Distribucija razlika pod nultom hipotezom (simulacija permutacijom)



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 $P(\text{ne } A) = 1 - P(A)$, zbrajanje za isključive $P(A \text{ ili } B) = P(A) + P(B)$, množenje za nezavisne $P(A \text{ i } B) = P(A) \times P(B)$. 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: `dbinom()`, `pbinom()`, `rbinom()`.

5. Normalna distribucija je definirana prosjekom μ i standardnom devijacijom σ . Pravilo 68-95-99.7 daje postotak podataka unutar 1, 2 i 3 SD od prosjeka.
6. Z-score $z = (x - \mu) / \sigma$ izražava koliko je vrijednost udaljena od prosjeka u jedinicama SD. Omogućuje usporedbu varijabli na različitim skalamama.
7. R funkcije za normalnu: `dnorm()` za gustoću, `pnorm()` za kumulativnu vjerojatnost ($P(X \leq x)$), `qnorm()` za kvantile (obrnuto od `pnorm`), `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 skalamama.
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, mjerenu 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:

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 zadatu 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.
Komplementarno pravilo	$P(\text{ne } A) = 1 - P(A)$.

Pojam	Objašnjenje
Pravilo zbrajanja	Za isključive: $P(A \text{ ili } B) = P(A) + P(B)$. Za neisključive: oduzeti $P(A \text{ i } B)$.
Pravilo množenja	Za nezavisne: $P(A \text{ i } B) = P(A) \times P(B)$.
Međusobno isključivi	Događaji koji se ne mogu dogoditi istovremeno.
Nezavisni događaji	Jedan ne utječe na vjerojatnost drugoga.
Uvjetna vjerojatnost	$P(A B) = P(A \text{ i } B) / P(B)$. 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 μ i standardnom devijacijom σ . Najvažnija distribucija u statistici.
Standardna normalna	Normalna distribucija sa $\mu = 0$ i $\sigma = 1$. Piše se $N(0, 1)$.
Pravilo 68-95-99.7	U normalnoj distribuciji, 68% podataka je unutar $\pm 1\sigma$, 95% unutar $\pm 2\sigma$, 99.7% unutar $\pm 3\sigma$ od prosjeka.
Z-score	Standardizirani rezultat: $z = (x - \mu) / \sigma$. Izražava udaljenost od prosjeka u jedinicama SD.
<code>scale()</code>	R funkcija koja izračunava z-score (oduzima prosjek, dijeli s SD).
<code>dbinom()</code>	Točna vjerojatnost binomne distribucije.
<code>pbinom()</code>	Kumulativna vjerojatnost binomne distribucije $P(X \leq q)$.
<code>rbinom()</code>	Generiranje slučajnih uzoraka iz binomne distribucije.
<code>dnorm()</code>	Gustoća normalne distribucije u zadanoj točki.
<code>pnorm()</code>	Kumulativna vjerojatnost normalne distribucije $P(X \leq x)$.
<code>qnorm()</code>	Kvantili normalne distribucije (obrnuta funkcija od pnorm).
<code>rnorm()</code>	Generiranje slučajnih uzoraka iz normalne distribucije.
<code>set.seed()</code>	Fiksira generator slučajnih brojeva za ponovljivost simulacija.
QQ-plot	Graf koji uspoređuje kvantile podataka s teorijskim kvantilima. Služi za provjeru normalnosti.

Pojam	Objašnjenje
<code>stat_qq()</code>	ggplot2 funkcija za QQ-plot. Kombinira se s <code>stat_qq_line()</code> .
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 nullom hipotezom.

9 Tjedan 9: Uzorkovanje, procjena i intervali pouzdanosti

Kako iz dijela saznati nešto o cijelini

```
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: 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 uzorka.

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: μ (prosjek), σ (standardna devijacija), ili p (proporcija).

Uzorak je podskup populacije koji zaista mjerimo. Mjere uzorka zovemo **statistike** i označavamo ih latinskim slovima: \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 neizbjegna 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

```
cat("Uzorak 3 (n=100):", u3, "\n")
```

Uzorak 3 (n=100): 4.8

Svaki uzorak daje malo drugačiji prosjek. Ovo je normalno i neizbjegno. Pitanje je: koliko ti prosjeci variraju? I kako ta varijacija ovisi o veličini uzorka?

9.5 Distribucija uzorkovanja

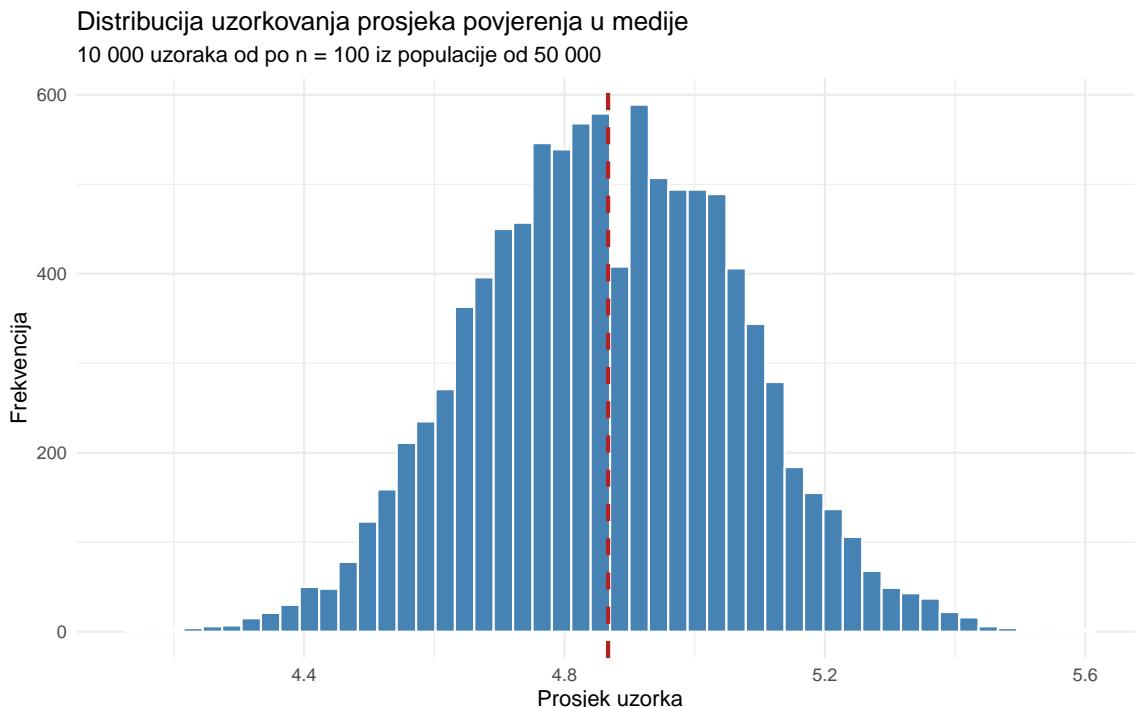
Distribucija uzorkovanja je distribucija statistike (npr. prosjeka) kroz mnogo ponovljenih uzoraka. Zamislite da uzimate 10 000 uzoraka od po 100 osoba iz populacije. Svaki uzorak daje jedan prosjek. Distribucija tih 10 000 prosjeka je distribucija uzorkovanja.

```
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: 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: 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: on 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 σ standardna devijacija populacije, a n veličina uzorka. U praksi ne znamo σ , 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 $SE = s/\sqrt{n}$ dobro procjenjuje stvarnu varijabilnost prosjeka uzoraka.

9.6.1 Veličina uzorka i preciznost

Formula $SE = \sqrt{n}$ odmah govori nešto fundamentalno: preciznost procjene raste s korijenom veličine uzorka, ne linearno.

```
set.seed(42)

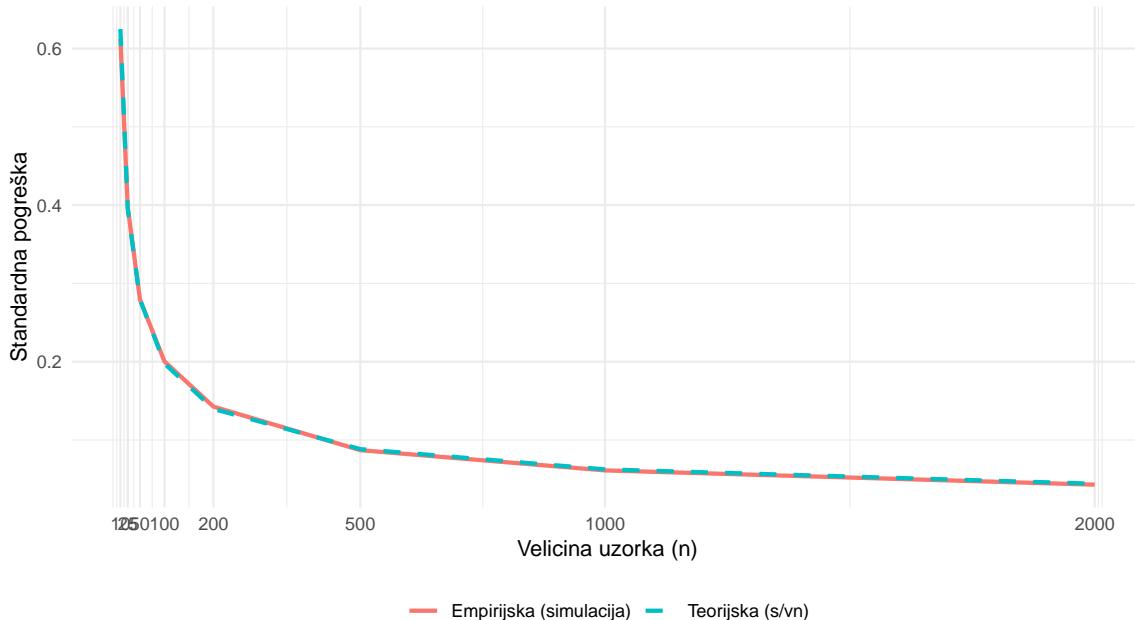
velicine <- c(10, 25, 50, 100, 200, 500, 1000, 2000)

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 (\sqrt{n})", linewidth = 1, linetype = "dashed")) +
  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")
```

Standardna pogreška pada s veličinom uzorka

Ali zakon opadajućih prinosa: od 100 do 1000 nije 10x preciznije



Pad je strm na početku (od $n=10$ do $n=100$ je ogromno poboljšanje) ali se postepeno usporava. Da biste prepolovili SE, morate učetverostručiti uzorak. Ovo objašnjava zašto su anketni uzorci obično između 500 i 2000: 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 opravдан.

9.7 Centralni granični teorem

Centralni granični teorem (Central Limit Theorem, CLT) je najvažniji teorem u cijeloj statistici. Kaže sljedeće:

Distribucija uzorkovanja prosjeka je približno normalna, neovisno o obliku izvorne distribucije, pod uvjetom da je uzorak dovoljno velik.

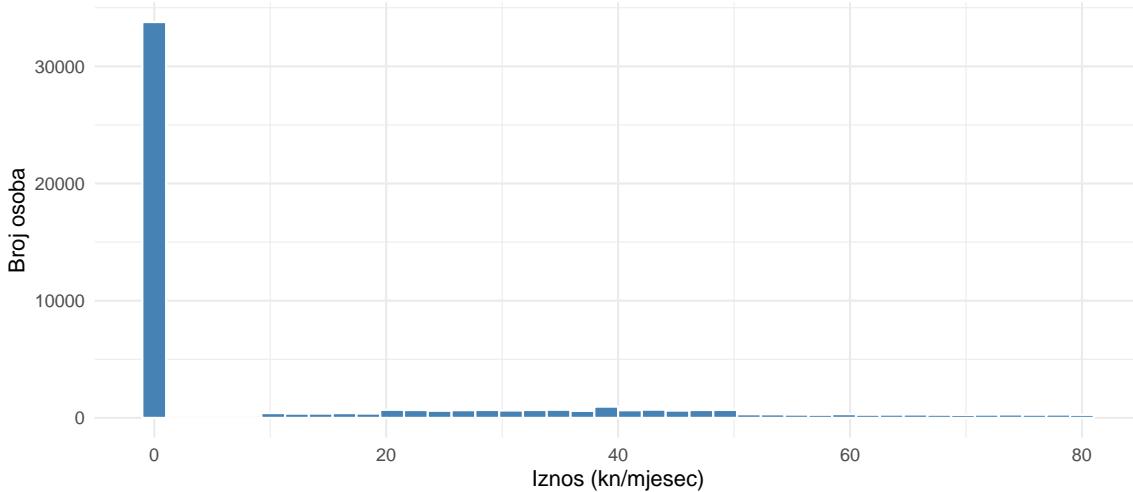
Ovo je izuzetno moćno jer znači da možemo koristiti normalnu distribuciju za donošenje zaključaka o prosjecima čak i kad izvorna varijabla nije normalna.

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()
```

Distribucija spremnosti za placanje vijesti (populacija)

Jako iskrivljena: mnogo nula, dugacak desni rep



Ovo definitivno nije normalna distribucija. Većina ljudi nije spremna platiti ništa, a oni koji jesu spremni plaćaju različite iznose.

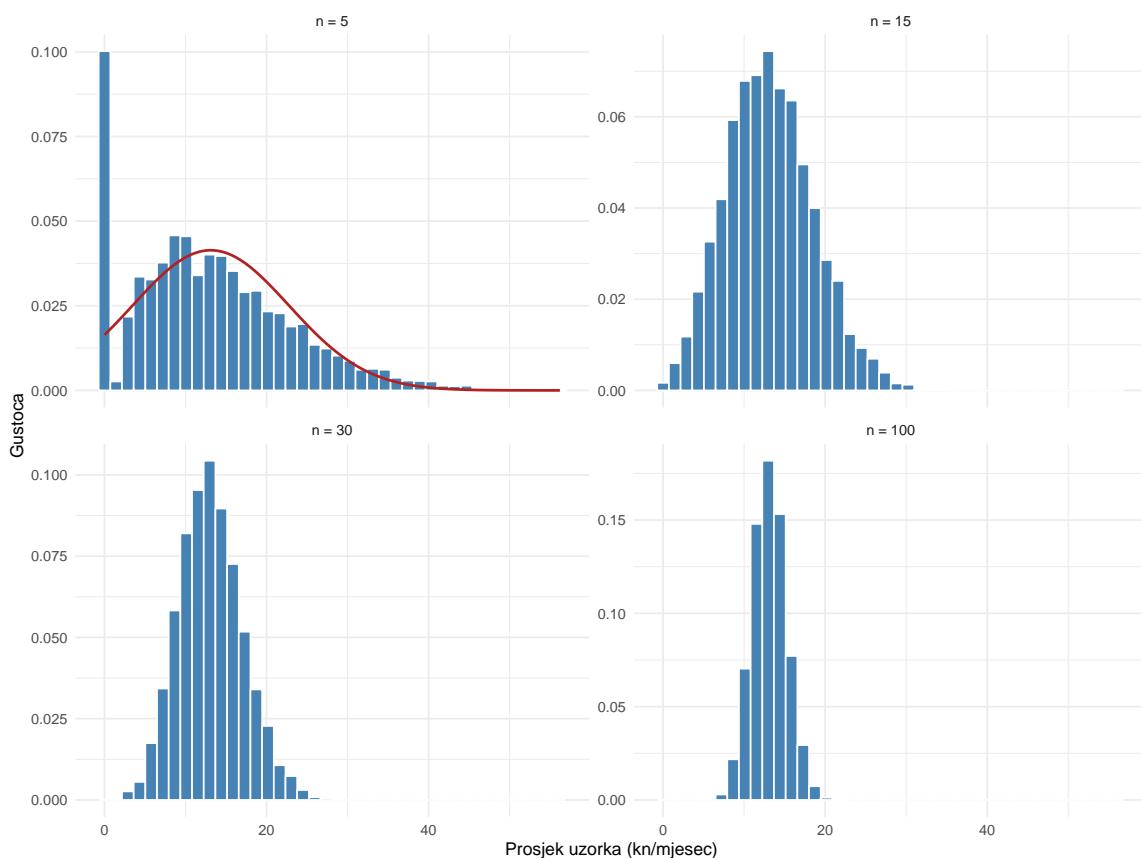
```
set.seed(42)

# 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 = "Prosječak uzorka (kn/mjesec)",
    y = "Gustoća"
) +
```

```
theme_minimal()
```

Centralni granicni teorem na djelu
Što veci uzorak, to normalnija distribucija uzorkovanja



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 prepostavlja da je distribucija prosjeka približno normalna. Ne prepostavlja 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**: svaka osoba u populaciji ima jednaku šansu biti odabrana. U praksi, ova pretpostavka je često narušena, i to 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 mlađi (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
slučajni_uzorak <- pop |> slice_sample(n = 100)

tibble(
  izvor = c("Populacija", "Slučajni uzorak (n=100)", "Pristrani uzorak (n=100)"),
  prosjek_trust = c(
    round(mean(pop$media_trust), 2),
    round(mean(slučajni_uzorak$media_trust), 2),
    round(mean(pristrani_uzorak$media_trust), 2)
  ),
  udio_portal = c(
    round(mean(pop$primary_news_source == "portal"), 3),
    round(mean(slučajni_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: 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 neproporcionalno 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$primary_news_source == "portal") * 100, 1),
  "Udio društvene mreže", round(mean(pop$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.

```
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)
```

```
cat("Populacijski udio portala:", round(p_populacija, 3), "\n")
```

Populacijski udio portala: 0.304

```
cat("Procjena iz uzorka (n=500):", round(p_hat, 3), "\n")
```

Procjena iz uzorka (n=500): 0.318

```
cat("SE proporcije:", round(se_p, 3), "\n")
```

SE proporcije: 0.021

```
cat("Margina pogreške (95%):", round(1.96 * se_p, 3), "\n")
```

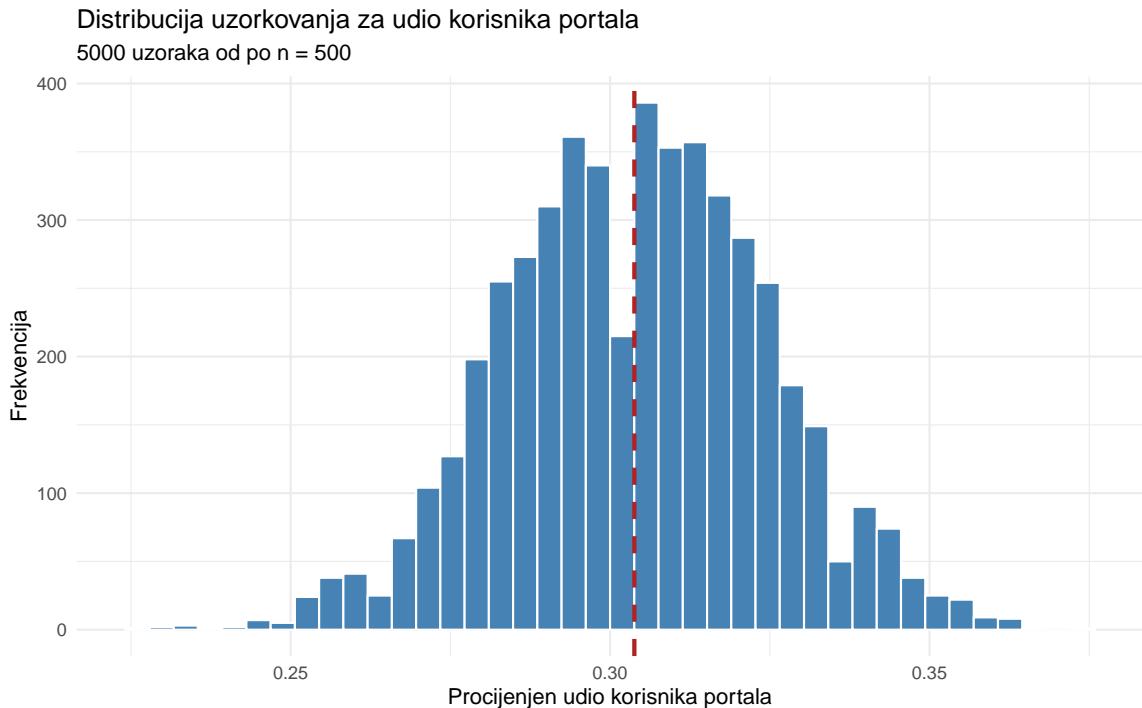
Margina pogreške (95%): 0.041

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%.

```
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)
```

```

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

```

cat("SE:", round(se, 3), "\n")

```

SE: 0.145

```

cat("95% CI: [", round(ci_lower, 2), ",",
    round(ci_upper, 2), "] \n")

```

95% CI: [4.79 , 5.35]

```

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: 5% intervala iz ponovljenih uzoraka neće pokriti pravi parametar. Zato se zove 95% interval, ne 100%.

9.10.1 Vizualizacija: 100 intervala pouzdanosti

```

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
  )
}

```

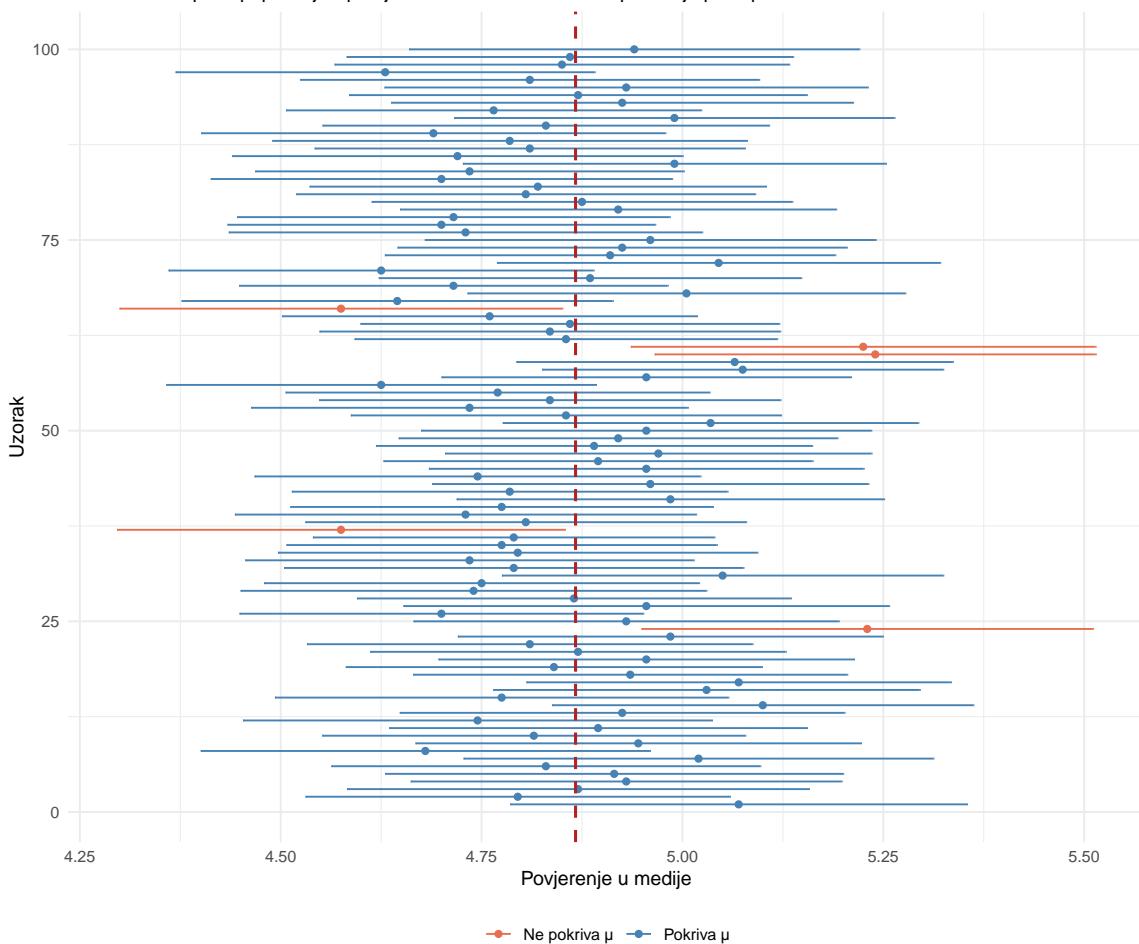
```
})

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 pokrivaju",
    x = "Povjerenje u medije",
    y = "Uzorak",
    color = NULL
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

100 intervala pouzdanosti (95%) iz 100 razlicitih uzorka
 Crvena crta = pravi populacijski prosjek. Narancasti intervali ne pokrivaju pravi parametar.



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 je: “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 fiksni 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 $SE = s/\sqrt{n}$ i da 95% interval pouzdanosti pokriva prosjek $\pm 1.96 \times SE$. U ovom dijelu prelazimo na alate koji se koriste u praksi: t-distribuciju, funkciju `t.test()` i planiranje veličine uzorka.

9.11 Od z do t: mali uzorci

Do sada smo koristili $z = 1.96$ za 95% CI. Ovo 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, ova dodatna nesigurnost znači da trebamo širi interval.

t-distribucija rješava ovaj problem. Izgleda poput normalne distribucije, ali ima deblje repove (veću 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)

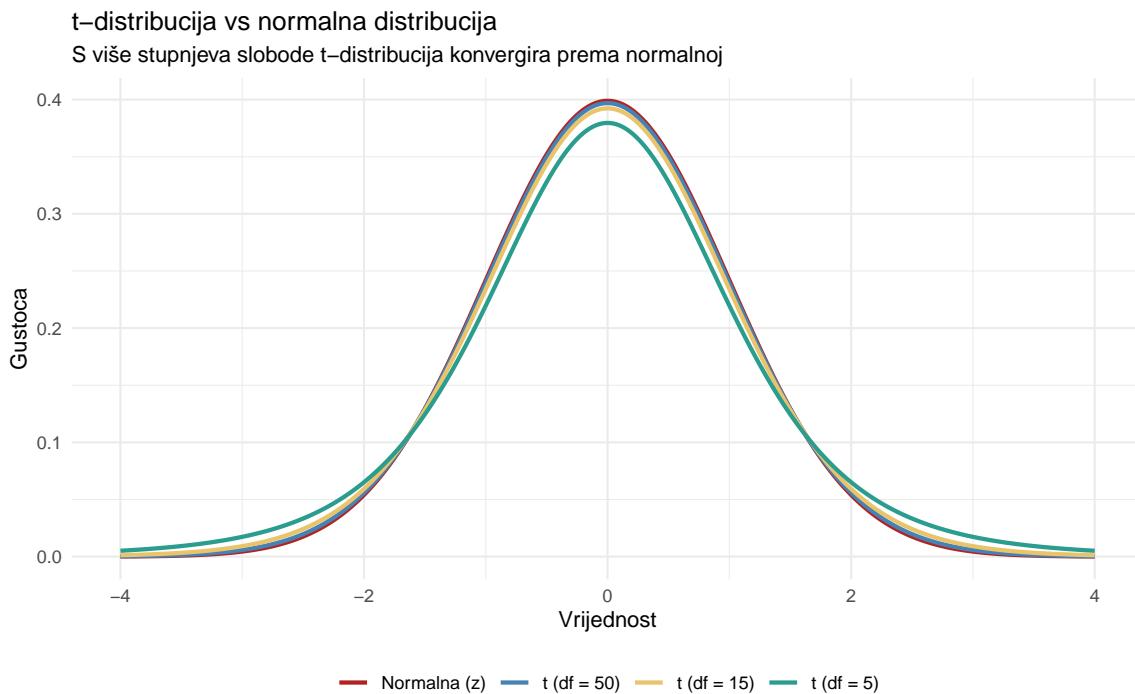
t_usporedba <- tibble(x = x) |>
  mutate(
    `Normalna (z)` = dnorm(x),
    `t (df = 5)` = dt(x, df = 5),
    `t (df = 15)` = dt(x, df = 15),
    `t (df = 50)` = dt(x, df = 50)
  ) |>
  pivot_longer(-x, names_to = "distribucija", values_to = "gustoca") |>
  mutate(distribucija = fct_relevel(distribucija,
    "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 $df = 5$ (uzorak od 6), t-distribucija ima znatno deblje repove od normalne. S $df = 50$, 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 ω (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).

```

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`.

```

# Pristup pojedinim elementima
cat("Prosjek uzorka:", round(rezultat$estimate, 3), "\n")

```

Prosjek uzorka: 5.07

```

cat("95% CI: [", round(rezultat$conf.int[1], 3), ",",
    round(rezultat$conf.int[2], 3), "]\n")

95% CI: [ 4.784 , 5.356 ]

cat("Stupnjevi slobode:", rezultat$parameter, "\n")

Stupnjevi slobode: 199

# Usporedba s populacijom
cat("\nProvi populacijski prosjek:", round(mean(pop$media_trust), 3), "\n")

Provi populacijski prosjek: 4.867

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.

```

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)
)

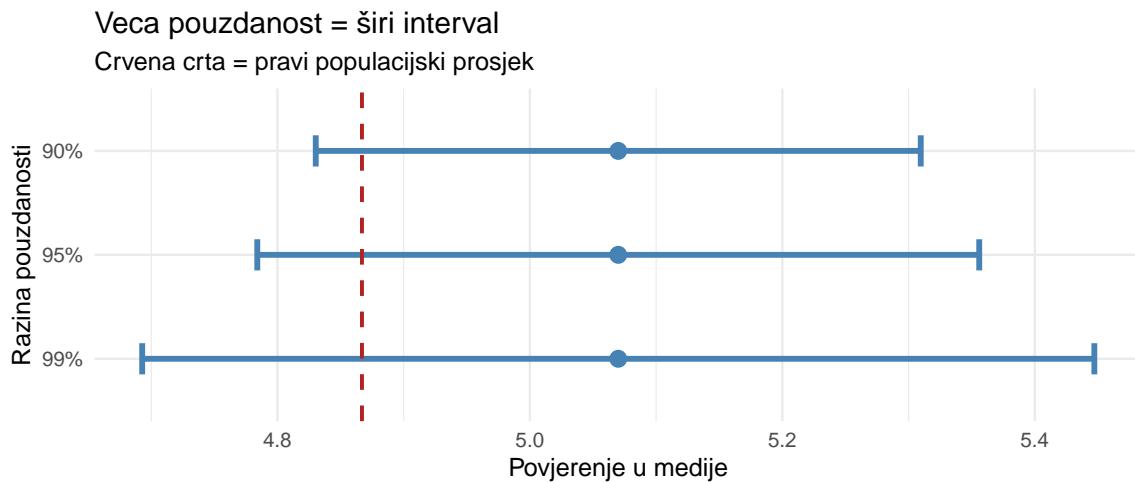
# A tibble: 3 x 4
  razina donja gornja sirina
  <chr>   <dbl>   <dbl>   <dbl>
1 90%     4.83    5.31   0.479
2 95%     4.78    5.36   0.572
3 99%     4.69    5.45   0.754

```

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 $+\infty$ (potpuno beskoristan ali potpuno siguran). 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 = "steelblue") +
  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.

```

set.seed(42)
uzorak_500 <- pop |> slice_sample(n = 500)

ci_po_izvoru <- uзорак_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"
  )

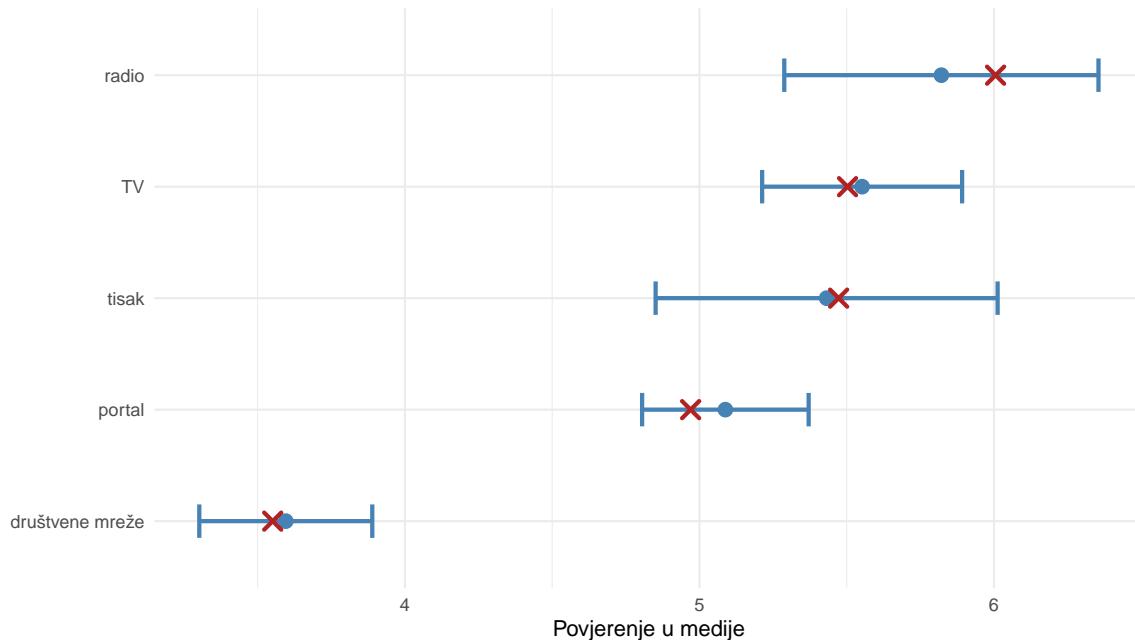
# Pravi populacijski prosjeci za usporedbu
pop_prosjeci <- pop |>
  group_by(primary_news_source) |>
  summarise(mu = mean(media_trust), .groups = "drop")

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 = "steelblue") +
  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()

```

95% CI za povjerenje u medije po primarnom izvoru vijesti

Plavi krug = prosjek uzorka. Crveni X = pravi populacijski prosjek.



Ovaj graf je izuzetno koristan za prezentaciju rezultata. Kad se intervali dviju grupa ne preklapaju, to sugerira statistički značajnu razliku (o čemu ćemo formalno govoriti 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)
ci_lo <- p_hat - 1.96 * se_p
ci_hi <- p_hat + 1.96 * se_p

cat("Procjena  $\hat{p}$ : ", round(p_hat, 3), "\n")

```

Procjena \hat{p} : 0.318

```

cat("SE:", round(se_p, 3), "\n")

```

SE: 0.021

```

cat("95% CI: [", round(ci_lo, 3), ", ", round(ci_hi, 3), "] \n")

```

95% CI: [0.277 , 0.359]

```

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.

```

# 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

```

cat("95% CI: [", round(prop_rez$conf.int[1], 3), ", ", round(prop_rez$conf.int[2], 3), "] \n")

```

95% CI: [0.278 , 0.361]

```

# CI za sve izvore vijesti
izvore <- unique(pop$primary_news_source)

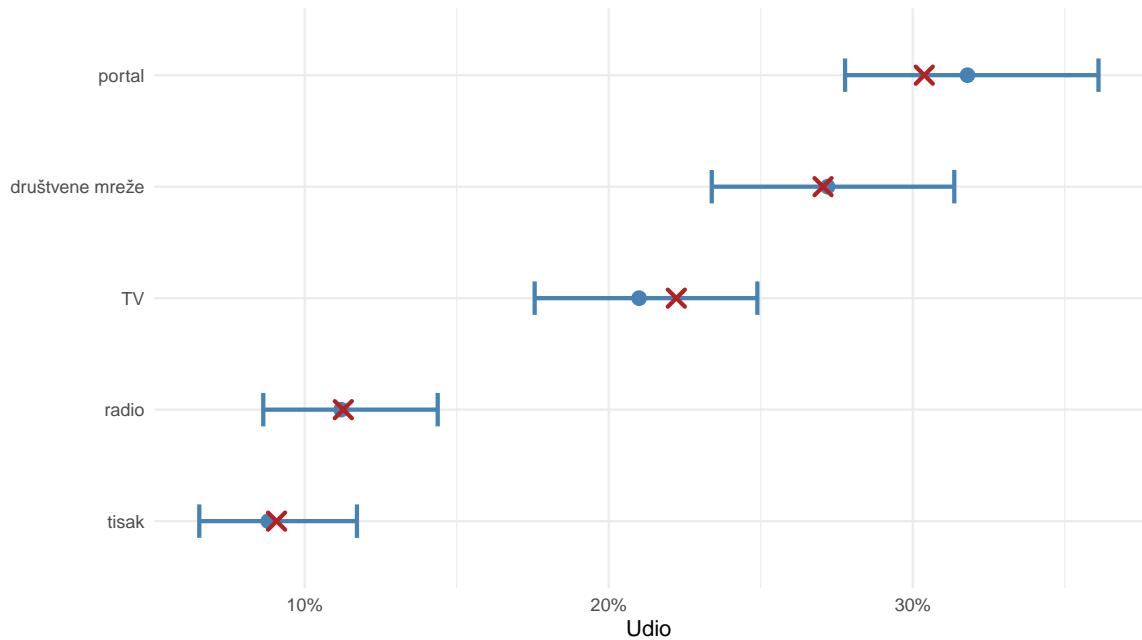
ci_izvore <- map_df(izvore, \(izvor) {
  n_da <- sum(uzorak_500$primary_news_source == izvor)
  test <- prop.test(n_da, n = 500, conf.level = 0.95)
  tibble(
    izvor = izvor,
    p_hat = test$estimate,
    ci_lo = test$conf.int[1],
    ci_hi = test$conf.int[2]
  )
}

# 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 = "steelblue") +
  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 udjeli",
    x = "Udio",
    y = NULL
  ) +
  theme_minimal()

```

95% CI za udio korisnika po primarnom izvoru vijesti
 Plavi krug = procjena iz uzorka (n=500). Crveni X = pravi populacijski udio.



Ovaj graf otkriva nešto što medijske ankete rijetko prikazuju: nesigurnost oko svakog broja. Portal i društvene mreže se ne mogu jasno razlučiti (intervali se preklapaju). TV i radio se mogu jasno razlučiti (intervali se ne preklapaju). Ovo je razlog zašto 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 $\pm 3\%$ ". 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 \hat{p} unaprijed, koristimo najgori slučaj $\hat{p} = 0.5$ (koji daje najširu marginu):

$$MoE_{max} = \frac{z^*}{2\sqrt{n}}$$

Za 95% CI: $MoE_{max} = 1.96 / (2\sqrt{n}) = 1/\sqrt{n}$.

```
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 $\pm 3.1\%$. S $n = 2000$, pada na $\pm 2.2\%$. 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 ($\hat{p} = 0.5$):

$$n = \frac{z^*^2}{4 \times MoE^2}$$

```
# 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

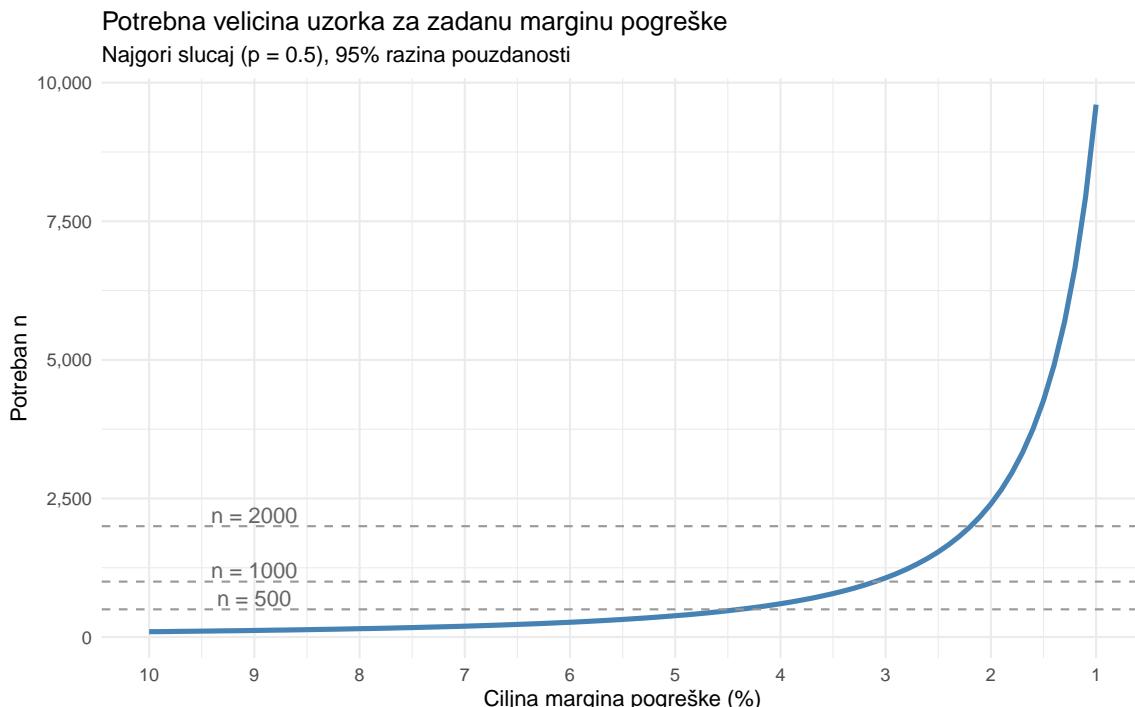
```

Za marginu od $\pm 3\%$ trebate 1068 ispitanika. Za $\pm 2\%$ trebate 2401. Za $\pm 1\%$ trebate čak 9604. Ovo ponovno potvrđuje zakon opadajućih prinosa: svako sljedeće poboljšanje je 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 zadalu marginu pogreške",
    subtitle = "Najgori slučaj ( $p = 0.5$ ), 95% razina pouzdanosti",
    x = "Ciljna marginu 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 mom kontekstu?" Ako vas zanima razlikuje li se popularnost dviju platformi za 5 postotnih bodova, trebate marginu manju od 5%, 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)))
```

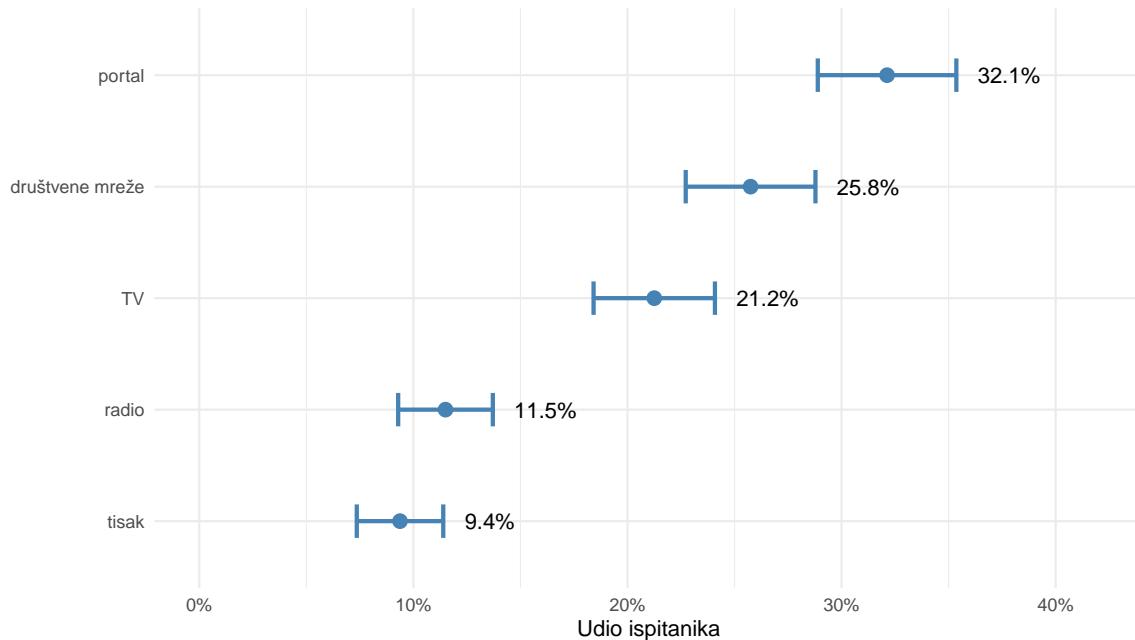
	primary_news_source	n	udio	se	moe	ci_lo	ci_hi
1	portal	257	32.1	1.7	3.2	28.9	35.4
2	društvene mreže	206	25.8	1.5	3	22.7	28.8
3	TV	170	21.2	1.4	2.8	18.4	24.1
4	radio	92	11.5	1.1	2.2	9.3	13.7
5	tisak	75	9.4	1	2	7.4	11.4

Novinar piše: "Portal je najpopularniji izvor vijesti (31%), a društvene mreže su na drugom mjestu (26%)." Tehnički 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 = "steelblue") +
  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()
```

Anketni rezultati S intervalima pouzdanosti
 n = 800. Preklapajući intervali znace da razlike nisu statisticki jasne.



9.15.1 Kontrolna lista za čitanje anketa

Kad srenete 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 margin pogreške? Ako je navode samo na dnu stranice, obratite posebnu pažnju. Jesu li razlike veće od margine pogreške? Ako su razlike manje od dvostrukog margina, 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).

```

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]

```
cat("t-test 95% CI: [", round(t_ci[1], 3), ",",
    round(t_ci[2], 3), "]\n")
```

t-test 95% CI: [4.02 , 5.26]

```
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 fleksibilnost: možemo ga koristiti za bilo koju statistiku.

```

# Bootstrap CI za MEDIJAN (za koji nema jednostavne formule za SE)
boot_medijski <- map_dbl(1:5000, \(\i) {
  uzorak_50 |>
    slice_sample(n = 50, replace = TRUE) |>
    pull(daily_media_min) |>
    median()
}

boot_ci_medijski <- quantile(boot_medijski, 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: 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

```
cat("Prosjek:", round(ukupno$estimate, 2), "\n")
```

Prosjek: 5.02

```
cat("95% CI: [", round(ukupno$conf.int[1], 2), ", ", round(ukupno$conf.int[2], 2), "] \n\n")
```

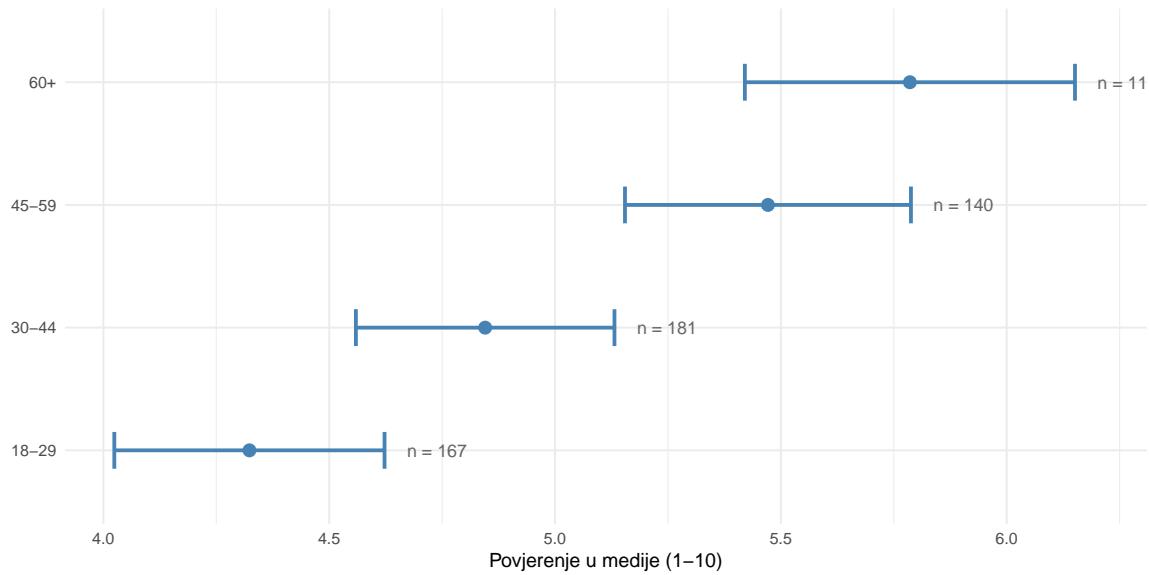
95% CI: [4.86 , 5.18]

```
# 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 = "steelblue") +
  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 = "steelblue") +
  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()
```

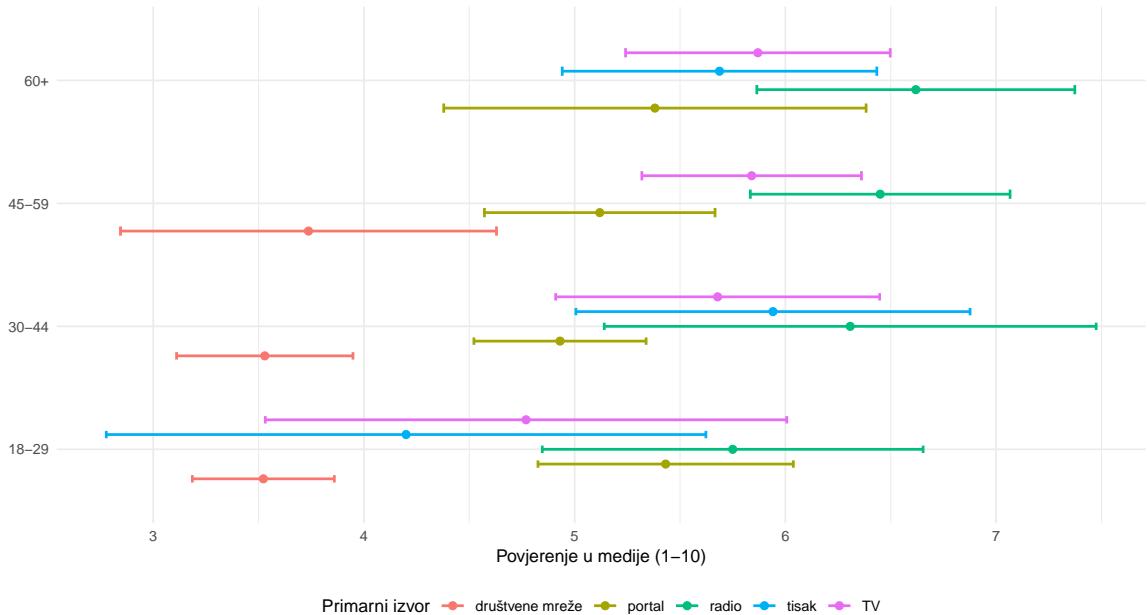
Povjerenje u medije po dobnim skupinama
95% intervali pouzdanosti. Starije skupine imaju više povjerenja.



```
# 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")
```

Povjerenje u medije: izvor vijesti x dobna skupina
Kombinacije s manje od 10 ispitanika isključene



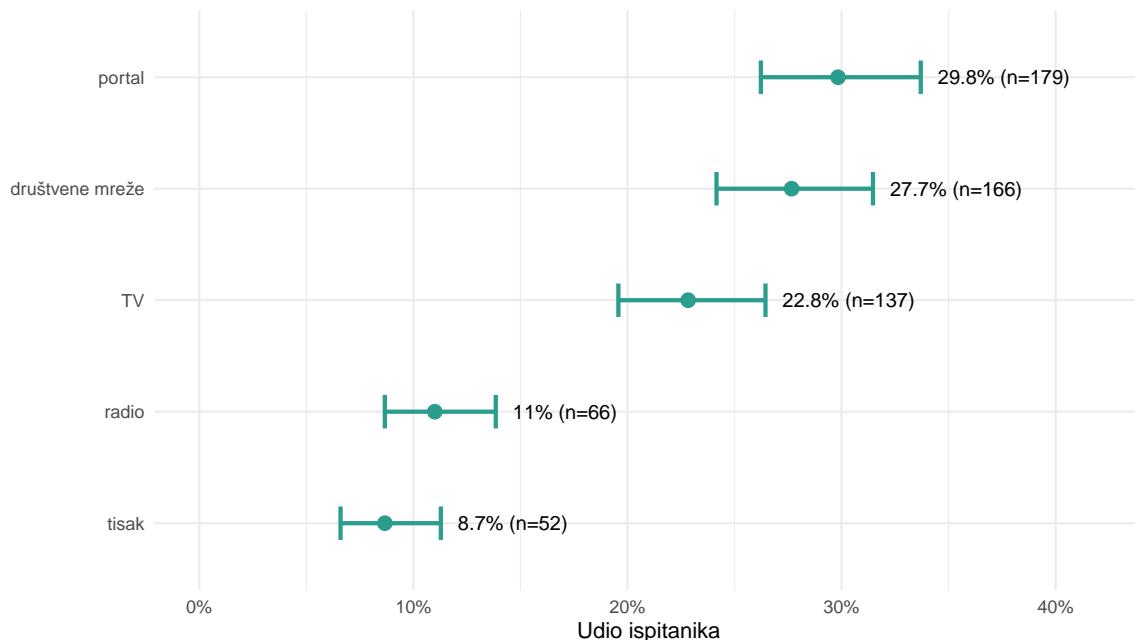
```
# Korak 4: Proporcije po izvoru vijesti s CI
prop_rezultati <- anketa |>
  count(primary_news_source) |>
  mutate(
    p_hat = n / sum(n),
    test = map2(n, sum(n), \((x, nn)) prop.test(x, nn, conf.level = 0.95)),
    ci_lo = map_dbl(test, \((t)) t$conf.int[1]),
    ci_hi = map_dbl(test, \((t)) 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 = "#2a9d8f") +
  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()
```

Preferirani izvor vijesti s intervalima pouzdanosti

Anketa na 600 ispitanika. Portali i društvene mreže statistički nerazlucivi.



```
# 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 (projek), 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 (o čemu ćemo na predavanju o t-testu).

Pogrešno: “Širok CI znači da je mjerjenje loše provedeno.” **Ispravno:** Širok CI obično znači mali uzorak ili veliku varijabilnost u podacima. To nije greška, nego realnost podataka.

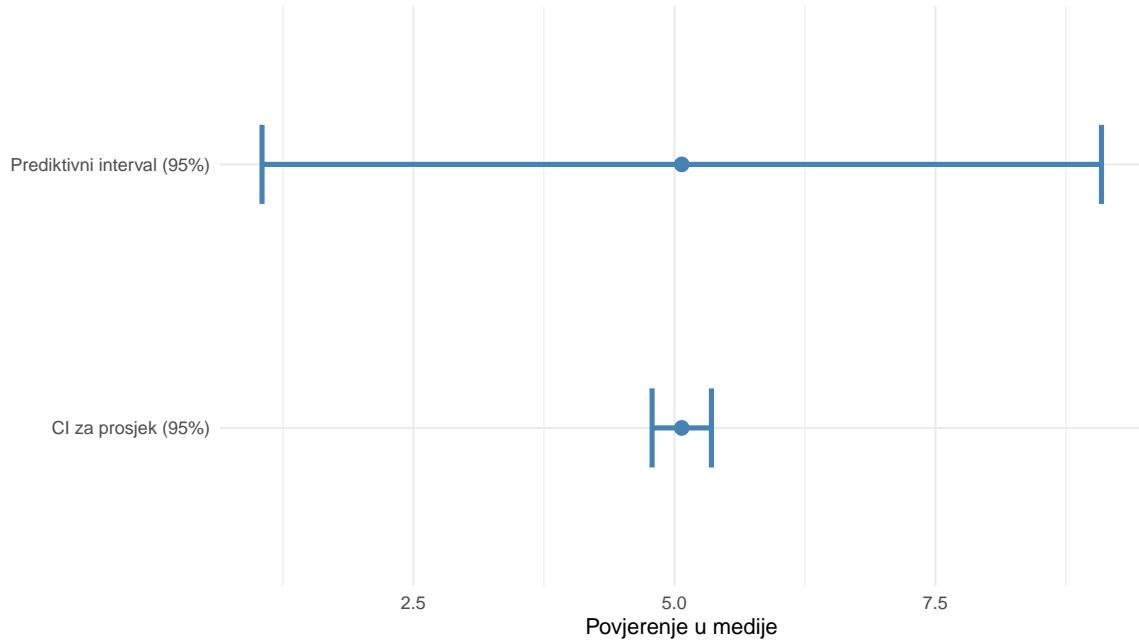
```
set.seed(42)
uzorak_200 <- pop |> slice_sample(n = 200)

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 = "steelblue") +
  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 se",
    x = "Povjerenje u medije",
    y = NULL
  ) +
  theme_minimal()
```

CI za prosjek vs prediktivni interval

CI govori gdje je populacijski prosjek. Prediktivni interval govori gdje su pojedinacna očekivanja.



CI za prosjek je uzak (± 0.28). Prediktivni interval je širok (± 3.9). Ovo su dva potpuno različita pitanja: "Gdje je pravi prosjek?" vs "Gdje će pasti sljedeće očekivanje?" Ne miješajte ih.

! Ključni zaključci

1. Populacija je cjelina o kojoj zaključujemo. Uzorak je dio koji mjerimo. Parametri (μ, σ) opisuju populaciju. Statistike (\bar{x}, s) opisuju uzorak i procjenjuju parametre.
2. Svaki uzorak daje malo drugačiju procjenu. Distribucija tih procjena kroz ponovljene uzorce naziva se distribucija uzorkovanja. Njezina standardna devijacija je standardna pogreška (SE).
3. Standardna pogreška $SE = s/\sqrt{n}$ 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: 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 $1/\sqrt{n}$. Za $n = 1000$ je $\pm 3.1\%$. Za $n = 400$ je $\pm 4.9\%$. Ovo objašnjava uobičajene veličine uzorka u anketama.
10. Kad planirate istraživanje, odredite ciljanu preciznost unaprijed i iz nje izvedite potrebnu veličinu uzorka. Formula: $n = z^{*2} \times p(1-p) / MoE^2$.
11. Bootstrap je računalna alternativa za CI kad nemamo formulu za SE željene statistike. Ideja: 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 margeine. Ako razlika između dvaju postotaka nije veća od dvostrukog margeine pogreške, zaključak o "vodstvu" jedne opcije nije opravдан.

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 $\pm 2.5\%$ 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

Wheeler, 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 (μ, σ, ρ). U praksi nepoznat.
Statistika	Mjera uzorka. Označava se latinskim slovima (x̄, s, p̂). Procjena parametra.
Pogreška uzorkovanja	Razlika između statistike i parametra. Neizbjegljiva 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/√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.
<code>t.test()</code>	R funkcija za t-test i t-interval pouzdanosti za prosjek.
<code>prop.test()</code>	R funkcija za test i CI za proporcije. Koristi Wilsonov interval.
<code>slice_sample()</code>	dplyr funkcija za slučajno uzorkovanje redova iz tibble. Argument <code>replace = TRUE</code> za bootstrap.
<code>qt()</code>	R funkcija za kritične vrijednosti t-distribucije. $qt(0.975, df)$ za 95% CI.
<code>qnorm()</code>	R funkcija za kritične vrijednosti normalne distribucije. $qnorm(0.975) = 1.96$.

10 Tjedan 10: Testiranje hipoteza

Kako donijeti odluku na temelju podataka

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći ćete

1. Formulirati nullu 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 Motivacija: A/B test na Instagramu

Radite kao analitičar u redakciji medijske kuće. Urednica vas pita: "Imamo li dokaz da carousel objave generiraju veći angažman od običnih slika?" Zadnjih nekoliko mjeseci objavljujete oba formata i sada imate podatke.

Prosjek angažmana za carousel je 10.1%. Za obične slike je 7.5%. Razlika postoji. Ali je li to prava razlika ili samo posljedica slučajnosti? Možda bismo, da ponovimo eksperiment, dobili obrnuti rezultat?

Ovo je temeljno pitanje testiranja hipoteza: je li opažena razlika dovoljno velika da isključimo slučajnost kao objašnjenje?

```
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~

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. Pitanje je: kolika je šansa da bismo vidjeli ovakvu ili veću razliku i da carousel zapravo NIJE bolji?

10.2 Logika testiranja hipoteza

Testiranje hipoteza slijedi logiku koja je slična suđenju u kaznenom pravu.

Na sudu, optuženik je nevin dok se ne dokaže krivnja. Ne trebate dokazati nevinost. Trebate dokazati krivnju izvan razumne sumnje. Ako nema dovoljno dokaza, presuda je “nedokazano” (ne “nevin”).

U statistici, početna pretpostavka je da nema učinka (nema razlike, nema veze). Ovu pretpostavku zovemo **nulta hipoteza** (H_0). Istraživač pokušava prikupiti dovoljno dokaza da odbaci nultu hipotezu u korist **alternativne hipoteze** (H_1) koja tvrdi da učinak postoji.

Za naš Instagram primjer:

H_0 : Nema razlike u angažmanu između carousel i single image formata. Svaka opažena razlika je posljedica slučajnosti.

H_1 : Postoji razlika u angažmanu između dva formata. Opažena razlika odražava stvarnu razliku u populaciji.

Matematički:

$$H_0 : \mu_{carousel} = \mu_{single}$$
$$H_1 : \mu_{carousel} \neq \mu_{single}$$

! Važna napomena

Nulta hipoteza uvijek sadrži znak jednakosti (= ili \leq ili \geq). Alternativna hipoteza sadrži znak nejednakosti (\neq ili $>$ ili $<$). Nikad obrnuto. Testiramo nultu hipotezu i tražimo dokaze protiv nje.

10.3 Od hipoteze do odluke: korak po korak

Cijeli postupak testiranja hipoteza ima pet koraka.

Korak 1: Postavite hipoteze (H_0 i H_1).

Korak 2: Odaberite razinu značajnosti (α). To je prag ispod kojeg ćete odbaciti H_0 . Konvencija je $\alpha = 0.05$ (5%).

Korak 3: Izračunajte testnu statistiku iz podataka.

Korak 4: Izračunajte p-vrijednost (vjerojatnost dobivanja ovakve ili ekstremnije testne statistike pod pretpostavkom da je H_0 istinita).

Korak 5: Donesite odluku. Ako je $p < \alpha$, odbacite H_0 . Ako je $p \geq \alpha$, ne možete odbaciti H_0 .

Krenimo korak po korak na jednostavnijem primjeru prije nego se vratimo na Instagram podatke.

10.4 Jednouzorački t-test

Počnimo s najjednostavnijim slučajem: imamo jedan uzorak i želimo testirati je li prosjek jednak nekoj zadanoj vrijednosti.

Situacija: medijska kuća tvrdi da njihovi korisnici provode prosječno 3 minute čitajući članak. Vi ste skeptični i mislite da je stvarno vrijeme čitanja drugačije. Provedete mjerjenje na uzorku 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?

10.4.1 Testna statistika

Testna statistika za jednouzorački t-test je:

$$t = \frac{\bar{x} - \mu_0}{SE} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Ovo mjeri koliko je prosjek uzorka udaljen od pretpostavljene vrijednosti, u jedinicama standardne pogreške.

```

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")

```

$\bar{x} = 2.54$

```
cat("μ₀ =", 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 μ_0 . Što veći $|t|$, to jači dokaz protiv H_0 .

10.4.2 P-vrijednost

P-vrijednost je vjerojatnost dobivanja testne statistike jednake ili ekstremnije od opažene, pod pretpostavkom da je H_0 istinita. To nije vjerojatnost da je H_0 istinita. To nije vjerojatnost da smo pogriješili. To je vjerojatnost podataka (ili ekstremnijih) pod H_0 .

```

# Dvosmjerni test: gledamo obje strane
p_value <- 2 * pt(abs(t_stat), df = n - 1, lower.tail = FALSE)

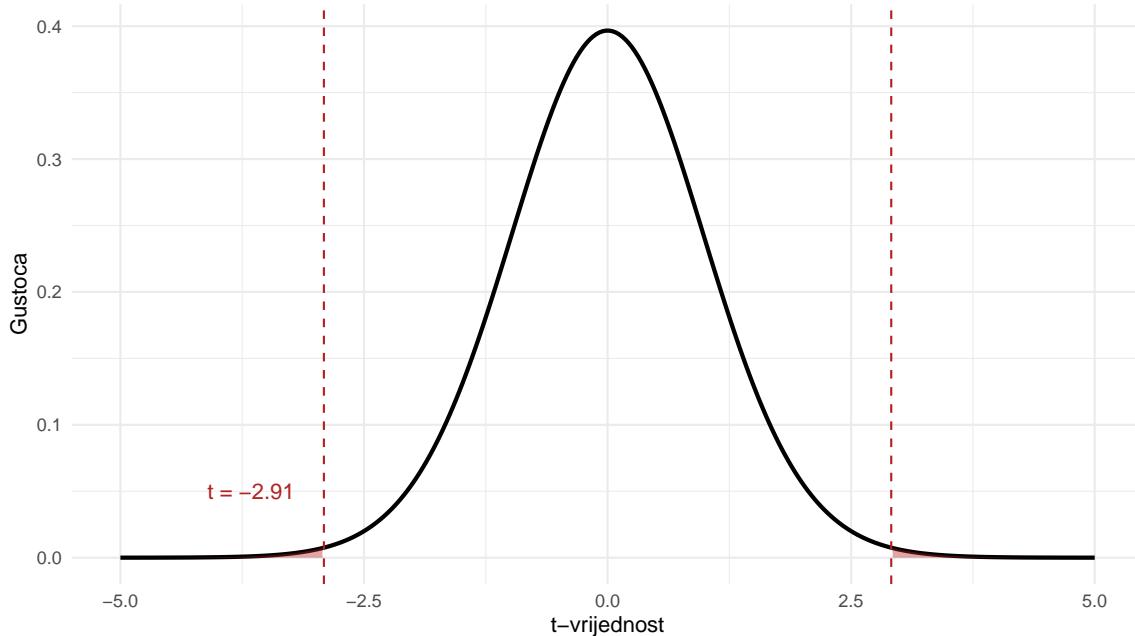
x_vals <- seq(-5, 5, length.out = 300)

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()

```

P-vrijednost je crveno osjencano područje
Dvosmjerni test. $p = 0.0056$. Ako je $p < 0.05$, odbacujemo H_0 .



```
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()` radi sve automatski

```
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 `t.test()` daje sve potrebne informacije: testnu statistiku, stupnjeve slobode, p-vrijednost, interval pouzdanosti i prosjek uzorka. P-vrijednost je ispod 0.05, što znači da imamo dovoljno dokaza da odbacimo tvrdnju medijske kuće da je prosječno vrijeme čitanja 3 minute.

💡 Veza između CI i testiranja hipoteza

Pogledajte 95% CI iz gornjeg rezultata. Ne sadrži vrijednost 3. Ovo nije slučajnost: odbacivanje H_0 na razini $\alpha = 0.05$ je ekvivalentno tome da 95% CI ne sadrži testiranu vrijednost 3. Ovo su dva načina gledanja na isti problem.

10.5 Dvosmjerni vs jednosmjerni test

U prethodnom primjeru koristili smo **dvosmjerni test** (two-tailed): testirali smo je li prosjek različit od 3, u bilo kojem smjeru. Hipoteze su bile $H_0: \mu = 3$ vs $H_1: \mu \neq 3$.

Ponekad unaprijed znamo smjer. Ako Instagram tim očekuje da su carousel objave BOLJE (ne samo različite), možemo koristiti **jednosmjerni test** (one-tailed): $H_0: \mu_{\text{carousel}} \leq \mu_{\text{single}}$ vs $H_1: \mu_{\text{carousel}} > \mu_{\text{single}}$.

```
# 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.

Upozorenje

Jednosmjerni test koristite SAMO ako ste smjer hipoteze odredili PRIJE nego ste pogledali podatke. Ako pogledate podatke, vidite da je prosjek manji od 3 i onda odlučite testirati “je li manji”, to je pristranost istraživača. Kad ste u sumnji, koristite dvosmjerni test. Većina publiciranih istraživanja koristi dvosmjerne testove.

10.6 Dvouzorački t-test: Instagram A/B test

Sada se vraćamo na naš motivacijski primjer. Želimo testirati razlikuje li se angažman između carousel i single image objava. Ovo zahtijeva **dvouzorački t-test** (two-sample t-test) jer uspoređujemo prosjeke dviju nezavisnih skupina.

$$H_0 : \mu_{carousel} = \mu_{single}$$

$$H_1 : \mu_{carousel} \neq \mu_{single}$$

```

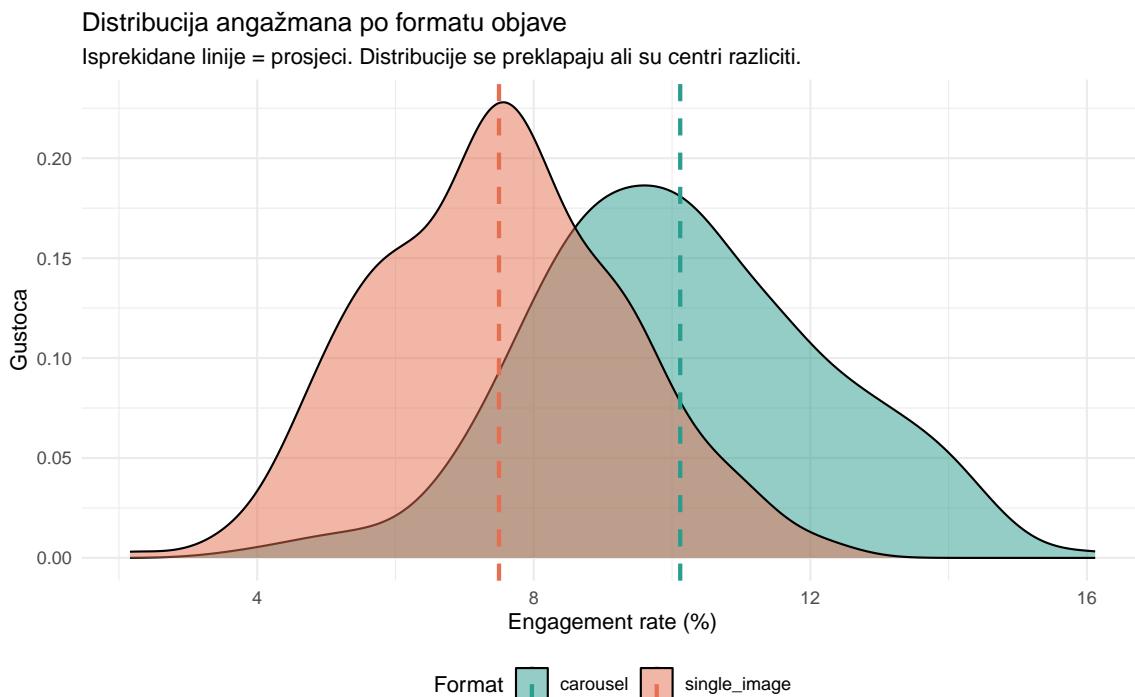
ig |>
  ggplot(aes(x = engagement_rate * 100, fill = format)) +
  geom_density(alpha = 0.5) +
  geom_vline(data = ig |> group_by(format) |> summarise(M = mean(engagement_rate) * 100),
             aes(xintercept = M, color = format), linewidth = 1, linetype = "dashed") +
  scale_fill_manual(values = 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 = projekci. 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, ali carousel objave su pomaknute udesno. Testna statistika za dvouzorački t-test mjeri razliku prosjeka u jedinicama zajedničke standardne pogreške:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{SE_{razlika}}$$

```

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 bodova")
```

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 manja od 0.05). Imamo snažne dokaze da se angažman razlikuje između dva formata. Carousel objave imaju statistički značajno viši angažman.

10.6.1 Welchov t-test

R po defaultu koristi **Welchov t-test** koji ne prepostavlja jednake varijance u dvjema skupinama. Ovo je gotovo uvijek bolja opcija od klasičnog Studentovog t-testa (koji prepostavlja jednake varijance). Welchov test je robusniji i gubi malo snage čak i kad su varijance jednake.

```
# Welchov (default)  
welch <- t.test(carousel, single, var.equal = FALSE)  
  
# Studentov (prepostavlja 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 vrlo slični u ovom slučaju. Ali Welchov test ima nerunde stupnjeve slobode (jer ih prilagodava za razliku u varijancama). Koristite Welchov test kao default jer ne zahtijeva dodatnu pretpostavku.

10.7 Simulacija: što znači p-vrijednost?

Mnogi studenti (i istraživači) krivo razumiju p-vrijednost. Simulacija pomaže izgraditi ispravnu intuiciju.

Prepostavimo da H_0 JEST istinita: carousel i single image imaju identičan angažman. Ako mnogo puta uzorkujemo i testiramo, koliko ćemo često dobiti $p < 0.05$?

```

set.seed(42)

# Simulacija: H_0 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_0 je ISTINITA. Od 10 000 testova:\n")

```

H_0 je ISTINITA. Od 10 000 testova:

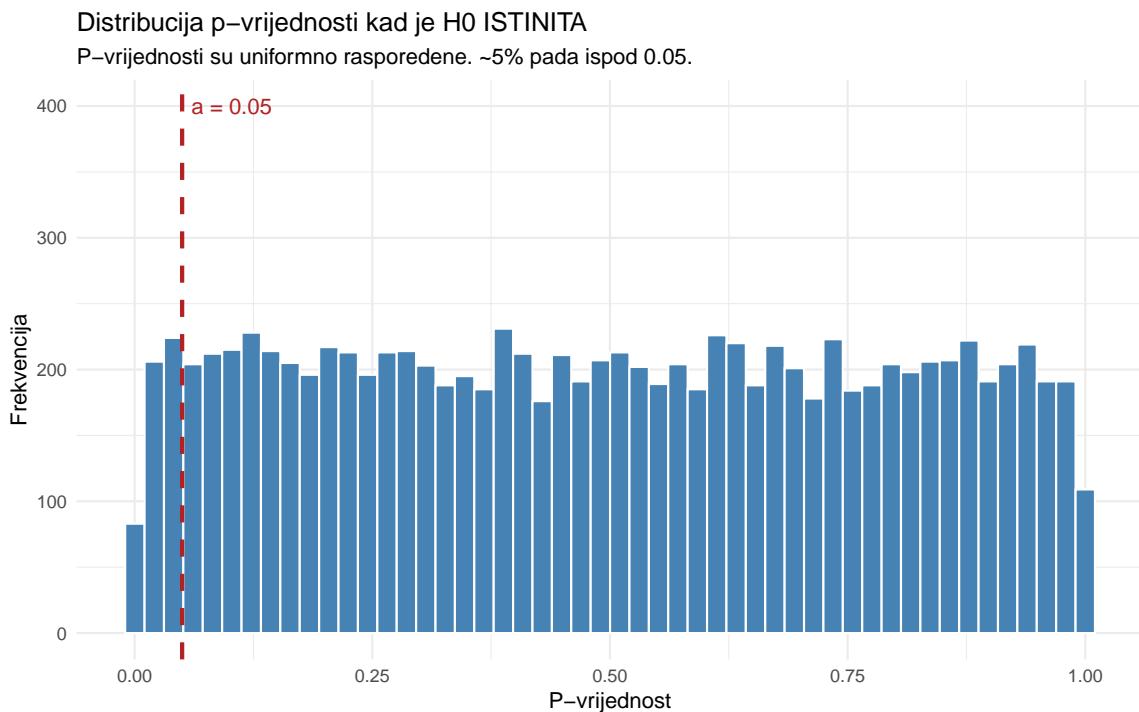
```
cat("p < 0.05:", sum(sim_p < 0.05), "(", round(mean(sim_p < 0.05) * 100, 1), "%)\n")
```

$p < 0.05: 497 (5 %)$

```
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 H0 ISTINITA",
    subtitle = "P-vrijednosti su uniformno raspoređene. ~5% pada ispod 0.05.",
    x = "P-vrijednost",
    y = "Frekvencija"
  ) +
  theme_minimal()
```



Kad je H₀ istinita, p-vrijednosti su uniformno raspoređene između 0 i 1. Točno 5% pada ispod 0.05 (po definiciji). Ovo znači da ćemo u 5% slučajeva pogrešno odbaciti H₀ čak i kad je istinita. Ovo je **greška tipa I** (false positive, lažno pozitivni rezultat).

10.7.1 Kad je H_0 istinita

```
set.seed(42)

# Simulacija:  $H_0$  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_0 je ISTINITA (razlika = 0.02). Od 10 000 testova:\n")
```

H_0 je ISTINITA (razlika = 0.02). Od 10 000 testova:

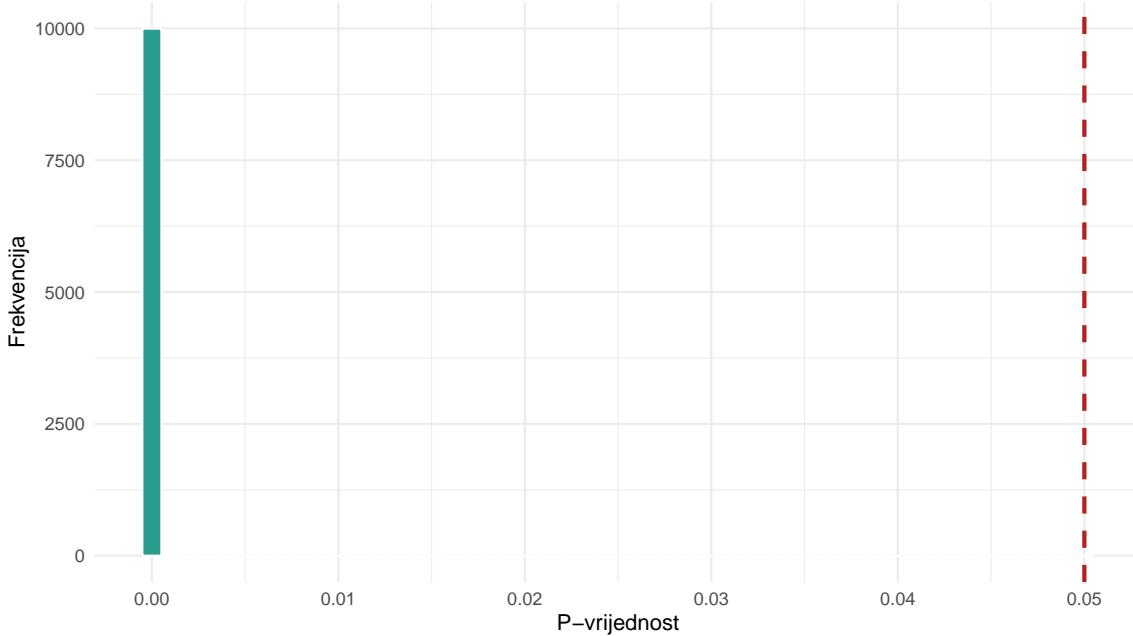
```
cat("p < 0.05:", sum(sim_p_h1 < 0.05), "(", round(mean(sim_p_h1 < 0.05) * 100, 1), "%)\n")
```

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  $H_0$  ISTINITA",
    subtitle = "Većina p-vrijednosti je blizu nule. Test je imao snagu detektirati razliku",
    x = "P-vrijednost",
    y = "Frekvencija"
  ) +
  theme_minimal()
```

Distribucija p-vrijednosti kad je H₁ ISTINITA

Vecina p-vrijednosti je blizu nule. Test je imao snagu detektirati razliku.



Kad razlika stvarno postoji, p-vrijednosti su koncentrirane blizu nule. Većina testova uspješno detektira razliku ($p < 0.05$). Ali ne svi. Postotak testova koji uspješno detektiraju pravu razliku naziva se **statistička snaga** (power). Testovi koji ne detektiraju pravu razliku ($p > 0.05$ iako razlika postoji) čine **grešku tipa II** (false negative, lažno negativni rezultat).

10.8 Greška tipa I i greška tipa II

Postoje četiri moguća ishoda testiranja hipoteza, ovisno o tome je li H_0 stvarno istinita i kakvu smo odluku donijeli.

```
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 (): Odbacimo H_0 iako je istinita. Zaključimo da razlika postoji kad je zapravo nema. Kontroliramo je postavljanjem (obično 0.05). Analogija sa suđenjem: osuditi nevinu osobu.

Greška tipa II (): Ne odbacimo H_0 iako je lažna. Propustimo pravu razliku. Ovisi o veličini uzorka, veličini učinka i α . Analogija sa suđenjem: oslobođiti krivca.

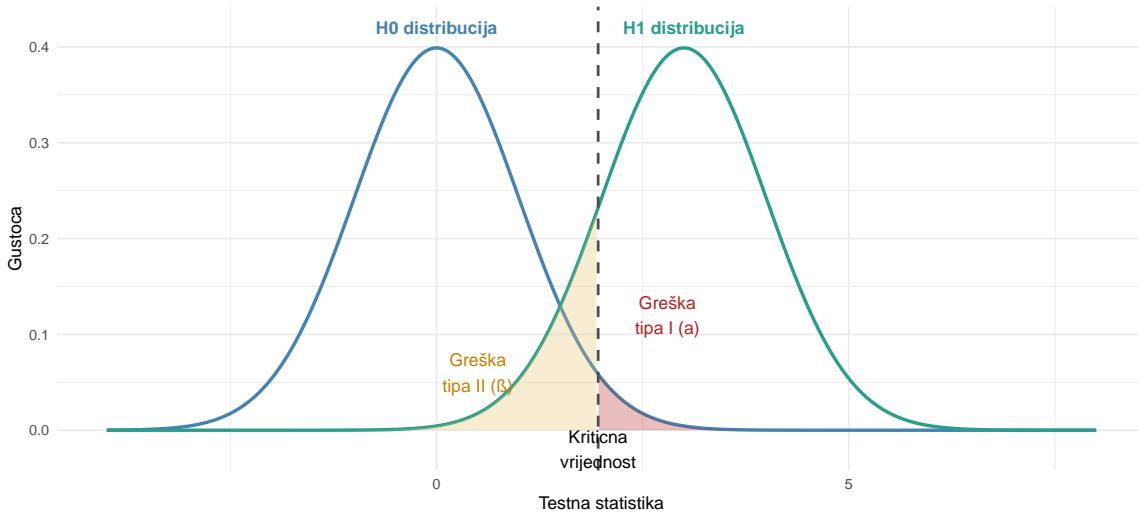
```
x <- seq(-4, 8, length.out = 500)
h0 <- dnorm(x, mean = 0, sd = 1)
h1 <- dnorm(x, mean = 3, sd = 1)

crit <- qnorm(0.975)

error_data <- tibble(x = x, H0 = h0, H1 = h1)

ggplot(error_data, aes(x = x)) +
  #  $H_0$  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) +
  #  $H_1$  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 = "H0 distribucija", color = "steelblue", fontface = "italic") +
  annotate("text", x = 3, y = 0.42, label = "H1 distribucija", color = "#2a9d8f", fontface = "italic") +
  annotate("text", x = 2.8, y = 0.12, label = "Greška\t tipa I ()", color = "firebrick") +
  annotate("text", x = 0.5, y = 0.06, label = "Greška\t tipa 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 možemo smanjiti oboje",
    x = "Testna statistika",
    y = "Gustoća"
  ) +
  theme_minimal()
```

Vizualizacija grešaka tipa I i tipa II
Crveno = lažno pozitivno. Žuto = lažno negativno. Ne možemo smanjiti oboje istovremeno.



Ovaj graf pokazuje ključan kompromis. Ako pomaknemo kritičnu vrijednost udesno (strožiji), smanjujemo crveno područje (manje lažno pozitivnih) ali povećavamo žuto područje (više lažno negativnih). Jedini način da smanjimo oboje istovremeno je povećati uzorak (što razdvaja dvije distribucije) ili imati veći učinak.

! Važna napomena

"Ne možemo odbaciti H_0 " nije isto što i " H_0 je istinita." Odsutnost dokaza nije dokaz odsutnosti. Možda razlika postoji ali je naš uzorak premalen da je detektira. Zato nikad ne kažemo "prihvaćamo H_0 " nego "ne možemo odbaciti H_0 na temelju dostupnih podataka."

10.9 P-vrijednost: što jest, a što nije

P-vrijednost je jedan od najčešće korištenih ali i najčešće pogrešno interpretiranih koncepta u statistici. Razjasnimo.

P-vrijednost JEST: Vjerovatnosc dobivanja testne statistike jednake ili ekstremnije od opažene, pod pretpostavkom da je H_0 istinita. Drugim riječima: koliko bi bili iznenadjeni ovakvim podacima kad H_0 ne bi bila istinita?

P-vrijednost NIJE:

P-vrijednost NIJE vjerovatnosc da je H_0 istinita. Ne možemo reći "postoji samo 3% šanse da nema razlike." P-vrijednost ne govori ništa o vjerovatnosti hipoteza.

P-vrijednost NIJE vjerojatnost da smo pogriješili. Mala p-vrijednost znači da su podaci neobični pod H_0 . Ne znači da smo sigurno u pravu.

P-vrijednost NIJE mjera veličine učinka. $p = 0.001$ ne znači da je učinak velik. Velik uzorak može proizvesti malu p-vrijednost za trivijalno malu razliku.

```
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) je statistički značajna. S 20 opažanja, razlika od 0.5 bodova (potencijalno važna) nije statistički značajna. Ovo pokazuje zašto p-vrijednost sama nije dovoljna. Uvijek trebamo i mjeru veličine učinka.

i Podsjetnik

U prvom dijelu naučili smo logiku testiranja hipoteza, formuliranje H_0 i H_1 , jednouzorački i dvouzorački t-test, p-vrijednost i greške tipa I i II. U ovom dijelu 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 ne govori koliko je učinak velik. Za to nam treba mjera **veličine učinka** (effect size).

Najčešća mjera za razliku dvaju prosjeka je **Cohenov d**, koji izražava razliku u jedinicama zajedničke standardne devijacije:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{pooled}}$$

gdje je s_{pooled} zajednička (pooled) standardna devijacija dviju grupa.

```
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 bodova")
```

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 Interpretacija Cohenovog d

Cohen (1988) je predložio smjernice za interpretaciju:

```

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 $d = 1.35$ je velik učinak. Carousel objave generiraju značajno viši angažman, i to u praktično važnoj mjeri. Ovo je informacija koju p-vrijednost sama ne daje.

```

# 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))

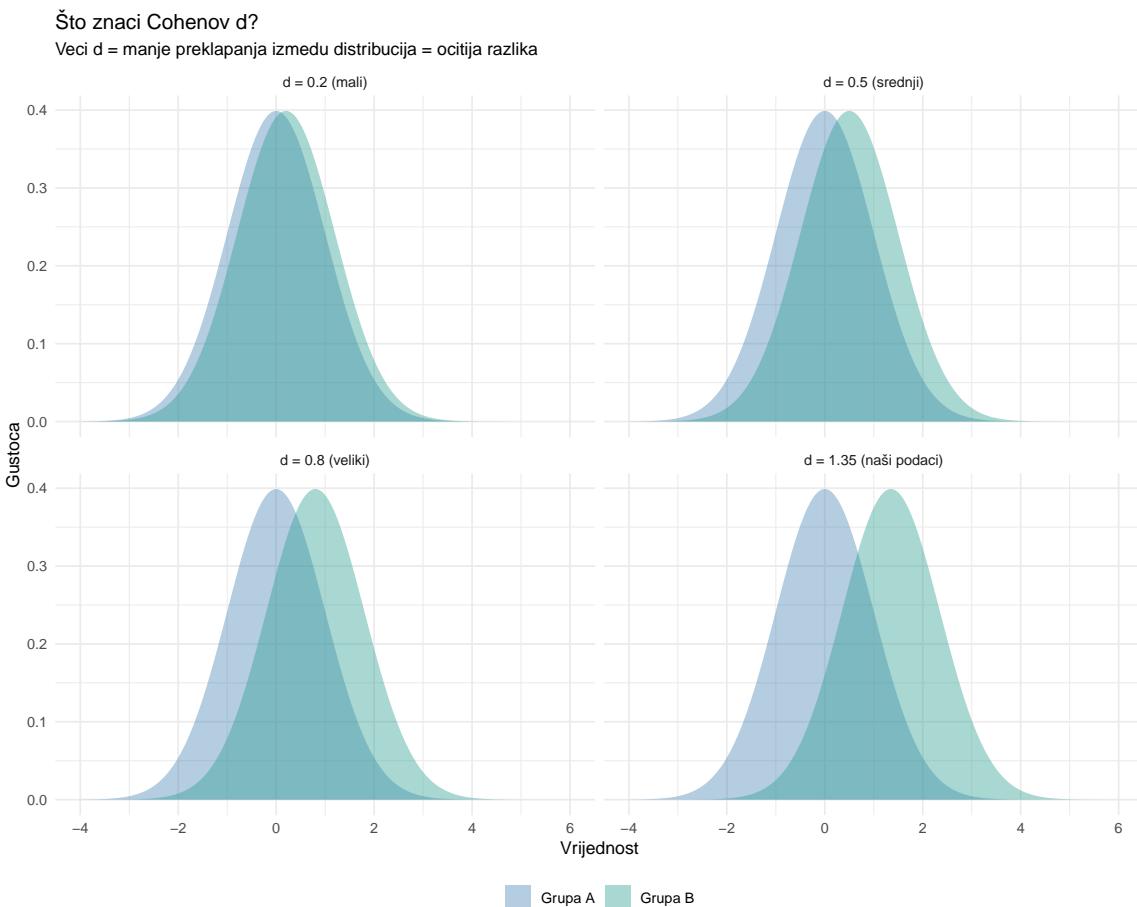
d_viz |>
  ggplot(aes(x = x, y = gustoca, fill = grupa)) +
  geom_area(alpha = 0.4, position = "identity") +
  facet_wrap(~panel, ncol = 2) +
  scale_fill_manual(values = c("Grupa_A" = "steelblue", "Grupa_B" = "#2a9d8f"),
                    labels = c("Grupa A", "Grupa B")) +
  labs(
    title = "Što 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 i razliku je teško primijetiti. S $d = 0.8$, razdvajanje je očito. Naš $d = 1.3$ pokazuje vrlo jasno razdvajanje: carousel i single image su očigledno različite kategorije po angažmanu.

💡 Praktični savjet

Uvijek izvještavajte veličinu učinka uz p-vrijednost. 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.

10.11 Statistička snaga

Statistička snaga (power) je vjerojatnost da test odbaci H_0 kad je H_1 istinita. Drugim riječima: koliko je vjerojatno da ćemo detektirati pravu razliku ako ona postoji?

Snaga ovisi o četiri faktora: veličini učinka (veći učinak je lakše detektirati), veličini uzorka (više podataka daje veću snagu), razini značajnosti (veći α daje veću snagu ali više lažno pozitivnih), i varijabilnosti podataka (manja varijabilnost daje veću snagu).

Konvencija je da snaga treba biti barem 0.80 (80%). Ovo znači da ako razlika postoji, želimo je detektirati barem u 8 od 10 pokušaja.

10.11.1 Analiza snage: koliki uzorak trebam?

Najčešća primjena analize snage je planiranje istraživanja: koliki uzorak mi treba da detektiram 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

```

Za mali učinak ($d = 0.2$) trebate čak 394 ispitanika po grupi. 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.

```

# 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))
  })
}

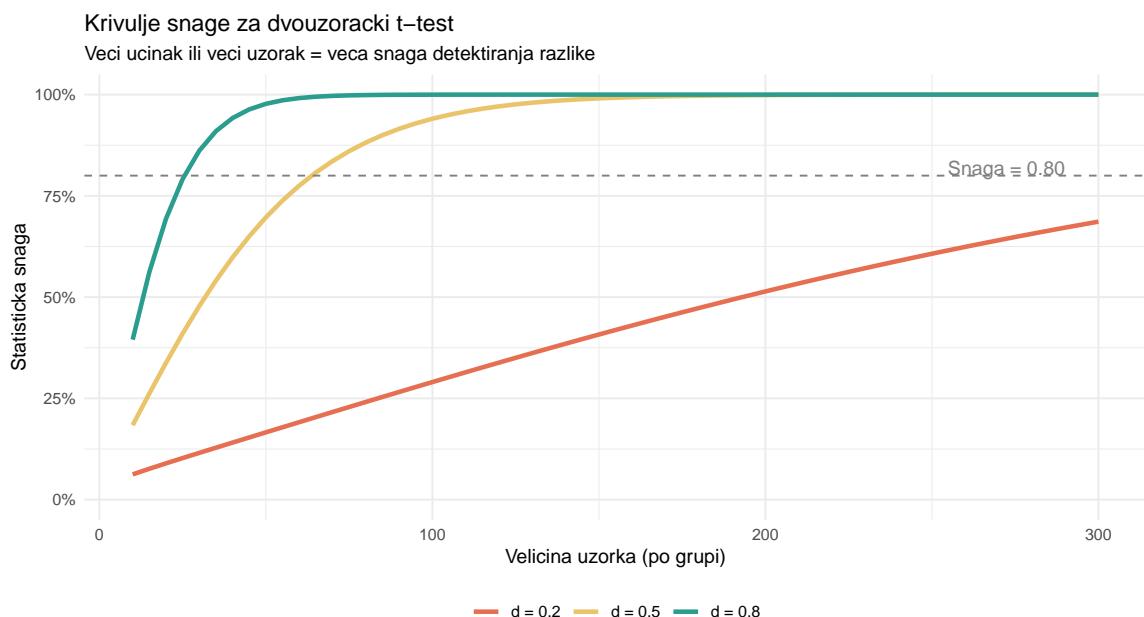
power_curves |>
  ggplot(aes(x = n, y = power, color = d)) +
  geom_line(linewidth = 1.2) +
  geom_hline(yintercept = 0.80, linetype = "dashed", color = "grey50") +
  annotate("text", x = 290, y = 0.82, label = "Snaga = 0.80", color = "grey50", hjust = 1)

```

```

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" = "#0072bc"))
  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 $n = 300$ po grupi ne dostiže 100%.
Za veliki učinak ($d = 0.8$, zelena), snaga brzo raste i s $n = 30$ 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 1.0 (100%). S ovakvom veličinom učinka i ovako velikim uzorkom, gotovo nema šanse 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

Dosad smo uspoređivali dvije **nezavisne** skupine (carousel vs single image su različite objave). Ali ponekad mjerimo istu jedinicu u dva uvjeta. Na primjer: angažman istih pratitelja prije i poslije redizajna profila, ili rating istih članaka od strane dva različita urednika.

Kad su opažanja u parovima, koristimo **upareni t-test** (paired t-test). Umjesto da uspoređujemo dva prosjeka, računamo razliku za svaki par i testiramo je li prosjek razlika različit od nule.

```

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
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

# Uporedni 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

```

```

# Usporedba: upareni vs nepareni test na istim podacima
paired_p <- t.test(urednicki_rating$urednik_B, urednicki_rating$urednik_A, paired = TRUE)$p.value
unpaired_p <- t.test(urednicki_rating$urednik_B, urednicki_rating$urednik_A, paired = FALSE)$p.value

```

```
cat("Upareni test p-vrijednost: ", round(paired_p, 5), "\n")
```

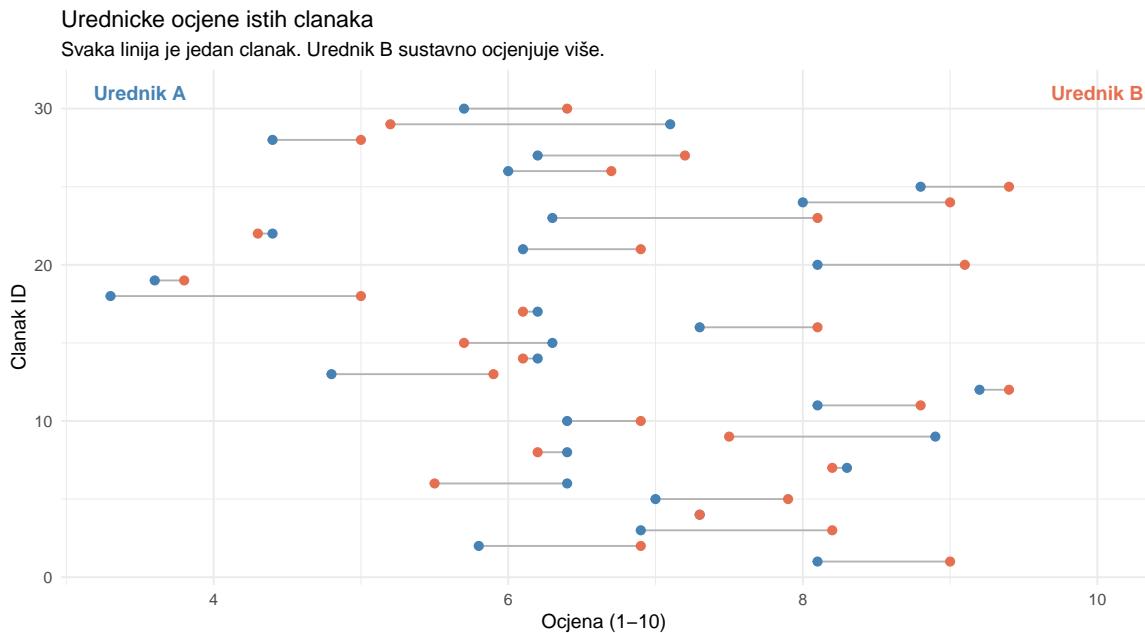
Upareni test p-vrijednost: 0.01296

```
cat("Nepareni test p-vrijednost: ", round(unpaired_p, 5), "\n")
```

Nepareni test p-vrijednost: 0.30785

Upareni test je osjetljiviji (manja p-vrijednost) jer uklanja varijabilnost između parova. Neki članci su općenito bolji, neki lošiji. Kad to kontroliramo (parenjem), preostaje samo varijabilnost u razlici između dva urednika, što je puno manja.

```
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 = "bold") +
  annotate("text", x = 10, y = 31, label = "Urednik B", color = "#e76f51", fontface = "bold") +
  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()
```



! Kada koristiti koji test?

Nepareni (nezavisni) t-test: Dvije različite skupine ispitanika. 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.

10.13 Statistička vs praktična značajnost

Statistička značajnost ($p < 0.05$) govori da razlika vjerojatno nije slučajnost. Ali ne govori je li razlika dovoljno velika da je važna u praksi. Ovo razlikovanje je ključno za donošenje odluka.

```
set.seed(42)

# Scenarij 1: Statistički značajno ali praktički bezznačajno
# 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

```
cat("Statistički značajno:", test1$p.value < 0.05, "\n")
```

Statistički značajno: FALSE

```
cat("Isplati li se redizajn? Vjerojatno ne.\n\n")
```

Isplati li se redizajn? Vjerojatno ne.

```
# 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 ===

```
cat("Razlika CTR:", round((mean(ctr_novi2) - mean(ctr_stari2)) * 100, 2), "postotnih bodova")
```

Razlika CTR: 0 postotnih bodova

```
cat("P-vrijednost:", round(test2$p.value, 4), "\n")
```

P-vrijednost: 1

```
cat("Statistički značajno:", test2$p.value < 0.05, "\n")
```

Statistički značajno: FALSE

```
cat("Zaslužuje li daljnje istraživanje? Vjerojatno da.\n")
```

Zaslužuje li daljnje istraživanje? Vjerojatno da.

Donošenje odluka na temelju samo p-vrijednosti je nepotpuno. Morate uzeti u obzir veličinu učinka, praktične posljedice (trošak redizajna vs dobitak od povećanog CTR-a), interval pouzdanosti (koliki raspon razlika je konzistentan s podacima) i kontekst (kolika je razlika "dovoljno velika" u vašoj industriji).

```
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 Potpuna analiza: Instagram A/B test izvještaj

Spojimo sve u koherentan izvještaj za urednicu. Slijedimo strukturu: opisna statistika, vizualizacija, statistički test, veličina učinka, podanalize, 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"
)

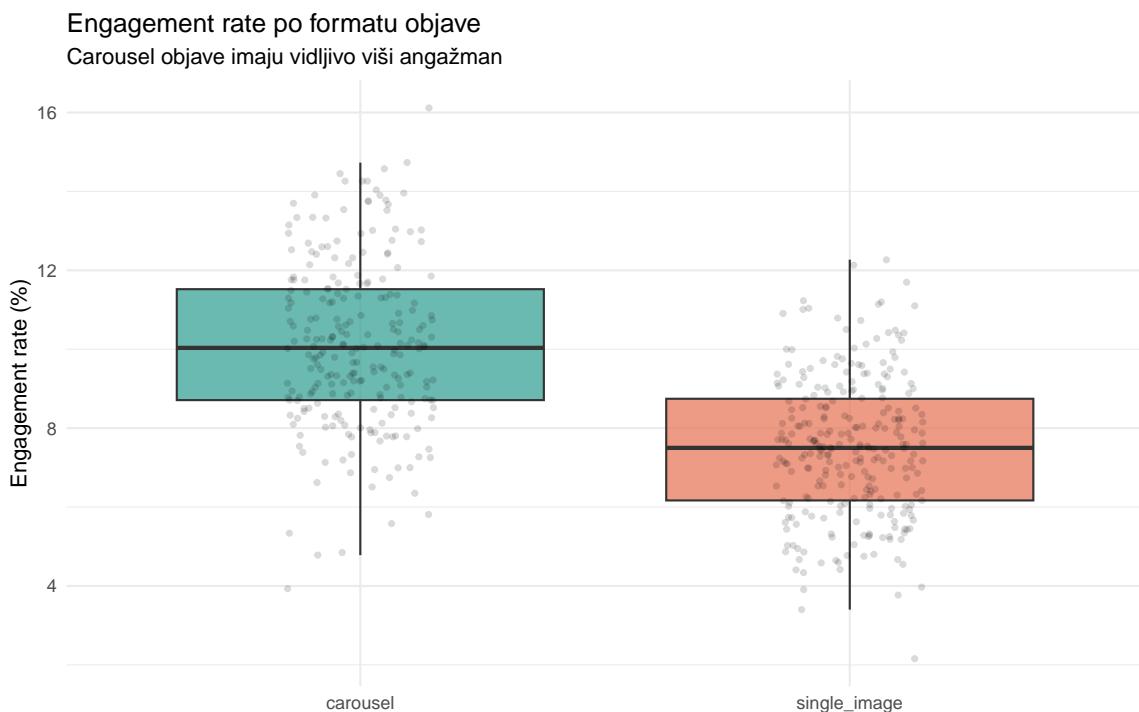
# A tibble: 2 x 8
#>   format      n M_engagement SD_engagement M_likes M_comments M_shares M_saves
#>   <chr>     <int>        <dbl>          <dbl>       <dbl>       <dbl>       <dbl>       <dbl>
#> 1 carousel    236         10.1        2.11      180        36        25        42
#> 2 single_i~   264         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

cat("==== REZULTAT DVOUZORAČKOG T-TESTA ====\n")

==== REZULTAT DVOUZORAČKOG T-TESTA ====

cat("t(", round(test_ig$parameter, 1), " ) = ", round(test_ig$statistic, 2), "\n", sep = "")

t(461.6) = 14.94

cat("p < 0.001\n")

p < 0.001

cat("Razlika prosjeka: ", round((mean(carousel) - mean(single)) * 100, 2), " postotnih bodova")

Razlika prosjeka: 2.62 postotnih bodova

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

cat("Cohenov d:", round(d_ig, 2), "(veliki učinak)\n")

Cohenov d: 1.35 (veliki učinak)

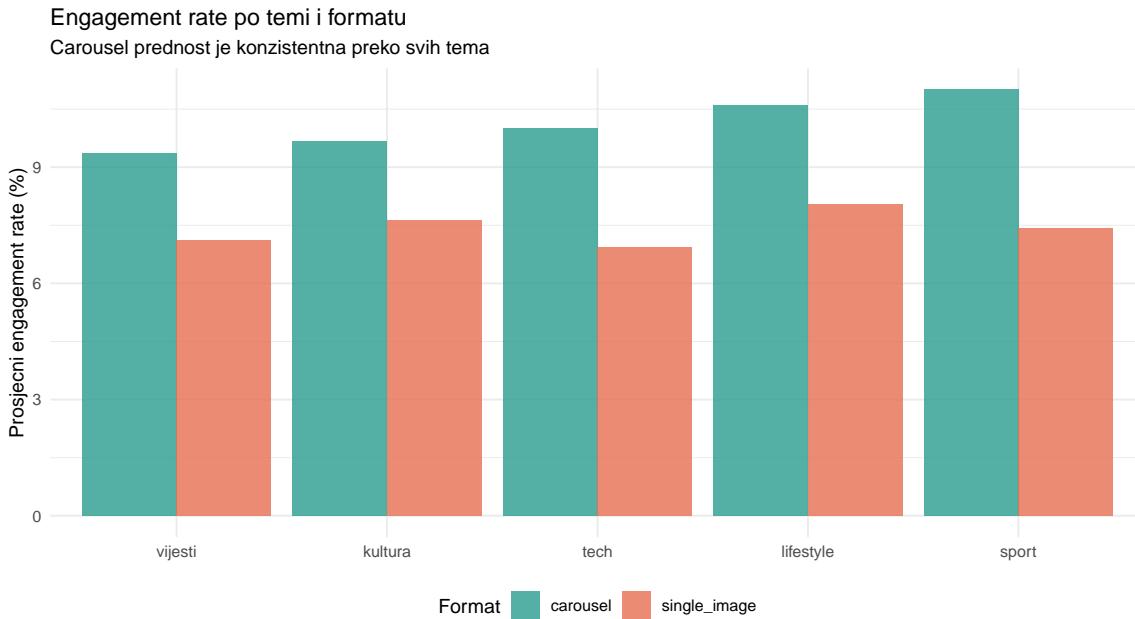
# Korak 4: Je li prednost carousela konzistentna po temama?
ig |>
  group_by(topic, format) |>
  summarise(M = mean(engagement_rate) * 100, .groups = "drop") |>
  ggplot(aes(x = fct_reorder(topic, M, .fun = max), y = M, fill = format)) +
  geom_col(position = "dodge", alpha = 0.8) +
  scale_fill_manual(values = 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"
  )

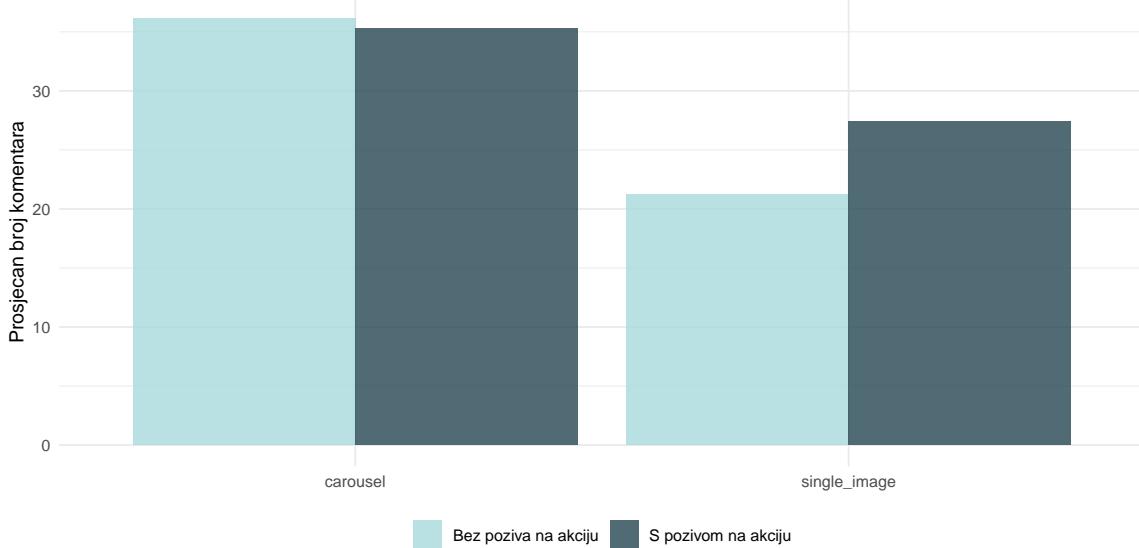
```

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. Učinak je konzistentan, što pojačava naše povjerenje u zaključak.

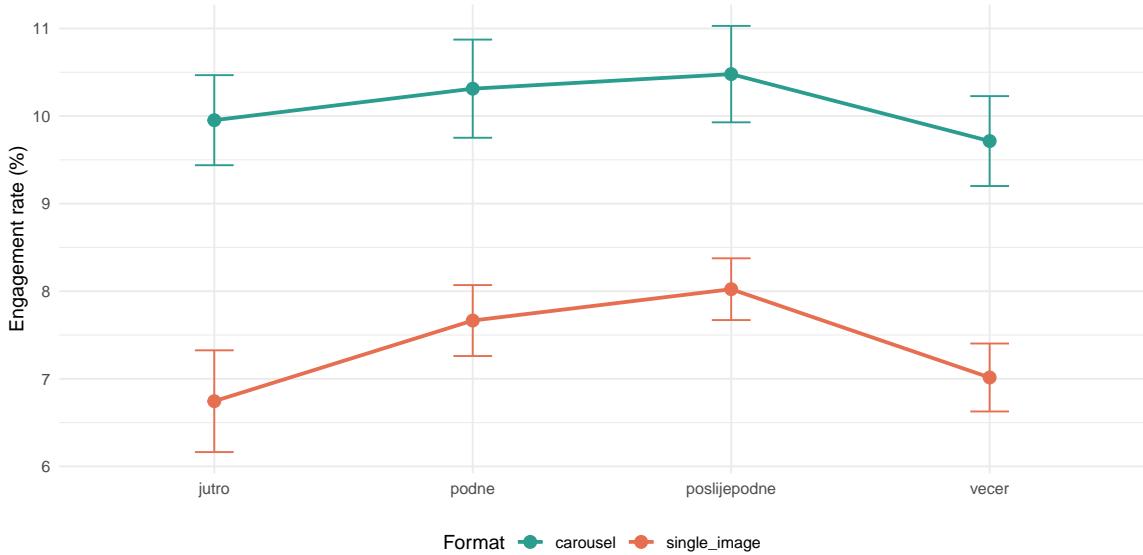
```
# Korak 5: Utjecaj CTA na komentare
ig |>
  group_by(format, has_cta) |>
  summarise(M_comments = mean(comments), .groups = "drop") |>
  mutate(has_cta = if_else(has_cta, "S pozivom na akciju", "Bez poziva na akciju")) |>
  ggplot(aes(x = format, y = M_comments, fill = has_cta)) +
  geom_col(position = "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")
```

Prosječan broj komentara: format x poziv na akciju
 CTA povećava komentare, osobito kod carousel objava



```
# Korak 6: Engagement po dobu dana
ig |>
  mutate(time_of_day = factor(time_of_day, levels = c("jutro", "podne", "poslijepodne", "vrijeme"),
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")
```

Engagement rate po dobu dana s intervalima pouzdanosti
Carousel prednost prisutna u svim terminima. Poslijepodne je optimalno za oba formata.



```
# 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 Još o p-vrijednostima: ASA izjava i višestruko testiranje

Američka statistička asocijacija (ASA) je 2016. izdala izjavu o p-vrijednostima. Šest principa vrijeti ponoviti.

P-vrijednosti mogu pokazati koliko su podaci nekompatibilni sa specificiranim statističkim modelom. P-vrijednosti ne mjere vjerojatnost da je hipoteza istinita niti vjerojatnost da su podaci proizvedeni samo slučajnošću. Znanstveni zaključci i poslovne odluke 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. Sama p-vrijednost ne pruža dobru mjeru dokaza za model ili hipotezu.

10.15.1 Višestruko testiranje

Kad provodimo mnogo testova istovremeno, povećava se šansa da barem jedan bude lažno pozitivan.

```
# 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):

```
cat("Statistički značajnih:", sum(sim_20_testova$znacajno), "\n\n")
```

Statistički značajnih: 1

```
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 može biti “statistički značajan.” Ovo je problem višestrukog testiranja. Postoje korekcije (Bonferronijeva: podijelite s brojem testova), ali svjesnost o problemu je prvi korak.

```
# 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 (mnogo lažno negativnih), ali štiti od lažno pozitivnih kad provodimo mnogo testova.

Upozorenje

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 $p < 0.05$.

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 prosjek jednouzorački određen?",  
  "Dvouzorački (nezavisni)", "Dvije nezavisne skupine", "t.test(x, y)", "Carousel vs single",  
  "Upareni", "Iste jedinice, dva mjerjenja", "t.test(x, y, paired = TRUE)", "Ocjene istih članova",  
)  
  
# 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 mjerjenja         t.test(x, ~ Ocjene~
```

Sva tri testa imaju istu logiku (testiramo H_0 , računamo t , gledamo p). Razlikuju se u formulaciji H_0 i načinu izračuna SE. R funkcija `t.test()` pokriva sva tri slučaja.

! Ključni zaključci

1. Testiranje hipoteza postavlja nultu hipotezu (H_0 : nema učinka) i traži dokaze protiv nje. Ako su podaci dovoljno neobični pod H_0 ($p < \alpha$), odbacujemo H_0 .
2. Testna statistika mjeri koliko su podaci neobični pod H_0 . Za t-test: $t = \frac{\text{razlika}}{\text{SE}}$. Veći $|t|$ = jači dokaz protiv H_0 .
3. P-vrijednost je vjerojatnost podataka (ili ekstremnijih) pod H_0 . NIJE vjerojatnost da je H_0 istinita. NIJE mjera veličine učinka.
4. `t.test()` pokriva jednouzorački ($\mu = \text{vrijednost}$), dvouzorački (dva vektora) i upareni test (`paired = TRUE`). Welchov test (default) ne prepostavlja jednakost varijance.
5. Greška tipa I (α) je lažno pozitivni rezultat. Greška tipa II (β) je propuštena prava razlika. Snaga = $1 - \beta$ trebala bi biti barem 0.80.
6. Cohenov d izražava razliku u SD jedinicama: $d = 0.2$ mali, $d = 0.5$ srednji, $d = 0.8$ veliki učinak. Uvijek ga izvijestite uz p-vrijednost.
7. `power.t.test()` računa potrebnu veličinu uzorka za zadanu snagu, učinak i α . Planirajte istraživanje PRIJE prikupljanja podataka.

8. Upareni t-test je osjetljiviji od nezavisnog jer kontrolira varijabilnost između parova. Koristite ga kad iste jedinice mjerite dva puta.
9. Statistička značajnost ($p < 0.05$) ne znači praktičnu važnost. Velik uzorak može detektirati trivijalne razlike. Mali uzorak može propustiti važne razlike.
10. Višestruko testiranje inflacionira grešku tipa I. Ako provodite mnogo testova, koristite korekciju (Bonferroni, BH) ili barem transparentno izvijestite koliko ste testova proveli.
11. Kompletan izvještaj uključuje opisnu statistiku, vizualizaciju, testnu statistiku s p-vrijednošću, veličinu učinka (d) s interpretacijom, interval pouzdanosti za razliku i praktičnu preporuku.
12. Odsutnost statistički značajnog rezultata ($p > 0.05$) ne znači da učinak ne postoji. Možda uzorak nije bio dovoljno velik. Zato nikad ne kažemo "prihvaćamo H_0 ."

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 $\alpha = 0.01$.
 3. Simulirajte 1000 t-testova gdje su oba uzorka iz iste distribucije (H_0 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., Cetinkaya-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_0)	Početna pretpostavka da nema učinka ili razlike. Sadrži znak jednakosti.
Alternativna hipoteza (H_1)	Tvrđnja da učinak ili razlika postoji. Sadrži znak nejednakosti.
Testna statistika	Broj koji mjeri koliko su podaci neobični pod H_0 . Za t-test: $t = \text{razlika} / \text{SE}$.
P-vrijednost	Vjerojatnost podataka (ili ekstremnijih) pod pretpostavkom da je H_0 istinita.
Razina značajnosti (α)	Prag za odbacivanje H_0 . Obično 0.05.
Greška tipa I	Kontrolira grešku tipa I.
Greška tipa II	Odbacimo H_0 kad je istinita. Lažno pozitivni rezultat. Vjerojatnost = .
Statistička snaga	Ne odbacimo H_0 kad je lažna. Propušteni pravi učinak. Vjerojatnost = .
Cohenov d	Vjerojatnost detektiranja pravog učinka. Snaga = 1 - . Cilj 0.80.
Pooled SD	Standardizirana mjera veličine učinka: razlika prosjeka / pooled SD. $d = 0.2$ mali, 0.5 srednji, 0.8 veliki.
Jednouzorački t-test	Zajednička standardna devijacija dviju grupa, ponderirana njihovim veličinama. Usporedba jednog prosjeka s poznatom vrijednošću. <code>t.test(x, mu = ...)</code> .
Dvouzorački t-test	Usporedba prosjeka dviju nezavisnih skupina. <code>t.test(x, y)</code> .
Welchov t-test	Default u R-u. Ne prepostavlja jednakе varijance. Robusniji od Studentovog.
Upareni t-test	Usporedba parova (iste jedinice, dva mjerjenja). <code>t.test(x, y, paired = TRUE)</code> .
Dvosmjerni test	H_0 : . Testira razliku u oba smjera. Default u R-u.

Pojam	Objašnjenje
Jednosmjerni test	$H_0: \leq$ ili $H_1: >$. Osjetljiviji u jednom smjeru ali slijep za drugi.
Višestruko testiranje	Provodenje 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.
<code>power.t.test()</code>	R funkcija za analizu snage: izračun potrebnog n, snage ili detektabilnog učinka.
<code>p.adjust()</code>	R funkcija za korekciju p-vrijednosti za višestruko testiranje (Bonferroni, BH, itd.).

Dio IV

Inferencijalna statistika

11 Tjedan 11: 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. Primjeniti 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 Motivacija: pitanja na koja t-test ne može odgovoriti

Prošla dva tjedna bavili smo se pitanjima poput "je li prosjek angažmana veći za carousel nego za single image." To su pitanja o prosjecima numeričkih varijabli. Ali mnoga istraživačka pitanja u komunikologiji uključuju **kategoričke varijable** (varijable čije su vrijednosti kategorije, ne brojevi).

Postoji li veza između dobne skupine i preferiranog tipa medija? Preferiraju li mladi streaming a stariji TV ili je distribucija ista u svim dobnim skupinama? Odgovara li distribucija obrazovanja u našem uzorku populacijskim podacima? Ovisi li zadovoljstvo medijem o regiji?

Za ovakva pitanja trebamo **hi-kvadrat testove** (² testove). Oni testiraju postoji li statistički značajna veza između dviju kategoričkih varijabli ili odstupa li opažena distribucija od očekivane.

11.2 Naši podaci

Koristimo podatke ankete o medijskim navikama provedene na 800 ispitanika iz različitih dobnih skupina i regija.

```
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, 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` (4 kategorije) i `media_type` (6 kategorija). Temeljno pitanje je: postoji li veza između dobi i preferiranog medija?

11.3 Kontingencijska tablica

Prvo što napravimo s dvjema kategoričkim varijablama je **kontingencijska tablica** (contingency table, cross-tabulation). Ona prikazuje frekvencije za svaku kombinaciju kategorija.

```
# 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

```
# 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

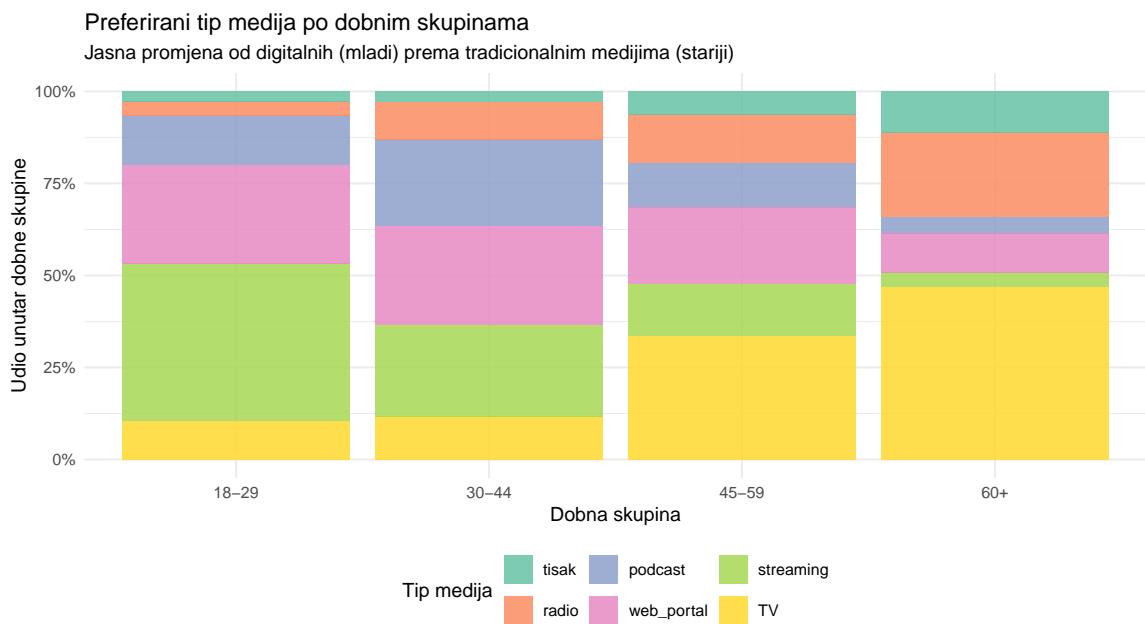
Proporcije po retku govore jasnu priču. Među osobama 18-29, 43% preferira streaming i 27% web portale. Među osobama 60+, 47% preferira TV i 23% radio. Obrazac je očit: mlađi preferiraju digitalne medije, stariji tradicionalne.

Ali je li ovaj obrazac statistički značajan ili bi mogao nastati slučajno?

```

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 (stariji)",
    x = "Dobna skupina",
    y = "Udio unutar dobne skupine",
    fill = "Tip medija"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

```



11.4 Hi-kvadrat test za dobrotu prilagodbe (goodness-of-fit)

Prije nego prijedemo na vezu dviju varijabli, krenimo s jednostavnijim pitanjem: odgovara li distribucija jedne kategoričke varijable nekoj očekivanoj distribuciji?

Situacija: medijska kuća tvrdi da ima podjednaku publiku iz svih pet regija. Naša anketa pokazuje drugačiju sliku. Želimo testirati odstupa li opažena distribucija od uniformne.

H_0 : Distribucija regija u uzorku odgovara uniformnoj distribuciji (20% svaka)

H_1 : 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 Kako funkcioniра hi-kvadrat statistika?

Hi-kvadrat statistika mjeri ukupno odstupanje opaženih frekvencija od očekivanih:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

gdje je O_i opažena frekvencija, a E_i očekivana frekvencija za kategoriju i . Što je veća razlika između opaženih i očekivanih, veća je χ^2 .

```

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)
))

```

	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 χ^2 ovisno o tome koliko njezina opažena frekvencija odstupa od očekivane. Zadnji redak (UKUPNO) je testna statistika χ^2 .

```

# 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))

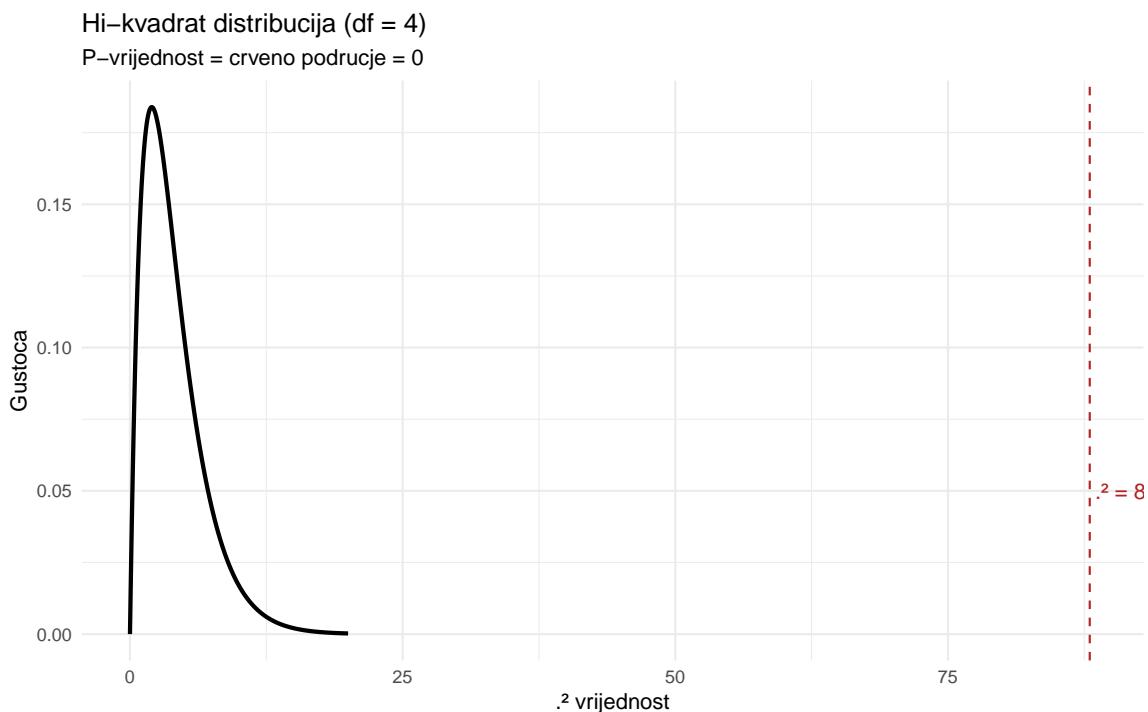
ggplot(chi_data, aes(x = x, y = density)) +
  geom_line(linewidth = 1) +
  geom_area(data = chi_data |> filter(x >= gof_test$statistic), fill = "firebrick", alpha

```

```

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()

```



P-vrijednost je manja od 0.05, što znači da distribucija regija u uzorku statistički značajno odstupa od uniformne. Zagreb je nadreprezentiran, a ostale regije su podreprezentirane.

11.4.2 Testiranje s poznatim proporcijama

Češći slučaj je testiranje protiv poznatih populacijskih proporcija, ne uniformne distribucije.

```

# 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 (gdje Zagreb čini 30% populacije), rezultat je drugačiji. P-vrijednost je veća, što znači da naš uzorak zapravo dobro odražava populacijsku distribuciju regija. Ovo je važna razlika: isti podaci mogu biti “odstupajući” od jedne distribucije a “konzistentni” s drugom.

11.5 Hi-kvadrat test nezavisnosti

Sada prelazimo na glavno pitanje: postoji li **veza** (asocijacija) između dviju kategoričkih varijabli? Konkretno: ovisi li preferirani tip medija o dobnoj skupini?

H_0 : Tip medija i dobna skupina su nezavisni (nema veze)

H_1 : Tip medija i dobna skupina NISU nezavisni (postoji veza)

“Nezavisni” znači da poznavanje nečije dobne skupine ne pomaže u predviđanju njihovog preferiranog medija. Ako su nezavisni, distribucija medijskog tipa trebala bi biti ista u svim dobnim skupinama.

```

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 ($p < 2.2\text{e-}16$). Imamo izuzetno snažne dokaze da postoji veza između dobi i preferiranog tipa medija. Ali χ^2 statistika sama ne govori koji parovi kategorija su odgovorni za vezu.

11.5.1 Očekivane frekvencije

Očekivane frekvencije su ono što bismo očekivali da vidimo KAD NE BI BILO veze između varijabli. Računaju se kao:

$$E_{ij} = \frac{\text{ukupno u retku } i \times \text{ukupno u stupcu } j}{\text{ukupno}}$$

```
# 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

Pod H (nema veze), sve dobne skupine bi imale istu distribuciju medijskih tipova. Na primjer, ako 22% cijelog uzorka preferira streaming, onda bi i u skupini 18-29 i u skupini 60+ oko 22% preferiralo streaming. Ali u stvarnosti, 43% mlađih preferira streaming, a samo 4% starijih. Ovo veliko odstupanje od očekivanih je ono što test detektira.

11.5.2 Provjera prepostavki

Hi-kvadrat test ima važnu prepostavku: sve očekivane frekvencije trebaju biti dovoljno velike (konvencija: barem 5).

```
# 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 test valjan. Kad ovo nije slučaj (male frekvencije), trebamo koristiti Fisherov egzaktni test.

11.6 Standardizirani reziduali: gdje je veza najjača?

Ukupna ² statistika govori da veza postoji, ali ne govori gdje. Za to koristimo **standardizirane reziduale** (Pearsonove reziduale):

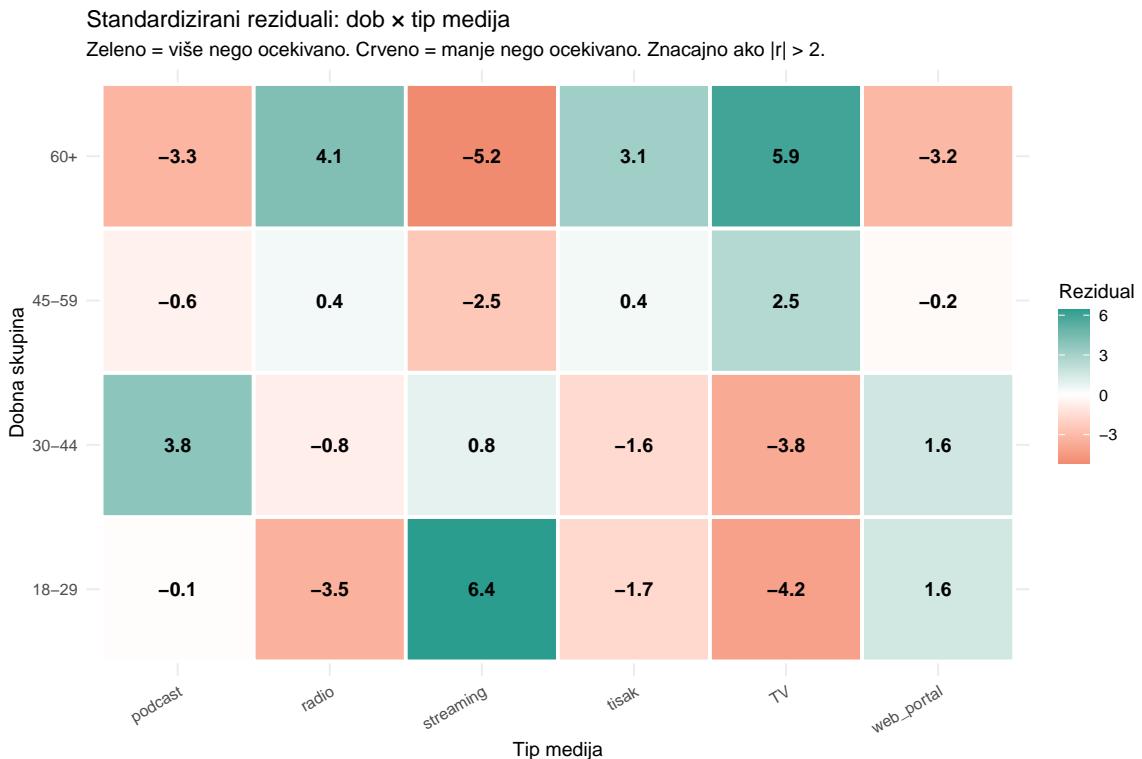
$$r_{ij} = \frac{O_{ij} - E_{ij}}{\sqrt{E_{ij}}}$$

Rezidual veći od +2 znači da taćelija ima značajno VIŠE opažanja nego što očekujemo pod H_0 . Rezidual manji od -2 znači značajno MANJE.

```
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

```
# 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 informacija. Najjače zelene ćelije (puno više nego očekivano) su streaming u skupini 18-29 (+6.2) i TV u skupini 60+ (+6.6). Najjače crvene ćelije (puno manje nego očekivano) su TV u skupini 18-29 (-4.3) i streaming u skupini 60+ (-5.1). Reziduali jasno potvrđuju generacijski jaz u medijskim navikama.

💡 Praktični savjet

Kad izvještavate rezultate hi-kvadrat testa, ne stajte samo na “ χ^2 je značajan.” Koristite reziduale da identificirate specifične kombinacije kategorija koje najviše doprinose vezi. Urednica ne želi znati samo da “postoji veza između dobi i medija.” Želi znati KOJA je ta veza: mladi biraju streaming, stariji TV.

11.7 Mjera veličine učinka: Cramérovo V

Kao i kod t-testa, sama statistička značajnost ne govori koliko je veza jaka. **Cramérovo V** je mjera veličine učinka za hi-kvadrat test nezavisnosti:

$$V = \sqrt{\frac{\chi^2}{n \times (k - 1)}}$$

gdje je n ukupan broj opažanja, a k manji od broja redova ili stupaca kontingencijske tablice. V varira od 0 (potpuna nezavisnost) do 1 (savršena asocijacija).

```
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 Interpretacija Cramérovog V

Cohen (1988) je predložio smjernice ovisne o stupnjevima slobode. 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
```

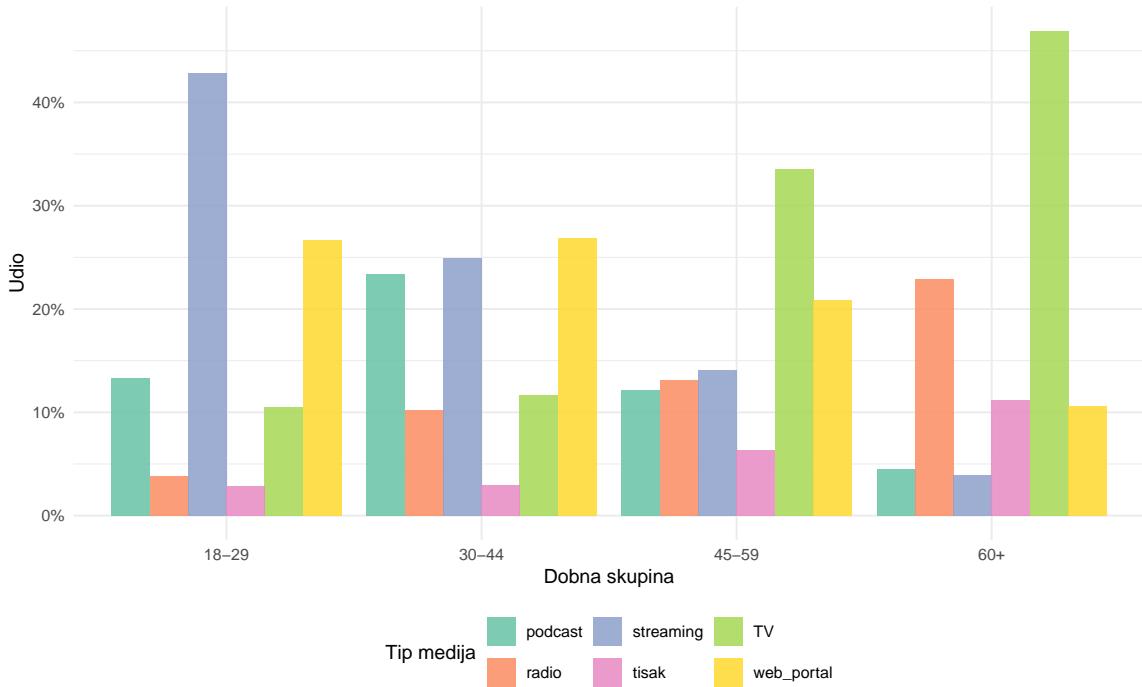
Naše $V = 0.31$ je veliki učinak. Veza između dobi i medijskog tipa je ne samo statistički značajna nego i praktično snažna.

11.8 Vizualizacija kategoričkih podataka

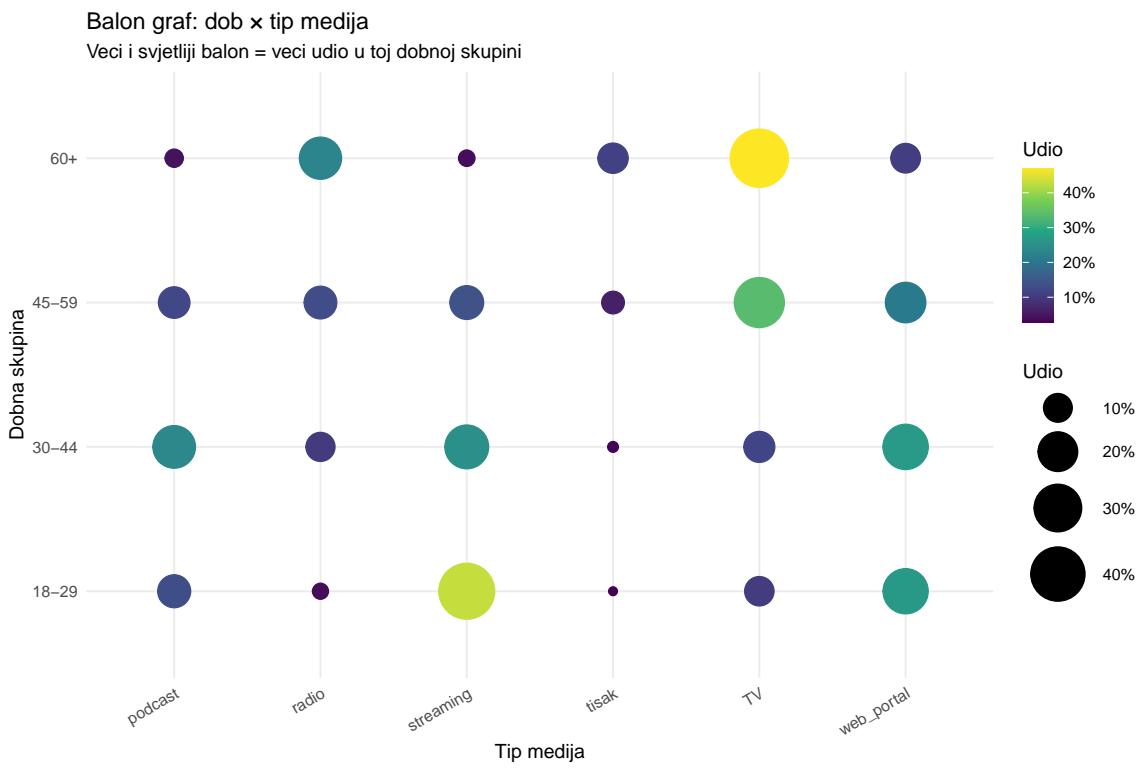
Kontingencijske tablice su korisne ali teške za brzo čitanje. Vizualizacije pomažu.

```
# 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 dobним 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")
```

Preferirani tip medija po dobnim skupinama
 $\chi^2 = 233.6$, $p < 0.001$, Cramérovo $V = 0.31$ (veliki ucinak)



```
# 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 svjetlijiji 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 jasne korake koje možete primijeniti na bilo koji par kategoričkih varijabli.

Korak 1: Formulirajte hipoteze.

Korak 2: Napravite kontingencijsku tablicu i vizualizirajte podatke.

Korak 3: Provjerite pretpostavke (sve očekivane frekvencije ≥ 5).

Korak 4: Provedite test i izračunajte p-vrijednost.

Korak 5: Izračunajte veličinu učinka (Cramérovo V).

Korak 6: Pogledajte reziduale za specifična odstupanja.

Korak 7: Interpretirajte u kontekstu.

```
# 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

```
# 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

```
# Korak 4: test  
chi_spol
```

Pearson's Chi-squared test

```
data: tablica_spol  
X-squared = 0.40693, df = 4, p-value = 0.9819
```

```
# 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

```
# Korak 6: reziduali  
cat("\nStandardizirani reziduali:\n")
```

Standardizirani reziduali:

```

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

```

U ovom primjeru, odnos između spola i preferencije sadržaja možda nije statistički značajan (ovisi o p-vrijednosti). Ovo je potpuno normalan rezultat: ne mora svaka veza biti značajna. Podatke treba pustiti da govore.

i Podsjetnik

U prvom dijelu naučili smo hi-kvadrat test za dobrotu prilagodbe i test nezavisnosti, očekivane frekvencije, standardizirane reziduale i Cramérovo V. U ovom dijelu pokrivamo situacije kad standardni hi-kvadrat test nije primjenjen te provodimo potpunu analizu kategoričkih podataka.

11.10 Fisherov egzaktni test

Hi-kvadrat test je aproksimacija koja radi dobro kad su očekivane frekvencije dovoljno velike (sve > 5). Kad to nije slučaj (mali uzorak, rijetke kategorije), aproksimacija postaje nepouzdana. **Fisherov egzaktni test** računa točnu p-vrijednost bez aproksimacije.

Situacija: testirate vezu između tipa CTA (call-to-action) i konverzije na malom uzorku od 40 newsletter kampanja.

```

# 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 (ili ispod). Hi-kvadrat aproksimacija nije pouzdana. Koristimo 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

Fisherov test daje nešto drugačiju p-vrijednost od hi-kvadrata. Za male uzorke, Fisherov test je pouzdaniji. R po defaultu primjenjuje **Yatesovu korekciju kontinuiteta** na 2x2 tablice (`chisq.test` s `correct = TRUE`), što daje konzervativniju p-vrijednost, ali Fisherov test je ipak bolji izbor kad su očekivane frekvencije male.

11.10.1 Odds ratio iz Fisherovog testa

Fisherov test za 2x2 tablice automatski računa **odds ratio** (omjer šansi), što je korisna mjera veličine učinka za binarne ishode.

```
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 > 1 znači da je šansa konverzije veća uz direktni CTA. Odds ratio $= 1$ bi značio da nema razlike. CI koji ne sadrži 1 sugerira statistički značajnu razliku.

💡 Kada koristiti koji test?

Hi-kvadrat test: Kad su sve očekivane frekvencije ≥ 5 i tablica je veća od 2x2. Brz i dovoljno točan za velike uzorke.

Fisherov egzaktni test: Kad su neke očekivane frekvencije < 5 ili je ukupni uzorak mali (< 50). Funkcionira za bilo koju veličinu tablice, ali je računalno zahtjevniji za veće tablice.

U praksi, Fisherov test možete koristiti uvijek. Za velike uzorke daje identične rezultate kao hi-kvadrat. Razlika se pojavljuje samo za male uzorke, gdje je Fisherov test bolji.

11.11 Spajanje kategorija

Ponekad kontingencijska tablica ima kategorije s vrlo malo opažanja. Umjesto da odmah pribjegnemo Fisherovom testu, možemo spojiti (kolapsirati) slične kategorije.

```
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 + tisak)  
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")
```

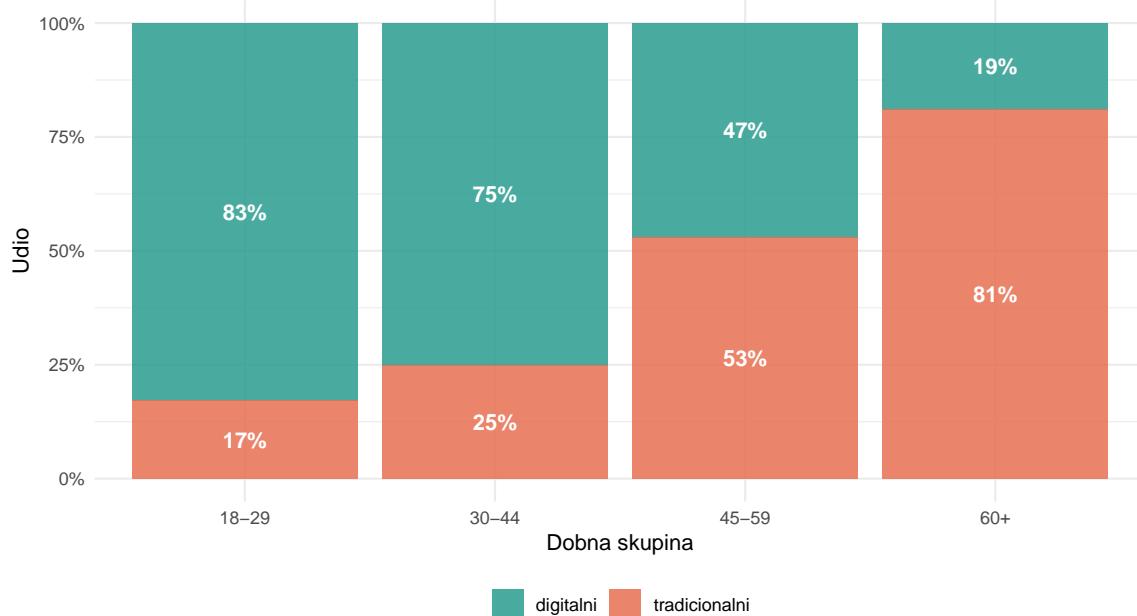
Cramerovo V: 0.499

Sažeta tablica je jasnija za komunikaciju rezultata. Umjesto šest kategorija, imamo jednu binarnu podjelu: digitalni vs tradicionalni mediji. 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")
```

Digitalni vs tradicionalni mediji po dobnim skupinama

chi² = 198.9, p < 0.001, V = 0.5



Priča je sada kristalno jasna: 85% osoba 18-29 preferira digitalne medije, ali samo 26% osoba 60+. Generacijski jaz je masivan.

11.12 Stratificirana analiza

Ponekad nas zanima je li veza između dviju varijabli konzistentna unutar podgrupa treće varijable. Na primjer: postoji li veza između medijskog tipa i zadovoljstva, ali odvojeno za svaku dobnu skupinu?

```
# 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). Ovo je povezano s Simpsonovim paradoksom: agregirani rezultati mogu biti obmanjujući.

11.13 Simpsonov paradoks

Simpsonov paradoks nastaje kad se smjer veze promijeni kad kontroliramo treću varijablu. Ovo je jedan od najvažnijih koncepata za razumijevanje kategoričkih podataka u komunikologiji.

```

# Konstruirani primjer: dva portala, CTR po tipu uređaja
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 uređaju
simpson |>
  select(portal, uređaj, ctr, prikazi)

# A tibble: 4 x 4
#>   portal  uređaj     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

```

Portal B ima viši CTR na SVAKOM uređaju (mobitel: 6.0% vs 8.0%, desktop: 36.0% vs 40.0%). Ali Portal A može imati viši ukupni CTR! Razlog je u tome što Portal A ima mnogo više mobilnog prometa (1000 od 1500 prikaza) koji ima nizak CTR, dok Portal B ima relativno više desktop prometa s visokim CTR-om. Kad agregiramo, neravnomjerna raspodjela uređaja “prevagne” i stvori obmanjujuće ukupne brojke.

Upozorenje

Simpsonov paradoks nije rijedak kuriozitet. U komunikološkim istraživanjima je čest jer se grupe (dobne, regionalne, platformske) razlikuju po veličini i karakteristikama. Kad god uspoređujete kategoričke podatke agregirano, zapitajte se: bi li se zaključak promijenio kad biste razdvojili po nekoj trećoj varijabli?

11.14 McNemarov test za uparene proporcije

Kad imate uparene kategoričke podatke (isti ispitanici mjereni dva puta), standardni hi-kvadrat test nije primjerен. Za tu situaciju koristimo **McNemarov test**.

Situacija: 200 ispitanika je pitano preferiraju li online ili tiskane vijesti, jednom na početku i jednom na kraju semestra.

```
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
promjenili <- sample(1:n_mc, 40)
for (i in promjenili[1:30]) poslije[i] <- "online"
for (i in promjenili[31:40]) poslije[i] <- "tisak"

mc_tablica <- table(Prije = prije, Poslije = poslije)
mc_tablica
```

		Poslije	
Prije	online	tisak	
online	95	4	
tisak	15	86	

Ova tablica se čita ovako: na dijagonali su ispitanici koji NISU promjenili mišljenje. Izvan dijagonale su oni koji jesu. McNemarov test testira je li broj promjena u jednom smjeru (tisak na online) značajno veći od broja promjena u drugom (online na tisak).

```
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

# 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

```
cat("Presli online na tisak:", online_na_tisak, "\n")
```

Presli online na tisak: 4

```
cat("Neto pomak prema onlineu:", tisak_na_online - online_na_tisak, "\n")
```

Neto pomak prema onlineu: 11

McNemarov test pokazuje je li pomak prema online vijestima statistički značajan. Više ispitanika je prešlo s tiska na online nego obrnuto.

! Kada koristiti McNemarov test?

Kad imate **iste ispitanike** mjerene dva puta na istoj binarnoj varijabli. Primjeri: preferencija prije i poslije kampanje, stav prije i poslije vijesti, izbor prije i poslije redizajna. McNemarov test je za kategoričke podatke ono što je upareni t-test za numeričke.

11.15 Potpuna analiza: medijske navike i demografija

Provđemo kompletну analizu kategoričkih podataka na našem datasetu. Istražujemo tri pitanja: (1) Ovisi li tip medija o dobi? (2) Ovisi li o regiji? (3) Ovisi li preferencija sadržaja o obrazovanju?

```
# 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.

```
# 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("Celiye s |rezidual| > 2:\n")
```

Celiye s |rezidual| > 2:

```
rez_df |>  
  mutate(r = round(r, 1), smjer = if_else(r > 0, "VISE nego ocekivano", "MANJE nego ocekivano")) |>  
  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
```

```
# 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.

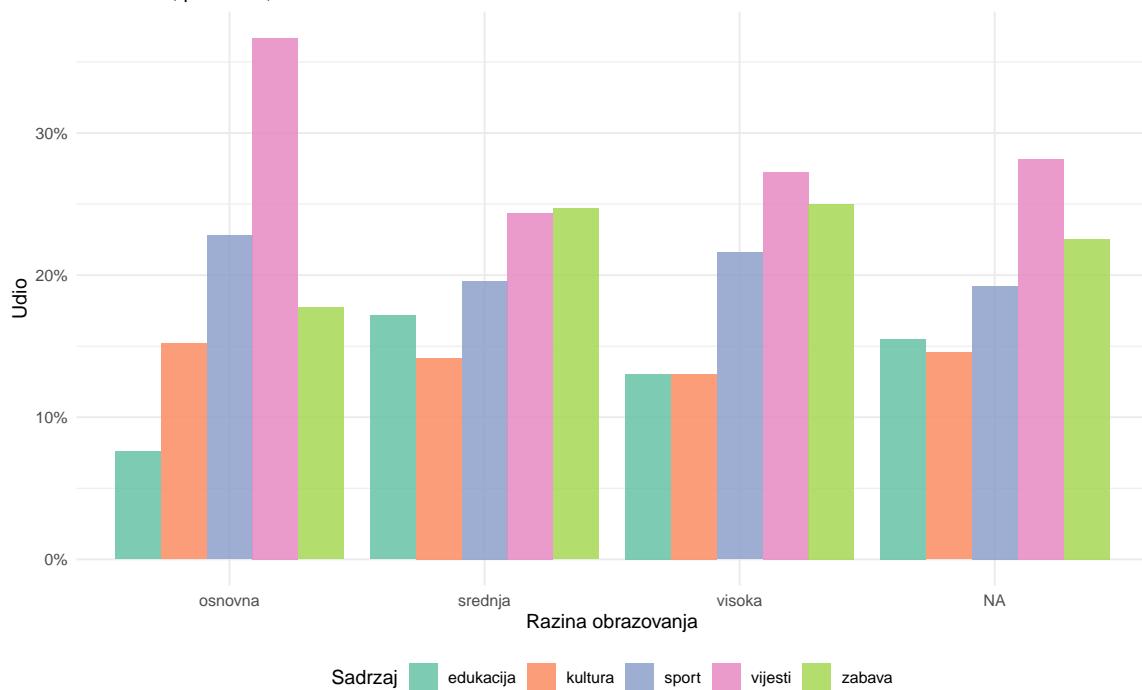
```

# 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")

```

Preferencija sadržaja po razini obrazovanja

chi2 = 10.5, p = 0.574, V = 0.07



```
# 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$statistic), 1),
  df = c((nrow(tab1)-1)*(ncol(tab1)-1), (nrow(tab2)-1)*(ncol(tab2)-1), (nrow(tab3)-1)*(ncol(tab3)-1)),
  p = c(format(test_q1$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 mnogo manje (ako uopće postoje). Veza obrazovanja i preferencije sadržaja varira ovisno o uzorku.

11.16 Uobičajene pogreške i ograničenja

Hi-kvadrat testovi su jednostavnici za provesti ali lako ih je krivo primijeniti ili interpretirati.

Pogreška 1: Korištenje postotaka umjesto frekvencija. Chi2 test zahtijeva frekvencije (apsolutne brojeve), ne postotke ili proporcije. `chisq.test(c(30, 20, 50))` je ispravno. `chisq.test(c(0.30, 0.20, 0.50))` će dati besmislene rezultate jer misli da ukupno imate 1 opažanje.

```

# 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

```

```
cat("\nISPRAVNO (frekvencije):\n")
```

ISPRAVNO (frekvencije):

```
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
```

Pogreška 2: Pretjerana granularnost. Tablica 10×8 s ukupno 100 opažanja imat će mnogo celija s malim frekvencijama. Bolje je spojiti kategorije i imati manje ali punije celije.

Pogreška 3: Kauzalna interpretacija. Chi2 test detektira asocijaciju (vezu), ne kauzalnost. Činjenica da su dob i medijski tip povezani ne znači da dob uzrokuje preferenciju (možda je obrazovanje, socioekonomski status ili kohorta efekt). Za kauzalne zaključke trebate eksperimentalni dizajn ili napredne metode.

Pogreška 4: Zaboravljanje veličine učinka. Kao i kod t-testa, p-vrijednost ovisi o veličini uzorka. S dovoljno velikim uzorkom, i trivijalna veza postaje "statistički značajna." Uvijek izvijestite Cramérovo V uz chi2 i p.

Pogreška 5: Testiranje iste hipoteze više puta. Ako testirate 10 parova varijabli, imate isti problem višestrukog testiranja kao kod t-testa. Koristite korekciju (Bonferroni ili BH).

```
# 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, možda se neki marginalno značajni rezultati izgube. Ovo je cijena korektnog pristupa višestrukom testiranju.

11.17 Pomoćna funkcija za kompletne hi-kvadrat izvještaj

U praksi često ponavljamo iste korake za svaki par varijabli. Napišimo funkciju koja automatizira cijeli postupak.

```

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.

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

1. Hi-kvadrat testovi se koriste kad su varijable kategoričke. Test za dobrotu prilagodbe uspoređuje jednu varijablu s očekivanom distribucijom. Test nezavisnosti testira vezu dviju varijabli.
2. Chi² statistika mjeri ukupno odstupanje opaženih od očekivanih frekvencija: $\chi^2 = \text{suma } (O - E)^2 / E$. Veći χ^2 = jači dokaz protiv H₀.
3. Očekivane frekvencije pod H₀ su $E = (\text{ukupno u retku puta ukupno u stupcu}) / \text{ukupno}$. Pretpostavka: sve $E \geq 5$.
4. Standardizirani reziduali identificiraju specifične celije koje najviše doprinose vezi. $|r| > 2$ je značajno. Uvijek ih izvijestite jer χ^2 sam ne govori gdje je veza najjača.
5. Cramérovo V mjeri jačinu veze (0 do 1). Za $k = 4$: V oko 0.06 mali, V oko 0.17 srednji, V oko 0.29 veliki učinak. Uvijek ga izvijestite uz χ^2 i p.
6. Fisherov egzaktni test koristite kad su očekivane frekvencije < 5 ili je uzorak mali. Za 2x2 tablice automatski daje odds ratio.
7. Spajanje kategorija (kolapsiranje) je legitiman pristup kad su neke kategorije prerijetke. Digitalni vs tradicionalni mediji je informativnije od šest odvojenih kategorija.
8. Simpsonov paradoks upozorava da agregirani rezultati mogu biti obmanjujući. Uvijek provjerite rezultate stratificirane po relevantnoj trećoj varijabli.
9. McNemarov test je za uparene kategoričke podatke (isti ispitanici, dva mjerenja). Ekvivalent uparenom t-testu za kategoričke varijable.
10. Hi-kvadrat test detektira asocijaciju, ne kauzalnost. Veza dobi i medijskog tipa ne znači da dob uzrokuje preferenciju.
11. Višestruko testiranje je problem i kod hi-kvadrat testova. Koristite BH ili Bonferroni korekciju kad testirate mnogo parova varijabli.
12. Kompletni izvještaj uključuje kontingencijsku tablicu, vizualizaciju (stacked/dodged barplot), χ^2 s df i p, Cramérovo V s interpretacijom, reziduale za specifična odstupanja i praktičnu interpretaciju u kontekstu istraživačkog pitanja.

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.

Pojam	Objašnjenje
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. (O minus E) / korijen(E). Doprinos svake ćelije ukupnom chi2. Značajno ako
Standardizirani rezidual	Frekvencija pod H0. Za test nezavisnosti: E = (redak total puta stupac total) / ukupno. (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 asocijacija 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 <code>chisq.test()</code> .
McNemarov test	Test za uparene kategoričke podatke (isti ispitanici, dva mjerena).
Simpsonov paradoks	Smjer veze se promjeni kad kontroliramo treću varijablu. Agregirani rezultati obmanjuju.
Stratificirana analiza	Provodenje testa odvojeno za podgrupe treće varijable. Otkriva Simpsonov paradoks.
Spajanje kategorija	Kolapsiranje rijetkih kategorija u šire grupe. Povećava očekivane frekvencije i preglednost.
<code>chisq.test()</code>	R funkcija za hi-kvadrat test. Prima vektor frekvencija (gof) ili kontingencijsku tablicu.
<code>fisher.test()</code>	R funkcija za Fisherov egzaktni test. Prima kontingencijsku tablicu.
<code>mcnemar.test()</code>	R funkcija za McNemarov test. Prima 2x2 tablicu uparenih podataka.
<code>table()</code>	R funkcija za kreiranje kontingencijske tablice. <code>table(x, y)</code> za dvije varijable.
<code>prop.table()</code>	Pretvara frekvencije u proporcije. <code>margin = 1</code> za retke, <code>margin = 2</code> za stupce.
<code>p.adjust()</code>	Korekcija p-vrijednosti za višestruko testiranje. <code>method = "BH"</code> je preporučen.

12 Tjedan 12: Usporedba prosjeka t-testovima

Kad prepostavke drže i kad ne drže

```
library(tidyverse)
```

Ishodi učenja

Nakon ovog predavanja moći će

1. Odabrat odgovarajući t-test (jednouzorački, nezavisni, upareni) za zadano istraživačko pitanje.
2. Provjeriti prepostavke 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 kompletну analizu usporedbe dvaju uvjeta na stvarnim podacima.

12.1 Motivacija: vizuali u člancima

Uredništvo web portala razmatra redizajn članka. Dosad su objavljali pretežno tekstualne članke, ali razmišljaju o dodavanju infografika, slika i vizualnih elemenata. Pitanje je: kako vizuali utječu na čitateljsko iskustvo? Mijenaju li vrijeme čitanja, razumijevanje, namjeru dijeljenja, percipiranu vjerodostojnost?

Proveli ste eksperiment: 120 članaka je prezentirano ispitanicima u dva uvjeta (s vizualima i bez vizuala). Svaki članak je testiran u oba uvjeta (within-subjects dizajn), što znači da ćemo koristiti **upareni t-test**.

Na predavanju o testiranju hipoteza (tjedan 10) naučili smo logiku t-testa, p-vrijednost i greške tipa I i II. Ovo predavanje se fokusira na praktičnu primjenu: provjeru prepostavki, odabir pravog testa i pravilno izvještavanje rezultata.

```

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, 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

```

Članci s vizualima imaju duže vrijeme čitanja, bolje razumijevanje i višu namjeru dijeljenja. Ali prije nego izvučemo zaključke, moramo provjeriti pretpostavke.

12.2 Podsjetnik: tri vrste t-testa

```

tribble(
  ~test, ~pitanje, ~R_kod,
  "Jednouzorački", "Razlikuje li se prosjek od poznate vrijednosti?", "t.test(x, mu = vrijednost, var.equal = TRUE)"
  "Nezavisni (dvouzorački)", "Razlikuju li se prosjeci dviju nezavisnih grupa?", "t.test(x1, x2, var.equal = TRUE)"
  "Upareni", "Razlikuje li se prosjek u dva uvjeta za iste jedinice?", "t.test(x, y, paired = TRUE)"
)

# A tibble: 3 x 3
#> # ... with 3 variables:
#> #   test     <chr>    pitanje <chr>    R_kod <chr>
#> 1 Jednouzorački  Razlikuje li se prosjek od poznate vrijednosti? t.test(x, mu = vrijednost, var.equal = TRUE)
#> 2 Nezavisni (dvouzorački) Razlikuju li se prosjeci dviju nezavisnih grupa? t.test(x1, x2, var.equal = TRUE)
#> 3 Upareni        Razlikuje li se prosjek u dva uvjeta za iste jedinice? t.test(x, y, paired = TRUE)

```

Za naše podatke (isti članci u dva uvjeta), koristimo upareni t-test. Ali prvo moramo provjeriti pretpostavke.

12.3 Pretpostavke t-testa

T-test ima nekoliko pretpostavki koje moramo provjeriti prije nego mu vjerujemo.

Za sve t-testove:

Podaci su barem intervalna skala (razlike između vrijednosti imaju smisla). Opažanja su nezavisna jedno od drugoga (osim parova u uparenom testu). Distribucija je približno normalna (ili je uzorak dovoljno velik da CLT kompenzira).

Dodatno za nezavisni t-test:

Varijance dviju grupa su približno jednake (Studentov t-test) ili koristimo Welchovu korekciju koja ne zahtijeva ovu pretpostavku (default u R-u).

Najvažnija pretpostavka za provjeru je **normalnost**. Za upareni t-test, trebamo provjeriti normalnost RAZLIKA (ne pojedinačnih mjerjenja).

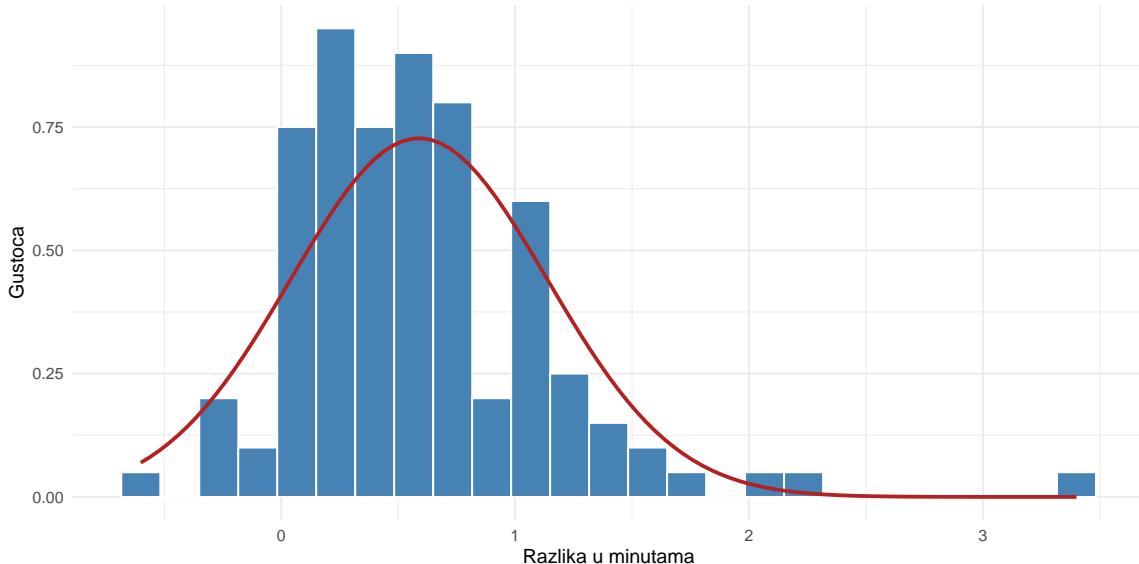
12.4 Provjera normalnosti

12.4.1 Vizualna provjera: histogram i QQ plot

```
# 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)

articles |>
  ggplot(aes(x = diff_time)) +
  geom_histogram(aes(y = after_stat(density)), fill = "steelblue", color = "white", bins = 30,
                 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()
```

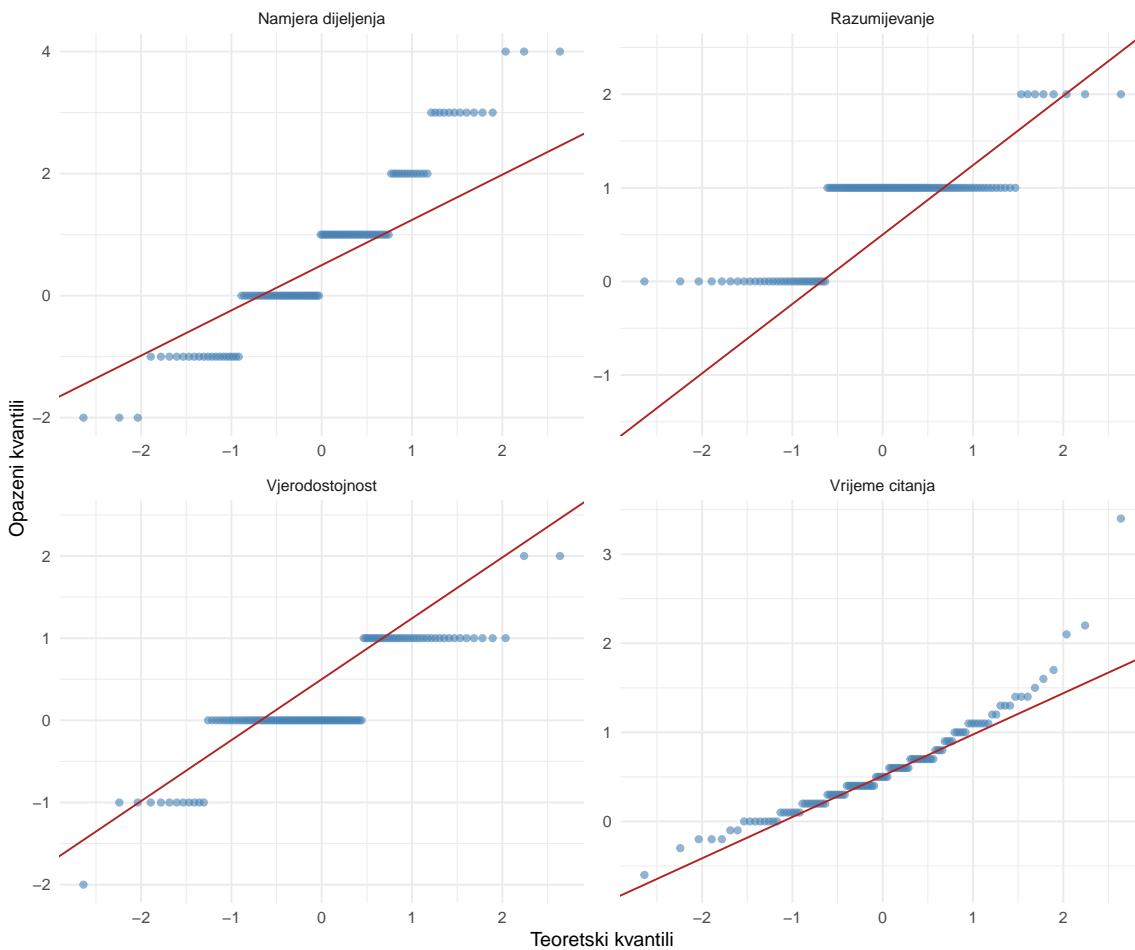
Distribucija razlika u vremenu citanja (s vizualima minus bez)
Crvena krivulja = normalna distribucija s istim M i SD



```
# 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()
```

QQ plotovi za razlike (upareni uvjeti)

Tocke blizu linije = normalna distribucija. Odstupanja na repovima su cesta.



QQ plot je najinformativniji dijagnostički alat za normalnost. Ako točke leže blizu dijagonalne linije, distribucija je približno normalna. Blaga odstupanja na repovima su prihvatljiva, osobito za $n > 30$. Sustavna zakrivljenost (S-oblik ili J-oblik) sugerira problem.

Razlike u vremenu čitanja i razumijevanju izgledaju razumno normalno. Razlike u dijeljenju i vjerodostojnosti su diskretnije (Likert skala) pa QQ plot pokazuje stepeničast uzorak, što je očekivano.

12.4.2 Shapiro-Wilkov test

Shapiro-Wilkov test je formalni test normalnosti. H je da su podaci normalno distribuirani. Ako je $p < 0.05$, odbacujemo normalnost.

```
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 ($p > 0.05$). Razlike u Likert varijablama (dijeljenje, vjerodostojnost) možda ne prolaze jer su diskretne (samo cjelobrojne vrijednosti). Za te varijable razmotrimo neparametrijske alternative.

! Važna napomena o Shapiro-Wilkovom testu

Shapiro-Wilkov test ima isti problem kao i svaki statistički test: s velikim uzorkom detektira i trivijalna odstupanja od normalnosti. S $n = 500$, gotovo svaka distribucija “ne prolazi” Shapiro-Wilkov test. S $n = 15$, gotovo svaka distribucija “prolazi.” Zato je vizualna procjena (QQ plot) jednako važna kao formalni test. Pravilo palca: ako QQ plot izgleda razumno i $n > 30$, t-test je dovoljno robustan čak i za umjerena odstupanja od normalnosti (zahvaljujući CLT-u).

12.5 Provjera homogenosti varijance

Za nezavisni t-test, Studentova verzija pretpostavlja jednake varijance u dvjema grupama. Levenov test (ili Bartlettov test) provjerava ovu pretpostavku.

```
# 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

```
# 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
```

Welchov test ima nerunde stupnjeve slobode jer ih prilagođava za razliku u varijancama. Kad su varijance jednake, oba testa daju iste rezultate. Kad su varijance različite, Welchov test je točniji.

Praktični savjet

Koristite Welchov t-test (default u R-u) uvek. Ne morate provjeravati homogenost varijance jer Welchov test radi jednako dobro kad su varijance jednake i bolje kad su različite. Studentov t-test s `var.equal = TRUE` koristite samo ako imate specifičan razlog (npr. usporedba s objavljenim rezultatima koji koriste Studentov test).

12.6 Upareni t-test na podacima o vizualima

Sada provodimo upareni t-test za naše glavno pitanje: utječu li vizuali na vrijeme čitanja?

```
# 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

# Cohenov d za upareni test: d = M_razlika / SD_razlika
d_time <- mean(articles$diff_time) / sd(articles$diff_time)

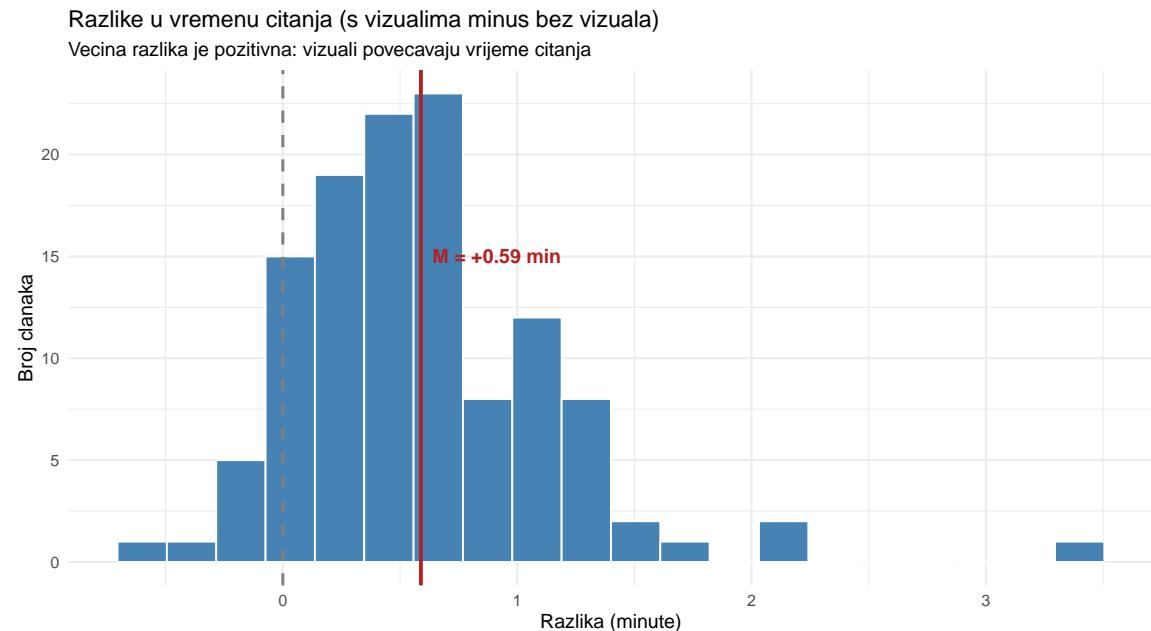
cat("==== Upareni t-test: Vrijeme citanja ===\n")

==== 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  
  
cat("Cohenov d:", round(d_time, 2), "\n")  
  
Cohenov d: 1.07  
  
cat("Interpretacija: Veliki ucinak. Vizuali povecavaju vrijeme citanja za ~35 sekundi.\n")  
  
Interpretacija: Veliki ucinak. Vizuali povecavaju vrijeme citanja za ~35 sekundi.
```

12.6.1 Vizualizacija uparenih podataka

```
# 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()
```



```
# 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) +
```

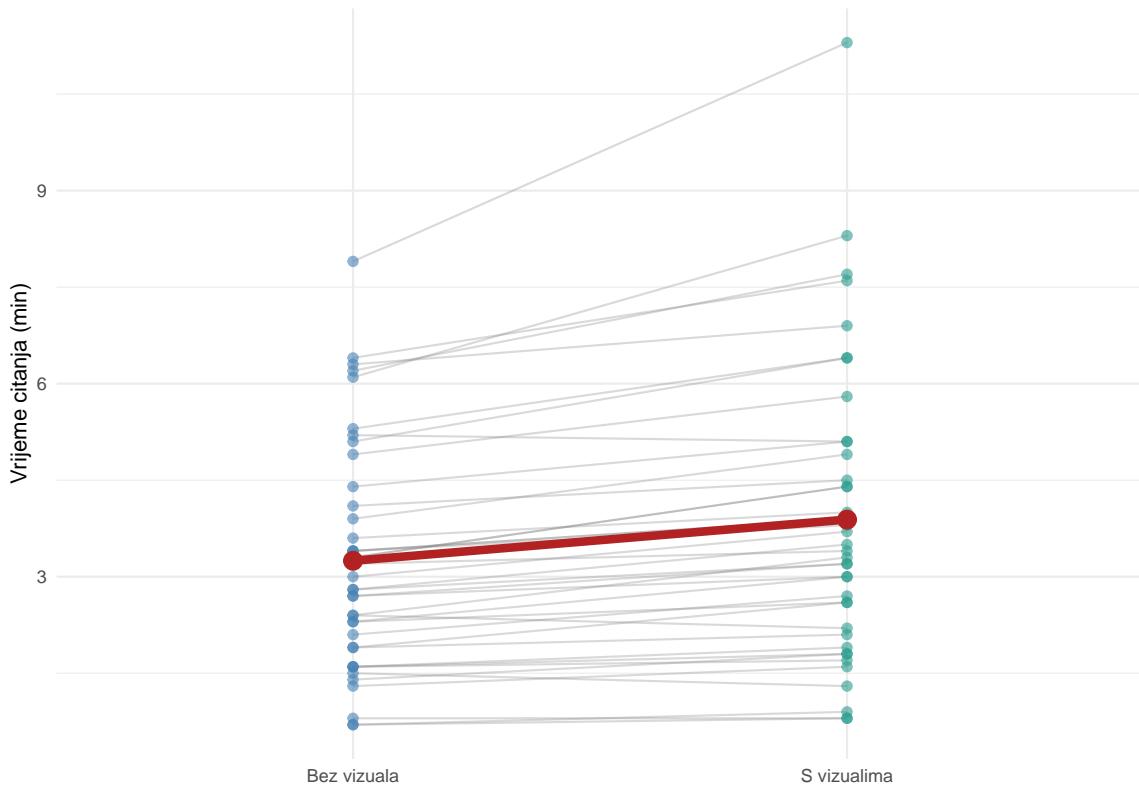
```

stat_summary(aes(group = 1), fun = mean, geom = "line", linewidth = 2, color = "firebrick")
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 = pojedinačni clanci. Crvena linija = prosjek.",
    x = NULL,
    y = "Vrijeme citanja (min)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

```

Vrijeme citanja po uvjetu (prvih 40 clanaka)

Sive linije = pojedinačni clanci. Crvena linija = prosjek.



Slope chart jasno pokazuje da većina linija ide prema gore (duže čitanje s vizualima). Crvena linija (prosjek) potvrđuje ukupni trend.

12.7 Sva četiri ishoda: sustavni pristup

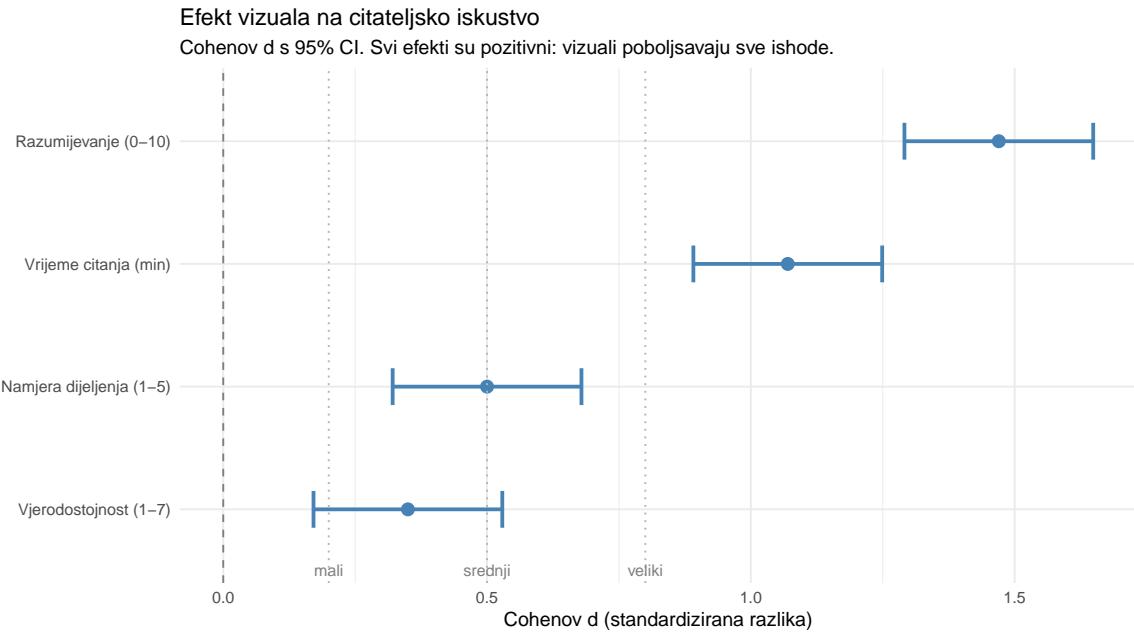
Provodimo upareni t-test za sve četiri mjere istovremeno.

```
# Funkcija za kompletну 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, "Vrijeme čitanja s vizuelnim sadržajem"),  
  uparena_analiza(articles$comprehension_with_visual, articles$comprehension_no_visual, "Rozumevanje teksta s vizuelnim sadržajem"),  
  uparena_analiza(articles$sharing_with_visual, articles$sharing_no_visual, "Namjera dijeljenja sadržaja s vizuelnim sadržajem"),  
  uparena_analiza(articles$credibility_with_visual, articles$credibility_no_visual, "Vjerojatnost da je sadržaj ispravan"))  
  |>  
  mutate(  
    znacajno = p < 0.05,  
    velicina = case_when(  
      abs(d) >= 0.8 ~ "veliki",  
      abs(d) >= 0.5 ~ "srednji",  
      abs(d) >= 0.2 ~ "mali",  
      .default = "zanemarivo"  
)  
    ,  
    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)
```

	ishod	M_bez	M_s	M_razlika	t	p	d	velicina	normalnost	ok
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<dbl>	<chr>	<lgl>	
1	Vrijeme citanja~	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 dijeljenja~	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	

Tablica daje kompletну sliku. Vrijeme čitanja i razumijevanje imaju normalne razlike (Shapiro $p > 0.05$) i značajne razlike s velikim učincima. Namjera dijeljenja i vjerodostojnost imaju diskretne distribucije (Shapiro može biti < 0.05) i manje učinke.

```
# 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 = "steelblue") +
  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 ishode",
    x = "Cohenov d (standardizirana razlika)",
    y = NULL
  ) +
  theme_minimal()
```



Forest plot je standardni način prikaza višestrukih veličina učinka. Sve točke su desno od nule (vizuali poboljšavaju sve ishode), a CI-evi ne prelaze nulu (svi efekti su statistički značajni). Najveći efekt je na razumijevanje, najmanji na vjerodostojnost.

12.8 Neparametrijske alternative: Wilcoxonov test

Kad normalnost nije zadovoljena i uzorak je premalen da CLT kompenzira, koristimo **neparametrijske testove**. Oni ne prepostavljaju normalnost jer rade s rangovima umjesto s izvornim vrijednostima.

```
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 test rangira sve razlike po apsolutnoj vrijednosti, dodaje im predznake i testira je li suma pozitivnih rangova značajno veća (ili manja) od očekivane pod H_0 .

```
# Namjera dijeljenja: Likert skala, možda 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 = TRUE)

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

# 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,
                     paired = TRUE)$p.value,
         p_w = wilcox.test(articles$reading_time_with_visual, articles$reading_time_no_visual,
                           paired = TRUE)$p.value),
  tibble(ishod = "Razumijevanje",
         p_t = t.test(articles$comprehension_with_visual, articles$comprehension_no_visual,
                     paired = TRUE)$p.value,
         p_w = wilcox.test(articles$comprehension_with_visual, articles$comprehension_no_visual,
                           paired = TRUE)$p.value),
  tibble(ishod = "Namjera dijeljenja",
         p_t = t.test(articles$sharing_with_visual, articles$sharing_no_visual, paired = TRUE)$p.value,
         p_w = wilcox.test(articles$sharing_with_visual, articles$sharing_no_visual, paired = TRUE)$p.value),
  tibble(ishod = "Vjerodostojnost",
         p_t = t.test(articles$credibility_with_visual, articles$credibility_no_visual, paired = TRUE)$p.value,
         p_w = wilcox.test(articles$credibility_with_visual, articles$credibility_no_visual, paired = TRUE)$p.value)
) |>
  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 Vjerodostojnjost 2.2e-04 3.2e-04
```

Za naše podatke, t-test i Wilcoxon daju konzistentne zaključke. Ovo je čest slučaj kad je $n > 30$ (CLT čini t-test robusnim). Kad rezultati nisu konzistentni, to je signal da treba opreznije interpretirati.

 Kada koristiti neparametrijski test?

Koristite Wilcoxon kad: (1) uzorak je mali ($n < 30$) i distribucija je jasno nenormalna, (2) podaci su ordinalni (Likert skala s malo kategorija), (3) postoje ekstremni outlieri koji iskrivljuju prosjek, (4) želite robusniju analizu kao provjeru (report oba rezultata). Za $n > 50$ s umjerenim normalnim podacima, t-test je gotovo uvijek dobar izbor zahvaljujući CLT-u.

12.9 Efekt vizuala po kategoriji članka

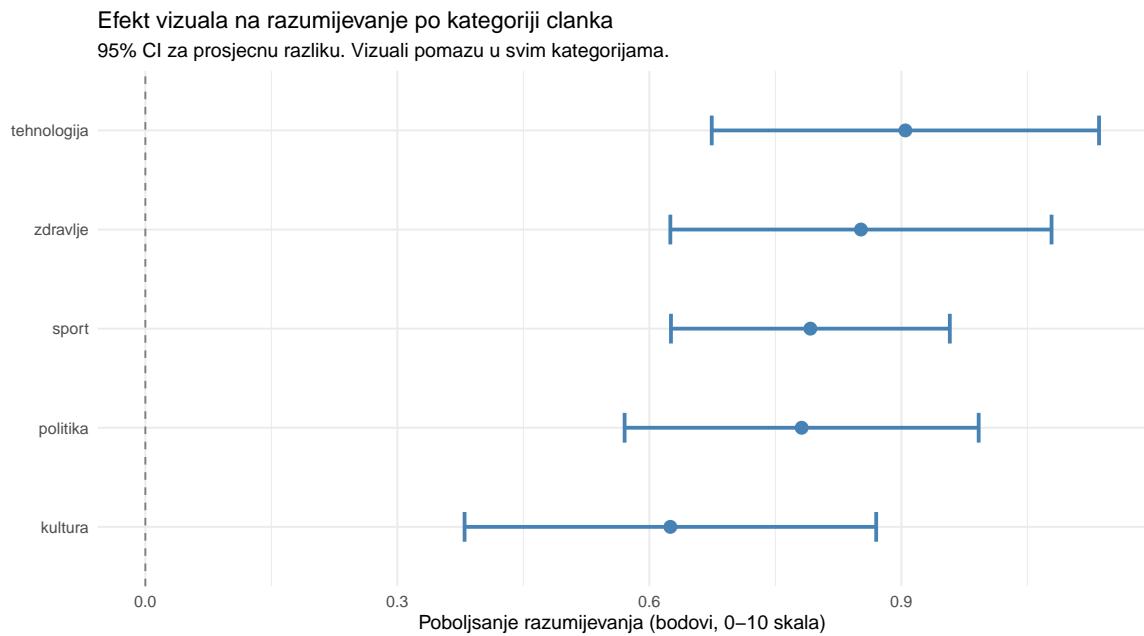
Prosječni efekt vizuala na razumijevanje je velik, ali je li konzistentan za sve teme? Možda vizuali pomažu više kod tehničkih tema nego kod sporta?

```
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)$statistic, 2),
    p = round(t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$p.value, 3),
    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 1          1000 -0.000100    0.000100 -0.000100 0.000100 0.000100
```

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()
```



i Podsjetnik

U prvom dijelu 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 ovom dijelu pokrivamo APA izvještavanje, nezavisni t-test na primjeru, utjecaj outliera i kompletну analizu.

12.10 Izvještavanje u APA formatu

Američka psihološka asocijacija (APA) propisuje standardizirani format za izvještavanje statističkih rezultata koji se koristi u većini društvenih znanosti, uključujući komunikologiju. Format omogućuje čitatelju da brzo razumije što je testirano, koliko je jaki dokaz i koliki je učinak.

12.10.1 Upareni t-test

```
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 = TRUE)
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),
    ")\n", 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 Struktura APA izvještaja

APA format za t-test uvijek sadrži sljedeće elemente: opis smjera razlike riječima, prosjeku i standardne devijacije obiju grupa, vrstu testa s df u zagradi, t-vrijednost zaokruženu na 2 decimale, p-vrijednost (točnu ili " $< .001$ " za vrlo male), mjeru veličine učinka (Cohenov d).

```
# 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-

```
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

Koristite ovu `apa_paired()` funkciju kao predložak i prilagodite je za svoje izvještaje. Konzistentno formatiranje štedi vrijeme i smanjuje greške. Mnogi R paketi (npr. `report`, `papaja`) automatiziraju APA izvještavanje, ali razumijevanje strukture je važnije od korištenja paketa.

12.11 Nezavisni t-test: kratki vs dugi članci

Upareni t-test je bio prikladan za usporedbu uvjeta (s/bez vizuala) jer je svaki članak bio u oba uvjeta. Ali ponekad uspoređujemo dvije odvojene grupe. Na primjer: razlikuje li se razumijevanje između kratkih i dugih članaka?

```
# 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"
  )

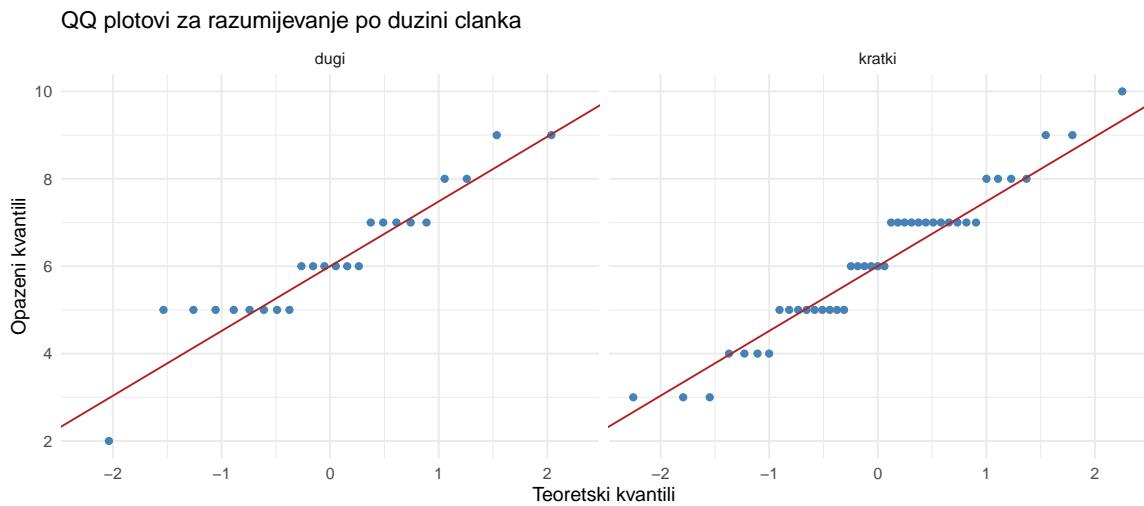
# A tibble: 2 x 5
#>   length_category     n   M_comp   SD_comp   M_time
#>   <chr>           <int>    <dbl>     <dbl>    <dbl>
#> 1 dugi              24     6.12     1.54     6.03
#> 2 kratki            41     6.1      1.69     1.91

# 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()

```



```

kratki <- articles_kd |> filter(length_category == "kratki") |> pull(comprehension_no_visual)
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

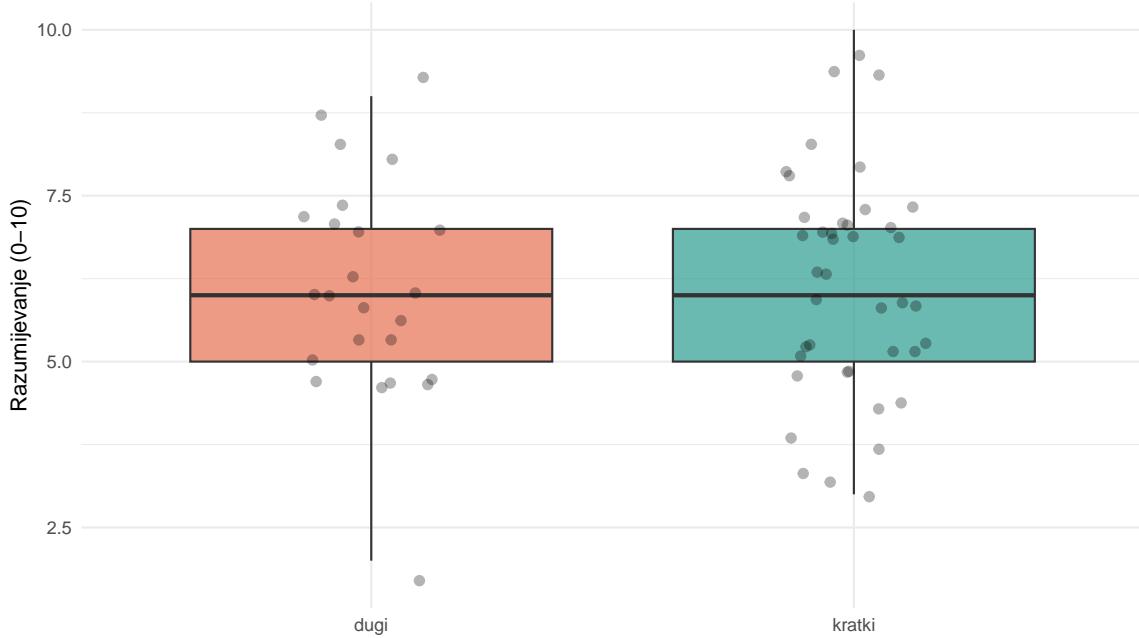
```

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")

```

Razumijevanje po duzini clanka (bez vizuala)

Welchov t-test: $t(51.9) = -0.07$, $p = 0.947$, $d = -0.02$



Ako je $p > 0.05$, to ne znači da dužina članka nema utjecaja. Možda je uzorak premali da detektira razliku, ili je stvarna razlika vrlo mala. Provjerimo snagu testa.

```
# Kolika je snaga ovog testa za detektiranje srednjeg učinka?  
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 učinak (d = 0.5) s n =", min(n1, n2), "po grupi:",  
    round(power_kd$power, 2), "\n")
```

Snaga za srednji učinak ($d = 0.5$) 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 Utjecaj outliera

T-test koristi prosjek koji je osjetljiv na ekstremne vrijednosti. Jedan outlier može značajno promijeniti rezultat.

```
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

```
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

```
cat("S OUTLIEROM:\n")
```

S OUTLIEROM:

```
cat(" M =", round(mean(s_outlierom), 2), ", SD =", round(sd(s_outlierom), 2), "\n")
```

M = 5.71 , SD = 3.79

```
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

Jedan outlier je pomaknuo prosjek i potpuno promijenio SD. Rezultat testa može biti sasvim drugačiji.

12.12.1 Detekcija outliera

```
# 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:

```
cat("Broj outliera:", nrow(outlieri), "od", nrow(articles), "\n\n")
```

Broj outliera: 3 od 120

```
if (nrow(outlieri) > 0) {
  outlieri |>
    select(article_id, category, diff_time, z) |>
    mutate(z = round(z, 2), diff_time = round(diff_time, 2))
}
```

	article_id	category	diff_time	z
1	10	kultura	3.4	5.12
2	31	tehnologija	2.2	2.94
3	62	sport	2.1	2.75

```
# 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 = TRUE)
t_bez_o <- t.test(bez_outliera$reading_time_with_visual, bez_outliera$reading_time_no_visual, paired = TRUE)

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. Ali važno je provjeriti, jer u nekim situacijama može.

12.12.2 Robusne alternative: trimmed means

Umjesto uklanjanja outliera (što je subjektivno), možemo koristiti **trimmed means** (podrezane prosjekе) koji automatski ignoriraju ekstremne vrijednosti.

```
# Obični prosjek vs 10% trimmed mean
cat("Obični prosjek razlika:", round(mean(articles$diff_time), 3), "\n")
```

Obični 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[2], 3), "]")
```

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

1. Identificirajte outliere vizualno (boxplot, histogram) i numerički ($|z| > 2.5$ ili $|z| > 3$).
2. Pokušajte razumjeti zašto su ekstremni (greška u podacima? 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 (trimmed means, bootstrap, Wilcoxon) i izvijestite oboje.

12.13 Formula pristup: t-test s formulom

Do sada smo koristili `t.test(x, y)` sintaksu. R podržava i **formula pristup** koji je elegantniji za nezavisni t-test i konzistentan s budućim modelima (ANOVA, regresija).

```
# 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_vizualima
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 “y ovisi o grupi.” Ova sintaksa će biti ključna na sljedećem predavanju o ANOVI, gdje ćemo uspoređivati više od dvije grupe.

Upozorenje

Formula pristup `t.test(y ~ grupa)` provodi NEZAVISNI t-test. Za upareni test morate koristiti `t.test(x, y, paired = TRUE)` sintaksu. Ne postoji formula pristup za upareni t-test u base R-u. Ovo je česta greška: korištenje formule na uparenim podacima daje pogrešne rezultate jer ignorira parove.

12.14 Potpuna analiza: izvještaj za uredništvo

Spojimo sve u profesionalni izvještaj. Cilj: dati uredništvu jasnu 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_visual),
  round(mean(articles$reading_time_with_visual), 2), round(sd(articles$reading_time_with_visual),
  "Razumijevanje (0-10)",
  round(mean(articles$comprehension_no_visual), 2), round(sd(articles$comprehension_no_visual),
  round(mean(articles$comprehension_with_visual), 2), round(sd(articles$comprehension_with_visual),
  "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),
  "Vjerodostojnjost (1-7)",
  round(mean(articles$credibility_no_visual), 2), round(sd(articles$credibility_no_visual),
  round(mean(articles$credibility_with_visual), 2), round(sd(articles$credibility_with_visual),
))
```

```

# 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

# 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)") )
)

kompletни_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)
  shapiro_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(shapiro_p, 3)
  )
})

```

```

kompletни_reзултати |>
  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
# ... with 9 variables:
#   ishod      <chr>    M_bez     <dbl>    M_s     <dbl>    razlika    <dbl>    t     <dbl>    p_t      <chr>    d     <dbl>    CI_lo     <dbl>    CI_hi     <dbl>
# 1 Vrijeme citanja (min)  "3.43"    3.43    4.02    0.59    11.8    1.1e-21  1.07    0.49    0.69
# 2 Razumijevanje (0-10)  "5.94"    5.94    6.74    0.8     16.1    1.1e-31  1.47    0.7     0.9
# 3 Namjera dijeljenja (1-5)  "2.48"    2.48    3.15    0.67    5.49    2.3e-07  0.5     0.43    0.91
# 4 Vjerodostojnost (1-7)   "4.41"    4.41    4.64    0.23    3.81    2.2e-04  0.35    0.11    0.35

```

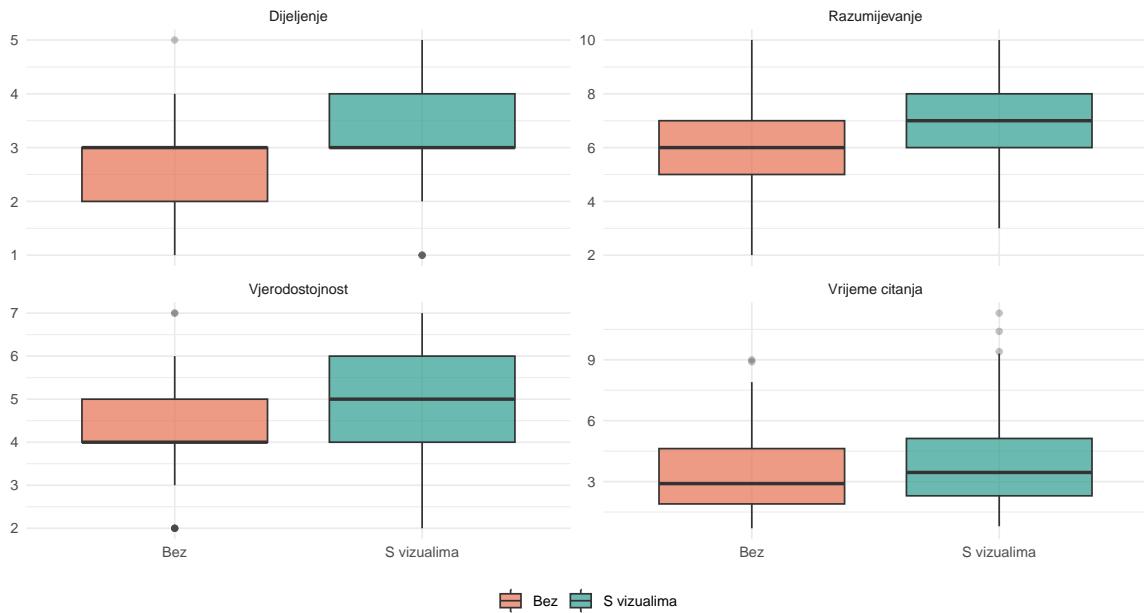
```

# 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")

```

Efekt vizuala na cetiri ishoda

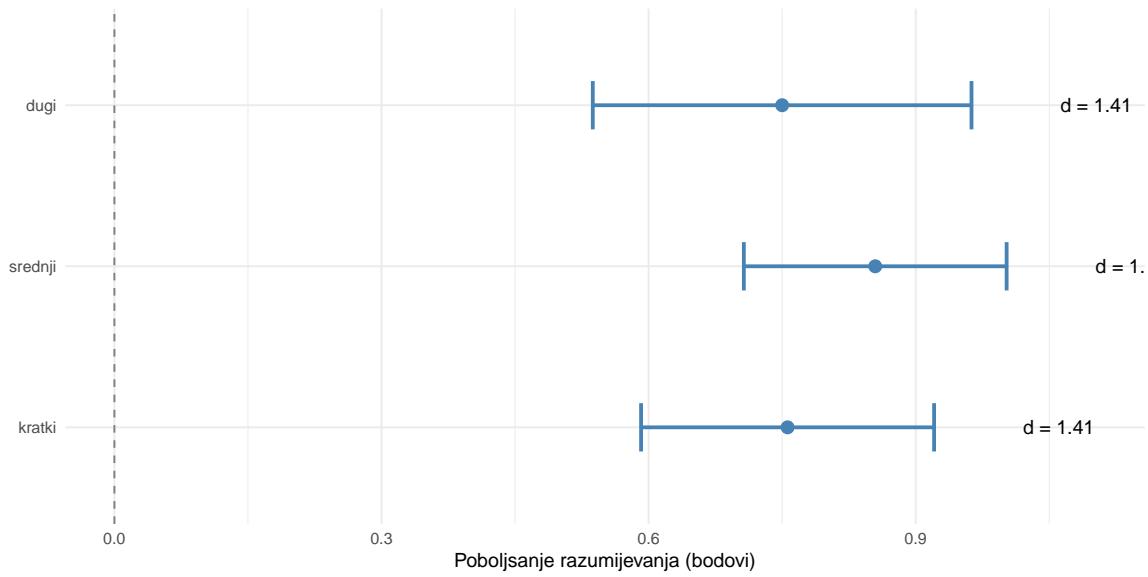
Vizuali poboljsavaju sve ishode. Najaci efekt na razumijevanje.



```
# 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()
```

Efekt vizuala na razumijevanje po duzini clanca

Vizuali pomazu kod svih duzina, s Cohenovim d za svaku kategoriju.

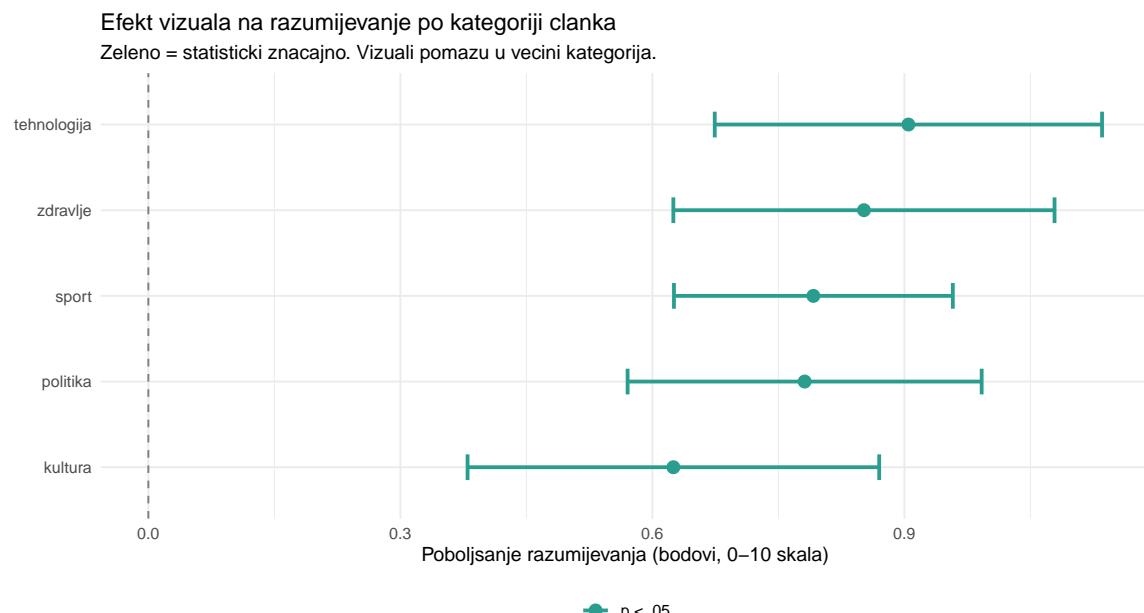


```
# Interakcija: kategorija clanca x uvjet za razumijevanje
articles |>
  group_by(category) |>
  summarise(
    n = n(),
    M_diff_comp = mean(diff_comp),
    SE = sd(diff_comp) / sqrt(n()),
    t = round(t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$stat,
    p = t.test(comprehension_with_visual, comprehension_no_visual, paired = TRUE)$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

```
cat("u dva uvjeta (s vizualima i bez). Cetiri ishoda mjerena.\n\n")
```

u dva uvjeta (s vizualima i bez). Cetiri ishoda mjerena.

```
cat("GLAVNI NALAZI:\n\n")
```

GLAVNI NALAZI:

```
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]

```
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 Dijagram odlučivanja: koji test odabrati?

```
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 mjerena", "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 mjerena       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 Usporedba svih t-testova na jednom mjestu

```
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 = TRUE)",  
  "Primjer", "Proshek 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      Proshek vs norma Muski vs zenski Prije vs poslije
```

! Ključni zaključci

1. Odaberite pravi t-test prema dizajnu: jednouzorački (jedna grupa vs poznata vrijednost), nezavisni (dvije nezavisne grupe), upareni (iste jedinice u dva uvjeta).
2. Prepostavke t-testa: normalnost (ili $n > 30$ za CLT), nezavisnost, intervalna skala. Za nezavisni test: jednake varijance (ili Welchov test kao default).
3. Normalnost provjeravajte vizualno (QQ plot) i formalno (Shapiro-Wilk). Za upareni test, provjeravajte normalnost RAZLIKA. Shapiro-Wilk s velikim n detektira trivijalna odstupanja pa je vizualna procjena jednako važna.
4. Welchov t-test (default u R-u) ne zahtijeva jednake varijance i trebao bi biti standardni izbor za nezavisne uzorke. Nema razloga koristiti Studentov test.
5. Wilcoxonov test je neparametrijska alternativa za mali n s nenormalnim podacima, ordinalne podatke ili podatke s ekstremnim outlierima. Za $n > 50$ s umjerenom normalnosti, t-test je dovoljno robustan.

6. Cohenov d: za upareni test $d = M_{\text{razlika}} / SD_{\text{razlika}}$, za nezavisni test $d = (M_1 \text{ minus } M_2) / s_{\text{pooled}}$. Smjernice: 0.2 mali, 0.5 srednji, 0.8 veliki.
7. APA format: opis smjera riječima, prosjeci i SD obiju grupa, $t(df) = \text{vrijednost, p-vrijednost}$, Cohenov d. `apa_paired()` funkcija automatizira ovaj format.
8. Outlieri mogu dramatično utjecati na rezultate. Identificirajte ih ($|z| > 2.5$), analizirajte ih bez njih, koristite robusne metode (trimmed means, bootstrap) ako se zaključci razlikuju.
9. Formula pristup `t.test(y ~ grupa)` koristi se SAMO za nezavisni test. Ne koristite ga za upareni test jer ignorira parove i daje pogrešne rezultate.
10. Forest plot je standardni način prikaza višestrukih veličina učinka s intervalima pouzdanosti. Idealan za usporedbu efekata na više ishoda u jednom grafu.
11. Snaga testa ovisi o veličini uzorka, veličini učinka i alfa. Uvijek provjerite imate li dovoljno snage prije nego zaključite da "nema razlike." `power.t.test()` računa potreban n.
12. Kompletan izvještaj uključuje dizajn studije, opisnu statistiku, provjeru pretpostavki (normalnost, outlieri), statističke testove s veličinom učinka i CI, moderačijske analize (po kategorijama, po dužini) i praktičnu preporuku.

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 razlike QQ plotom.
 3. Napišite funkciju `kompletни_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. <code>t.test(x, y, paired = TRUE)</code> .
Nezavisni t-test	Test za usporedbu prosjeka dviju nezavisnih grupa. Default u R-u je Welchov. <code>t.test(x, y)</code> .
Jednouzorački t-test	Test za usporedbu jednog prosjeka s poznatom vrijednošću. <code>t.test(x, mu = vrijednost)</code> .
Welchov t-test	Varijanta nezavisnog t-testa koja ne pretpostavlja jednakost 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. <code>shapiro.test(x)</code> . H0: podaci su normalni. S velikim n previše osjetljiv.
QQ plot	Dijagnostički graf: točke blizu linije = normalno. <code>stat_qq() + stat_qq_line()</code> 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. <code>wilcox.test(x, y, paired = TRUE)</code> .

Pojam	Objašnjenje
Mann-Whitney U test	Neparametrijska alternativa nezavisnom t-testu. <code>wilcox.test(x, y)</code> . Isto kao rank-sum test.
Cohenov d (upareni)	$d = M_{\text{razlika}} / SD_{\text{razlika}}$. Standardizirana mjera veličine učinka za uparene podatke.
Cohenov d (nezavisni)	$d = (M_1 \text{ minus } M_2) / s_{\text{pooled}}$. Standardizirana mjera veličine učinka za nezavisne grupe.
Pooled SD	Zajednička SD dviju grupa ponderirana njihovim veličinama uzorka.
Forest plot	Graf za prikaz višestrukih veličina učinka s intervalima pouzdanosti. Standardan u meta-analizama.
Trimmed mean	Projek koji ignorira ekstremne vrijednosti (npr. 10% s oba kraja). <code>mean(x, trim = 0.10)</code> .
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	<code>t.test(y ~ grupa)</code> sintaksa za nezavisni test. Konzistentno s ANOVA i regresijom. NE za upareni test.
Bootstrap	Metoda ponovnog uzorkovanja s vraćanjem za robustan CI. Ne prepostavlja normalnost.
<code>shapiro.test()</code>	R funkcija za Shapiro-Wilkov test normalnosti. Prima vektor numeričkih podataka.
<code>wilcox.test()</code>	R funkcija za Wilcoxonov test. Argumenti isti kao <code>t.test()</code> : mu, paired, alternative.
<code>power.t.test()</code>	R funkcija za analizu snage t-testa. Računa n, snagu ili detektabilni učinak.

13 Tjedan 13: 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 prepostavke ANOVA-e (normalnost, homogenost varijance).
5. Primjeniti post-hoc testove (Tukey HSD) za identifikaciju specifičnih razlika.
6. Izračunati eta-kvadrat kao mjeru veličine učinka za ANOVA-u.
7. Primjeniti Kruskal-Wallisov test kao neparametrijsku alternativu.
8. Provesti kompletну 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

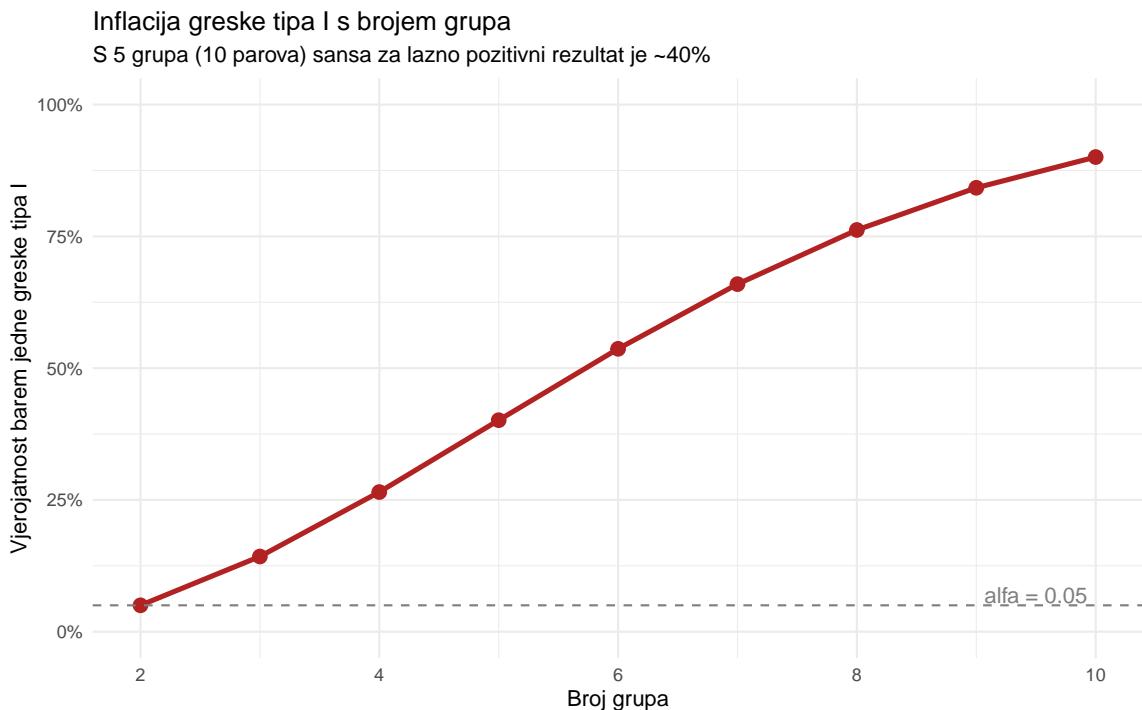
S pet grupa imamo 10 mogućih parova ($5 \text{ choose } 2 = 10$). Ako svaki test provodimo na razini alfa = 0.05, vjerojatnost barem jednog lažno pozitivnog rezultata je:

$$P(\text{barem 1 greska}) = 1 - (1 - 0.05)^{10} = 1 - 0.95^{10} \approx 0.40$$

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 lažno 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)
```

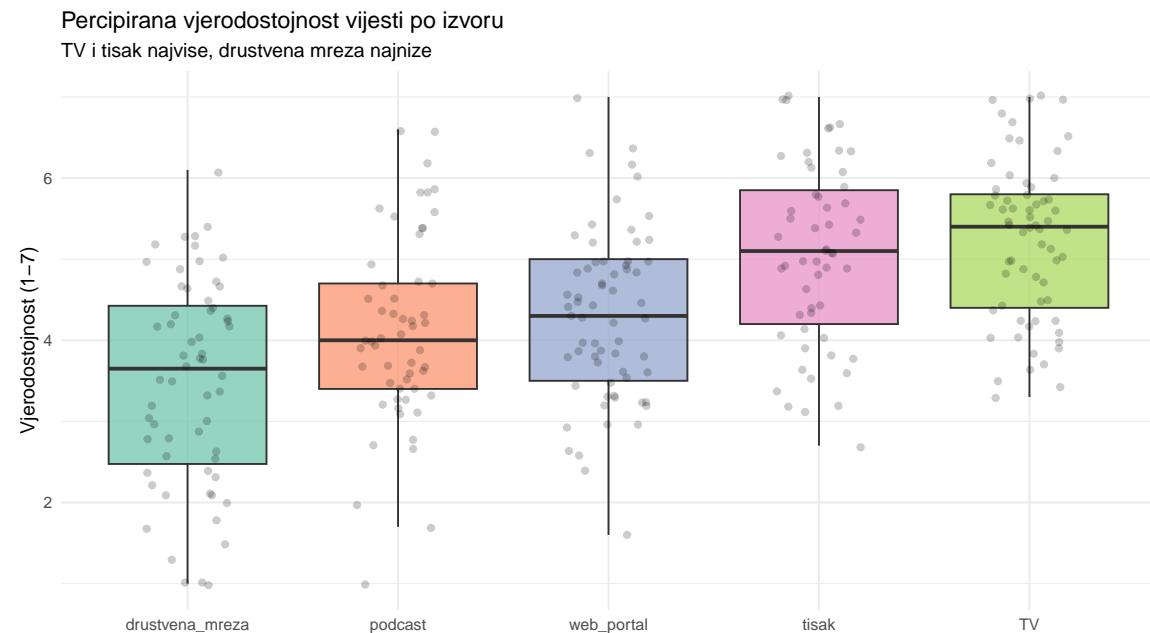
```
cred |>
  group_by(news_source) |>
  summarise(
    n = n(),
    M = round(mean(credibility), 2),
    SD = round(sd(credibility), 2),
    Min = min(credibility),
    Max = max(credibility),
    .groups = "drop"
  ) |>
  arrange(desc(M))
```

```
# 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 uzmemu 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 mreža 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:

$$SS_{total} = SS_{between} + SS_{within}$$

```
# 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

```

cat("SS_between:", round(ss_between, 1), "\n")

SS_between: 122.8

cat("SS_within: ", round(ss_within, 1), "\n")

SS_within: 369.2

cat("Provjera: ", round(ss_between + ss_within, 1), " (treba biti = SS_total)\n\n")

Provjera: 492 (treba biti = SS_total)

# 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

cat("df_within: ", df_within, "\n")

df_within: 295

cat("MS_between:", round(ms_between, 2), "\n")

MS_between: 30.69

```

```
cat("MS_within: ", round(ms_within, 2), "\n")
```

MS_within: 1.25

```
cat("F =", round(f_stat, 2), "\n")
```

F = 24.52

```
cat("p =", format(p_val, scientific = TRUE, digits = 3), "\n")
```

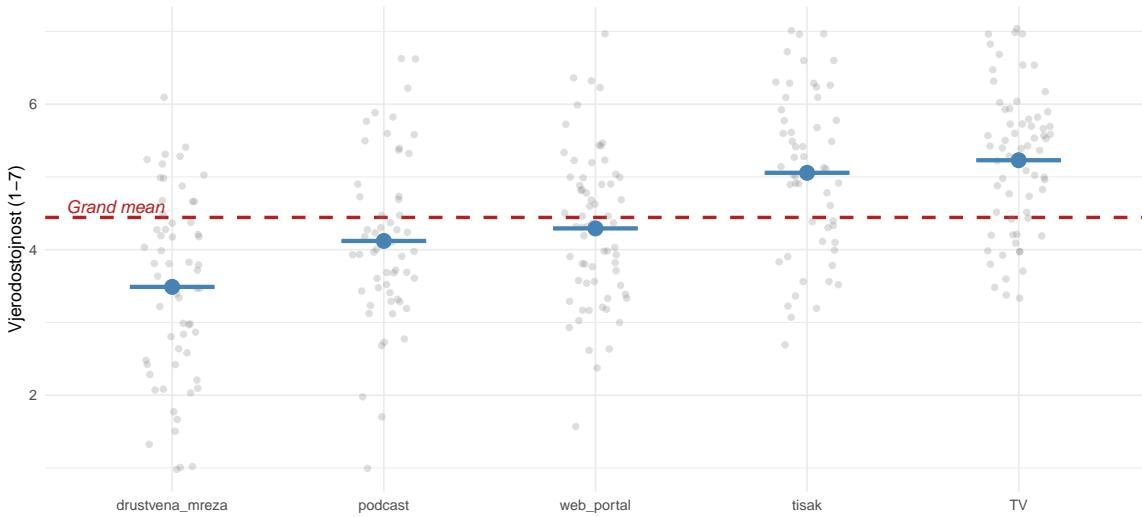
p = 1.55e-17

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.

```
# 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 točke",
    x = NULL, y = "Vjerodostojnjost (1-7)"
  ) +
  theme_minimal()
```

ANOVA logika: razlike između grupnih prosjeka vs varijabilnost unutar grupa
 Plavi crossbar = grupni prosjek. Crvena linija = ukupni prosjek. Sive točke = pojedinci.



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: df, SS (Sum Sq), MS (Mean Sq), F vrijednost i p-vrijednost (Pr(>F)). Rezultat potvrđuje: postoji statistički značajna razlika u percipiranoj vjerodostojnosti ovisno o izvoru vijesti.

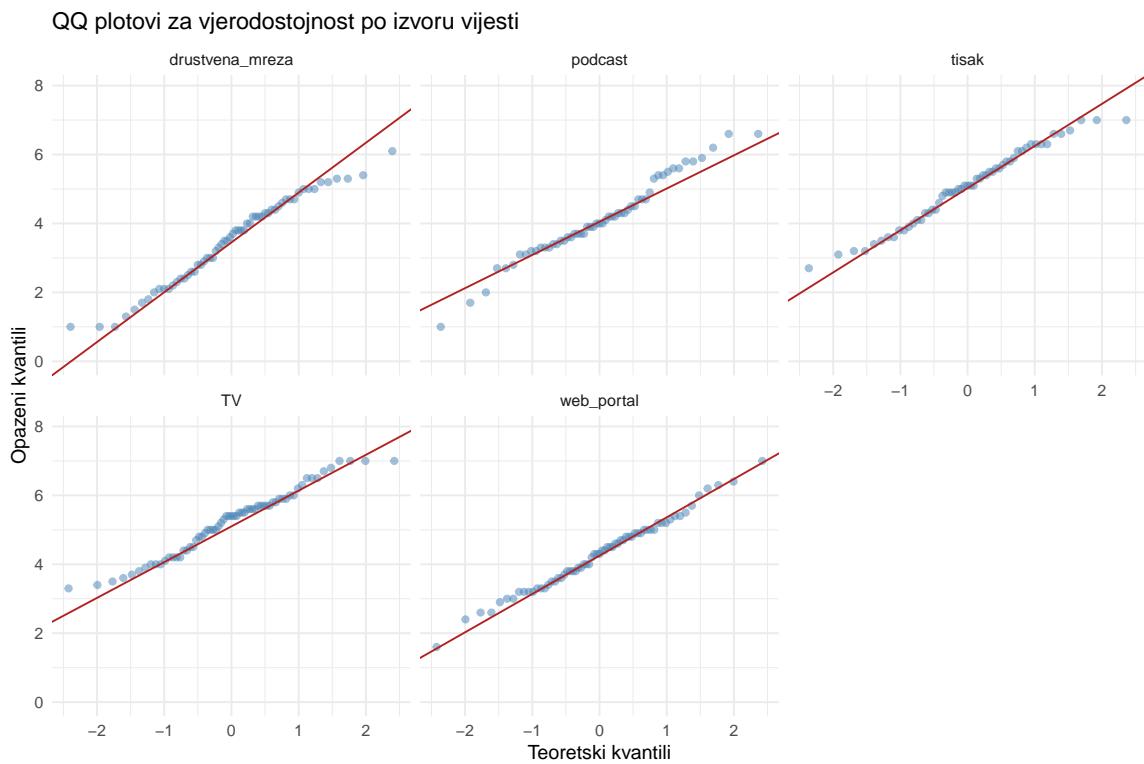
Ali ANOVA je **omnibus test**: govori da se barem dvije grupe razlikuju, ali ne govori KOJE. Za to trebamo post-hoc testove (drugi dio).

13.5 Prepostavke ANOVA-e

ANOVA ima tri prepostavke: nezavisnost opažanja (dizajn istraživanja), normalnost distribucije unutar svake grupe (ili dovoljno velik n), 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()
```



```

# 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

```

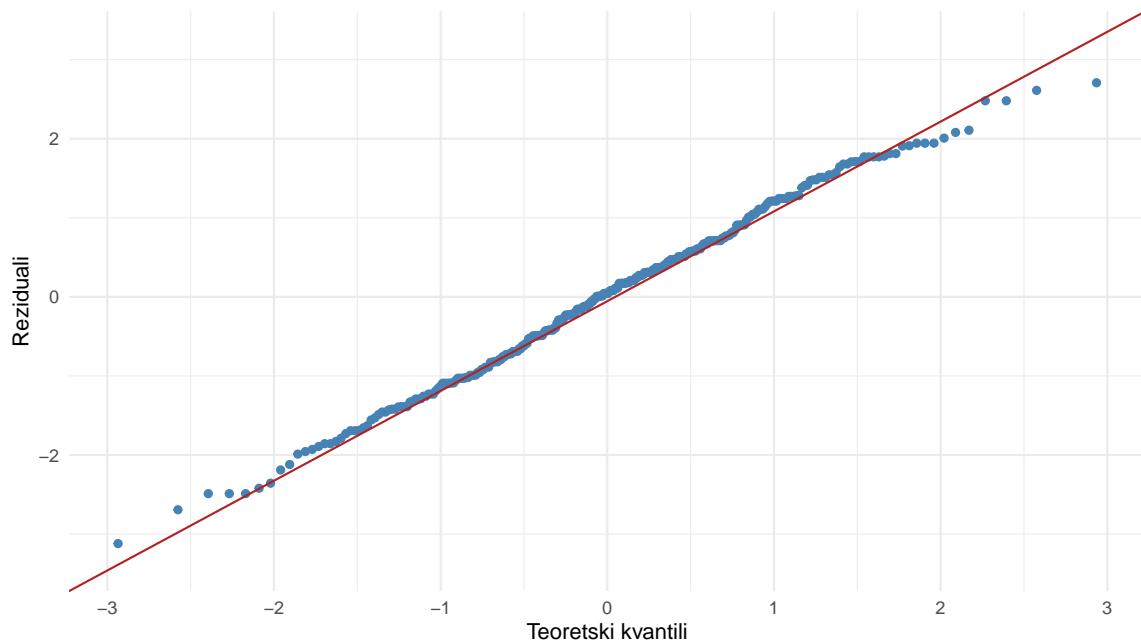
Neke grupe možda ne prolaze Shapiro-Wilk test. Ali s $n > 50$ po grupi, ANOVA je robustna na umjerena odstupanja od normalnosti (CLT). Također možemo provjeriti normalnost **reziduala** modela:

```

# 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()

```

QQ plot reziduala ANOVA modela
 Ako su reziduali normalni, pretpostavka je zadovoljena



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

```
cat("Pravilo palca: ako je omjer < 3, homogenost je prihvatljiva.\n")
```

Pravilo palca: ako je omjer < 3, homogenost je prihvatljiva.

```
# 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):

```
cat("p =", round(levene_p, 4), "\n")
```

p = 0.1895

```
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 prepostavlja jednake varijance:

```
# Welchova ANOVA (ne prepostavlja 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

# 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 zaključak.\n")
```

Oba pristupa daju isti zaključak.

💡 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.

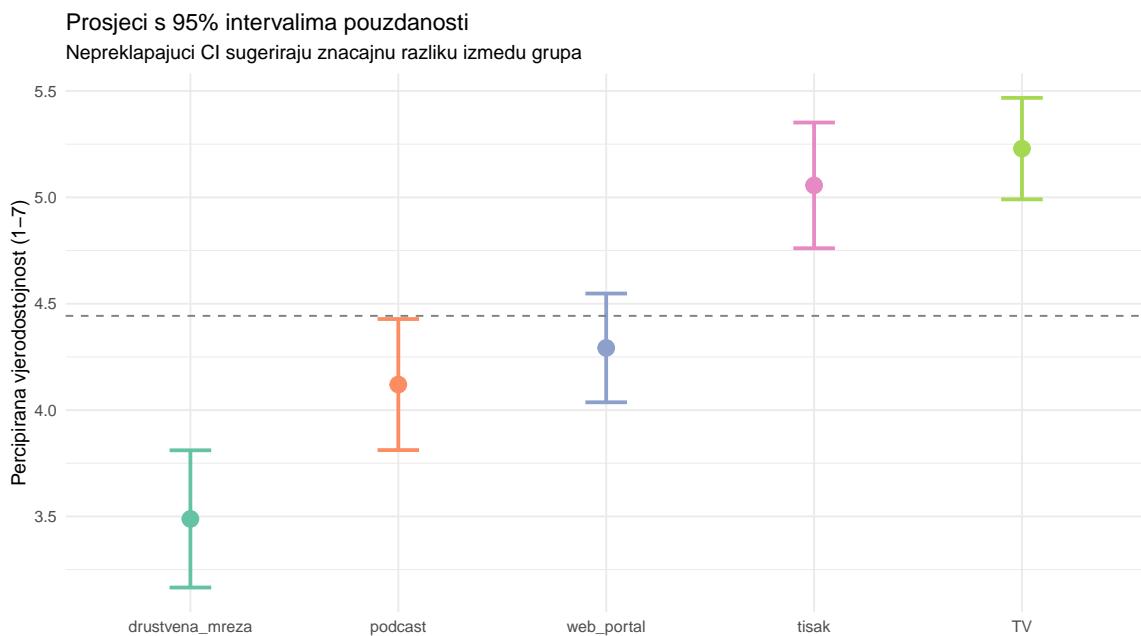
13.6 Vizualizacija ANOVA rezultata

```
# 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 = "Nepreklapajući CI sugeriraju znacajnu razliku između grupa",
    x = NULL,
    y = "Percipirana vjerodostojnost (1-7)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

```

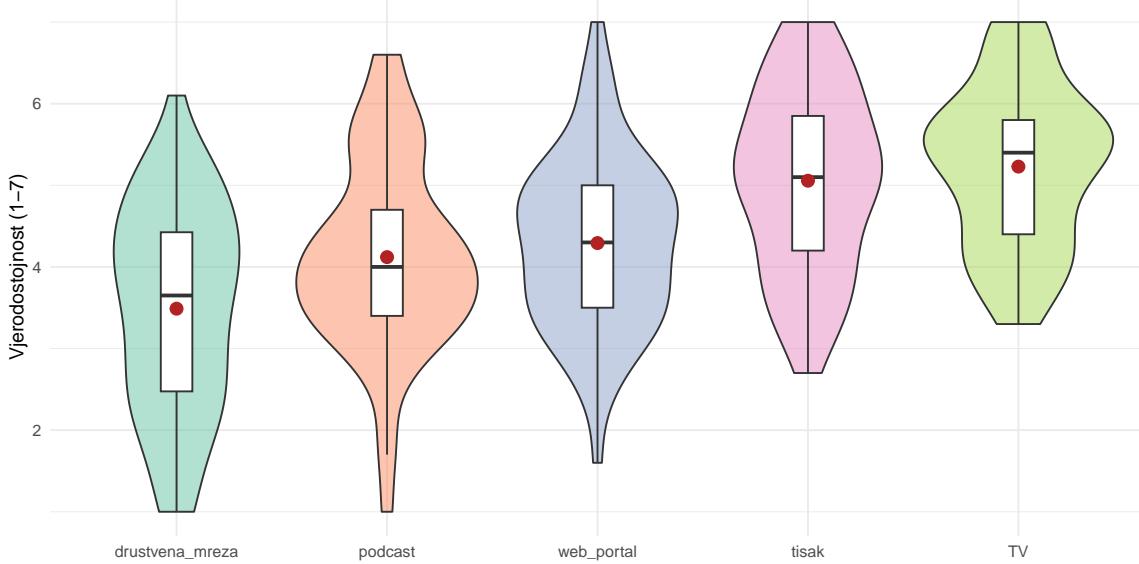


```

# 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 točka = prosjek. Bijeli box = medijan",
    x = NULL,
    y = "Vjerodostojnost (1-7)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

```

Distribucija vjerodostojnosti po izvoru
 Violin = oblik distribucije. Crvena točka = prosjek. Bijeli box = medijan i IQR.



i Podsjetnik

U prvom dijelu naučili smo logiku ANOVA-e (F = varijabilnost između / varijabilnost unutar), dekompoziciju varijance ($SS_{total} = SS_{between} + SS_{within}$), prepostavke (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

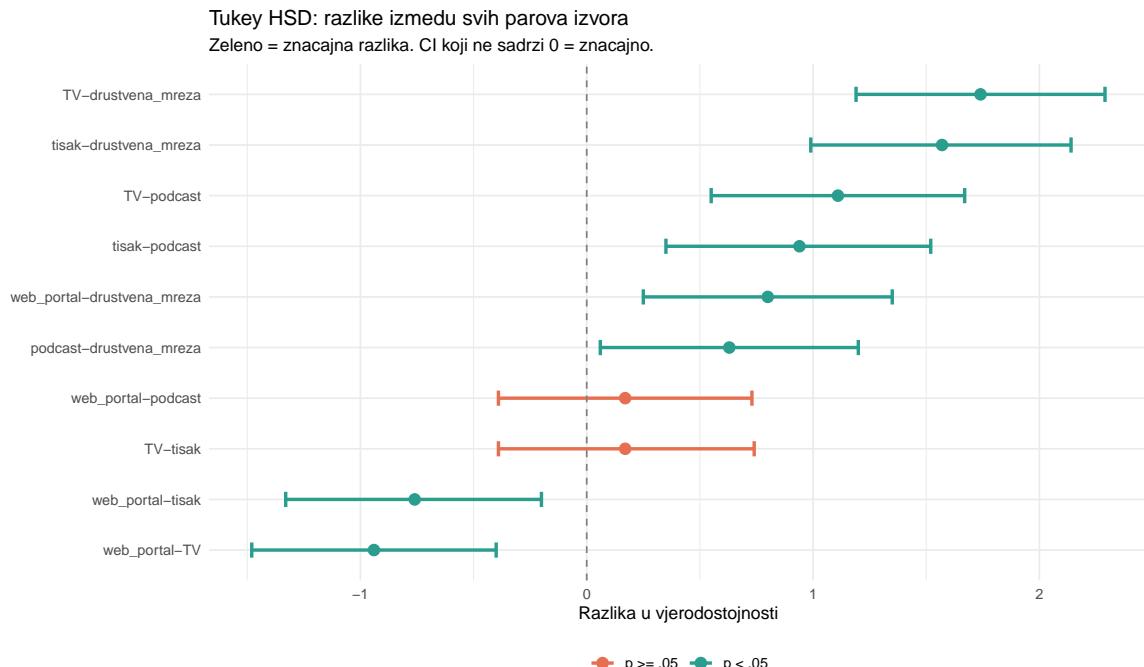
```
# Preglednja tablica
tukey_df <- as_tibble(tukey_rez$news_source, rownames = "par") |>
  mutate(
    razlika = round(diff, 2),
    CI_lo = round(lwr, 2),
    CI_hi = round(upr, 2),
    p = round(`p adj`, 4),
    znacajno = p < 0.05
  ) |>
  select(par, razlika, CI_lo, CI_hi, p, znacajno) |>
  arrange(p)

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 = značajna razlika. CI koji ne sadrži 0 = značajno.",
    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 dijeli 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 mjeru 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 smjernice:

```
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 točnija procjena.\n")
```

Razlika je mala za velike uzorke, ali omega2 je točnija 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_mediya = if_else(
    news_source %in% c("TV", "tisak"), "tradicionalni", "digitalni"
  ))

t.test(credibility ~ tip_mediya, data = cred)
```

```
Welch Two Sample t-test

data: credibility by tip_mediya
t = -8.9741, df = 278.64, p-value < 2.2e-16
alternative hypothesis: true difference in means between group digitalni and group tradicionalni
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_mediya == "tradicionalni") |> pull(credibility)
dig <- cred |> filter(tip_mediya == "digitalni") |> pull(credibility)
s_p <- sqrt(((length(trad)-1)*sd(trad)^2 + (length(dig)-1)*sd(dig)^2) / (length(trad)+length(dig)))
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 zaključak.\n")
```

Oba testa daju isti zaključak.

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).

```
# 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
```

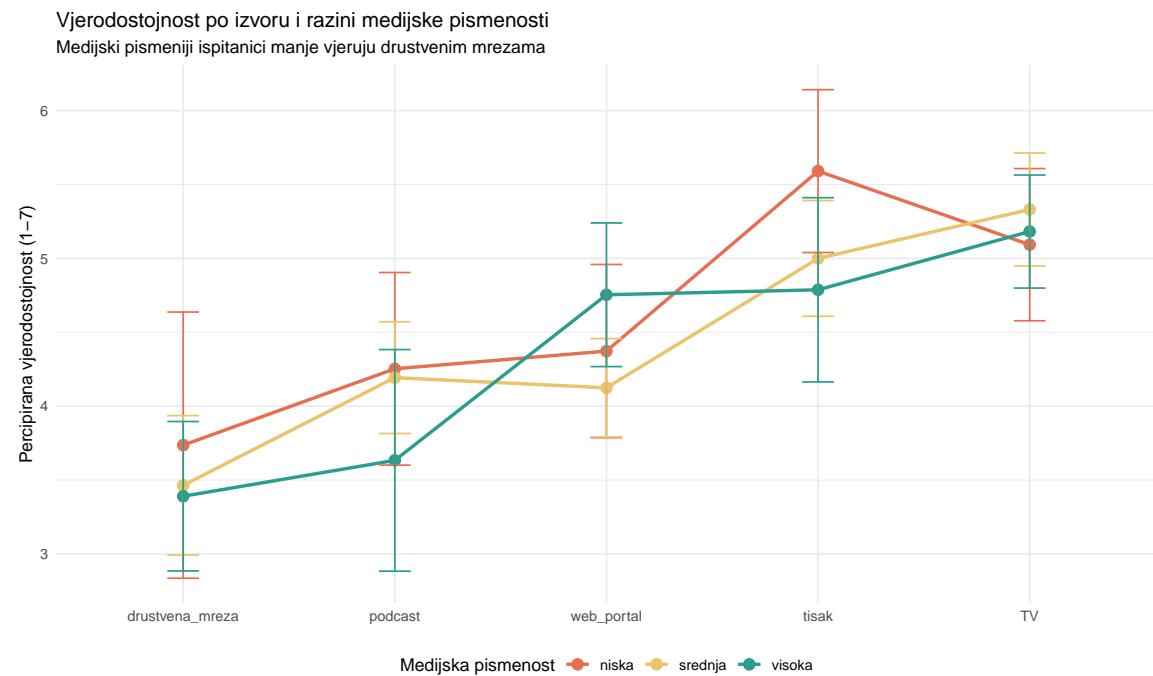
13.11 Potpuna analiza: izvještaj

```
# 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

```
# 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" = "#2ca02c")) +
  labs(
    title = "Vjerodostojnjost po izvoru i razini medijske pismenosti",
    subtitle = "Medijski pismeniji ispitanici manje vjeruju drustvenim mrezama",
    x = NULL,
    y = "Percipirana vjerodostojnjost (1-7)",
    color = "Medijska pismenost"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```

# 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.

```
cat("=====\\n")
```

```
=====
```

```
cat("  APA IZVJESTAJ: PERCIPIRANA VJERODOSTOJNOST PO IZVORU\\n")
```

APA IZVJESTAJ: PERCIPIRANA VJERODOSTOJNOST PO IZVORU

```
cat("=====\\n\\n")
```

```
=====
```

```
cat("Jednosmjerna ANOVA pokazala je statisticki znacajnu razliku u\\n")
```

Jednosmjerna ANOVA pokazala je statisticki znacajnu razliku u

```
cat("percipiranoj vjerodostojnosti vijesti ovisno o izvoru,\n")
```

percipiranoj vjerodostojnosti vijesti ovisno o izvoru,

```
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.

```
cat("Tukey HSD post-hoc testovi pokazali su da:\n")
```

Tukey HSD post-hoc testovi pokazali su da:

```
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

```
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.

```
cat("Medijska pismenost moderira efekt: visoko pismeni\n")
```

Medijska pismenost moderira efekt: visoko pismeni

```
cat("ispitanici manje vjeruju vijestima s drustvenih mreza.\n")
```

ispitanici manje vjeruju vijestima s drustvenih mreza.

13.12 Dijagram odlučivanja: ANOVA ili nešto drugo?

```
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, nejednakе 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, nejednakе 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 prosjekе tri ili više grupa jednim testom, kontrolirajući grešku tipa I. Višestruki t-testovi inflacioniraju alfa i nisu prihvatljivi.
2. F statistika = $MS_{\text{between}} / MS_{\text{within}}$. Velik F znači da su razlike između grupa veće od varijabilnosti unutar grupe.
3. Dekompozicija varijance: $SS_{\text{total}} = SS_{\text{between}} + SS_{\text{within}}$. 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 prilagodene p-vrijednosti za svaki par.
6. Eta-kvadrat = $SS_{\text{between}} / SS_{\text{total}}$. Proporcija varijabilnosti objašnjena grupom. Smjernice: 0.01 mali, 0.06 srednji, 0.14 veliki učinak. Omega-kvadrat je manje pristrana alternativa.
7. Pretpostavke: nezavisnost, normalnost unutar grupe (ili $n > 30$), homogenost varijance. Welchova ANOVA (`oneway.test(var.equal = FALSE)`) ne zahtijeva jednakе 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: $F(df_{\text{between}}, df_{\text{within}}) = \text{vrijednost}, p, \eta^2$. 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 mjerjenja 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: 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 $MS_{\text{between}} / MS_{\text{within}}$. $F \gg 1$ sugerira značajne razlike između grupa.
SS (Sum of Squares)	Mjera varijabilnosti. $SS_{\text{total}} = SS_{\text{between}} + SS_{\text{within}}$.

Pojam	Objašnjenje
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. $SS_{between} / SS_{total}$. Proporcija varijabilnosti objašnjena grupnom pripadnošću. $0.01/0.06/0.14$ mali/srednji/veliki.
Eta-kvadrat	Manje pristrana alternativa eta-kvadratu. Preporučena za publikacije.
Omega-kvadrat	Porast greške tipa I pri višestrukim usporedbama. ANOVA to kontrolira.
Inflacija alfa	Prepostavka jednakih varijanci u svim grupama. Levenov test je provjerava. <code>oneway.test(var.equal = FALSE)</code> .
Homogenost varijance	ANOVA koja ne zahtijeva jednake varijance. Neparametrijska alternativa jednosmjernoj ANOVA-i. Koristi rangove.
Welchova ANOVA	<code>kruskal.test()</code> .
Kruskal-Wallisov test	Post-hoc test za Kruskal-Wallis. Neparametrijski ekvivalent Tukeyu.
Dunnov test	Unaprijed definirane specifične usporedbe. Snažniji od post-hoc testova jer testiraju manje hipoteza.
Planirane usporedbe	R funkcija za klasičnu ANOVA-u. <code>aov(y ~ grupa, data = ...)</code> .
<code>aov()</code>	R funkcija za Tukey post-hoc test. Prima <code>aov()</code> objekt.
<code>TukeyHSD()</code>	R funkcija za Welchovu ANOVA-u. <code>var.equal = FALSE</code> za robusnu verziju.
<code>oneway.test()</code>	R funkcija za Kruskal-Wallisov test. Sintaksa kao <code>aov()</code> .
<code>kruskal.test()</code>	R funkcija za pairwise Wilcoxon testove s korekcijom p-vrijednosti.
<code>pairwise.wilcox.test()</code>	

14 Tjedan 14: Linearna regresija

Predvidjanje i objašnjavanje s modelima

```
library(tidyverse)
```

i 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, prilagodenog R-kvadrata i AIC-a.
7. Prepoznati uobičajene probleme (multikolinearnost, utjecajne točke, nelinearnost).
8. Napisati kompletan izvještaj regresijske analize.

14.1 Motivacija: što pokreće angažman?

Social media menadžerica želi znati: koji faktori utječu na angažman Instagram objava? Duljina teksta? Broj hashtagova? Tip sadržaja? Vrijeme objave? I koliko dobro možemo predvidjeti angažman na temelju tih faktora?

ANOVA (prošli tjedan) odgovara na pitanje "razlikuju li se grupe." Regresija ide dalje: modelira odnos između jedne ili više nezavisnih varijabli (prediktora) i zavisne varijable (ishoda), kvantificira snagu i smjer svakog odnosa, i omogućuje predvidjanje novih opažanja.

```
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", "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, 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

Korelacija mjeri jačinu i smjer linearne veze između dviju varijabli. Regresija ide korak dalje: definira jednadžbu pravca koja opisuje tu vezu.

```
# Korelacijske matrice numeričkih prediktora s engagement_rate
posts |>
  select(engagement_rate, text_length, num_hashtags, num_mentions, followers) |>
  cor() |>
  round(3)
```

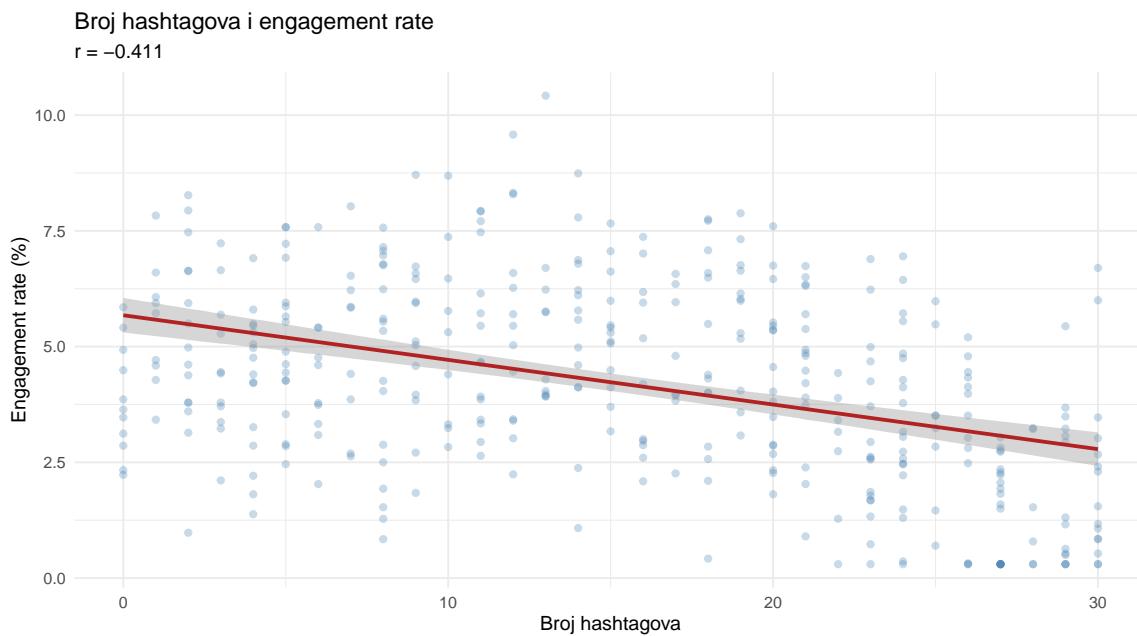
	engagement_rate	text_length	num_hashtags	num_mentions	followers
engagement_rate	1.000	0.040	-0.411	-0.016	-
text_length	0.040	1.000	0.008	0.061	-
num_hashtags	-0.411	0.008	1.000	0.031	0.062
num_mentions	-0.016	0.061	0.031	1.000	0.092
followers	-0.132	-0.025	0.062	0.092	1.000

```
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()

```



Korelacija između broja hashtagova i engagement ratea je negativna ($r = -0.41$). Ali pozor: ovo ne znači nužno da više hashtagova uzrokuje niži angažman. Možda je veza nelinearna (optimalni broj hashtagova postoji). Ovo ćemo istražiti u dijagnostici.

14.3 Jednostavna linearna regresija

Jednostavna linearna regresija modelira vezu jednog prediktora (X) i ishoda (Y):

$$Y = b_0 + b_1 X + \varepsilon$$

gdje je b_0 odsječak (intercept, vrijednost Y kad je X = 0), b_1 nagib (slope, promjena Y za svaku jedinicu promjene u X), a ε greška (rezidual, ono što model ne objašnjava).

```

# 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 Čitanje regresijskog outputa

```

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.

R-kvadrat je nizak jer duljina teksta sama nije dobar prediktor angažmana. Trebamo više prediktora (višestruka regresija).

14.3.2 Vizualizacija regresijskog pravca

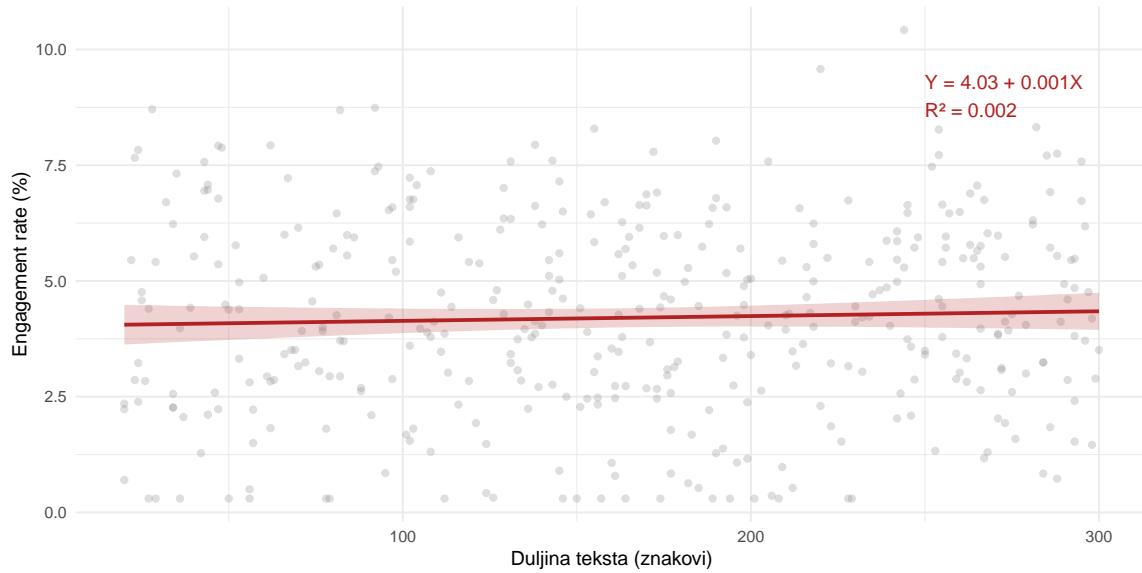
```

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 = 0.25)
  annotate("text", x = 250, y = 9,
           label = paste0("Y = ", round(koef[1], 2), " + ", round(koef[2], 4), "X\nR² = ",
                         round(r2, 2)),
           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()

```

Jednostavna linearna regresija: duljina teksta i angažman

Sivi pojas = 95% CI za regresijski pravac



14.4 Što su reziduali?

Reziduali su razlike između opaženih i predviđenih vrijednosti:

$$e_i = Y_i - \hat{Y}_i$$

Regresija minimizira sumu kvadriranih reziduala (OLS, Ordinary Least Squares). Reziduali nam govore koliko model grijesi za svako pojedino opažanje.

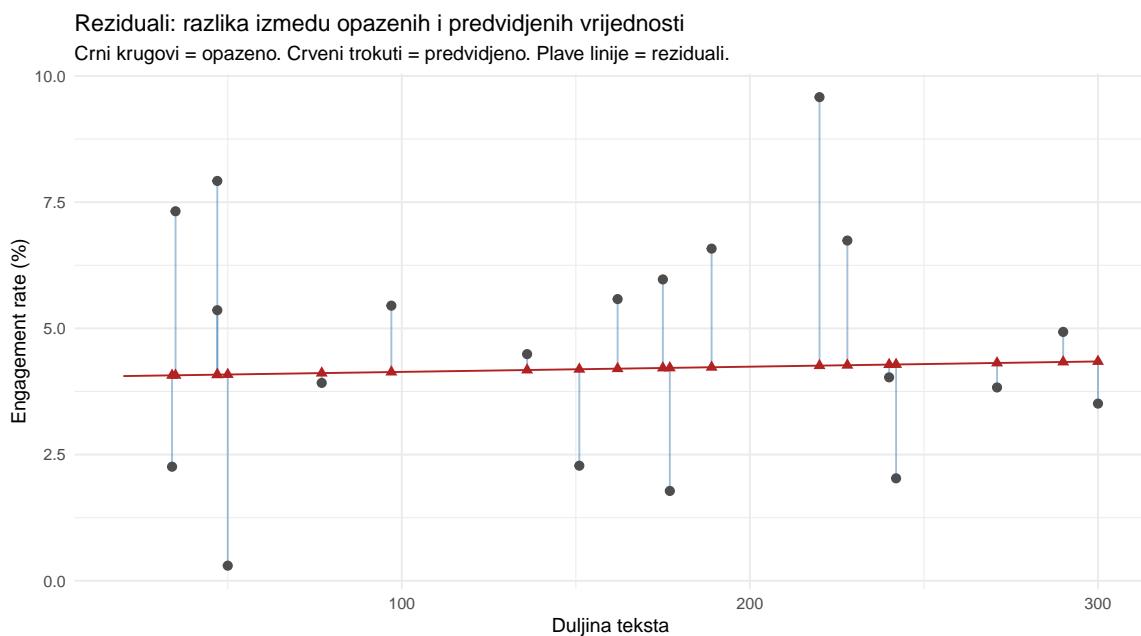
```
# Dodajmo predviđene vrijednosti i reziduale u podatke
posts_pred <- posts |>
  mutate(
    predicted = predict(model1),
    residual = residuals(model1)
  )

# 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 = 0.5)
labs(
  title = "Reziduali: razlika izmedu opazenih i predvidjenih vrijednosti",
  subtitle = "Crni krugovi = opazeno. Crveni trokuti = predvidjeno. Plave linije = reziduali",
  x = "Duljina teksta",
  y = "Engagement rate (%)"
) +
theme_minimal()

```



14.5 Prepostavke linearne regresije

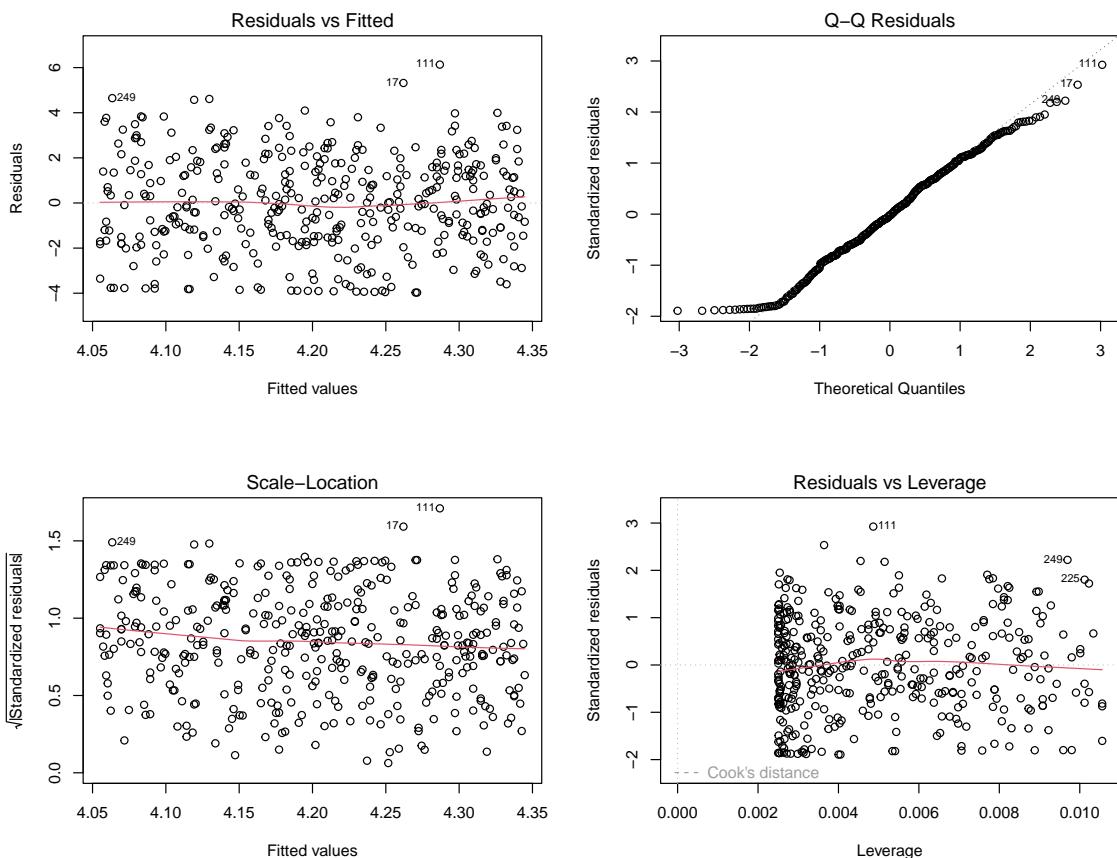
Linearna regresija ima četiri ključne prepostavke: linearnost (veza između X i Y je linearna), nezavisnost reziduala, homoskedastičnost (jednaka varijanca reziduala za sve vrijednosti X) i normalnost reziduala.

14.5.1 Dijagnostički grafovi

```

par(mfrow = c(2, 2))
plot(model1)

```



```
par(mfrow = c(1, 1))
```

Četiri standardna dijagnostička grafa:

Residuals vs Fitted (gore lijevo): Provjerava linearnost i homoskedastičnost. Želimo ravnu crvenu liniju oko nule i ravnomjerno raspršene točke. Ako vidimo oblik lijevka ili krivulju, pretpostavka je narušena.

Normal Q-Q (gore desno): Provjerava normalnost reziduala. Točke trebaju ležati blizu dijagonale.

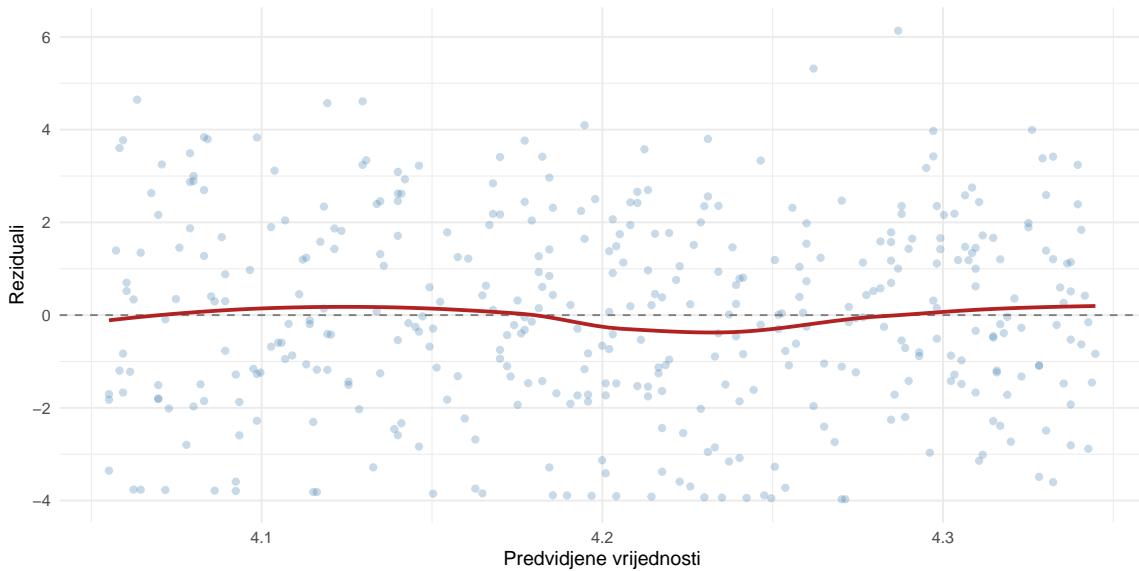
Scale-Location (dolje lijevo): Provjerava homoskedastičnost. Želimo ravnu liniju i ravnomjerno raspršene točke.

Residuals vs Leverage (dolje desno): Identificira utjecajne točke. Opažanja s visokim leverageom i velikim rezidualima (izvan Cookovih udaljenosti) mogu neprimjereno utjecati na model.

```
# 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 = linearost zadovoljena. Lijevak = heteroskedastnost",
    x = "Predvidjene vrijednosti",
    y = "Reziduali"
  ) +
  theme_minimal()
```

Reziduali vs predvidjene vrijednosti

Crvena linija blizu nule = linearost zadovoljena. Lijevak = heteroskedasticnost.



14.6 Višestruka regresija

Jednostavna regresija s jednim prediktorom rijetko je dovoljna. **Višestruka regresija** uključuje više prediktora:

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_k X_k + \varepsilon$$

Svaki koeficijent b_i sada predstavlja **parcijalni efekt**: promjenu Y za jediničnu promjenu X_i , uz kontrolu svih ostalih prediktora (držeći ih konstantnima).

```
# 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

```
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 ===

```
cat("Jednadžba:\\n")
```

Jednadžba:

```

cat("engagement = ", round(koef2[1], 2), "\n", sep = "")

engagement = 5.31

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("Prilagoden R-kvadrat:", round(adj_r2_m2, 3), "\n")
```

Prilagoden 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.

14.6.1 Usporedba modela

```
# 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, m 0.183  0.175 1659.
3 M3: + content_type      0.376  0.365 1558.
4 M4: + topic              0.406  0.389 1546.
```

Svaki dodani prediktor povećava R-kvadrat. **Prilagođeni R-kvadrat** penalizira dodavanje prediktora koji ne poboljšavaju model dovoljno. **AIC** (Akaike Information Criterion) je još jedna mjera: niži AIC = bolji model.

```
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

```

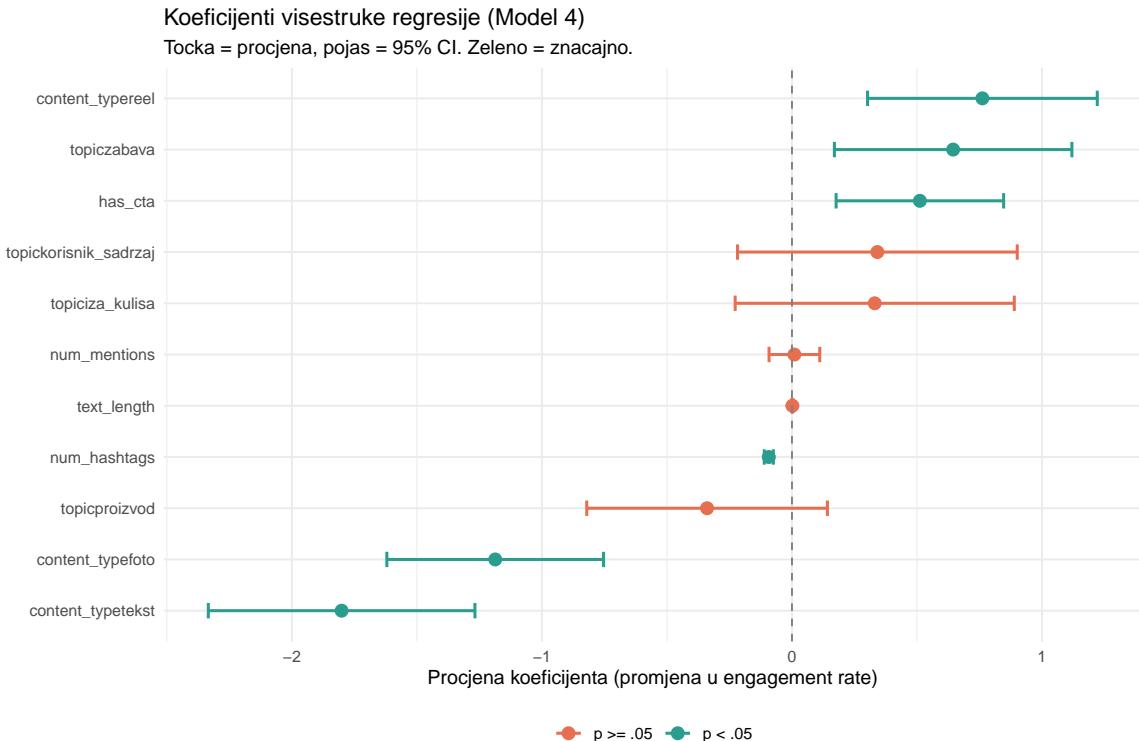
Residual standard error: 1.642 on 388 degrees of freedom
 Multiple R-squared: 0.4061, Adjusted R-squared: 0.3892
 F-statistic: 24.12 on 11 and 388 DF, p-value: < 2.2e-16

```

# 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)
  )

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 prilagodeni R-kvadrat

R-kvadrat (R^2) je proporcija varijabilnosti Y-a objašnjena modelom:

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} = \frac{SS_{model}}{SS_{total}}$$

Problem: R^2 uvijek raste (ili ostaje isti) kad dodamo prediktor, čak i ako je beskoristan.

Prilagodeni R^2 penalizira dodavanje nepotrebnih prediktora:

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1}$$

gdje je n broj opažanja, a k broj prediktora.

```
# 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_demo)
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

```
cat("Model S random prediktorima:\n")
```

Model S random prediktorima:

```
cat("  R2 =", round(summary(model_rand)$r.squared, 4), "(veci! ali lazno)\n")
```

R2 = 0.3807 (veci! ali lazno)

```
cat("  Adj R2 =", round(summary(model_rand)$adj.r.squared, 4), "(korigira za lazno poboljsanje)\n")
```

Adj R2 = 0.368 (korigira za lazno poboljsanje)

! Važna napomena o R-kvadratu

R^2 nije "ocjena" modela. $R^2 = 0.20$ može biti odličan za predviđanje ljudskog ponašanja (koje je inherentno varijabilno) a $R^2 = 0.90$ može biti loš za fizikalni zakon. Kontekst je ključan. U komunikološkim istraživanjima, R^2 između 0.10 i 0.30 je uobičajen i prihvatljiv.

Podsjetnik

U prvom dijelu naučili smo jednostavnu i višestruku linearnu regresiju, interpretaciju koeficijenata, R-kvadrat, dijagnostičke grafove i usporedbu modela. U ovom dijelu pokrivamo naprednu dijagnostiku, standardizirane koeficijente i potpunu analizu.

14.8 Multikolinearnost i VIF

Multikolinearnost nastaje kad su prediktori međusobno jako korelirani. Kad je to slučaj, model teško razdvaja njihove individualne efekte, koeficijenti postaju nestabilni i standardne greške narastu.

VIF (Variance Inflation Factor) mjeri koliko je varijanca koeficijenta povećana zbog korelacije s drugim prediktorima.

```
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, data = posts)

# 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
```

Pravilo palca: $VIF < 5$ je prihvatljiv. $VIF > 10$ ukazuje na ozbiljnu multikolinearnost. Naši prediktori imaju niske VIF-ove, što znači da su relativno nezavisni jedni od drugih.

⚠ Što učiniti kad je VIF visok?

Opcije su: ukloniti jedan od koreliranih prediktora, kombinirati korelirane prediktore u jednu mjeru (npr. prosjek ili faktor), koristiti regulariziranu regresiju (ridge, lasso), ili prihvatiti šire CI i opreznije interpretirati.

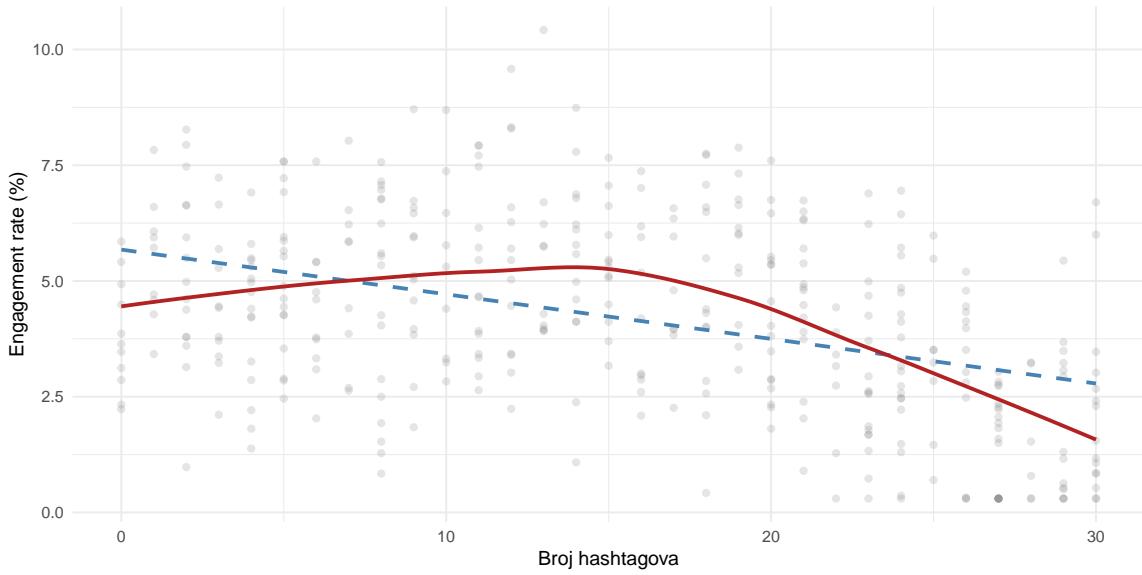
14.9 Nelinearni odnosi

Linearna regresija prepostavlja linearne odnose. Ali stvarni odnosi često nisu linearni. Na primjer, broj hashtagova i angažman: previše hashtagova može biti jednako loše kao premalo.

```
# 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 sugerira nelinearnost",
    x = "Broj hashtagova",
    y = "Engagement rate (%)"
  ) +
  theme_minimal()
```

Linearni vs nelinearni odnos: hashtagovi i angažman

Plava = linearni model. Crvena = LOESS (fleksibilni). Oblik obrnuto U sugerira nelinearnost.



LOESS krivulja sugerira obrnuto U: angažman je najveći oko 8-12 hashtagova i opada s previše ili premalo. Linearni model ovo ne može uhvatiti.

14.9.1 Polinomijalna regresija

Možemo dodati kvadratni član da uhvatimo zakrivljenost:

```
# Model s kvadratnim clanom za hashtagove
model_poly <- lm(engagement_rate ~ num_hashtags + I(num_hashtags^2) +
                  has_cta + content_type + topic, data = posts)

# 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")

cat("Polinomijalni model: Adj R2 =", round(summary(model_poly)$adj.r.squared, 3),
    ", AIC =", round(AIC(model_poly), 1), "\n")
```

Linearni model: Adj R² = 0.388 , AIC = 1544.4

```
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

```

# Koeficijenti za hashtag efekt
koef_poly <- coef(model_poly)
cat("num_hashtags:    ", round(koef_poly["num_hashtags"], 4), "\n")

num_hashtags:    0.2033

cat("num_hashtags^2:  ", round(koef_poly["I(num_hashtags^2)"], 5), "\n\n")

num_hashtags^2:  -0.00977

# 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

```

# 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())))
  )

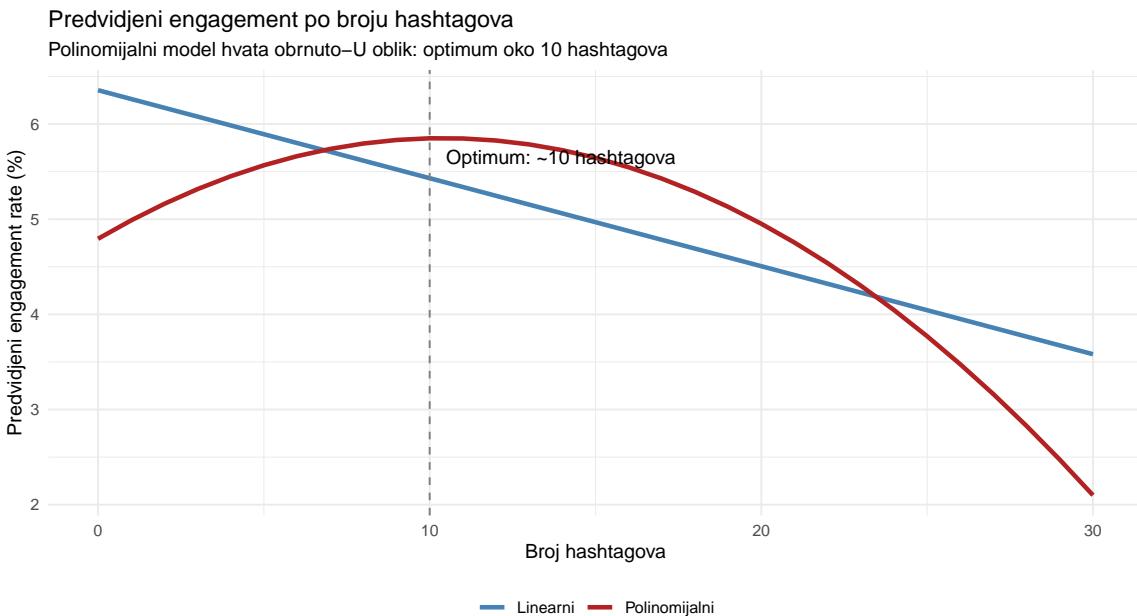
hashtag_pred |>
  pivot_longer(c(pred_lin, pred_poly), names_to = "model", values_to = "predicted") |>
  mutate(model = if_else(model == "pred_lin", "Linearni", "Polinomijalni")) |>
  ggplot(aes(x = num_hashtags, y = predicted, color = model)) +
  geom_line(linewidth = 1.2) +
  geom_vline(xintercept = 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")

```



14.10 Standardizirani koeficijenti

Nestandardizirani koeficijenti su u originalnim jedinicama (postotni bodovi angažmana po dodatnom hashtagu). Problem: ne možemo direktno usporediti koeficijente različitih prediktora jer su na različitim skalaima.

Standardizirani koeficijenti (beta koeficijenti) izražavaju promjenu Y u SD jedinicama za promjenu od 1 SD u X. Ovo omogućuje usporedbu relativne važnosti prediktora.

```

# 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
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

```

Standardizirani koeficijenti (beta) omogućuju usporedbu: koji prediktor ima najveći efekt na angažman? Rangiranje po absolutnoj vrijednosti beta otkriva koje varijable su najvažnije.

14.11 Utjecajne točke: Cookova udaljenost

Neka opažanja mogu neprimjereno utjecati na regresijski pravac. **Cookova udaljenost** mjeri koliko bi se model promijenio kad bismo uklonili to opažanje.

```

posts_diag <- posts |>
  mutate(
    cook = cooks.distance(model4),
    leverage = hatvalues(model4),
    std_residual = rstandard(model4)
  )

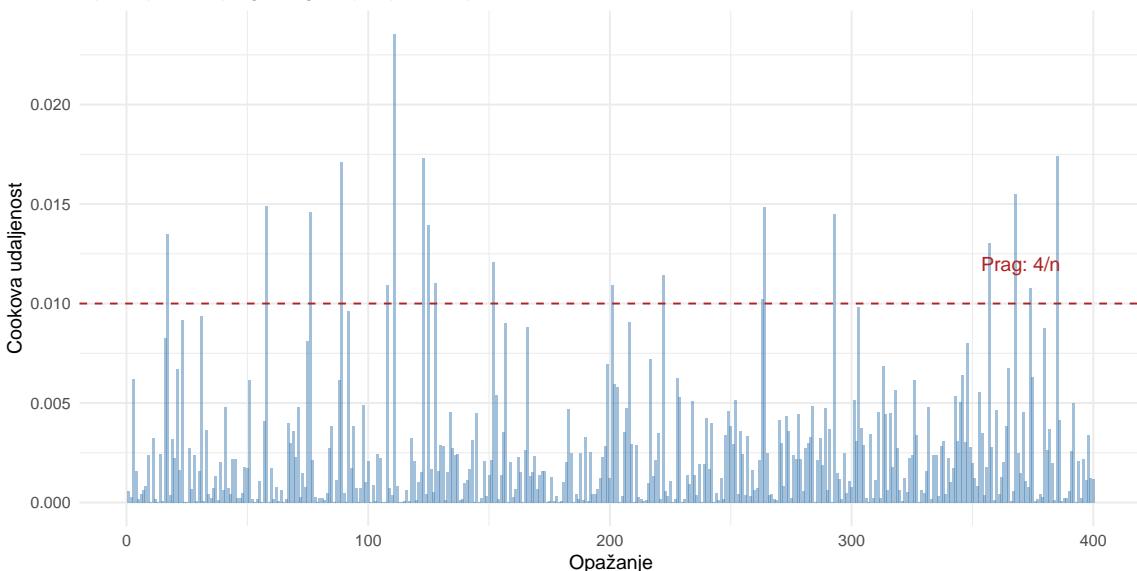
```

```

# 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 = "Opažanja iznad praga mogu neprimjereno utjecati na model",
    x = "Opažanje",
    y = "Cookova udaljenost"
  ) +
  theme_minimal()

```

Cookova udaljenost za svako opažanje
Opažanja iznad praga mogu neprimjereno utjecati na model



```

# Koje objave su najutjecajnije?
prag_cook <- 4 / nrow(posts)
utjecajne <- posts_diag |> filter(cook > prag_cook) |> nrow()

cat("Prag Cook's distance:", round(prag_cook, 4), "\n")

```

Prag Cook's distance: 0.01

```
cat("Broj utjecajnih tocaka:", utjecajne, "od", nrow(posts), "\n")
```

Broj utjecajnih tocaka: 19 od 400

```
# 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

```
cat("Bez utjecajnih tocaka: Adj R2 =", round(summary(model4_clean)$adj.r.squared, 3), "\n")
```

Bez utjecajnih tocaka: Adj R² = 0.466

14.12 Potpuna analiza: izvještaj za menadžericu

```
# 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_typerest    -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

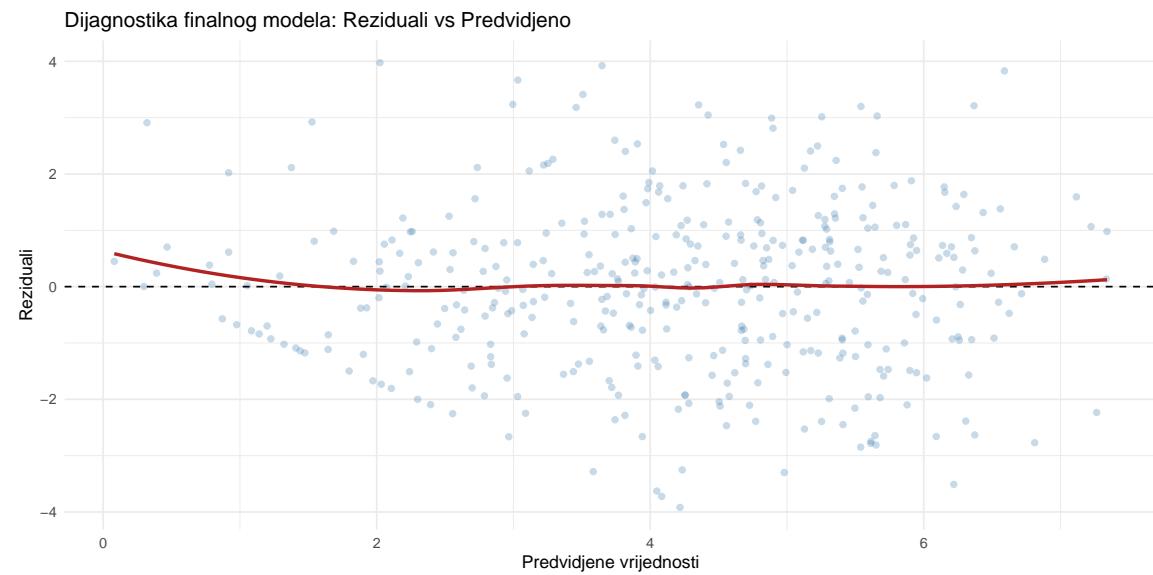
```

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

```

# 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()

```



```

# Predvidjeni engagement po content_type (kontrolirajući 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")

```



```
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.

14.13 Ograničenja regresije

Linearna regresija je moćan alat ali ima jasna ograničenja.

Korelacija nije kauzalnost. 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 su reelovi novi pa dobivaju algoritamski boost, ili ih koriste samo za određene teme.

Model je dobar koliko i podaci. Regresija ne može uhvatiti prediktore koje nismo mjerili (kvaliteta fotografije, trending tema, algoritamske promjene). To je razlog zašto R^2 nikad neće biti 1.

Ekstrapolacija je opasna. Model je treniran na podacima s 0-30 hashtagova. Predvidjanje za 50 hashtagova bi bilo ekstrapolacija izvan raspona podataka i nepouzdano.

Prepostavke moraju biti zadovoljene. Ako dijagnostički grafovi pokazuju ozbiljna odstupanja (nelinearnost, heteroskedastičnost, nenormalne reziduale), rezultati mogu biti nepouzdani. Transformacije ili alternativni modeli mogu pomoći.

! Ključni zaključci

1. Linearna regresija modelira vezu između prediktora (X) i ishoda (Y). Jednostavna: $Y = b_0 + b_1 X$. Višestruka: $Y = b_0 + b_1 X_1 + b_2 X_2 + \dots$
2. `lm(y ~ x1 + x2, data)` provodi regresiju. `summary()` daje koeficijente, SE, t, p, 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.
4. R-kvadrat je proporcija varijabilnosti objašnjena modelom. Prilagodeni R^2 korigira za broj prediktora. U društvenim znanostima, $R^2 = 0.10$ do 0.30 je uobičajen.
5. AIC omogućuje usporedbu modela: niži AIC = bolji model. Penalizira dodavanje nepotrebnih prediktora.
6. Prepostavke: linearnost, nezavisnost, homoskedastičnost, normalnost reziduala.

Dijagnostički grafovi (`plot(model)`) ih provjeravaju.

7. VIF mjeri multikolinearnost. VIF < 5 je prihvatljiv. VIF > 10 je problematičan. Visok VIF znači da su prediktori previsoko korelirani.
8. Nelinearne odnose možemo uhvatiti polinomom: $I(x^2)$ dodaje kvadratni član. LOESS krivulja otkriva nelinearnost vizualno.
9. Standardizirani koeficijenti (beta) omogućuju usporedbu relativne važnosti prediktora na istoj skali (SD jedinice).
10. Cookova udaljenost identificira utjecajne točke. Prag: $4/n$. Provedite analizu s i bez utjecajnih točaka za provjeru stabilnosti.
11. Regresija otkriva asocijacije, ne kauzalnost. Ekstrapolacija izvan raspona podataka je nepouzdana. Model je dobar koliko i podaci.
12. Kompletan izvještaj: opis uzorka i modela, F-test i R^2 , značajni koeficijenti s interpretacijom, dijagnostika pretpostavki, vizualizacija efekata i praktične preporuke.

14.14 Zadaci za pripremu

1. Učitajte `social_engagement.csv`. Provedite jednostavnu regresiju `engagement_rate ~ num_hashtags`. Interpretirajte koeficijent i R^2 . Pogledajte dijagnostičke grafove. Dodajte kvadratni član $I(num_hashtags^2)$ i usporedite modele.
 2. Izradite višestruki model s barem 4 prediktora. Izračunajte VIF za numeričke prediktore. Napišite rezultate u APA formatu.
 3. Kreirajte graf koji prikazuje predvidjeni 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.

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.

Fox, J. & Weisberg, S. (2019). *An R Companion to Applied Regression* (3rd edition). SAGE. Referentni priručnik za regresijsku dijagnostiku u R-u.

14.16 Pojmovnik

Pojam	Objašnjenje
Linearna regresija	Modeliranje linearne veze između prediktora (X) i ishoda (Y). $Y = b_0 + b_1 X + e$.
Jednostavna regresija	Regresija s jednim prediktorom.
Višestruka regresija	Regresija s dva ili više prediktora.
Koeficijent (b, slope)	Koeficijenti su parcijalni efekti.
Intercept (b0)	Promjena Y za jediničnu promjenu X, uz kontrolu ostalih prediktora.
Parcijalni efekt	Predviđeni Y kad su svi prediktori jednaki nuli. Često bez praktičnog značenja.
Rezidual (e)	Efekt jednog prediktora uz kontrolu (držanje konstantnima) svih ostalih.
R-kvadrat (R^2)	Razlika između opaženog i predviđenog Y. $e = Y - \hat{Y}$.
Prilagodeni R^2	Proporcija varijabilnosti Y-a objašnjena modelom. 0 = model ne objašnjava ništa. 1 = savršeno.
AIC	R^2 korigiran za broj prediktora. Koristi se za usporedbu modela s razlicitim brojem prediktora.
OLS	Akaike Information Criterion. Niži = bolji model. Penalizira kompleksnost.
Dummy varijabla	Ordinary Least Squares. Metoda koja minimizira sumu kvadriranih reziduala.
VIF	Binarna (0/1) varijabla za kodiranje kategorija. R ih automatski kreira u <code>lm()</code> .
Multikolinearnost	Variance Inflation Factor. Mjeri multikolinearnost. $VIF < 5$ prihvatljivo, > 10 problematicno.
	Visoka korelacija između prediktora. Čini koeficijente nestabilnima.

Pojam	Objašnjenje
Cookova udaljenost	Mjera utjecaja pojedinog opažanja na model. Prag: 4/n.
Leverage	Koliko je opažanje ekstremno u prostoru prediktora. Visok leverage = potencijalno utjecajno.
Standardizirani koeficijent (beta)	Koeficijent izražen u SD jedinicama. Omogućuje usporedbu prediktora.
Polinomijalna regresija	Dodavanje kvadratnog (ili višeg) člana za uhvatiti nelinearne odnose. $I(x^2)$.
Homoskedastičnost	Jednaka varijanca reziduala za sve predviđene vrijednosti. Pretpostavka regresije.
Ekstrapolacija	Predviđanje izvan raspona podataka. Nepouzdano jer model nije treniran za te vrijednosti.
<code>lm()</code>	R funkcija za linearnu regresiju. <code>lm(y ~ x1 + x2, data = ...)</code> .
<code>summary()</code>	Na <code>lm()</code> objektu daje koeficijente, SE, t, p, R ² i F-test.
<code>predict()</code>	Generira predviđene vrijednosti. <code>predict(model, newdata = ...)</code> za nova opažanja.
<code>residuals()</code>	Izvlači reziduale iz modela.
<code>broom::tidy()</code>	Pretvara model output u tibble s koeficijentima, SE, t, p i CI.
<code>AIC()</code>	R funkcija za izračun AIC-a. <code>AIC(model1, model2)</code> za usporedbu.