Statistički praktikum 1

1 Uvod u R

1.1 Što je R i kako ćemo ga koristiti?

R je programski jezik i okruženje namijenjeno statističkoj obradi podataka, simulacijama i grafičkim prikazima. Besplatan je i otvorenog koda, a osnovne funkcije moguće je proširivati dodatnim paketima. U ovim vježbama najprije ćemo upotrebljavati funkcije koje dolaze sa samim R-om kako bismo jasno vidjeli što svaka naredba radi.

RStudio je razvojno okruženje koje olakšava rad s R-om, ali nije sam programski jezik. U konzoli se naredba izvršava odmah nakon pritiska tipke Enter. Skripta je tekstualna datoteka s nastavkom `.R` u kojoj čuvamo naredbe. Analizu pišemo u skripti, a zatim odabrane retke šaljemo u konzolu. Tako se cijeli postupak može ponoviti, provjeriti i ispraviti bez ponovnog ručnog unošenja.

Za izvršavanje jednog retka u RStudiju postavimo pokazivač bilo gdje u taj redak i pritisnemo `Ctrl+Enter`. Na računalu s operacijskim sustavom macOS koristimo `Cmd+Enter`. Za izvršavanje više naredbi najprije ih označimo mišem ili tipkovnicom, a zatim upotrijebimo isti prečac. Poslane naredbe pojavljuju se u konzoli, a ispod njih R ispisuje rezultat ili poruku o pogrešci. Naredbu možemo poslati i gumbom `Run` iznad prozora skripte, ali je rad tipkovničkim prečacem obično brži.

Rezultati računanja mogu se samo ispisati u konzoli ili spremiti pod odabranim imenom. Spremljeni rezultat zovemo *objekt*. Trenutačno postojeće objekte možemo vidjeti funkcijom `ls()`, a pojedini objekt ukloniti funkcijom `rm()`. Nije potrebno brisati objekte tijekom svake analize; važnije je upotrebljavati jasna imena i skriptu koja se može izvršiti od početka.

```
2 + 3                  # rezultat se samo ispisuje
zbroj <- 2 + 3         # rezultat se sprema u objekt zbroj
zbroj
ls()
```

R učitava i sprema datoteke u odnosu na trenutačnu radnu mapu. Nju možemo provjeriti funkcijom `getwd()`. Na početku rada dobro je postaviti radnu mapu koristeći funkciju `setwd()`. Na primjer, ako je put do podataka koje trebamo `"C:/podaci/pozivi.csv"`, onda je dobro postaviti radnu mapu `setwd("C:/podaci")`.[1] Ovo je posebno korisno kada se u istoj mapi nalazi više datoteka koje trebamo.

Funkcije u R-u pozivamo njihovim imenom, a argumente navodimo u oblim zagradama. Pomoć za poznatu funkciju otvaramo pozivom funkcije `help`, primjerice `help(mean)` će nam dati informaciju o funkciji `mean`. Poruku o pogrešci treba pročitati: ona je informacija o tome gdje R nije mogao izvršiti zadanu naredbu, a ne razlog za napuštanje analize.

R ima klasične tipove podataka:

- `numeric` (10.55, 34, -2.3)
- `integer` (10L, -12L) - slovo "L" označava podatak cijelim brojem

[1] Alternativno, kada učitavamo podatke možemo dati cijeli put do datoteke koju trebamo.

- `complex`
- `character` ("jutro", "dan", "10.55")
- `logical` (TRUE, FALSE)

Pozivom funkcije `class()` možemo otkriti koji tip podataka je spremljen u varijabli.

1.2 Objekti i unos podataka

R razlikuje velika i mala slova. Operator `<-` pridružuje vrijednost imenu objekta, a znak `#` započinje komentar. Vektor kreiramo koristeći funkciju `c()` (*combine*).

Zadatak 1. *Što ispisuju zadnje dvije naredbe u donjem kodu?*

```
x <- c(4, 7, 7, 9, 12)  # numericki vektor
smjena <- c("jutro", "noc", "jutro", "popodne", "jutro")
length(x)
class(x)
```

Funkciju `c()` možemo koristiti i kada želimo spojiti nekoliko vektora ili dodati element u vektor. Na primjer, za vektore `x_1`, `x_2`, možemo kreirati novi vektor koji će sadržavati elemente iz `x_1`, potom `x_2` i na kraju 0 na sljedeći način

```
x_novi <- c(x_1, x_2, 0)
```

S numeričkim vektorima možemo jednostavno postupati. Na primjer, dozvoljene su sljedeće operacije

```
y <- c(2, -3, 1, 7, -1)
x-y
x+2
x/4
```

Rezultat zbroja vektora i broja 2 će biti vektor kojemu je svaki element veći za 2. Analogno vrijedi za množenje ili dijeljenje vektora nekim brojem.

Zadatak 2. *Provjerite što će biti spremljeno u* `a`*,* `b` *i* `c`*.*

```
a <- -4:5
b <- seq(from = 0, to = 1, by = 0.1)
c <- seq(from = 0, to = 1, length.out = 11)
```

Funkciju `seq()` koristimo za kreiranje niza (`sequence`). Možemo zadati inkrement ili zadati finalnu duljinu. Ako nam treba niz cijelih brojeva od `a_1` do `a_2` možemo samo pozvati `a_1:a_2`. Ako želimo pristupiti 3. elementu ili 2. – 5. ili svima osim 3. i 6. pišemo sljedeće:

```
a[3]
a[2:5]
a[c(-3, -6)]
```

Jedan stupac podataka najčešće spremamo kao vektor. Ako imamo više podataka za koje ima smisla da su objedinjeni (npr. pratimo letove, kašnjenje i broj putnika na pojedinom letu), spremamo ih kao kao `data.frame`; svi njezini stupci moraju imati jednak broj elemenata. Pozivom sljedeće funkcije dobit ćemo podatkovni okvir sa stupcima pod nazivom `id, smjena` i `broj_poziva`, a podaci će biti brojevi od 1 do 5, podaci iz vektora `smjena` te podaci iz vektora `x`, redom.

```
uzorak <- data.frame(
  id = 1:5,
  smjena = smjena,
  broj_poziva = x
)
uzorak
```

Za ispravnu analizu moramo znati što predstavlja jedan redak, što predstavlja svaki stupac i koja je mjerna skala svakog obilježja.

Osim tzv. `data.frame`-a, ako su svi podaci istog tipa (npr. svi podaci su tipa `integer` ili, češće, svi podaci su tipa `numeric`) možemo koristiti matrice. Matrice su puno manji objekt i samim time efikasniji u računanju – razlika koja se znatno osjeti kada radimo s velikim brojem podataka. Za stvaranje matrice koristimo funkciju `matrix()`. Na primjer, ako želimo stvoriti matricu $\left(\begin{smallmatrix}1&2&3\\4&5&6\end{smallmatrix}\right)$, možemo to učiniti izvršavanjem sljedećih linija koda

```
podaci <- c(1,2,3,4,5,6)
matrica <- matrix(podaci, nrow=2, ncol=3, byrow=True)
```

Prvo unosimo podatke koje želimo staviti u matricu, potom pišemo broj redaka (`nrow`) i/ili broj stupaca (`col`). Pretpostavlja se da ćemo podatke unositi po stupcima. Ako želimo podatke unijeti po retcima moramo staviti `byrow` na `TRUE`. Tijekom kolegija ćemo često imati podatke različitih tipova pa ćemo primarno koristiti podatkovni okvir (`data.frame`).

1.3 Učitavanje i prvi pregled

Za učitavanje podataka koristimo funkciju `read.table()` ili `read.csv()`. Ako pozovemo `help()` za ove funkcije vidimo koje argumente primaju.

```
read.table(file, header = FALSE, sep = "", dec = ".")
read.csv(file, header = TRUE, sep = ",", dec = ".")
```

Obje imaju istu funkciju (učitavanje tablica), ali različite zadane parametre. Funkciju `read.table()` obično koristimo kada stupci nemaju naziv (prvi redak u file-u su podaci), kada su odvojeni *whitespace*-om, te kada je oznaka za decimalu točka. To je najčešće slučaj za podatke spremljene u .txt formatu. Ako je u prvom retku spremljen naziv pojedinih stupaca (podaci kreću od drugog retka), ako su podaci odvojeni zarezom, a oznaka za decimalni dio je i dalje točka, koristimo `read.csv()`. To je najčešće slučaj za podatke spremljene u .csv formatu. Bez problema možemo učitati .csv format pozivom `read.table()` funkcije – samo moramo postaviti `header=TRUE` i `sep=","`.

Datoteka `pozivi.csv` koristi točku-zarez za razdvajanje podataka i zarez za oznaku decimalnih mjesta. Zato je učitavamo na sljedeći način

```
pozivi <- read.csv("podaci/pozivi.csv", sep=";", dec=",")
head(pozivi)
str(pozivi)
summary(pozivi)
dim(pozivi)
names(pozivi)
```

`head()` prikazuje prvih nekoliko redaka, `str()` strukturu i tipove stupaca, a `summary()` sažetak prilagođen tipu stupca. Na kraju treba provjeriti jesu li brojčane vrijednosti doista učitane kao brojevi. Funkcija `dim()` vraća dimenzije podatkovnog okvira, dok `names()` vraća nazive stupaca. Ako želimo promijeniti naziv stupca, radimo sljedeće:

```
names(pozivi) <- c("stupac_1", "stupac_2", "stupac_3", "stupac_4", "stupac_5")
```

1.4 Redci, stupci i izmjena podataka

Stupcu u podatkovnom okviru pristupamo operatorom `$` (ako stupac ima naziv) ili uglatim zagradama. Zapis `podaci[redci, stupci]` uvijek navodi retke prije zareza, a stupce poslije zareza.

Zadatak 3. *Izvršite sljedeće linije koda u R-u i komentirajte što dobijete.*

```
pozivi$broj_poziva
pozivi[, c("smjena", "broj_poziva")]
pozivi[1:4, ]
```

Postojeći stupac možemo izmijeniti, a novi nastaje pridruživanjem novom imenu. Promjene treba raditi na kopiji ako želimo sačuvati izvorne podatke. Na primjer, ako dobijemo podatke za novi stupac spremljene kao vektor `duga_smjena`, možemo ga dodati u postojeći podatkovni okvir koristeći

```
pozivi$duga_smjena <- duga_smjena
```

U drugu ruku, ako želimo izbaciti stupac `smjena`, možemo to napraviti na sljedeći način

```
pozivi$smjena <- NULL
```

1.4.1 Odabir redaka prema uvjetu

Usporedbe vraćaju logičke vrijednosti `TRUE` i `FALSE`. Logički operatori su

- `==` (provjerava jesu li dva objekta jednaka)
- `!=` (provjerava jesu li dva objekta različita
- `&` (i)
- `|` (ili)
- `!` (ne)

U R-u možemo provjeravati je li vektor jednak nekom elementu. Na primjer ako napišemo

```
x == 7
```

kao rezultat ćemo dobiti vektor s logičnim vrijednostima

```
FALSE  TRUE  TRUE FALSE FALSE.
```

To možemo iskoristiti za izdvajanje redaka u vektoru (ili data frameu) koji zadovoljavaju neki uvjet. Ako nas zanima pripada li vrijednost zadanom skupu, koristimo `%in%`. Koji rezultat će dati prva, a koji druga linija koda? Provjerite u R-u.

```
x %in% c(7,12)
x[x %in% c(7,12)]
```

Zadatak 4. *Kreirajte novi data frame s nazivom* ***pozivi_jutro*** *koji će sadržavati informacije samo o pozivima u jutarnjoj smjeni. Iz nje izbrišite stupac* ***smjena****.*

Rješenje.
```
pozivi_jutro <- pozivi[pozivi$smjena == "jutro", ]
View(pozivi_jutro)
pozivi_jutro$smjena = NULL
```

□

Zadatak 5. *Koja informacija je spremljena u data frameu* ***zauzeto_jutro****, a koja u* ***dan****?*

```
zauzeto_jutro <- pozivi[
  pozivi$smjena == "jutro" & pozivi$broj_poziva >= 10, ]
dan <- pozivi[pozivi$smjena %in% c("jutro", "popodne"), ]
```

1.5 Nedostajuće vrijednosti

`NA` označava nepoznatu ili nezabilježenu vrijednost. Ona nije nula i nije prazan tekst. Budući da nepoznata vrijednost nije jednaka ni samoj sebi, ne tražimo je izrazom `x == NA`, nego funkcijom `is.na()`. Izvršite sljedeće linije koda.

```
is.na(pozivi$trajanje_min)
sum(is.na(pozivi$trajanje_min))
colSums(is.na(pozivi))
```

Prva će nam vratiti bool vektor s vrijednosti `TRUE` na mjestima gdje se nalazi `NA` u stupcu, a `FALSE` inače. Druga linija će napisati broj `NA` vrijednosti u tom stupcu. Ako nas zanima koliko je `NA` vrijednosti u pojedinom stupcu čitave tablice koristimo naredbu `colSums` – ta naredba daje sumu vrijednosti po stupcima. Pozivom `is.na(pozivi)`, dobijemo bool tablicu, pa treća linija koda sumira broj `TRUE` vrijednosti po stupcima.

Funkcija `!is.na()` označava vrijednosti koje ne nedostaju. To je važno pri selekciji: ako uvjet sadrži `NA`, rezultat može sadržavati redak ispunjen nedostajućim vrijednostima. Zato uvjet dopunjujemo zahtjevom da je vrijednost poznata.

```
dugi_pozivi <- pozivi[
  !is.na(pozivi$trajanje_min) & pozivi$trajanje_min > 10, ]
```

Postoje dva česta, ali različita postupka. Argument `na.rm=TRUE` izostavlja nedostajuće vrijednosti samo iz konkretnog računa. Funkcija `complete.cases()` odabire retke koji nemaju `NA` ni u jednom promatranom stupcu.

```
mean(pozivi$trajanje_min, na.rm = TRUE)
potpuni <- pozivi[complete.cases(pozivi), ]
bez_NA_u_trajanju <- pozivi[!is.na(pozivi$trajanje_min), ]
```

Ove tablice nemaju nužno jednak broj redaka. Redak kojem nedostaje ocjena može i dalje biti valjan za analizu trajanja. Brisanje svih nepotpunih redaka zato ne smije biti automatska odluka. Također, uklanjanje `NA` rješava tehnički problem računa, ali ne dokazuje da nedostajanje nije pristrano.

2 Deskriptivna statistika

Za opažanja $x_1, \ldots, x_n$ aritmetička sredina (`mean()`) je

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i, \tag{1}$$

a uzoračka standardna devijacija (`sd()`) je

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}. \tag{2}$$

Aritmetička sredina je dobar procjenitelj za očekivanje. Ona nam pokazuje prosječnu vrijednost u uzorku. Dva uzroka s aritmetičkom sredinom mogu biti veoma različita (npr. uzorci $\{-100, 100\}$ i $\{-0.5, 0.5\}$ imaju istu aritmetičku sredinu, ali su poprilično različiti). Standardna devijacija mjeri prosječno odstupanje od aritmetičke sredine. Uparena sa aritmetičkom sredinom daje malo širu sliku o podacima. Obje su osjetljive na *outliere*. Otpornija mjera sredina na outliere je medijan (`median`). Medijan je središnja vrijednost u nizu podataka koji su složeni po

veličini od najmanjega do najvećeg. Osim aritmetičke sredine i medijana, koristimo i mod (primarno za ne-numeričke podatke). Često nas zanima raspon podataka – za to koristimo funkciju `range()`. Ona vraća najmanji i najveći podatak (njima možemo pristupiti i pozivima `min()` i `max()`).

```
mean(pozivi$broj_poziva)
sd(pozivi$broj_poziva)
median(pozivi$broj_poziva)
quantile(pozivi$broj_poziva)
```

2.1 Frekvencije i stupčasti dijagram

Frekvencija kategorije jest broj opažanja u toj kategoriji. Koristimo ih kada imamo **diskretne** podatke. Funkcija `table()` računa apsolutne frekvencije.

```
f_smjena <- table(pozivi$smjena)
f_smjena
```

Zadatak 6. *Prisjetite se kako dobivamo relativne frekvencije iz apsolutnih. Izračunajte relativne frekvencije za smjene iz podatkovnog okvira pozivi.*

Rješenje. `uk_broj_smjena <- sum(f_smjena) %ili length(pozivi$smjena)`
`rel_f_smjena <- f_smjena/uk_broj_smjena`

□

Stupčasti dijagram prikazuje frekvencije kategorija. Ako želimo nacrtati stupčasti dijagram zovemo funkciju `barplot()`. **Funkcija barplot kao podatke očekuje visine stupića odnosno frekvencije** (relativne ili apsolutne)! Pogledajte što dobijete pozivom `barplot(pozivi$smjena)`. Ispravan pristup je izračunati (apsolutne ili relativne) frekvencije, a potom pozvati funkciju `barplot()` nad frekvencijama.

```
barplot(f_smjena,
        xlab = "Smjena", ylab = "Broj opažanja",
        main = "Broj opažanja po smjenama",
        col = "steelblue")
```

Frekvencije i njihov vizualni prikaz (stupčasti dijagram) nam daju malo širu sliku o podacima. Možemo vidjeti postoje li neke vrijednosti koje dominirau u rezultatima ili neke koje odskaču.

Za dvije kategorijske varijable `table()` daje kontingencijsku tablicu.

```
f_dan_smjena <- table(pozivi$dan, pozivi$smjena)
f_dan_smjena
barplot(f_dan_smjena, beside = TRUE, legend.text = TRUE,
        xlab = "Smjena", ylab = "Broj opažanja")
```

2.2 Histogram

Histogram prikazuje distribuciju *neprekidnog* obilježja. Raspon vrijednosti dijeli se na intervale, a stupci se dodiruju jer intervali čine uzastopne dijelove brojčane osi. To ga razlikuje od stupčastog dijagrama, koji prikazuje odvojene kategorije i zato između stupaca obično postoje razmaci. Crtamo ga pozivom funkcije `hist()` nad podacima za koje želimo napraviti histogram. Na primjer,

```
hist(pozivi$trajanje_min,
     breaks = 5,
     xlab = "Trajanje (min)",
     freq=FALSE,
     main = "Distribucija trajanja",
     col = "skyblue")
```

Argument `breaks` određuje podjelu na intervale. Premalo intervala može sakriti važna obilježja distribucije, a previše ih može naglasiti slučajne razlike u malom uzorku. Zato treba isprobati nekoliko razumnih podjela i jasno navesti odabranu. Argument `breaks` može primiti broj (kao u gornjem primjeru) ili vektor koji izričito kaže gdje su rubovi stupića. Ako primi broj, to je samo *sugestija* R-u i nije nužno da ćete dobiti traženi broj stupaca na grafu. Ako primi vektor, toga će se strogo držati.

Argument `freq` određuje hoćemo li na vertikalnoj osi vidjeti apsolutne frekvencije (`freq = TRUE`; ovo je zadana vrijednost) ili relativne (`freq = FALSE`). Alternativno, možemo podestiti argument `probability` koji se ponaša obrnuto od `freq`. Često nas zanimaju relativne frekvencije - u tom slučaju je površina ispod histograma 1 i, crtanjem funkcije gustoće pojedinih distribucija, možemo naslutiti kojoj distribuciji podaci pripadaju.

```
hist(pozivi$trajanje_min,
     breaks = 5, freq = FALSE,
     xlab = "Trajanje (min)",
     main = "Histogram na skali gustoce")
```

Iz histograma opisujemo položaj većine opažanja, raspršenost, simetričnost ili asimetričnost, moguće praznine i izdvojene vrijednosti. Izgled histograma sa relativnim frekvencijama možemo smatrati aproksimacijom funkcije distribucije podataka. Histogram ipak ne prikazuje svako pojedino opažanje i njegov izgled ovisi o granicama intervala.

Zadatak 7. *Provjerite kako se histogram mijenja s različitim brojem stupaca sljedećim kodom.*

```
podaci <- rnorm(10000)
hist(podaci, freq=F, breaks=20)
hist(podaci, freq=F, breaks=50)
hist(podaci, freq=F, breaks=100)
```

2.3 Kutijasti dijagram

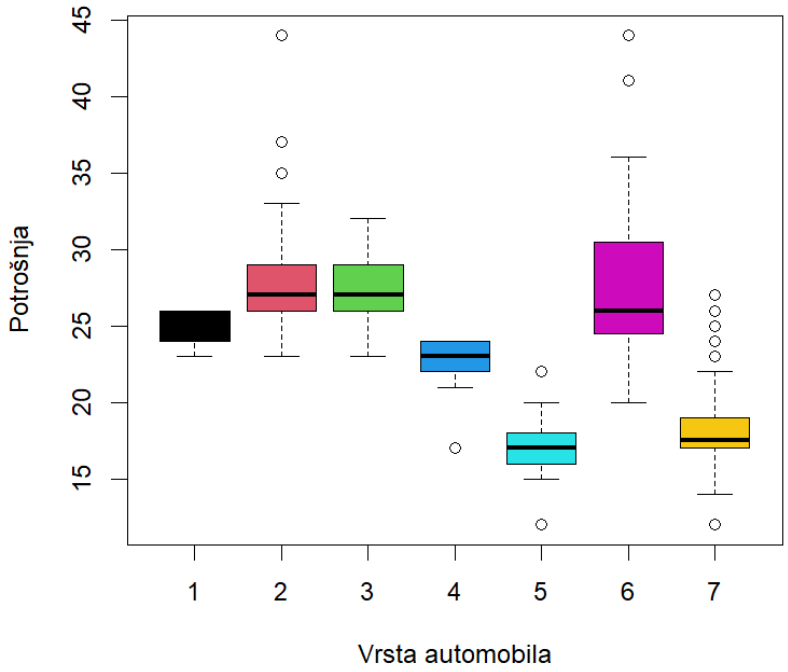
Kutijasti dijagram sažima numeričko obilježje pomoću medijana, kvartila i raspona vrijednosti; posebne točke označavaju potencijalno izdvojena opažanja. One nisu automatski pogreške. Graf je posebno koristan za usporedbu skupina.

```
boxplot(trajanje_min ~ smjena, data = pozivi,
        xlab = "Smjena", ylab = "Trajanje (min)",
        main = "Trajanje po smjenama",
        col = c("gold", "skyblue", "grey70"))
```

Graf treba čitati zajedno s brojem opažanja i numeričkim sažecima. Kod malih skupina izgled kutijastog dijagrama može se snažno promijeniti zbog samo jednog opažanja.

Promotrite donji dijagram o potrošnji automobila ovisno o vrsti. Iz njega možemo isčitati puno informacija. Na primjer, vrste 1,2,3,4 i 6 imaju otprilike jednaku prosječnu potrošnju, u prosjeku višu nego vrste 5 i 6, koje također imaju podjednaku potrošnju. Vrsta 6 ima puno veći raspon vrijednosti potrošnje u odnosu na ostale vrste. Koji je mogući razlog tome što ima veći

raspon? Vrsta 7 ima puno *outliera*, pogotovo s gornje strane. Razmislite je li za pojedine vrste smislenije koristiti aritmetičku sredinu ili medijan kao mjeru sredine.



2.4 Dijagram raspršenja

Usporedba kutijastih dijagrama daje odnos između nekoliko statistika izračunatih iz podataka. Ako želimo vidjeti kako se kreće odnos između dvije ili tri varijable, odnosno kakav je odnos između dvije varijable za svaki redak (osobu, objekt) zasebno, koristimo dijagram raspršenja (*scatter plot*). Pozivamo ga naredbom `plot()`. Na primjer možemo pozvati

```
plot(pozivi$trajanje_min, pozivi$broj_poziva)
```

da bismo vidjeli odnos između trajanja poziva u minutama i broja poziva. Funkcija `plot` prima dva vektora - prvi vektor će dati x-koordinate, a drugi pripadne y-koordinate.

3 Petlje i grananje

Petlja ponavlja isti blok naredbi. Korisna je kada isti postupak treba provesti više puta, osobito ako rezultat jednog koraka određuje što se događa u sljedećem. U R-u ipak najprije treba provjeriti postoji li jednostavnija vektorizirana funkcija. Primjerice, zbroj vektora računamo sa `sum(x)`, a broj nedostajućih vrijednosti po stupcima s `colSums(is.na(podaci))`; za te postupke nije potrebno pisati petlju.

3.1 Petlja `for`

Petlju `for` koristimo kada unaprijed znamo po kojim vrijednostima ili objektima želimo prolaziti. Opći oblik je:

```
for (element in skup_vrijednosti) {
  naredbe
}
```

U svakom prolasku varijabla `element` poprima sljedeću vrijednost iz zadanog skupa. Promotrimo dva primjera. Prvo, klasični način prolaska kroz vektor (stupac),

```
br_redaka <- dim(pozivi)[1]
for (i in 1:br_redaka){
    print(pozivi$trajanje_min)
}
```

Sljedeći primjer za *više numeričkih stupaca* ispisuje broj raspoloživih opažanja i prosjek:

```
varijable <- c("broj_poziva", "trajanje_min", "ocjena")

for (ime in varijable) {
  x <- pozivi[[ime]]
  cat("\nVarijabla:", ime, "\n")
  cat("Broj opazanja:", sum(!is.na(x)), "\n")
  cat("Prosjek:", mean(x, na.rm = TRUE), "\n")
}
```

Izraz `pozivi[[ime]]` potreban je zato što se ime stupca mijenja u svakom prolasku. Zapis `pozivi$ime` tražio bi stupac koji se doslovno zove `ime`, a ne stupac čiji je naziv trenutačno spremljen u objektu `ime`.

3.2 Petlja `while`

Petlja `while` ponavlja naredbe dok je zadani logički uvjet `TRUE`. Primjerice, sljedeći postupak zbraja uzastopne prirodne brojeve dok zbroj ne dosegne 20:

```
i <- 0
zbroj <- 0

while (zbroj < 20) {
  i <- i + 1
  zbroj <- zbroj + i
}

c(broj_clanova = i, konacni_zbroj = zbroj)
```

Prije pisanja petlje `while` treba odrediti što će mijenjati uvjet i osigurati da uvjet jednom postane `FALSE`. U suprotnom nastaje beskonačna petlja, koju u RStudiju možemo prekinuti tipkom `Esc` ili gumbom za zaustavljanje u konzoli.

3.3 Naredba grananja

Ako donosimo različite odluke ovisno o tome što se nalazi u varijabli, koristimo grananje putem naredbe `if`. Jedan primjer korištenja naredbi `if` i `else` je dan u nastavku.

```
kasnjenje<-45.6
if(kasnjenje<15){
  koliko_kasnjenje <- "malo"
} else if (kasnjenje < 30){
  koliko_kasnjenje <- "srednje"
} else {
  koliko_kasnjenje <- "veliko"
}
```

Naredba `else if` (i `else`) mora biti u istoj liniji kao zagrada } koja zatvara `if`.

4 Zadaci za vježbu

U zadatku koristimo podatke iz datoteke `diabetes.csv`. Svaki redak predstavlja jednog sudionika istraživanja. Rješenja spremite u jednu R skriptu. Uz svaki graf dodajte naslov i oznake osi, a uz numeričke rezultate napišite kratku interpretaciju. Ne uklanjajte sve nepotpune retke unaprijed: za svaki račun najprije odredite koje su mu varijable potrebne.

1. Učitajte podatke iz datoteke `diabetes.csv` i spremite ih u podatkovni okvir (`data.frame`) naziva `dijabetes`.

2. Upoznajte podatke prije analize:

   (a) prikažite prvih šest redaka;

   (b) odredite broj redaka i stupaca te ispišite imena stupaca;

   (c) pomoću `str()` provjerite tip svakog stupca;

   (d) objasnite što predstavljaju stupci `subject`, `gender`, `height`, `weight` i `glyhb`.

3. Za svaki stupac odredite broj vrijednosti koje nedostaju. Ispišite retke u kojima nedostaje vrijednost varijable `glyhb`. Koliko bi redaka ostalo kada biste zadržali samo potpune retke svih osam stupaca?

4. Iz podatkovnog okvira `dijabetes` izdvojite sljedeće podtablice i ispišite ih u konzoli:

   (a) samo stupce `subject`, `gender`, `age` i `glyhb`;

   (b) samo ženske sudionice;

   (c) sudionike mlađe od 40 godina;

   (d) sudionike mlađe od 40 godina kojima je `glyhb` zabilježen.

5. Odredite apsolutne i relativne frekvencije za varijablu `gender`. Prikažite odnos broja muških i ženskih sudionika stupčastim dijagramom.

6. Odredite apsolutne i relativne frekvencije za varijablu `frame`, uključujući eventualne `NA` vrijednosti. Nacrtajte stupčasti dijagram apsolutnih frekvencija.

7. Napravite kopiju podataka naziva `dijabetes2`. U njoj stupce `gender`, `frame` i `location` pretvorite u faktore te ponovno provjerite strukturu podataka. Izvorni objekt `dijabetes` ostavite nepromijenjenim.

8. U objektu `dijabetes2` pretvorite visinu iz inča u centimetre tako da vrijednosti stupca `height` pomnožite s 2.54. Stupac preimenujte u `height_cm`.

9. U istom objektu pretvorite težinu iz funti u kilograme tako da vrijednosti stupca `weight` pomnožite s 0.454. Stupac preimenujte u `weight_kg`.

10. Provjerite raspon pretvorenih visina i težina te odredite broj nedostajućih vrijednosti u oba stupca. Procijenite jesu li rezultati pretvorbe razumni prije računanja BMI-ja.

11. Nacrtajte dijagram raspršenja u kojem svaka točka predstavlja jednog sudionika, visina u centimetrima je na osi x, a težina u kilogramima na osi y.

12. U podatke dodajte stupac `BMI`. Indeks tjelesne mase računa se formulom

$$\text{BMI} = \frac{\text{težina u kg}}{(\text{visina u m})^2}. \tag{3}$$

    Pazite da je visina prije računanja izražena u metrima. Prikažite prvih šest vrijednosti novog stupca zajedno s visinom i težinom.

13. Za varijable `age`, `glyhb`, `height_cm`, `weight_kg` i `BMI` pomoću petlje `for` ispišite:

    - naziv varijable;
    - broj raspoloživih opažanja;
    - aritmetičku sredinu;
    - standardnu devijaciju.

    Imena varijabli najprije spremite u znakovni vektor, a stupcu u svakom prolasku pristupite pomoću dvostrukih uglatih zagrada. Pri računanju pravilno obradite nedostajuće vrijednosti. Objasnite zašto izraz `dijabetes2[[ime]]` u ovom zadatku nije moguće zamijeniti izrazom `dijabetes2$ime`.

14. Nacrtajte histogram za glikirani hemoglobin `glyhb`. Odaberite smislen broj razreda i opišite oblik distribucije.

15. Za `glyhb` izračunajte aritmetičku sredinu bez argumenta `na.rm`. Usporedite rezultat s aritmetičkom sredinom dobivenom u 13. zadatku i objasnite razliku. Nakon toga odredite medijan, minimum i maksimum uz pravilnu obradu nedostajućih vrijednosti.

16. Nacrtajte dijagram raspršenja s BMI-jem na osi x i vrijednošću `glyhb` na osi y. Jesu li na grafu prisutna sva 354 sudionika? Objasnite odgovor.

17. Odredite aritmetičku sredinu varijable `glyhb` posebno za muškarce, a posebno za žene. Uz svaki prosjek navedite broj opažanja na kojem je izračunat.

18. U `dijabetes2` dodajte stupac `rezultat` čije su kategorije:

    - `"normalno"` ako je `glyhb` manji od 5.7;
    - `"predijabetes"` ako je `glyhb` između 5.7 i 6.5, uključujući obje granice;
    - `"dijabetes"` ako je `glyhb` veći od 6.5.

    Ako `glyhb` nedostaje, i `rezultat` mora ostati `NA`. Pretvorite novi stupac u faktor s navedenim redoslijedom kategorija i provjerite frekvencije.

19. Odredite prosječnu dob sudionika čiji je rezultat u kategoriji `"dijabetes"`. Usporedite je s prosječnom dobi svih sudionika dobivenom u 13. zadatku.

20. Za svaku kategoriju stupca `rezultat` zasebno odredite broj sudionika te prosječnu težinu u kilogramima.

21. Za muškarce i žene zasebno odredite varijancu, standardnu devijaciju i koeficijent varijacije težine u kilogramima. Koeficijent varijacije definiramo kao

$$\mathrm{CV} = \frac{s}{\bar{x}} \cdot 100\%. \tag{4}$$

    Grupne standardne devijacije usporedite s ukupnom standardnom devijacijom težine dobivenom u 13. zadatku.

22. Odredite karakterističnu petorku za `glyhb`: minimum, prvi kvartil, medijan, treći kvartil i maksimum. Usporedite je s rezultatom funkcije `summary()`.

23. Nacrtajte kutijasti dijagram za BMI. Opišite medijan, raspršenost i eventualne izdvojene vrijednosti. Pronađite retke koji odgovaraju točkama koje R na tom dijagramu označava kao izdvojene.

24. Nacrtajte usporedne kutijaste dijagrame za `glyhb` posebno za muškarce i žene. Usporedite središta, raspršenost i izdvojene vrijednosti dviju skupina, ali nemojte iz opisnog grafa izvoditi zaključak o cijeloj populaciji.

Zadaci za zadaću

Zadatak 1.1. *Učitajte podatke iz datoteke* `osiguranje.xls` *o dugovima za medicinske usluge. Koliko ima podataka? Ispišite kako izgleda prvih nekoliko podataka. Potom kako izgleda posljednjih nekoliko podatka. Koji je raspon vrijednosti dugova?*

Zadatak 1.2. *Učitajte podatke iz datoteke* `ucenici_uspjeh.xls`*. Izdvojite podatke o uspjehu studenata koji su završili neke pripreme za ispit u tablicu* `ucenici_uspjeh_pripremljeni`*. Iz nove tablice izbacite stupac koji kaže jesu li studenti završili neke pripreme za ispit jer nam više nije potreban.*

Zadatak 1.3. *Učitajte podatke iz* `kvaliteta_vina.csv` *Zanimaju nas podaci o stalnoj kiselosti promatranih vina. Odredite raspon tih podataka, aritmetičku sredinu, medijan i standardnu devijaciju.*

Zadatak 2.1. *Grafički prikažite distribuciju podataka o godinama iz tablice* `osiguranje`*. Kojoj distribuciji biste rekli da podaci pripadaju? Promotrite i distribuciju podataka o iznosu naplate. Kojoj distribuciji bi ti podaci mogli pripadati?*

Zadatak 2.2. *Kreirajte novi stupac u tablici* `osiguranje` *pod nazivom* `djeca`*. Neka u tom stupcu stoji* `FALSE` *ako osoba nema djecu, a* `TRUE` *ako osoba ima djecu. Usporedite naknade za cijeli uzorak, samo za osobe s djecom te za osobe bez djece.*

Zadatak 2.3. *Kreirajte novi stupac pod nazivom* `starost` *za tablicu* `osiguranje`*. Neka stupac sadržava*

- `"mladi"` *ako je osoba mlađa od 21,*
- `"odrasli"` *ako osoba ima između 22 i 55 godina,*
- `"stariji"` *ako osoba ima 56 godina ili više.*

Usporedite podatke iz te tri skupine.

5 Od točkaste procjene do pouzdanog intervala

Na prethodnim vježbama računali smo aritmetičku sredinu i standardnu devijaciju uzorka. Prosjek opisuje prikupljene podatke i može poslužiti kao procjena nepoznatog očekivanja populacije. Međutim, drugi slučajni uzorak obično bi dao drugi prosjek. Zato uz točkastu procjenu želimo navesti i koliko je precizna.

Pouzdani interval daje donju i gornju granicu za nepoznati parametar. **Razina pouzdanosti** $1-\alpha$ predstavlja vjerojatnost da je nepoznati parametar unutar slučajnog intervala. Primjerice, za razinu 95% je $\alpha = 0.05$, a u svakom repu razdiobe pri konstrukciji dvostranog intervala ostavljamo $\alpha/2 = 0.025$.

Neka je θ nepoznati parametar, a $X_1, \ldots, X_n$ slučajni uzorak duljine n. Za egzaktni postupak vrijedi

$$\mathbb{P}_\theta\big(L(X_1, ..., X_n) \leq \theta \leq U(X_1, ..., X_n)\big) = 1 - \alpha. \tag{5}$$

Parametar je fiksan, a uzorak i granice intervala slučajni su prije opažanja.

Interval $[L(X_1, ..., X_n), U(X_1, ..., X_n)]$ nazivamo $1-\alpha$-pouzdanim intervalom. To je slučajan interval i vjerojatnost da će parametar θ upasti unutra je $1 - \alpha$. Kada uvrstimo konkretan uzorak $x_1, .., x_n$, dobijemo realizaciju $1-\alpha$-pouzdanog intervala $[L(x_1, ..., x_n), U(x_1, ..., x_n)]$. On **nije slučajan** - ili sadrži nepoznati θ koji procjenjujemo ili ne. Ako bismo mnogo puta prikupili uzorak iste veličine i svaki put izračunali interval istim postupkom, približno 95% tako dobivenih 95%-pouzdanih intervala sadržavalo bi pravi parametar.

Pouzdani interval za očekivanje nije interval koji sadrži 95% pojedinačnih opažanja. Veći uzorak u pravilu omogućuje precizniju procjenu; veća razina pouzdanosti uz iste podatke zahtijeva širi interval.

5.1 Kvantili u R-u

Kvantil reda $u \in [0, 1]$ jest vrijednost do koje razdioba ima vjerojatnost u. O kvantilu možemo razmišljati kao "obratu" vjerojatnosti, u kontekstu vjerojatnosne funkcije distribucije. Na primjer, ako za funkciju distribucije standardne normalne razdiobe Φ vrijedi

$$\Phi(0) = \frac{1}{2},$$

Onda je vrijednost kvantila reda 1/2 nula. Da bismo izračunali kvantil, koristimo `q` i naziv distribucije koju želimo. Na primjer, za standardnu normalnu razdiobu koristimo `qnorm()`, za Studentovu `qt()`, a za hi-kvadrat razdiobu `qchisq()`. Zadnje dvije funkcije zahtijevaju i broj stupnjeva slobode, `df`. Pregledajte ih koristeći naredbu `help`. Ono što ćete primjetiti je da osim `qnorm`, možete pozvati i funkcije `rnorm`, `pnorm`, `dnorm`. Funkcija `rnorm` dat će vam slučajan uzorak (**r**andom sample). Prvi parametar koji prima je duljina uzorka `n`. Funkcija `dnorm` dat će vam iznos funkcije gustoće (**d**ensity) za točku (i parametre) koje želite. Funkcija `pnorm` dat će vam iznos funkcije distribucije u točki (i parametrima) koje unesete. Pripazite, pri pozivu neke od tih funkcija za normalnu razdiobu, (npr. `rnorm`), ako želite parametre $N(2, 3^2)$, unosite `rnorm(1000, 2,3)` – standardnu devijaciju umjesto varijance.

Isprobajte sljedeće linije koda u R-u. Komentirajte što dobijete.

```
alpha <- 0.05
qnorm(c(alpha/2, 1 - alpha/2))
qt(c(alpha/2, 1 - alpha/2), df = 13)
qchisq(c(alpha/2, 1 - alpha/2), df = 13)
```

Zadatak 8. *Izračunajte kvantile standardne normalne razdiobe potreban za 90%, 95% i 99% pouzdani interval. Zašto koristimo* `1 - alpha/2`*, a ne* `1 - alpha`*? Usporedite simetričnost dobivenih parova kvantila. Uz pomoć naredbe* `help`*, saznajte čemu služi parametar* `lower.tail`*.*

6 Pouzdani interval za očekivanje

Pretpostavimo da su $X_1, \ldots, X_n$ nezavisne slučajne varijable iz iste normalne razdiobe $N(\mu, \sigma^2)$. Očekivanje μ je nepoznato. Razlikujemo slučaj kada znamo populacijsku standardnu devijaciju σ od slučaja kada je procjenjujemo iz uzorka.

6.1 Poznata varijanca

Nabrojimo neka svojstva normalne razdiobe. Neka je $X \sim N(\mu_1, \sigma_1^2)$, $Y \sim N(\mu_2, \sigma_2^2)$, te neka su X i Y nezavisne.

- Za $a, b \in \mathbb{R}$, slučajna varijabla $Z = aX + b$ ima $N(a\mu_1 + b, a^2\sigma_1^2)$ distribuciju.
- $X + Y \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$.
- Za slučajan uzorak duljine n iz $N(\mu_1, \sigma_1)$, $\overline{X}_n \sim N(\mu, \sigma^2/n)$.

Aritmetička sredina ima varijancu σ^2/n, pa koristeći gornju propoziciju (tvrdnju 1) možemo pokazati

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1). \tag{6}$$

Primjer: simulacija razdiobe statistike Z. Generiramo $B = 10000$ nezavisnih uzoraka veličine $n = 14$ iz $N(5, 2^2)$. Iz svakog uzorka računamo izraz s desne strane formule. Tako dobivamo 10000 vrijednosti statistike Z, čiji histogram na skali gustoće uspoređujemo s gustoćom $N(0, 1)$.

```
set.seed(302)
mu <- 5; sigma <- 2; n <- 14; B <- 10000
z_sim <- numeric(B)
for (b in 1:B) {
  x_sim <- rnorm(n, mean = mu, sd = sigma)
  z_sim[b] <- (mean(x_sim) - mu) / (sigma / sqrt(n))
}
hist(z_sim, breaks = 40, probability = TRUE,
     main = "Razdioba statistike Z", xlab = "Z",
     col = "grey85")
curve(dnorm(x), add = TRUE, col = "red", lwd = 2)
```

Naredba `set.seed(302)` postavlja početni broj za simulaciju pseudoslučajnih brojeva u programu. Konkretno, ako dva korisnika postave isti *seed*, dobit će iste rezultate npr. pozivom funkcije `rnorm(n)`. Crvena krivulja prikazuje teorijsku gustoću. Histogram se odnosi na statistiku iz ponovljenih uzoraka, a ne na pojedinačna opažanja iz jednog uzorka. Argument `probability = TRUE` omogućuje usporedbu histograma s funkcijom gustoće.

Neka je $z_{1-\alpha/2}$ kvantil standardne normalne razdiobe. Iz događaja $-z_{1-\alpha/2} \leq Z \leq z_{1-\alpha/2}$ želimo izdvojiti μ. To radimo na sljedeći način

$$\begin{aligned} 1 - \alpha &= \mathbb{P}_\mu\left(-z_{1-\alpha/2} \leq \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \leq z_{1-\alpha/2}\right) = \mathbb{P}_\mu\left(-z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \overline{X} - \mu \leq z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right) \\ &= \mathbb{P}_\mu\left(-\overline{X} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}} \leq -\mu \leq -\overline{X} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right) = \mathbb{P}_\mu\left(\overline{X} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}} \leq \mu \leq \overline{X} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right). \end{aligned}$$

Dakle, $1 - \alpha$-pouzdani interval je

$$\left[\overline{X} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \overline{X} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right],$$

a njegova realizacija, za opaženi prosjek $\bar{x}$,

$$\left[\bar{x} - z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}, \quad \bar{x} + z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}\right]. \tag{7}$$

Primjer. Na uzorku od 100 opažanja iz normalne populacije dobiveno je $\bar{x} = 172.7$. Populacijska varijanca poznata je i iznosi 7. Računamo realizaciju 90% pouzdani interval za μ.

```
n <- 100
xbar <- 172.7
sigma <- sqrt(7)
alpha <- 0.10
pogreska <- qnorm(1 - alpha/2) * sigma / sqrt(n)
xbar + c(-1, 1) * pogreska
```

Dobivamo približno $[172.265, 173.135]$. Broj `pogreska` jest poluširina intervala, odnosno dopušteno odstupanje od točkaste procjene. Standardna pogreška prosjeka $\sigma/\sqrt{n}$ mjeri raspršenost prosjeka pri ponovljenom uzorkovanju; σ mjeri raspršenost pojedinačnih opažanja.

6.2 Nepoznata varijanca

U praksi σ najčešće nije poznat. Zamijenimo ga uzoračkom standardnom devijacijom S, koju računamo naredbom `sd()`. Tada vrijedi

$$T = \frac{\overline{X} - \mu}{S/\sqrt{n}} \sim t_{n-1}. \tag{8}$$

Primjer: simulacija razdiobe statistike T. Ponovimo uzorkovanje, ali u svakom uzorku poznati σ u nazivniku zamijenimo izračunatim `sd(x_sim)`. Dobivene vrijednosti uspoređujemo s gustoćom Studentove razdiobe s $n - 1 = 13$ stupnjeva slobode.

```
set.seed(303)
mu <- 5; sigma <- 2; n <- 14; B <- 10000
t_sim <- numeric(B)
for (b in 1:B) {
  x_sim <- rnorm(n, mean = mu, sd = sigma)
  t_sim[b] <- (mean(x_sim) - mu) / (sd(x_sim) / sqrt(n))
}
hist(t_sim, breaks = 40, probability = TRUE,
     main = "Razdioba statistike T", xlab = "T",
     col = "grey85")
curve(dt(x, df = n - 1), add = TRUE, col = "red", lwd = 2)
```

Uočite da se uzoračka standardna devijacija mijenja u svakom ponavljanju. Crvena krivulja sada prikazuje gustoću t_{13}, a ne standardnu normalnu gustoću.

Analognim postupkom kao gore (raspišite sami:), dobijemo da je za opaženi uzorak s prosjekom $\bar{x}$ i uzoračkom standardnom devijacijom s koristimo interval

$$\left[\bar{x} - t_{1-\alpha/2,n-1}\frac{s}{\sqrt{n}}, \quad \bar{x} + t_{1-\alpha/2,n-1}\frac{s}{\sqrt{n}}\right]. \tag{9}$$

Zadatak 9. *Izvedite formulu za gornji pouzdani interval.*

Studentova razdioba ima deblje repove od standardne normalne razdiobe. Za isti brojčani iznos $\sigma = s$ i istu razinu pouzdanosti t-interval je širi od intervala (7). Time uzimamo u obzir da smo raspršenost procijenili.

6.3 Račun iz sažetaka i iz cijelog uzorka

Primjer. Za 14 nezavisnih rezultata iz normalne populacije dobiveni su prosjek 30.93 boda i uzoračka standardna devijacija 5.15 bodova. Odredimo 95% pouzdani interval za očekivani broj bodova.

```
n <- 14
xbar <- 30.93
s <- 5.15
alpha <- 0.05
se <- s / sqrt(n)
pogreska <- qt(1 - alpha/2, df = n - 1) * se
interval_t <- xbar + c(-1, 1) * pogreska
interval_t
```

Dobivamo približno $[27.956, 33.904]$ boda. Uz rezultat navedemo veličinu uzorka, razinu pouzdanosti i pretpostavku o normalnoj populaciji.

Ako imamo sva pojedinačna opažanja, iste granice možemo dobiti funkcijom `t.test()`. Ona vraća više rezultata, a `$conf.int` izdvaja pouzdani interval. Testiranjem hipoteza bavit ćemo se na kasnijim vježbama.

```
x <- c(12.1, 11.8, 12.4, 12.0, 11.9, 12.3, 12.2, 11.7)
n <- length(x)
se <- sd(x) / sqrt(n)
rucno <- mean(x) + c(-1, 1) * qt(0.975, n - 1) * se
rucno
t.test(x, conf.level = 0.95)$conf.int
```

Granice su približno $[11.845, 12.255]$. Argument `conf.level` jest razina pouzdanosti, dakle $1-\alpha$, a ne α.

Ako podaci sadrže `NA`, najprije izdvojimo poznata opažanja naredbom `x <- x[!is.na(x)]`, a tek zatim računamo `length(x)`, `mean(x)` i `sd(x)`. Tako sva tri računa koriste isti uzorak.

Zadatak 10. *Za gornji vektor* `x` *izračunajte 90%, 95% i 99% interval funkcijom* `t.test()`*. Izračunajte njihove duljine i objasnite razliku.*

Zadatak 11. *Za opažene vrijednosti* $n = 14$, $\bar{x} = 30.93$ *i* $s = 5.15$ *zamijenite* `qt()` *funkcijom* `qnorm()`*. Usporedite granice. Zašto dobiveni normalni interval nije egzaktni 95% interval u ovom modelu?*

7 Pouzdani interval za varijancu

Sada želimo procijeniti raspršenost populacije, odnosno σ^2. Ponovno pretpostavljamo nezavisna opažanja iz iste normalne razdiobe. Za uzoračku varijancu S^2 vrijedi

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1}. \tag{10}$$

Neka su $c_L = \chi^2_{\alpha/2,n-1}$ i $c_U = \chi^2_{1-\alpha/2,n-1}$ odgovarajući kvantili. Vjerojatnost da gornji omjer bude između c_L i c_U iznosi $1-\alpha$. Izdvajanjem σ^2 dobivamo interval s opaženim granicama

$$\left[\frac{(n-1)s^2}{\chi^2_{1-\alpha/2,n-1}}, \quad \frac{(n-1)s^2}{\chi^2_{\alpha/2,n-1}}\right]. \tag{11}$$

Veći kvantil ide u nazivnik donje granice. Hi-kvadrat razdioba nije simetrična pa interval nije simetričan oko uzoračke varijance.

Zadatak 12. *Izvedite formulu za gornji interval.*

Primjer. Za $n = 14$ i $s = 5.15$ odredimo 90% pouzdani interval za varijancu normalne populacije.

```
n <- 14
s <- 5.15
alpha <- 0.10
kvantili <- qchisq(c(1 - alpha/2, alpha/2), df = n - 1)
interval_var <- (n - 1) * s^2 / kvantili
interval_var
sqrt(interval_var)
```

Bez zaokruživanja međurezultata dobivamo približno $[15.419, 58.520]$ za varijancu. Kvadratnim korijenom obiju granica dobivamo interval $[3.927, 7.650]$ za standardnu devijaciju. Ako su podaci mjereni u bodovima, varijanca se izražava u kvadratnim bodovima, a standardna devijacija u bodovima.

Zadatak 13. *Za vektor* `x` *iz prethodnog odjeljka izračunajte 95% interval za varijancu i standardnu devijaciju. Koristite* `var(x)` *i* `length(x)`*. Je li uzoračka varijanca na sredini intervala?*

8 Pouzdani interval za proporciju

Kada opažamo ima li pojedina jedinica neko svojstvo, modeliramo opažanja nezavisnim Bernoullijevim varijablama s istim parametrom p. Ako je k od n jedinica imalo svojstvo, točkasta procjena jest $\hat{p} = k/n$. Bernoullijeva varijabla ima očekivanje p i varijancu $p(1-p)$, pa je standardna pogreška uzoračke proporcije $\sqrt{p(1-p)/n}$. Budući da je p nepoznat, u standardnoj pogrešci zamijenimo ga konzistentnom procjenom $\hat{p}$. Centralni granični teorem i ta zamjena za fiksni $0 < p < 1$ opravdavaju sljedeću aproksimaciju kada je uzorak dovoljno velik:

$$W = \frac{\hat{p} - p}{\sqrt{\hat{p}(1-\hat{p})/n}} \approx N(0,1). \tag{12}$$

Oznaka $\approx N(0,1)$ označava aproksimaciju razdiobe, a ne jednakost: za fiksni p između nule i jedinice izraz konvergira po distribuciji prema $N(0,1)$ kada n raste.

Primjer: simulacija razdiobe standardizirane proporcije. Generiramo 10000 Bernoullijevih uzoraka veličine 400 s pravom proporcijom $p = 0.135$. U svakom uzorku računamo $\hat{p}$ i izraz iz (12), a histogram uspoređujemo sa standardnom normalnom gustoćom. Funkcija `rbinom(n, size = 1, prob = p)` generira n nezavisnih Bernoullijevih opažanja.

```
set.seed(305)
p <- 0.135; n <- 400; B <- 10000
w_sim <- rep(NA_real_, B)
for (b in 1:B) {
  x_sim <- rbinom(n, size = 1, prob = p)
  p_sim <- mean(x_sim)
  if (p_sim > 0 & p_sim < 1) {
    w_sim[b] <- (p_sim - p) / sqrt(p_sim * (1-p_sim) / n)
  }
}
sum(is.na(w_sim))
hist(w_sim[!is.na(w_sim)], breaks = 40, probability = TRUE,
     main = "Standardizirana proporcija", xlab = "W",
     col = "grey85")
curve(dnorm(x), add = TRUE, col = "red", lwd = 2)
```

Ako uzorak sadrži samo nule ili samo jedinice, nazivnik je nula pa je izraz nedefiniran; takva ponavljanja brojimo i izostavljamo iz histograma. Za ovdje odabrane parametre ona su vrlo malo vjerojatna. Histogram približno prati crvenu krivulju, ali zbog diskretnosti proporcije i konačne veličine uzorka podudaranje nije potpuno.

Približno $1-\alpha$ vjerojatnosti standardne normalne razdiobe nalazi se između $-z_{1-\alpha/2}$ i $z_{1-\alpha/2}$. Uvrštavanjem izraza (12) i izdvajanjem nepoznatog p iz te nejednakosti dobivamo **Waldov interval**:

$$\left[\hat{p} - z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \quad \hat{p} + z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right]. \tag{13}$$

To je aproksimativni interval. Pri malom broju uspjeha ili neuspjeha, osobito blizu $p = 0$ ili $p = 1$, njegova pokrivenost može biti loša. Kao grubu provjeru tražimo barem 10 uspjeha i 10 neuspjeha; ni taj uvjet ne jamči točno zadanu pokrivenost.

Zadatak 14. *Izvedite formulu za gornji interval.*

8.1 Račun i usporedba postupaka

Primjer. U slučajnom uzorku od 400 osoba njih 54 ima krvnu grupu 0 minus. Procijenimo proporciju u populaciji i izračunajmo 95% interval.

```
n <- 400
k <- 54
p_hat <- k / n
alpha <- 0.05
wald <- p_hat + c(-1, 1) * qnorm(1 - alpha/2) *
  sqrt(p_hat * (1 - p_hat) / n)
p_hat
wald
binom.test(k, n, conf.level = 1 - alpha)$conf.int
```

Procjena je $\hat{p} = 0.135$, a Waldov interval približno $[0.1015, 0.1685]$, odnosno od 10.15% do 16.85%. Funkcija `binom.test()` daje egzaktni Clopper–Pearsonov interval. Zbog diskretnosti njegova pokrivenost iznosi barem zadanu razinu, a ne nužno točno 95%. Granice se zato ne moraju podudarati s Waldovima.

Zadatak 15. *Za $k = 1$ i $n = 20$ izračunajte oba 95% intervala. Je li Waldov interval unutar $[0, 1]$? Objasnite zašto zamjena negativne granice nulom sama po sebi ne jamči željenu pokrivenost.*

9 Aproksimativni pouzdani intervali

Što ako populacija nije normalna? Za nezavisna opažanja iz iste populacije s konačnom, pozitivnom varijancom, centralni granični teorem opravdava normalnu aproksimaciju prosjeka kada je uzorak dovoljno velik. Tada se često koristi aproksimativni interval $\bar{x} \pm z_{1-\alpha/2} s/\sqrt{n}$, ili t-interval (9). Ne postoji univerzalna veličina uzorka koja jamči dobru aproksimaciju: asimetrija i izdvojene vrijednosti mogu zahtijevati znatno više opažanja. Kod malog uzorka pretpostavka o normalnosti posebno je važna. Histogram pomaže u pregledu podataka, ali sam po sebi ne dokazuje normalnost.

10 Simulacija: što znači 95% pouzdanosti?

U simulaciji znamo pravi parametar pa možemo provjeriti sadrži li ga interval. Generiramo 100 nezavisnih uzoraka veličine 10 iz $N(5, 2^2)$ i za svaki računamo 95% t-interval. Koristimo petlju s prethodnih vježbi.

```
set.seed(301)
mu <- 5
n <- 10
B <- 100
donji <- gornji <- numeric(B)
for (b in 1:B) {
  x_sim <- rnorm(n, mean = mu, sd = 2)
  e <- qt(0.975, df = n - 1) * sd(x_sim) / sqrt(n)
  donji[b] <- mean(x_sim) - e
  gornji[b] <- mean(x_sim) + e
}
pokriva <- donji <= mu & mu <= gornji
mean(pokriva)
plot(range(c(donji, gornji)), c(1, B), type = "n",
     xlab = "Ocekivanje", ylab = "Redni broj uzorka")
segments(donji, 1:B, gornji, 1:B,
         col = ifelse(pokriva, "grey40", "red"))
abline(v = mu, lty = 2)
```

Sivi intervali sadrže pravi parametar, a crveni ga promašuju. `mean(pokriva)` jest udio intervala koji su ga sadržavali. Ne očekujemo točno 95 pokrivanja u svakih 100 ponavljanja.

Zadatak 16. *Ponovite simulaciju s `B <- 10000`, bez crtanja grafa. Što se mijenja kada povećamo `B`, a što kada povećamo `n`?*

Zadaci za samostalni rad

Za svaki zadatak navedite parametar koji procjenjujete, pretpostavke, razinu pouzdanosti, granice intervala i zaključak u kontekstu podataka. Koristite R za račun, a međurezultate nemojte unaprijed zaokruživati.

Zadatak 17. *Masa pakiranja modelira se normalnom razdiobom s poznatom standardnom devijacijom od 4 grama. Na slučajnom uzorku od 64 pakiranja dobivena je prosječna masa od 201 grama.*

1. *Odredite realizaciju 95% pouzdanog intervala za očekivanu masu.*
2. *Kolika bi bila duljina intervala uz 256 opažanja i istu poznatu standardnu devijaciju? Je li za račun duljine potreban prosjek novog uzorka?*
3. *Koliko je opažanja potrebno da polušširina 95% intervala bude najviše 0.5 grama? Konačan broj zaokružite naviše funkcijom* `ceiling()`.

Zadatak 18. *Trajanje jednog postupka, u minutama, izmjereno je na nezavisnom slučajnom uzorku. Pretpostavite normalnu populaciju.*

```
trajanje <- c(9.8, 10.3, 10.1, 9.9, 10.5, 10.2,
              9.7, 10.0, 10.4, 10.1, 9.9, 10.2)
```

1. *Izračunajte veličinu uzorka, prosjek i uzoračku standardnu devijaciju.*
2. *Odredite realizaciju 95% intervala za očekivanje iz formule i provjerite ga funkcijom* `t.test()`. *Potom izračunajte realizaciju 99% intervala.*
3. *Odredite realizaciju 95% intervala za varijancu i standardnu devijaciju.*
4. *Student tvrdi: "95% postupaka traje između granica intervala za očekivanje." Objasnite pogrešku i napišite ispravno tumačenje pouzdanosti.*

Zadatak 19. *U slučajnom uzorku od 200 proizvoda pronađeno je 18 neispravnih.*

1. *Izračunajte procijenjenu proporciju i realizaciju Waldovog 95% interval.*
2. *Provjerite broj uspjeha i neuspjeha. Usporedite granice s onima iz* `binom.test()`. *Izrazite oba intervala u postocima.*
3. *Možemo li interval za proporciju tumačiti kao interval za broj neispravnih proizvoda u sljedećih 200 proizvoda? Objasnite.*

11 Hipoteze, p-vrijednost i zaključivanje

Na prošlim smo vježbama procjenjivali parametre. Sada ispitujemo jesu li podaci u skladu s određenom tvrdnjom o parametru. Da bismo proveli test prvo postavljamo hipoteze - nultu (što bi trebalo vrijediti) i alternativnu (što želimo "pokazati"). **Nulta hipoteza** H_0 daje referentnu vrijednost, a **alternativna hipoteza** H_1 odstupanje koje želimo otkriti. Primjerice, promatramo masu čokoladice za koju je označeno da teži 40 grama, čokoladice i slutimo da očekivana težina nije 40 grama, možemo postaviti $H_0 : \mu = 40$ prema $H_1 : \mu \neq 40$. Ako je naša slutanja da je masa u prosjeku veća , alternativa je $\mu > 40$, a za smanjenje $\mu < 40$. Smjer biramo prema pitanju *prije* pregleda konkretnog uzorka na kojemu ćemo raditi.

P-vrijednost jest, uz pretpostavku H_0 i ostale uvjete modela, vjerojatnost rezultata kao opaženi ili ekstremnijeg (s obzirom na odabranu statistiku i alternativu). Za dvostrani test oba

smjera odstupanja su važna; za jednostrani samo unaprijed odabrani smjer. Mala p-vrijednost znači da su opaženi podaci teško uskladivi s H_0 i pretpostavkama modela, odnosno da je mala vjerojatnost ako vrijedi H_0 (i pretpostavke) da smo opazili dobiveni uzorak.

Što znači mala vjerojatnost? Je li to 10%, 5% ili 0.05%, ovisi o primjeni. Zato prije testiranja, određujemo tzv. **razinu značajnosti**. Razinu značajnosti α odabiremo unaprijed, a najčešće je 0.05. Ako je p-vrijednost manja ili jednaka α, odbacujemo H_0; inače je ne odbacujemo. Primjerice, uz p-vrijednost 0.03 odbacujemo na razini 0.05, ali ne i na razini 0.01. P-vrijednost se pritom ne mijenja. **P-vrijednost nije vjerojatnost da je H_0 istinita.** Velika p-vrijednost ne dokazuje H_0: uzorak možda nije dovoljno velik da otkrije stvarno odstupanje. S tim ćemo povezati pojam jakosti.

U svakom primjeru odredit ćemo parametar, hipoteze i uvjete, zatim izračunati p-vrijednost i zaključiti u kontekstu. Svi brojčani podaci konstruirani su za vježbu. Oznake $\overline{X}$ i S^2 predstavljaju uzorački prosjek i nepristranu uzoračku varijancu; S je njezin korijen. Uzorci imaju barem dva opažanja i pretpostavljamo pozitivne populacijske varijance. Općenito, opažanja unutar svakog uzorka imaju istu razdiobu.

Zadatak 20. *Pri $\alpha = 0.05$ dobili smo p-vrijednost 0.12. Student tvrdi: "Dokazali smo H_0 s vjerojatnošću 88%." Ispravite zaključak.*

12 Jednostavni t-test: masa čokoladica

Primjer. Na pakiranju čokoladice piše da joj je masa 40 grama. Iz proizvodnje je nezavisno i nasumično odabrano 14 čokoladica te su izvagane bez omota.

$$40.12, 39.94, 40.21, 40.05, 39.88, 40.16, 40.08, 39.97, 40.25, 40.02, 40.11, 39.91, 40.18, 40.07$$

Razlikuje li se očekivana masa od deklariranih 40 g? Koristimo $\alpha = 0.05$.

Rješenje. Testiramo $H_0 : \mu = 40$ prema $H_1 : \mu \neq 40$. Pretpostavljamo nezavisan uzorak. Prvo gledamo pretpostavke, a to će biti da podaci dolaze iz normalne populacije. Za sada to provjeravamo grafički.

```
masa <- c(40.12, 39.94, 40.21, 40.05, 39.88, 40.16, 40.08,
            39.97, 40.25, 40.02, 40.11, 39.91, 40.18, 40.07)
hist(masa, breaks = 5, main = "Masa cokoladica",
     xlab = "Masa (g)")
n <- length(masa)
c(n = n, prosjek = mean(masa), sd = sd(masa))
```

Histogramom tražimo jaku asimetriju, odvojene skupine i izdvojene vrijednosti. Pregledamo nekoliko razumnih podjela. Nezavisnost obrazlažemo načinom odabira, a ne histogramom.

Pod pretpostavkom da vrijedi H_0 statistika

$$T = \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \sim t_{n-1}. \tag{14}$$

ima Studentovu razdiobu s $n-1$ stupnjeva slobode. Razliku prosjeka i nul-vrijednosti mjerimo u jedinicama standardne pogreške $S/\sqrt{n}$. Za dvostranu alternativu računamo oba repa:

$$p_{\text{vr}} = 2\mathbb{P}_{t_{n-1}}(T \geq |t_{\text{obs}}|). \tag{15}$$

```
t_obs <- (mean(masa) - 40) / (sd(masa) / sqrt(n))
p_rucno <- 2 * pt(abs(t_obs), n - 1, lower.tail = FALSE)
rez <- t.test(masa, mu = 40, alternative = "two.sided")
c(t = t_obs, p_rucno = p_rucno, p_R = rez$p.value)
```

Dobivamo $\bar{x} = 40.06786$, $t \approx 2.2345$, 13 stupnjeva slobode i p-vrijednost 0.04363. Kako je ona manja od 0.05, odbacujemo H_0: podaci podupiru očekivanu masu različitu od 40 g. Prosječ-

no odstupanje iznosi oko 0.068 g; njegovu praktičnu važnost procjenjujemo prema dopuštenoj toleranciji, a ne samo p-vrijednosti.

Za $H_1 : \mu > \mu_0$ koristimo desni rep distribucije t_{obs}, a za $H_1 : \mu < \mu_0$ lijevi. U jednostranom računu ne uzimamo apsolutnu vrijednost statistike. U R-u su pripadni argumenti `alternative = "greater"` i `alternative = "less"`.

12.1 Veza s prethodnim tjednom

Isti bi se test mogao opisati **kritičnim područjem**: skupom vrijednosti statistike za koje odbacujemo H_0. Za test razine α vjerojatnost ulaska u taj skup mora biti najviše α pri svakoj vrijednosti parametara iz H_0. Za usklađene dvostrane t-testove i F-test isti se zaključak može dobiti odgovarajućim $(1-\alpha)$-pouzdanim intervalom: gledamo isključuje li nul-vrijednost. Za očekivanje je to μ_0, za razliku očekivanja 0, a za omjer varijanci 1. Jednostrani test traži odgovarajući jednostrani interval iste razine. Na granici treba uskladiti uključivanje rubova. Kod Welchova testa i interval i test temelje se na istoj aproksimaciji. **Ovdje sve odluke donosimo preko p-vrijednosti.**

13 Upareni t-test: predmemorija i učitavanje stranica

Primjer. Na 12 nezavisno odabranih stranica mjerimo vrijeme učitavanja bez predmemorije i s njom, u sekundama. Zanima nas smanjuje li predmemorija očekivano vrijeme. Uvjeti su usporedivi, a redoslijed dvaju mjerenja unutar stranice nasumičan. Prije svakog mjerenja postavljamo zadano stanje predmemorije kako prethodno mjerenje ne bi promijenilo uvjete sljedećeg. Koristimo $\alpha = 0.05$.

Rješenje. Dva mjerenja iste stranice povezana su. Formiramo $D_i = X_i - Y_i$, gdje je X_i vrijeme bez predmemorije, a Y_i vrijeme s njom. Pozitivna razlika znači uštedu. Testiramo $H_0 : \mu_D = 0$ prema $H_1 : \mu_D > 0$.

```
bez <- c(1.8, 2.4, 1.6, 3.1, 2.7, 1.9,
         2.2, 3.4, 2.5, 2.0, 2.9, 1.7)
sa <- c(1.68, 2.08, 1.68, 2.85, 2.29, 1.85,
        2.02, 3.12, 2.41, 1.80, 2.94, 1.48)
d <- bez - sa
hist(d, breaks = 5, main = "Usteda vremena",
     xlab = "Bez - s predmemorijom (s)")
```

Normalnost pregledavamo na *histogramu razlika*. Pretpostavljamo normalnu populaciju razlika s pozitivnom varijancom i nezavisnost *između parova*. Zavisnost unutar para dopuštena je; pojedinačni nizovi vremena ne moraju svaki zasebno biti normalni.

Upareni test jednostavni je t-test nad razlikama:

$$T = \frac{\overline{D}}{S_D/\sqrt{n}} \sim t_{n-1} \quad \text{pod } H_0. \tag{16}$$

Broj $n = 12$ označava parove, a ne 24 nezavisna opažanja.

```
upareni <- t.test(bez, sa, paired = TRUE, alternative = "greater")
c(usteda = mean(d), t = unname(upareni$statistic),
  p = upareni$p.value)
```

Dobivamo $t \approx 3.9766$ i p-vrijednost $0.001086 < 0.05$. Odbacujemo H_0: podaci podupiru smanjenje očekivanog vremena. Procijenjena prosječna ušteda je 0.1667 sekundi; ne tvrdimo da je svaka stranica postala brža.

Na istim mjestima u vektorima moraju biti mjerenja iste stranice. Kod nedostajućih podataka zadržavamo potpune parove, primjerice funkcijom `complete.cases()`. Jednake duljine vektora same po sebi nisu razlog za uparivanje.

Zadatak 21. *Bez novog računa: ako zamijenimo vektore, koju alternativu trebamo za isto pitanje o uštedi? Što se mijenja u predznaku statistike?*

14 Dva nezavisna uzorka: potrošnja monitora

Primjer. U istim uvjetima izmjerena je snaga u vatima na 12 nezavisno odabranih monitora modela A i 10 modela B. Razlikuju li se očekivane potrošnje? Koristimo $\alpha = 0.05$.

Rješenje. Testiramo $H_0 : \mu_A - \mu_B = 0$ prema $H_1 : \mu_A - \mu_B \neq 0$. Nema prirodnih parova. Pretpostavljamo nezavisnost unutar skupina i između njih te normalnost svake populacije.

```
a <- c(27.8, 28.3, 28.1, 27.6, 28.5, 28.0,
       28.4, 27.9, 28.2, 28.6, 27.7, 28.1)
b <- c(28.4, 27.9, 28.8, 28.2, 28.6,
       27.8, 28.5, 28.0, 28.7, 28.3)
par(mfrow = c(1, 2))
hist(a, breaks = 4, main = "Model A", xlab = "Snaga (W)")
hist(b, breaks = 4, main = "Model B", xlab = "Snaga (W)")
par(mfrow = c(1, 1))
```

Pregledavamo histograme zasebno.

14.1 Jednake varijance: `var.equal = TRUE`

Vraćamo se usporedbi očekivanja: $H_0 : \mu_A - \mu_B = 0$ prema $H_1 : \mu_A - \mu_B \neq 0$. Uz dodatnu pretpostavku $\sigma_A^2 = \sigma_B^2 = \sigma^2 > 0$ koristimo zajedničku procjenu varijance:

$$S_p^2 = \frac{(n-1)S_A^2 + (m-1)S_B^2}{n+m-2}, \tag{17}$$

$$T = \frac{\overline{X} - \overline{Y}}{S_p\sqrt{1/n + 1/m}} \sim t_{n+m-2} \quad \text{pod } H_0. \tag{18}$$

```
jednake <- t.test(a, b, var.equal = TRUE)
jednake$p.value
```

Dobivamo $t \approx -1.5649$, 20 stupnjeva slobode i p-vrijednost 0.1333. Kako je ona veća od 0.05, ne odbacujemo H_0. Račun vrijedi uz opravdanu pretpostavku zajedničke varijance.

14.2 Zasebne varijance: `var.equal = FALSE`

Welchov postupak ne pretpostavlja jednake varijance. Kao i prije, brojnik je razlika prosjeka, koja pod H_0 procjenjuje razliku očekivanja 0. Kako procijeniti njezinu raspršenost? Prosjek prvog uzorka ima varijancu σ_A^2/n, a drugog σ_B^2/m. Zbog nezavisnosti uzoraka pri oduzimanju prosjeka *varijance se zbrajaju*:

$$\mathrm{Var}(\overline{X} - \overline{Y}) = \frac{\sigma_A^2}{n} + \frac{\sigma_B^2}{m}. \tag{19}$$

Nepoznate varijance procijenimo zasebno s S_A^2 i S_B^2 te uzmemo korijen. Tako dobivamo standardnu pogrešku razlike u nazivniku:

$$T_W = \frac{\overline{X} - \overline{Y}}{\sqrt{S_A^2/n + S_B^2/m}}. \tag{20}$$

Statistika govori koliko je opažena razlika udaljena od nule u jedinicama svoje standardne pogreške. Veća varijanca ili manji uzorak povećava doprinos te skupine pogrešci. Za razliku od prethodne inačice, ne udružujemo procjene u jednu zajedničku varijancu.

Pod H_0 razdiobu statistike *aproksimiramo* Studentovom razdiobom. Procijenjeni broj stupnjeva slobode za tu aproksimaciju daje Welch–Satterthwaiteova formula:

$$\widehat{\nu} = \frac{(S_A^2/n + S_B^2/m)^2}{\dfrac{(S_A^2/n)^2}{n-1} + \dfrac{(S_B^2/m)^2}{m-1}}. \tag{21}$$

Ovisi o veličinama uzoraka i objema procjenama varijance; općenito nije $n + m - 2$ i ne mora biti cijeli. R ga računa automatski i ne zaokružuje na cijeli broj. Studentova razdioba definirana je i za necijeli pozitivan broj stupnjeva slobode.

```
se_w <- sqrt(var(a)/length(a) + var(b)/length(b))
t_w <- (mean(a) - mean(b)) / se_w
welch <- t.test(a, b, var.equal = FALSE)
c(se = se_w, t = t_w, df = unname(welch$parameter),
  p = welch$p.value)
```

Ovdje je razlika prosjeka -0.22 W, standardna pogreška 0.14166 W, pa je $t_W \approx -1.5530$. Uz $\widehat{\nu} \approx 18.634$ dvostrana p-vrijednost je $0.1372 > 0.05$: ne odbacujemo H_0. Welchov je test zadani izbor u R-u i razuman izbor bez opravdanja za zajedničku varijancu.

14.3 F-test: usporedba varijanci

F-testom možemo testirati jesu li varijance različite. Testiramo $H_0 : \sigma_A^2/\sigma_B^2 = 1$ prema $H_1 : \sigma_A^2/\sigma_B^2 \neq 1$, uz $\alpha = 0.05$. Za nezavisne normalne uzorke pod pretpostavkom da vrijedi H_0 je

$$F = \frac{S_A^2}{S_B^2} \sim F_{n-1,m-1}. \tag{22}$$

F-razdioba nije simetrična. Za dvostrani test uzimamo dvostruku manju repnu vjerojatnost, najviše 1; taj ačun provodi `var.test()`.

```
f_test <- var.test(a, b, alternative = "two.sided")
c(F = unname(f_test$statistic), p = f_test$p.value)
```

Dobivamo $F \approx 0.85227$, stupnjeve slobode 11 i 9 te p-vrijednost $0.7893 > 0.05$. Ne odbacujemo jednake varijance, ali ih nismo dokazali. F-test posebno je osjetljiv na odstupanja od normalnosti: vraćamo se histogramima obiju skupina. Njegovu p-vrijednost ne koristimo kao automatsko pravilo za izbor inačice t-testa. Pitanja o očekivanju i varijanci su različita.

14.4 Zašto koristiti Welchov test? Simulacijska usporedba

Simuliramo dva nezavisna uzorka iz normalnih populacija s jednakim stvarnim očekivanjima $\mu_1 = \mu_2 = 0$. Prvi uzorak uvijek ima $n = 10$ i $\sigma_1 = 10$, a drugi $m \in \{10, 20, 40\}$ i $\sigma_2 \in \{2, 5, 10, 20, 50\}$. Za svaku od 15 kombinacija ponavljamo pokus 10 000 puta. Normalnost i nezavisnost ovdje vrijede po konstrukciji; u stvarnim podacima normalnost procjenjujemo histogramima.

U svakom ponavljanju na *istim podacima* provodimo dva dvostrana t-testa za $H_0 : \mu_1 = \mu_2$ prema $H_1 : \mu_1 \neq \mu_2$, uz $\alpha = 0.05$. Jedan pretpostavlja zajedničku varijancu, drugi je Welchov. Odbacujemo samo kada je p-vrijednost manja ili jednaka 0.05. Budući da znamo da je H_0 istinita, **ispravna odluka u ovom pokusu jest ne odbaciti je**. Za test koji dobro održava zadanu

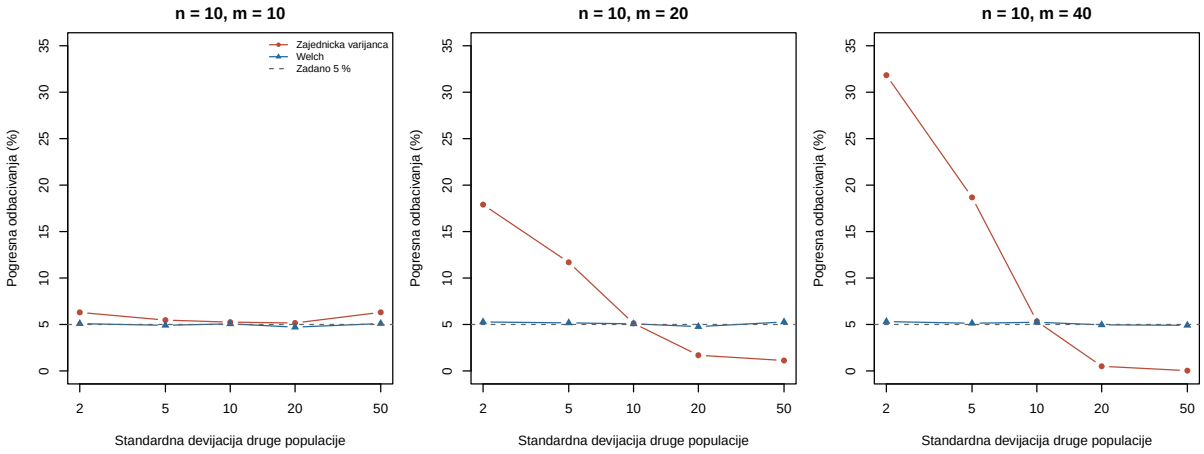
razinu očekujemo približno 95% ispravnih odluka i 5% pogrešnih odbacivanja. Neodbacivanje pritom ne znači da smo dokazali jednakost očekivanja.

Rezultati: postotak ispravnih odluka

U prvom je uzorku uvijek $n = 10$ i $\sigma_1 = 10$. Brojevi su postoci neodbacivanja istinite nul-hipoteze u 10 000 ponavljanja.

	$m = 10$		$m = 20$		$m = 40$	
σ_2	Zajednička	Welch	Zajednička	Welch	Zajednička	Welch
2	93,70	94,91	82,10	94,73	68,17	94,69
5	94,53	95,09	88,31	94,83	81,32	94,87
10	94,75	94,93	94,88	94,93	94,66	94,76
20	94,84	95,29	98,31	95,23	99,50	95,04
50	93,69	94,90	98,88	94,73	99,97	95,09

Svi su rezultati izraženi u postocima. Stupac *Zajednička* označava t-test koji pretpostavlja jednake varijance. Referentna je vrijednost približno **95%**, a ne 100%.



Zaključak. Kada manji uzorak dolazi iz populacije s većom varijancom, test sa zajedničkom varijancom prečesto odbacuje istinitu H_0. Primjerice, za $m = 40$ i $\sigma_2 = 2$ pogrešno odbacuje u 31.83% ponavljanja, a Welchov u 5.31%. Welchov test u svim prikazanim kombinacijama ostaje blizu zadanih 5% pogrešnih odbacivanja.

Više od 95% ispravnih odluka ovdje ne znači bolji test: test sa zajedničkom varijancom može biti previše konzervativan. Za usporedbu jakosti trebalo bi simulirati i različita očekivanja.

Na grafovima je na vodoravnoj osi σ_2, a na okomitoj postotak *pogrešnih odbacivanja*: on je 100% minus postotak iz tablice. Crvena linija prikazuje test sa zajedničkom varijancom, plava Welchov, a vodoravna isprekidana linija označava 5%. Kod je u `welch_simulacija.R`.

15 Jakost testa: koliko često otkrivamo stvaran učinak?

Neka slučajni uzorak X ima razdiobu određenu nepoznatim parametrom $\theta \in \Theta$. Parametar može biti vektor: u normalnom modelu obuhvaća i očekivanje i varijancu. Hipoteze zapisujemo kao $H_0 : \theta \in \Theta_0$ i $H_1 : \theta \in \Theta_1$, gdje su Θ_0 i Θ_1 disjunktni podskupovi prostora parametara. Za zadanu razinu α promatramo postupak koji odbacuje H_0 ako je p-vrijednost $p_{\text{vr}}(X) \leq \alpha$. Prije prikupljanja podataka uzorak, p-vrijednost i odluka su slučajni; stvarna vrijednost parametra θ je fiksna.

Alternativno, svaki test možemo zapisati kao funkciju $\tau : \mathbb{R}^n \to \{0, 1\}$, gdje je

$$\tau(\mathbf{x}) = \begin{cases} 1, & \text{odbacujemo } H_0 \\ 0, & \text{ne odbacujemo } H_0. \end{cases}$$

Test je u potpunosti određen svojim **kritičnim područjem**, odnosno skupom $C = \tau^{-1}(1)$. Na primjer, za t-test za jedan uzorak iz primjera s čokoladama, je $\tau(x_1, ..., x_n) = \mathbf{1}_{\left\{\frac{\bar{x}-40}{0.0304} \in C\right\}}$, za kritično područje $C = \langle -\infty, -t_{1-\alpha}(13)\rangle \cup \langle t_{1-\alpha}(13), \infty\rangle = \langle -\infty, -2.1603\rangle \cup \langle 2.1603, \infty\rangle$.

Zadatak 22. *Kako bi definirali funkciju τ za t-test za jedan uzorak uz alternativnu hipotezu $\mu > \mu_0$, a kako uz alternativnu hipotezu $\mu < \mu_0$?*

Stvarna vrijednost	Ne odbacujemo H_0	Odbacujemo H_0
$\theta \in \Theta_0$	Ispravna odluka	Pogreška prve vrste
$\theta \in \Theta_1$	Pogreška druge vrste	Ispravna odluka

Funkcija jakosti testa definirana je za svaku $\theta \in \Theta$ kao vjerojatnost odbacivanja H_0 kada podaci imaju razdiobu s parametrom θ:

$$\gamma(\theta) = \mathbb{P}_\theta(p_{\text{vr}}(X) \leq \alpha) = \mathbb{P}_\theta(\tau(X_1, ..., X_n) = 1), \qquad \theta \in \Theta. \tag{23}$$

Oznaka $\mathbb{P}_\theta$ znači da vjerojatnost računamo prema razdiobi uzorka pri toj stvarnoj vrijednosti parametra. Postupak testiranja, veličina uzorka i α pritom ostaju fiksni.

Ako je $\theta \in \Theta_0$, tada je $\gamma(\theta)$ vjerojatnost **pogreške prve vrste**. Kažemo da je test razine α ako vrijedi

$$\sup_{\theta \in \Theta_0} \gamma(\theta) \leq \alpha. \tag{24}$$

To je zahtjev na test, a ne automatsko svojstvo svakog postupka: za aproksimativne testove može vrijediti samo približno.

Ako je $\theta \in \Theta_1$, tada $\gamma(\theta)$ nazivamo **jakošću testa pri alternativi** θ. Vjerojatnost **pogreške druge vrste** pri istoj alternativi jest

$$\beta(\theta) = \mathbb{P}_\theta(p_{\text{vr}}(X) > \alpha) = 1 - \gamma(\theta), \qquad \theta \in \Theta_1. \tag{25}$$

Primjerice, jakost 0.80 pri određenoj alternativi znači da bismo pri toj stvarnoj vrijednosti parametra u približno 80% ponovljenih istraživanja odbacili H_0. U preostalih približno 20% počinili bismo pogrešku druge vrste. **Jakost nije vjerojatnost da je H_1 istinita**: ona opisuje odluke testa uz zadanu stvarnu alternativu.

15.1 Masa čokoladica: različita stvarna očekivanja

Promatramo nezavisan uzorak iz $N(\mu, 0.2^2)$ i dvostrani jednostavni t-test za $H_0 : \mu = 40$ prema $H_1 : \mu \neq 40$, uz $\alpha = 0.05$.

Prvo ćemo pokazati kako egzaktno računamo funkciju jakosti testa, a potom kako simulacijom procijeniti jakost testa.

Primjer. Funkcija jakosti za gornji t-test je

$$\gamma(\mu) = \mathbb{P}_\mu\left(\widetilde{T} \geq t_{1-\alpha/2}(n-1) - \frac{\mu - \mu_0}{S/\sqrt{n}}\right) + \mathbb{P}_\mu\left(\widetilde{T} \leq -t_{1-\alpha/2}(n-1) - \frac{\mu - \mu_0}{S/\sqrt{n}}\right), \quad \widetilde{T} \sim t(n-1)$$

Rješenje.

$$\gamma(\mu) = \mathbb{P}_\mu(\tau(X_1,...,X_n)=1) = \mathbb{P}_\mu\left(\left\{\frac{\overline{X}-\mu_0}{S/\sqrt{n}} \geq t_{1-\alpha/2}(n-1)\right\} \cup \left\{\frac{\overline{X}-\mu_0}{S/\sqrt{n}} \leq -t_{1-\alpha/2}(n-1)\right\}\right)$$
$$= \mathbb{P}_\mu\left(\frac{\overline{X}-\mu_0}{S/\sqrt{n}} \geq t_{1-\alpha/2}(n-1)\right) + \mathbb{P}_\mu\left(\frac{\overline{X}-\mu_0}{S/\sqrt{n}} \leq -t_{1-\alpha/2}(n-1)\right)$$
$$= \mathbb{P}_\mu\left(\frac{\overline{X}-\mu}{S/\sqrt{n}} + \frac{\mu-\mu_0}{S/\sqrt{n}} \geq t_{1-\alpha/2}(n-1)\right) + \mathbb{P}_\mu\left(\frac{\overline{X}-\mu}{S/\sqrt{n}} + \frac{\mu-\mu_0}{S/\sqrt{n}} \leq -t_{1-\alpha/2}(n-1)\right)$$
$$= \mathbb{P}_\mu\left(\frac{\overline{X}-\mu}{S/\sqrt{n}} \geq t_{1-\alpha/2}(n-1) - \frac{\mu-\mu_0}{S/\sqrt{n}}\right) + \mathbb{P}_\mu\left(\frac{\overline{X}-\mu}{S/\sqrt{n}} \leq -t_{1-\alpha/2}(n-1) - \frac{\mu-\mu_0}{S/\sqrt{n}}\right)$$
$$= \mathbb{P}_\mu\left(\widetilde{T} \geq t_{1-\alpha/2}(n-1) - \frac{\mu-\mu_0}{S/\sqrt{n}}\right) + \mathbb{P}_\mu\left(\widetilde{T} \leq -t_{1-\alpha/2}(n-1) - \frac{\mu-\mu_0}{S/\sqrt{n}}\right)$$

Nacrtati ćemo u R-u uz pomoć sljedećeg koda (koristimo podatke za čokoladice).

```
mu0 <- 40
uzorak <- c(40.12, 39.94, 40.21, 40.05, 39.88, 40.16, 40.08,
  39.97, 40.25, 40.02, 40.11, 39.91, 40.18, 40.07)
s <- sd(uzorak)
x_bar <- mean(uzorak)
t_alpha <- qt(0.05/2, length(uzorak)-1, lower.tail=F)
n<-length(uzorak)

fja_jakosti <- function(mu){
  lijevi <- pt(-t_alpha-(mu-mu0)/s*sqrt(n), df=n-1)
  desni <- pt(t_alpha-(mu-mu0)/s*sqrt(n), df=n-1, lower.tail=F)
  return (desni+lijevi)
}

mu <- seq(39.5, 40.5, by=0.01)
rez <- fja_jakosti(mu)

plot(mu, rez, type="l")
abline(0.05,0, col="red")
abline(v=40, col="gray")
```

Zadatak 23. *Odredite, a potom nacrtajte krivulju jakosti za analogan test uz alternativnu hipotezu* $H1 : \mu > 40$.

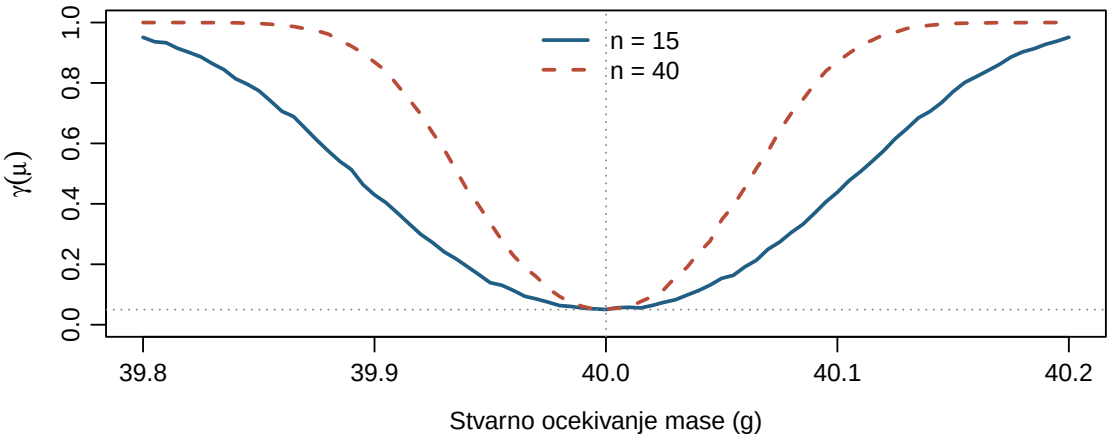
Pokažimo sada kako se ta krivulja dobije simulacijom. U simulaciji zadajemo stvarno očekivanje μ i standardnu devijaciju $\sigma = 0.2$ g. Test ne koristi poznatu vrijednost σ, nego je procjenjuje uzoračkom standardnom devijacijom, kao u poglavlju 2. Pri usporedbi jakosti mijenjamo μ, a $\sigma = 0.2$ ostaje fiksan; zato ovdje funkciju zapisujemo kao $\gamma(\mu)$. Za početak neka je stvarno $\mu = 40.1$ g. Tada je H_0 neistinita. Oznaka $B = 5000$ predstavlja broj ponavljanja simulacije: generiramo 5000 nezavisnih uzoraka veličine $n = 15$, za svaki izračunamo p-vrijednost i udjelom odbacivanja procjenjujemo $\gamma(40.1)$.

```
set.seed(405)
B <- 5000
jakost_sim <- function(mu, n) {
  p <- replicate(B, {
    x <- rnorm(n, mean = mu, sd = 0.2)
    t.test(x, mu = 40, alternative = "two.sided")$p.value
  })
  mean(p <= 0.05)
}
jakost_sim(40.1, 15)
jakost_sim(40.1, 40)
jakost_sim(40, 15)
```

Naredba `replicate(B, {...})` ponavlja zapisani postupak B puta i sprema p-vrijednosti; `mean(p <= 0.05)` računa udio odbacivanja. Prva dva rezultata procjenjuju jakost pri istoj stvarnoj razlici 0.1 g, za dvije veličine uzorka. Treći procjenjuje vjerojatnost pogreške prve vrste: pri $\mu = 40$ nul-hipoteza je istinita. Redom dobivamo približno 0.439, 0.870 i 0.052. Uz $n = 15$ ne otkrivamo stvarnu razliku u približno 56% ponavljanja, a uz $n = 40$ u približno 13%. Dobivene vrijednosti malo ovise o slučajnim uzorcima.

Zadatak 24. *U populaciji sa stvarnim očekivanjem* $\mu = 40.1$ *g dobili smo p-vrijednost* 0.18. *Koju smo odluku donijeli na razini* 0.05 *i koja se pogreška dogodila? Znamo li u stvarnom istraživanju pravi* μ*?*

15.2 Krivulje jakosti i veličina uzorka



Za svako stvarno očekivanje na vodoravnoj osi ponavljamo isti postupak. Graf prikazuje procijenjeni udio odbacivanja u 10 000 ponavljanja za $n = 15$ i $n = 40$, uz istu $\sigma = 0.2$ g. Kod za izradu grafa nalazi se u `graf_jakosti.R`.

Pri $\mu = 40$ obje su krivulje blizu 0.05: tada prikazuju vjerojatnost pogreške prve vrste. Za $\mu \neq 40$ prikazuju jakost. Udaljavanjem od 40 u bilo kojem smjeru jakost raste jer je test dvostran.

Pri istoj nenultoj razlici $\mu - 40$ veći uzorak ima veću jakost. Vjerojatnost pogreške druge vrste jest $1 - \gamma(\mu)$: čak i kada je $\mu \neq 40$, test može dati p-vrijednost veću od 0.05.

Što utječe na jakost? Veći uzorak i veća apsolutna razlika $|\mu - 40|$ povećavaju jakost ovog testa. Veća raspršenost podataka uz istu razliku otežava otkrivanje, a smanjenje α smanjuje jakost.

Broj opažanja i broj simulacija nisu isto. n je broj čokoladica u jednom uzorku, a B broj ponovljenih uzoraka. Veći n mijenja jakost testa. Veći B samo preciznije procjenjuje istu jakost. Neodbacivanje H_0 zato ne dokazuje da praktično važna razlika ne postoji.

Zadatak 25. *Bez novog računa predvidite kako se mijenja jakost ako povećamo n, smanjimo stvarni učinak ili smanjimo α. Što mijenja povećanje B?*

Zadaci za samostalni rad

Za svaki test navedite parametar, hipoteze i uvjete, pregledajte histogram kad je normalnost pretpostavka te izračunajte p-vrijednost i zaključite. Za račune pretpostavite normalne populacije (kod uparenog testa normalnu populaciju razlika), uz pozitivne varijance. Histogramom procijenite jesu li podaci očito protivni toj pretpostavci; histogram ne potvrđuje normalnost. Koristite $\alpha = 0.05$ ako nije drukčije zadano. Odluke donosimo samo preko p-vrijednosti. Dovoljni su gotovi R postupci; ne treba ponavljati ručni račun statistika ili izvode formula.

Zadatak 26. *Na nezavisnom uzorku žarulja izmjeren je svjetlosni tok u lumenima:*

```
svjetlo <- c(806, 794, 812, 803, 798, 809,
             801, 815, 797, 805, 810, 800)
```

1. *Razlikuje li se očekivanje od 800 lumena? Donesite odluke na razinama* 0.05 *i* 0.01 *koristeći istu p-vrijednost.*

2. *Na deset nezavisno odabranih perilica izmjerena je potrošnja vode u litrama u dvama odvojenim pranjima s jednakim opterećenjem. Između pranja vraćamo jednake početne uvjete. Redoslijed programa je nasumičan. Unaprijed pitamo troši li novi program manje vode.*

   ```
   standard <- c(62, 58, 71, 65, 60, 74, 68, 56, 70, 63)
   novi <- c(58, 57, 66, 62, 56, 69, 66, 53, 65, 61)
   ```

   Provedite prikladan test. Koji histogram pregledavate?

Zadatak 27. *Za dva načina očvršćivanja premaza izmjerena je tvrdoća na različitim, nezavisno odabranim uzorcima.*

```
tvrd_a <- c(69.8, 70.1, 69.9, 70.3, 70.0, 69.7,
            70.2, 70.1, 69.9, 70.4, 69.8)
tvrd_b <- c(68.6, 71.4, 69.2, 70.8, 72.0,
            68.9, 71.1, 69.7, 70.5, 68.4)
```

1. *Razlikuju li se očekivanja? Koristite Welchov test.*

2. *Razlikuju li se varijance? Provedite F-test.*

3. *Jesu li to ista pitanja? Koju bi dodatnu pretpostavku tražio t-test s* `var.equal = TRUE`*? Ne treba ga ponovno računati.*

Zadatak 28. *U simulaciji iz 5.1 postavite stvarno $\mu = 40$. Što sada procjenjuje udio odbacivanja i kojoj bi vrijednosti trebao biti blizu?*

Zadatak 29. *Neka su dani podaci iz normalne distribucije s nepoznatim očekivanjem i varijancom, gdje je* $n = 26$, $\overline{x} = 31.67$, $s^2 = 5.3$.

1. *Definirajte test (funkciju) za hipoteze* $H_0 : \mu = 33$, $H_1 : \mu < 33$.
2. *Odredite funkciju jakosti i nacrtajte ju u R-u.*
3. *Ako je stvarna vrijednost od* $\mu = 32$, *koja je vjerojatnost da smo napravili pogrešku druge vrste?*

Pete vježbe: Prilagodba modela podacima

Do sada smo za potrebe testova i računanja pouzdanih intervala, često pretpostavljali da podaci dolaze iz neke distribucije (obično normalne). Na ovim i idućim vježbama bavit ćemo se procjenom parametara distribucije temeljem dobivenog uzoraka.

16 Metoda momenata

Najjednostavnija metoda je metoda momenata. Ideja metode momenata je uspoređivati populacijske (teorijske) momente te uzoračke momente. Neka je $x_1, \ldots, x_n$ realizacija slučajnog uzorka iz $X \sim f(\cdot \mid \theta)$ gdje je $\theta \in \Theta \subseteq \mathbb{R}^d$. Izjednačavanjem prvih d teoretskih i uzoračkih momenata dobivamo sustav jednadžbi

$$\mathbb{E}(X) = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \mathbb{E}(X^2) = \frac{1}{n}\sum_{i=1}^{n} x_i^2, \ \ldots \ , \mathbb{E}(X^d) = \frac{1}{n}\sum_{i=1}^{n} x_i^d$$

gdje $\mathbb{E}(X^k), k = 1, \ldots, d$ ovise o $\theta = (\theta_1, \ldots, \theta_d)$. Rješavanjem tog sustava dobivamo procjenu metodom momenata

$$\hat{\theta} = (\hat{\theta_1}, \ldots, \hat{\theta_d}).$$

Primjer. Neka je dan uzorak duljine 20 iz normalne razdiobe $N(\mu, \sigma^2)$, takav da je $\overline{x} = 0.3$ i $s_x^2 = 0.05$. Koristeći metodu momenata dobijemo

$$\mu = \mathbb{E}(X) = \overline{x} = 0.3$$
$$\sigma^2 + \mu^2 = \mathbb{E}(X^2) = \frac{n-1}{n} s_x^2 + \overline{x}^2,$$

tj. procjene metodom momenata su $\widehat{\mu} = 0.3$ i $\widehat{\sigma}^2 = 0.0475$.

17 Procjenitelj maksimalne vjerodostojnosti (MLE)

Neka je $X_1, \ldots, X_n$ slučajni uzorak s gustoćom $f(\cdot \mid \theta)$ gdje je $\theta \in \Theta \subseteq \mathbb{R}^d$. Neka je $x_1, \ldots, x_n$ realizacija tog uzorka. Definiramo **funkciju vjerodostojnosti**

$$L(\theta | x_1, \ldots, x_n) = f_{X_1,\ldots,X_n}(x_1, \ldots, x_n \mid \theta) = \prod_{i=1}^{n} f(x_i|\theta).$$

Tada se **procjena metodom maksimalne vjerodostojnosti** definira kao

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L(\theta \mid x_1, \ldots, x_n)$$

a **procjenitelj maksimalne vjerodostojnost** je statistika

$$\hat{\theta}\,(X_1, \ldots, X_n)\,.$$

Primjer. Zanima nas je li novčić simetričan. Bacili smo ga 15 puta i dobili sljedeće rezultate

$$P, G, G, G, P, P, G, P, G, P, P, G, P, P, P.$$

Koristeći MLE procijenite parametar $p =$ vjerojatnost da će pasti P, $p \in \langle 0, 1 \rangle$.

Rješenje. Neka je X ishoda bacanja novčića i neka poprima vrijednosti 0 (ako je pala glava) i 1 (ako je palo pismo). To je slučajna varijabla iz $B(p)$ distribucije. Njena funkcija gustoće je

$$f(x_i|p) = \mathbb{P}_p(x_i = P) = p^{x_i}(1-p)^{1-x_i}.$$

Onda je

$$L(p|x_1, ..., x_{15}) = \prod_{i=1}^{15} p^{x_i}(1-p)^{1-x_i} = p^9(1-p)^6.$$

Maksimum tražimo putem stacionarne točke,

$$0 = \frac{d}{dp}L(p) = 9p^8(1-p)^6 - 6(1-p)^5p^9 = p^8(1-p)^5(9(1-p) - 6p).$$

S obzirom da su $p, 1-p \neq 0$, vrijedi $9 = 15p$, tj. $\widehat{p} = \frac{9}{15}$. Još treba argumentirati da je to zaista maksimum npr. promatranjem druge derivacije:

$$\frac{d^2}{dp^2}L(\widehat{p}) = -15\widehat{p}^8(1-\widehat{p})^5 < 0.$$

□

Dobili smo da je procijenjeni parametar 0.6. To nije nužno stvaran parametar distribucije i, posebno, ne znači da novčić nije simetričan. Da bi donijeli taj zaključak morali bismo provesti statistički test - i vidjeti je li razlika od 0.1 ($= 0.6 - 0.5$) dovoljno velika da zaključimo da novčić nije simetričan.

Radi jednostavnosti računa, često se promatra funkcija log-vjerodostojnosti definirana s

$$\ell(\theta \mid x_1, \ldots, x_n) = \ln L(\theta|x_1, \ldots, x_n) = \sum_{i=1}^{n} \ln f(x_i|\theta).$$

Kako je $y \mapsto \ln x$ strogo rastuća funkcija, $\hat{\theta}$ koji maksimizira $\ell(\cdot \mid x_1, \ldots, x_n)$ ujedno maksimizira i $L(\cdot|x_1, \ldots, x_n)$. [Invarijantnost MLE-a] Neka je $\hat{\theta}$ procjenitelj maksimalne vjerodostojnosti za θ te definiramo parametar $\alpha = g(\theta)$ gdje je g neka izmjeriva funkcija. Tada je $\hat{\alpha} = g(\hat{\theta})$ MLE procjenitelj za α.

Primjer. Neka je dan uzorak iz $N(\mu, \sigma^2)$ distribucije. Ako je $\widehat{\sigma}$ procjena standardne devijacije σ metodom maksimalne vjerodostojnosti, onda je $\widehat{\sigma}^2$ procjena varijance σ^2 (metodom maksimalne vjerodostrojnosti).

Da bismo otkrili procjenitelj metodom maksimalne vjerodostojnosti trebamo riješiti optimizacijski problem (minimum funkcije $L(\theta|x_1, ..., x_n)$. U R-u možemo iskoristiti funkciju `optimize()`. Na primjer, u primjeru s bacanjem novčića mogli smo pozvati

```
fja_bern <- function(br_p, n, p){
  return (p^br_p * (1-p)^(n-br_p))
}
optimize(fja_bern, interval = c(0.01, 0.99),
         maximum = T, n=15, br_p=9)
```

Funkcija `optimize` prima funkciju kojoj želimo naći minimum/maksimum. Po zadanim postavkama traži minimum, zato moramo postaviti `maximum = T`. Trebamo dati i interval na kojem traži optimizaciju zato postavljamo `interval = c(0.01, 0.99)`. Na kraju, s obzirom da naša funkcija prima 3 parametra (a `optimize` traži min/max po jednom parametru), moramo dati vrijednosti ostalih parametara. Alternativno, mogli smo definirati funkciju tako da ovisi samo o jednom parametru `p`, pa bi kod izgledao

```
fja_bern2 <- function(p){
  return (p^9 * (1-p)^6)
}
optimize(fja_bern2, interval = c(0.01, 0.99),
         maximum = T)
```

Općenito, jer je $L(\theta|x_1,...,x_n) = \prod_{i=1}^{1} 5f(x_i|\theta)$ umnožak funkcija gustoće, možemo koristiti funkcije gustoće dostupne u R-u. Konkretno, funkciju gustoće za $B(p)$ distribuciju u točki x_i (jer je $B(p) = Binom(1,p)$) možemo dobiti pozivom `dbinom(x_i, 1, p)`. Onda funkciju vjerodostojnosti definiramo kao

```
novcic <- c(1,0,0,0,1,1,0,1,0,1,1,0,1,1,1)
fja_bern3 <- function(p, podaci){
  L <- 1
  for (xi in podaci){
    L <- L*dbinom(xi, 1, p)
  }
  return(L)
}
optimize(fja_bern3, interval = c(0.01, 0.99),
         maximum = T, podaci=novcic)
```

Funkcija `optimize()` numerički traži vrijednost u kojoj funkcija postiže maksimum, zato za rješenje dobijemo $\widehat{p} = 0.6000004$ umjesto samo 0.6. Često nam je to dovoljno, a često i nije moguće egzaktno odrediti procjenitelj maksimalne vjerodostojnosti ručno pa je nužno koristiti funkciju `optimize()`.

Zadatak 30. *Nacrtajte histogram za sljedeće podatke*

$$4.499, 2.039, 0.166, 1.767, 3.900, 7.296, 1.559, 0.608, 0.046, 0.658, 6.689, 1.845.$$

Iz koje distribucije biste rekli da podaci dolaze? Procijenite parametare distribucije koristeći metodu momenata, a potom koristeći MLE.

18 Pearsonov hi-kvadrat test pripadnosti distribuciji

Promatramo statističko obilježje X te neka je $X_1, \ldots, X_n$ pripadni slučajni uzorak. Prvo promatramo slučaj jednostavne hipoteze

$$H_0 : X \sim F$$
$$H_1 : \text{ ne } H_0$$

gdje je F neka fiksna (diskretna) razdioba (npr. $F = B(6, 1/3)$). Podijelimo moguće realizacije distribucije F u k razreda $A_1, \ldots, A_k$ te promatramo testnu statistiku

$$H = \sum_{j=1}^{k} \frac{(N_j - n_j)^2}{n_j} \overset{H_0}{\sim} \chi(k-1)$$

gdje su $N_j = \sum_{i=1}^{n} 1_{\{X_i \in A_j\}}$ opažene frekvencije j-tog razreda, a $n_j = n \cdot p_j = n \cdot \mathbb{P}(X \in A_j|H_0)$ teoretske frekvencije j-tog razreda.

Ako podaci zaista dolaze iz distribucije F, očekujemo da će vrijediti $N_j \approx n_j$, tj. $|N_j - n_j| \approx 0$. U tom slučaju će realizacija H statistike (h) biti neki mali broj blizu 0. Ako podaci ne dolaze iz F razdiobe, h će biti veliki. Zato za danu razinu značajnosti α računamo kritično područje kao $C = [\chi^2_\alpha(k-1), +\infty\rangle$. Ako se testna statistika H realizira u C, odbacujemo nultu hipotezu. Drugi način donošenja odluke jest koristeći p-vrijednost. Računamo

$$p\text{-vrijednost} = \mathbb{P}_{H_0}(H > h)$$

te ako je ona manja od razine značajnosti α, odbacujemo nultu hipotezu u korist alternativne na razini značajnosti α.

Za provedbu hi-kvadrat testa **svaki razred mora imati očekivanu vrijednost bar** 5, ondnosno mora vrijediti $n_i \geq 5$, za svaki i. Ako to nije zadovoljeno, spajamo razrede dok ne dobijemo da to vrijedi. Inače test gubi snagu i ne možemo biti sigurni u ispravnost dobivenih rezultata.

Zadatak 31. *Generirajte* 50 *podataka iz Poissonove distribucije s parametrom* 3 *(postavite* `seed` *na* 450*). Nacrtajte barplot, izgledaju li podaci kao da su iz Poissonove distribucije? Napravite jednostavan Pearsonov hi-kvadrat test kojim provjervate dolaze li podaci iz* $P(\lambda)$ *distribucije.*

Rješenje.
```
set.seed(450)
podaci <- rpois(50, 3)
n<-50
barplot(table(podaci)/n) %ok, lici na Poissonovu distribuciju s lambda = 3

Ni <- table(podaci)
pi <- c(dpois(0:5, 3), ppois(5, 3, lower.tail=F))
ni <- n*pi
ni
```
Suma teorijskih vjerojatnosti grupa mora biti 1, zato su grupe ishodi $0, 1, ..., 5$ i ishod ≥ 6. Dobijemo da je

```
> ni
[1]  2.489353  7.468060 11.202090 11.202090  8.401568  5.040941  4.195897
```

Trebamo spojiti prve dvije grupe i zadnje dvije.

```
Ni <- c(Ni[1]+Ni[2], Ni[3:5], Ni[6]+Ni[7])
pi <- c(pi[1]+pi[2], pi[3:5], pi[6]+pi[7])
ni<-n*pi

h <- sum((Ni - ni)^2/ni)
k<-length(Ni)
pchisq(h, k-1, lower.tail=F)
```

Kao rezultat dobijemo 0.8449, odnosno ne možemo odbaciti H_0 u korist H_1 ni na jednoj razumnoj razini značajnosti. □

Možemo koristiti i naredbu `chisq.test()` u R-u. Naredba prima uzoračke (apsolutne) frekvencije grupa i pripadne teoretske vjerojatnosti (relativne frekvencije grupa).

```
chisq.test(Ni, p=pi)
```

Ako pokušate pozvati `chisq.test()` nad podacima bez grupiranja (gdje su neke frekvencije manje od 5) dobit ćete rezultat zajedno s upozorenjem da rezultat testa možda nije točan.

Zadatak 32. *Pozovite* `chisq.test()` *nad negrupiranim frekvencijama. Usporedite rezultat sa rezultatom dobivenim za grupirane.*

Pretpostavimo sada da želimo provesti Pearsonov χ^2-test, ali ovaj puta promatramo složenu hipotezu

$$H_0 : X \sim F(\cdot \mid \theta)$$
$$H_1 : \text{ ne } H_0$$

gdje je $F(\cdot \mid \theta)$ parametrizirana familija distribucija (npr. $B(n,p)$). Za $\theta_0 \in \Theta$, podijelimo moguće realizacije distribucije F u k razreda $A_1, \ldots, A_k$ te promatramo testnu statistiku

$$H = \sum_{j=1}^{k} \frac{(N_j - n_j(\theta_0))^2}{n_j(\theta_0)}$$

gdje su $N_j = \sum_{i=1}^{n} 1_{\{X_i \in A_j\}}$ opažene frekvencije j-tog razreda, a $n_j = n \cdot p_j(\theta_0) = n \cdot \mathbb{P}(X \in A_j \mid \theta_0)$ teoretske frekvencije j-tog razreda.

Koji parametar θ_0 odabrati? Ideja je uzeti θ_0 koji minimizira realizaciju testne statistike H. Takav se procjenitelj naziva procjeniteljem **minimum χ^2-metode**. Dakle

$$\hat{\theta} = \arg\min_{\theta_0 \in \Theta} \sum_{j=1}^{k} \frac{(N_j - n_j(\theta_0))^2}{n_j(\theta_0)}.$$

Tada testna statistika H ima $\chi^2(k-d-1)$ distribuciju (d je broj parametara koji smo procijenili). Zaključak o odbacivanju nulte hipoteze se donosi na sličan način kao i u slučaju jednostavne hipoteze.

Zadatak 33. *Za podatke iz prethodnog zadatka provedite Pearsonov hi-kvadrat test sa složenom hipotezom $H_0 : X \sim P(\lambda)$, H_1 : ne H_0.*

Rješenje. Prvi problem je kako grupirati podatke. Vjerojatno se sa različitim λ grupe različito mijenjaju. Za potrebe ovih vježbi ćemo grupirati s obzirom na dane uzoračke frekvencije. U početnom uzorku frekvencije su $7, 6, 10, 10, 9, 7, 1$ i potrebno je samo grupirati posljednje dvije. Tada je $N_i = \{7, 6, 10, 10, 9, 8\}$ za grupe $0, 1, ..., 4$ i ≥ 5. Onda je

$$\begin{cases} n_j(\lambda) = n \cdot \frac{e^{-\lambda}\lambda^j}{j!}, & j = 0, 1, ..., 4 \\ n_j(\lambda) = n \cdot (1 - \sum_{i=0}^{4} n_i), & j = 5 \end{cases}.$$

Minimum tražimo uz pomoć funkcije `optimize()`.

```
Ni_min <- table(podaci)
Ni_min <- c(Ni_min[1:5], sum(Ni_min[6:7]))
k<-length(Ni_min)

fja_min_chisq <- function(lambda){
  suma <- 0
  nj <- c()
  for (i in 0:4){
    nj[i+1] <- n*dpois(i, lambda)
  }
  nj[6]<- n*ppois(4, lambda, lower.tail=F)
  suma <- sum((Ni_min - nj)^2/nj)
  return (suma)
}

optimize(fja_min_chisq, c(1, 10))

#minimum
#[1] 2.643868
```

Dobijemo $\lambda = 2.643868$. Također, dobijemo i vrijednost H statistike u minimumum 5.929334. To možemo iskoristiti za odrediti p-vrijednost.

```
pchisq(5.929334, k-1-1, lower.tail=F)
```

Rezultat je 0.2045, odnosno ne odbacujemo pretpostavku da podaci dolaze iz $P(2.64)$ razdiobe. □

Zadatak 34. *Ponovite postupak, ali grupirajte i prve dvije grupe u jednu. Koju procjenu za λ dobijete?*

Pearsonov χ^2-test možemo provoditi i u slučaju kada su $F(\cdot \mid \theta)$ distribucije apsolutno neprekidnih slučajnih varijabli. Tipično, Kolmogorov-Smirnovljev test je jači i bolji odabir za apsolutno neprekidne slučajne varijable.

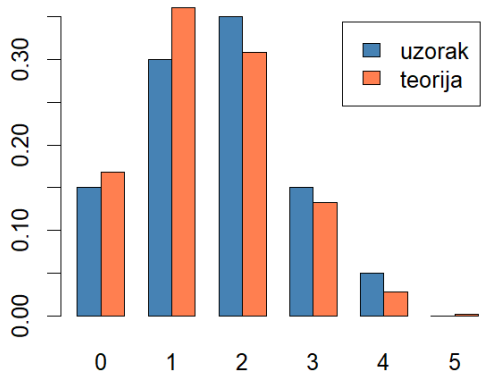
18.1 Vizualna usporedba uzoračke distribucije diskretne varijable s teorijskom

Ako želimo vizualno usporediti neprekidnu distribuciju s teorijskom, obično crtamo histogram, a potom funkciju gustoće na istom grafu. Ako dolaze iz iste distribucije, histogram bi trebao pratiti oblik funkcije gustoće. U diskretnom slučaju obično crtamo usporedni stupčasti dijagram. Za svaku vrijednost crtamo uzoračku vrijednost stupića pored teorijske visine stupića.

Primjer. Generirajmo uzorak duljine 20 iz $B(5, 0.3)$, te vizualno usporedimo s teorijskom distribucijom.

```
set.seed(6)
binomna <- rbinom(20, 5, 0.3)
uzoracka_d <- table(binomna)/20
uzoracka_d <- c(uzoracka_d, 0) #nije se dogodila 5
teorijska_d <- dbinom(0:5, 5, 0.3)
podaci <- matrix(c(uzoracka_d, teorijska_d), nrow=2, byrow=T)
barplot(podaci, beside=T, names=0:5, col=c("steelblue", "coral"),
        legend.text=c("uzorak", "teorija"))
```

Kao rezultat dobijemo sljedeći graf. Ključno je uzoračke i teorijske frekvencije spremiti u matricu



tipa $2 \times$ broj razreda. Potom se poziva naredba barplot nad tom matricom uz naredbu `beside=T`. Dodatno, dali smo imena stupićima (grupama) naredbom `names`, boje naredbom `col` te tekst za legendu (da bi osoba koja gleda graf znala što je uzoračka, a što teorijska frekvencija) naredbom `legend.text`.

Zadaci za samostalni rad

Proučavajući pravilnosti u pojavnosti niza oboljelih stabala, ekologistica E. C. Pielou mjerila je duljine neprekinutih nizova oboljelih stabala na plantažnim nasadima Douglasovih jela. Podaci

objavljeni u radu *Pielou, E. C. (1963) Runs of healthy and diseased trees in transects through an infected forest, BIOMETRICS, 19, 603-614* su

Duljina neprekinutog niza	1	2	3	4	5	6
Broj takvih nizova	71	28	5	2	2	1

1. Spremite zadane podatke u R-u.
2. Izračunajte ukupan broj podataka n.
3. Nacrtajte stupčasti dijagram relativnih frekvencija statističkog obilježja

$$X = \text{ Duljina neprekinutog niza oboljelih jela .}$$

4. Kojom distribucijom biste opisali distribuciju od X i zašto?
5. Odredite procjenu parametra distribuje θ metodom maksimalne vjerodostojnosti koristeći naredbu `optimize()`. Nazovite tu procjenu $\hat{\theta}$.
6. Nacrtajte stupčasti dijagram teorijske razdiobe s procjenom $\hat{\theta}$ od θ.
7. Usporedite dobiveni stupčasti dijagram sa stupčastim dijagramom empirijske razdiobe koristeći argument `beside`.
8. Odredite procjenu parametra distribucije θ minimalnom χ^2-metodom.
9. Provedite χ^2-test sa složenom hipotezom.
10. Provedite χ^2-test sa jednostavnom hipotezom $H_0 : X \sim G(0.7)$, $H_1 :$ ne H_0.