

# SAP - Projekt - Analiza teniskih mečeva

Benjak Petar, Bilić Ante, Kaštelan Niko, Paradžik Mario

## Motivacija i opis problema

Statistika i predviđanje sportskih rezultata mogu pomoći menadžerima, trenerima, procjeniteljima kvota i drugima u donošenju odluka. U tenisu je statistika kao alat dobila dodatnu popularnost zahvaljujući bivšem treneru Craigu O'Shaughnessyu, strategu s uporištem u statistici čija je analiza bila ključna u rezultatima Novaka Đokovića protiv njegovih najvećih rivala. Svojim zaključcima izvedenim iz povijesnih podataka mečeva tenisačima je moguće prilagoditi kondicijske pripreme, teniske treninge i strategiju protiv pojedinih protivnika, što rezultira boljom i konzistentnijom igrom.

## Učitavanje potrebnih paketa

```
library(dplyr)
library(tidyverse)
library(nortest)
```

## Učitavanje podataka

```
tennisMatches <- read.csv("tennis_atp_matches_full.csv")
```

## Dimenzija podataka

```
dim(tennisMatches)

## [1] 96602      53
```

## Nazivi varijabli

```
names(tennisMatches)

##  [1] "X.1"                  "X"                   "tourney_id"
##  [4] "tourney_name"          "surface"              "draw_size"
##  [7] "tourney_level"         "tourney_date"        "match_num"
## [10] "winner_id"             "winner_seed"         "winner_entry"
## [13] "winner_name"           "winner_hand"         "winner_ht"
## [16] "winner_ioc"            "winner_age"          "loser_id"
```

```

## [19] "loser_seed"           "loser_entry"          "loser_name"
## [22] "loser_hand"           "loser_ht"              "loser_ioc"
## [25] "loser_age"            "score"                "best_of"
## [28] "round"                 "minutes"              "w_ace"
## [31] "w_df"                  "w_svpt"               "w_1stIn"
## [34] "w_1stWon"              "w_2ndWon"             "w_SvGms"
## [37] "w_bpSaved"              "w_bpFaced"            "l_ace"
## [40] "l_df"                  "l_svpt"               "l_1stIn"
## [43] "l_1stWon"              "l_2ndWon"             "l_SvGms"
## [46] "l_bpSaved"              "l_bpFaced"            "winner_rank"
## [49] "winner_rank_points"     "loser_rank"            "loser_rank_points"
## [52] "winner_ohb"             "loser_ohb"             "winner_ohb"

```

### Prikaz podataka

```
view(tennisMatches)
```

### Tipovi varijabli u skupu podataka

```
sapply(tennisMatches, class)
```

X.1	X	tourney_id	tourney_name
## "integer"	"integer"	"character"	"character"
## surface	draw_size	tourney_level	tourney_date
## "character"	"integer"	"character"	"integer"
## match_num	winner_id	winner_seed	winner_entry
## "integer"	"integer"	"integer"	"character"
## winner_name	winner_hand	winner_ht	winner_ioc
## "character"	"character"	"integer"	"character"
## winner_age	loser_id	loser_seed	loser_entry
## "numeric"	"integer"	"integer"	"character"
## loser_name	loser_hand	loser_ht	loser_ioc
## "character"	"character"	"integer"	"character"
## loser_age	score	best_of	round
## "numeric"	"character"	"integer"	"character"
## minutes	w_ace	w_df	w_svpt
## "integer"	"integer"	"integer"	"integer"
## w_1stIn	w_1stWon	w_2ndWon	w_SvGms
## "integer"	"integer"	"integer"	"integer"
## w_bpSaved	w_bpFaced	l_ace	l_df
## "integer"	"integer"	"integer"	"integer"
## l_svpt	l_1stIn	l_1stWon	l_2ndWon
## "integer"	"integer"	"integer"	"integer"
## l_SvGms	l_bpSaved	l_bpFaced	winner_rank
## "integer"	"integer"	"integer"	"integer"
## winner_rank_points	loser_rank	loser_rank_points	winner_ohb
## "integer"	"integer"	"integer"	"logical"
## loser_ohb			
## "logical"			

Kod učitavanja podataka može doći do situacije gdje se tipovi podataka pogrešno prepoznaju pa ih je potrebno ručno izmjeniti. U ovom se slučaju krivo prepoznaju tipovi varijabli: tourney\_date, winner\_id te loser\_id.

```
tennisMatches <- tennisMatches %>% mutate(
  tourney_date = as.Date(as.character(tourney_date), "%Y%m%d"),
  winner_id = as.factor(winner_id),
  loser_id = as.factor(loser_id)
)

summary(tennisMatches)

##      X.1          X    tourney_id    tourney_name
##  Min.   : 1   Min.   : 0   Length:96602   Length:96602
##  1st Qu.:24151 1st Qu.:24150  Class :character  Class :character
##  Median :48302 Median :48301  Mode  :character  Mode  :character
##  Mean   :48302  Mean   :48301
##  3rd Qu.:72452 3rd Qu.:72451
##  Max.   :96602  Max.   :96601
##
##      surface      draw_size    tourney_level    tourney_date
##  Length:96602      Min.   : 4.00   Length:96602      Min.   :1990-12-31
##  Class :character  1st Qu.: 32.00   Class :character  1st Qu.:1997-04-21
##  Mode  :character  Median : 32.00   Mode  :character  Median :2004-06-14
##                      Mean   : 52.75
##                      3rd Qu.: 64.00
##                      Max.   :128.00
##
##      match_num     winner_id    winner_seed    winner_entry
##  Min.   : 1.00  103819 :1250   Min.   : 1.00  Length:96602
##  1st Qu.: 9.00  104745 :1012   1st Qu.: 3.00  Class :character
##  Median :22.00  104925 : 946   Median : 5.00  Mode  :character
##  Mean   :58.42   103970 : 740   Mean   : 6.85
##  3rd Qu.:48.00  101736 : 692   3rd Qu.: 8.00
##  Max.   :1701.00 101948 : 687   Max.   :35.00
##                      (Other):91275  NA's   :57628
##
##      winner_name    winner_hand    winner_ht    winner_ioc
##  Length:96602      Length:96602      Min.   :160.0  Length:96602
##  Class :character  Class :character  1st Qu.:180.0  Class :character
##  Mode  :character  Mode  :character  Median :185.0  Mode  :character
##                      Mean   :185.5
##                      3rd Qu.:190.0
##                      Max.   :211.0
##                      NA's   :4070
##
##      winner_age     loser_id    loser_seed    loser_entry
##  Min.   :14.35  103852 : 457   Min.   : 1.0  Length:96602
##  1st Qu.:22.96  104269 : 424   1st Qu.: 4.0  Class :character
##  Median :25.49  104022 : 423   Median : 6.0  Mode  :character
##  Mean   :25.74   102148 : 422   Mean   : 8.2
##  3rd Qu.:28.24  104312 : 404   3rd Qu.:11.0
##  Max.   :42.79  104259 : 388   Max.   :35.0
##  NA's   :58      (Other):94084  NA's   :75349
##
##      loser_name    loser_hand    loser_ht    loser_ioc
```

```

##  Length:96602      Length:96602      Min.   :160.0  Length:96602
##  Class :character  Class :character  1st Qu.:180.0  Class :character
##  Mode  :character  Mode  :character  Median :185.0  Mode  :character
##                                         Mean   :185.1
##                                         3rd Qu.:190.0
##                                         Max.   :211.0
##                                         NA's   :7671
##  loser_age       score       best_of      round
##  Min.   :14.51  Length:96602  Min.   :3.000  Length:96602
##  1st Qu.:23.00  Class :character 1st Qu.:3.000  Class :character
##  Median :25.63  Mode  :character Median :3.000  Mode  :character
##  Mean   :25.82                    Mean   :3.446
##  3rd Qu.:28.42                    3rd Qu.:3.000
##  Max.   :46.04                    Max.   :5.000
##  NA's   :127
##  minutes        w_ace       w_df       w_svpt
##  Min.   : 0.0  Min.   : 0.000  Min.   : 0.000  Min.   : 0.00
##  1st Qu.: 74.0 1st Qu.: 3.000  1st Qu.: 1.000  1st Qu.: 56.00
##  Median : 96.0 Median : 5.000  Median : 2.000  Median : 73.00
##  Mean   :102.8 Mean   : 6.493  Mean   : 2.745  Mean   : 78.03
##  3rd Qu.:124.0 3rd Qu.: 9.000  3rd Qu.: 4.000  3rd Qu.: 94.00
##  Max.   :1146.0 Max.   :113.000  Max.   :26.000  Max.   :491.00
##  NA's   :12410  NA's   :9793   NA's   :9793   NA's   :9793
##  w_1stIn        w_1stWon    w_2ndWon    w_SvGms
##  Min.   : 0.00  Min.   : 0.00  Min.   : 0.00  Min.   : 0.00
##  1st Qu.: 34.00 1st Qu.: 26.00 1st Qu.:12.00  1st Qu.: 9.00
##  Median : 44.00 Median : 33.00 Median :16.00  Median :11.00
##  Mean   : 47.44 Mean   : 35.76 Mean   :16.79  Mean   :12.38
##  3rd Qu.: 58.00 3rd Qu.: 43.00 3rd Qu.:21.00  3rd Qu.:15.00
##  Max.   :361.00 Max.   :292.00 Max.   :82.00  Max.   :90.00
##  NA's   :9793   NA's   :9793   NA's   :9793   NA's   :9793
##  w_bpSaved      w_bpFaced   l_ace       l_df
##  Min.   : 0.00  Min.   : 0.000  Min.   : 0.000  Min.   : 0.000
##  1st Qu.: 1.00  1st Qu.: 2.000  1st Qu.: 2.000  1st Qu.: 2.000
##  Median : 3.00  Median : 4.000  Median : 4.000  Median : 3.000
##  Mean   : 3.53  Mean   : 5.174  Mean   : 4.806  Mean   : 3.502
##  3rd Qu.: 5.00  3rd Qu.: 7.000  3rd Qu.: 7.000  3rd Qu.: 5.000
##  Max.   :24.00  Max.   :34.000  Max.   :103.000  Max.   :26.000
##  NA's   :9793   NA's   :9793   NA's   :9793   NA's   :9793
##  l_svpt         l_1stIn     l_1stWon    l_2ndWon
##  Min.   : 0.00  Min.   : 0.00  Min.   : 0.00  Min.   : 0.00
##  1st Qu.: 59.00 1st Qu.: 34.00 1st Qu.:22.00  1st Qu.: 10.00
##  Median : 76.00 Median : 44.00 Median :29.00  Median : 14.00
##  Mean   : 80.85 Mean   : 47.86 Mean   :31.78  Mean   : 15.02
##  3rd Qu.: 97.00 3rd Qu.: 58.00 3rd Qu.:39.00  3rd Qu.: 19.00
##  Max.   :489.00 Max.   :328.00 Max.   :284.00  Max.   :101.00
##  NA's   :9793   NA's   :9793   NA's   :9793   NA's   :9793
##  l_SvGms        l_bpSaved   l_bpFaced   winner_rank
##  Min.   : 0.00  Min.   :-6.000  Min.   : 0.000  Min.   : 1.00
##  1st Qu.: 9.00  1st Qu.: 2.000  1st Qu.: 6.000  1st Qu.: 18.00
##  Median :11.00  Median : 4.000  Median : 8.000  Median : 46.00
##  Mean   :12.18  Mean   : 4.813  Mean   : 8.752  Mean   : 81.35
##  3rd Qu.:15.00  3rd Qu.: 7.000  3rd Qu.:11.000 3rd Qu.: 89.00
##  Max.   :91.00  Max.   :28.000  Max.   :35.000  Max.   :2101.00

```

```

##  NA's    :9793    NA's    :9793    NA's    :9793    NA's    :1040
##  winner_rank_points  loser_rank    loser_rank_points  winner_ohb
##  Min.   : 1       Min.   : 1.0   Min.   : 1.0   Mode :logical
##  1st Qu.: 517     1st Qu.: 37.0   1st Qu.: 385.0  FALSE:72849
##  Median : 860     Median : 71.0   Median : 639.0  TRUE :23753
##  Mean   : 1387    Mean   : 119.9  Mean   : 867.6
##  3rd Qu.: 1551    3rd Qu.: 119.0  3rd Qu.: 1015.0
##  Max.   :16950    Max.   :2159.0  Max.   :16950.0
##  NA's   :2032     NA's   :2289    NA's   :3278
##  loser_ohb
##  Mode :logical
##  FALSE:77304
##  TRUE :19298
##
##
##
##

```

### Traženje nedostajućih vrijednosti

Dani skup podataka nerijetko sadrži nedostajuće podatke. Rad nad takvim podacima može dovesti do pogrešaka u testiranju hipoteza i zaključivanju. Varijable s velikim udjelom nedostajućih vrijednosti ćemo obraditi s ciljem da zadržimo informaciju koju sadrže.

```

for (col_name in names(tennisMatches)) {
  if (sum(is.na(tennisMatches[, col_name])) > 0) {
    cat("Ukupno nedostajućih vrijednosti za varijablu ", col_name, ":", ,
        sum(is.na(tennisMatches[, col_name])), "\n")
  }
}

```

```

## Ukupno nedostajućih vrijednosti za varijablu  winner_seed : 57628
## Ukupno nedostajućih vrijednosti za varijablu  winner_ht : 4070
## Ukupno nedostajućih vrijednosti za varijablu  winner_age : 58
## Ukupno nedostajućih vrijednosti za varijablu  loser_seed : 75349
## Ukupno nedostajućih vrijednosti za varijablu  loser_ht : 7671
## Ukupno nedostajućih vrijednosti za varijablu  loser_age : 127
## Ukupno nedostajućih vrijednosti za varijablu  minutes : 12410
## Ukupno nedostajućih vrijednosti za varijablu  w_ace : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_df : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_svpt : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_1stIn : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_1stWon : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_2ndWon : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_SvGms : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_bpSaved : 9793
## Ukupno nedostajućih vrijednosti za varijablu  w_bpFaced : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_ace : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_df : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_svpt : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_1stIn : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_1stWon : 9793
## Ukupno nedostajućih vrijednosti za varijablu  l_2ndWon : 9793

```

```

## Ukupno nedostajućih vrijednosti za varijablu l_SvGms : 9793
## Ukupno nedostajućih vrijednosti za varijablu l_bpSaved : 9793
## Ukupno nedostajućih vrijednosti za varijablu l_bpFaced : 9793
## Ukupno nedostajućih vrijednosti za varijablu winner_rank : 1040
## Ukupno nedostajućih vrijednosti za varijablu winner_rank_points : 2032
## Ukupno nedostajućih vrijednosti za varijablu loser_rank : 2289
## Ukupno nedostajućih vrijednosti za varijablu loser_rank_points : 3278

```

Varijabla winner\_seed ima 59% nedostajućih vrijednosti, a varijabla loser\_seed ima 78% nedostajućih vrijednosti što znači da se gubi znatno količina informacije koju sadrže. Obradit ćemo ih na način da zamjenimo NA vrijednosti sa (najveći\_seed + 1), što u ovom slučaju je 36.

```

tennisMatches$winner_seed[is.na(tennisMatches$winner_seed)] <- 36
tennisMatches$loser_seed[is.na(tennisMatches$loser_seed)] <- 36

```

## Problem 1

Možemo li nešto zaključiti iz distribucije visine najboljih deset igrača u posljednjih 30 godina u odnosu na distribuciju visine igrača koji nisu bili tako uspješni?

Potrebno je izdvojiti visine tenisača u dva različita skupa podataka. Prvi skup podataka sadrži jedinstven skup igrača koji su u posljednjih 30 godina bili u top deset najboljih u trenutku igranja meča, a drugi sadrži jedinstven skup igrača koji u posljednjih trideset godina u trenutku igranja meča nisu bili u top deset najboljih.

```

topTenW = tennisMatches[tennisMatches$winner_rank <= 10, c("winner_name", "winner_ht")]
topTenL = tennisMatches[tennisMatches$loser_rank <= 10, c("loser_name", "loser_ht")]

```

```

colnames(topTenW) = c("name", "ht")
colnames(topTenL) = c("name", "ht")

```

```

topTen = rbind(topTenW, topTenL)
topTen = topTen[!duplicated(topTen[, c("name")]), ]

```

```

cat("Pregled podataka o visini igrača u top deset najboljih\n")

```

```

## Pregled podataka o visini igrača u top deset najboljih

```

```

summary(topTen$ht)

```

```

##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.    NA's
## 170.0   183.0   188.0   186.8   190.0   206.0       1

```

```

length(topTen$ht)

```

```

## [1] 111

```

```

notTopTenW = tennisMatches[tennisMatches$winner_rank > 10, c("winner_name", "winner_ht")]
notTopTenL = tennisMatches[tennisMatches$loser_rank > 10, c("loser_name", "loser_ht")]

colnames(notTopTenW) = c("name", "ht")
colnames(notTopTenL) = c("name", "ht")

notTopTen = rbind(notTopTenW, notTopTenL)
notTopTen = notTopTen[!duplicated(notTopTen[, c("name")]), ]

cat("\nPregled podataka o visini igrača koji nisu u top deset najboljih\n")

##  

## Pregled podataka o visini igrača koji nisu u top deset najboljih  

summary(notTopTen$ht)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##    160.0    180.0   185.0    184.2    188.0    211.0    1287

length(notTopTen$ht)

## [1] 2637

```

Premda su u podatcima o visinama igrača koji nisu u top deset najboljih gotovo polovica nedostajeće vrijednosti, odlučujemo se za njihovo uklanjanje.

```

topTen = na.omit(topTen)
notTopTen = na.omit(notTopTen)

cat("Pregled podataka o visini igrača u top deset najboljih\n")

## Pregled podataka o visini igrača u top deset najboljih  

summary(topTen$ht)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    170.0    183.0   188.0    186.8    190.0    206.0

cat("\nPregled podataka o visini igrača koji nisu u top deset najboljih\n")

##  

## Pregled podataka o visini igrača koji nisu u top deset najboljih  

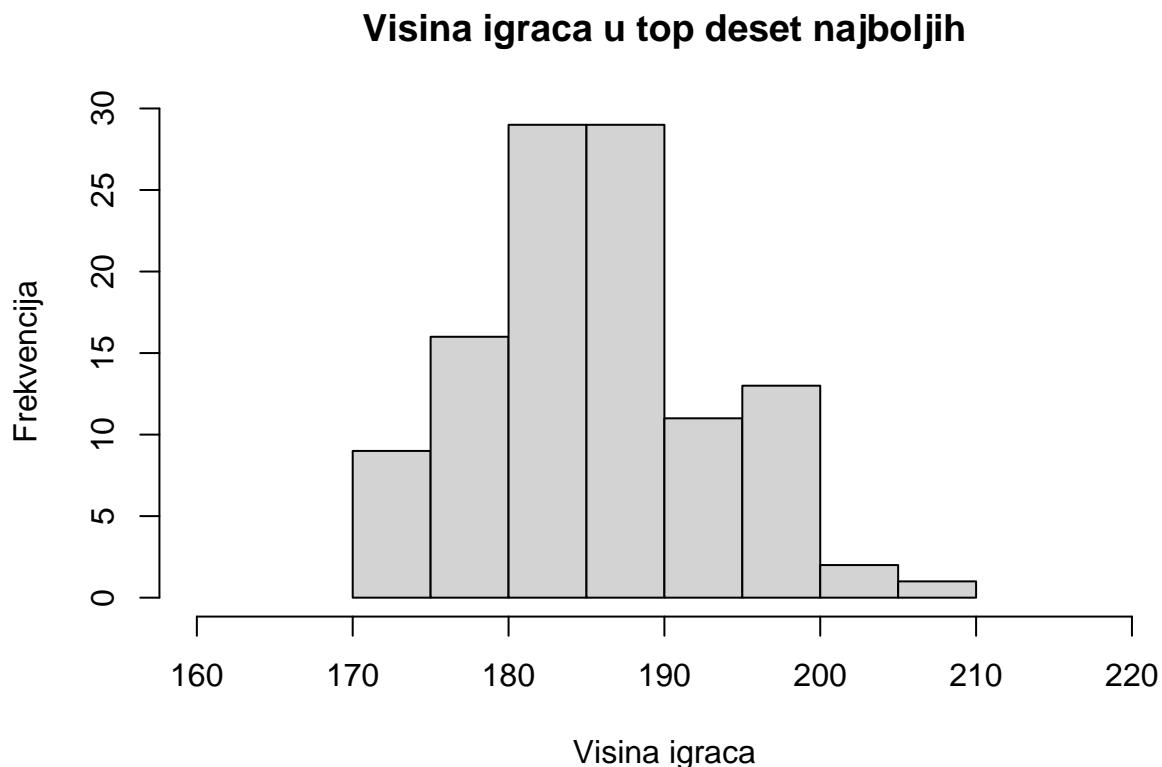
summary(notTopTen$ht)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    160.0    180.0   185.0    184.2    188.0    211.0

```

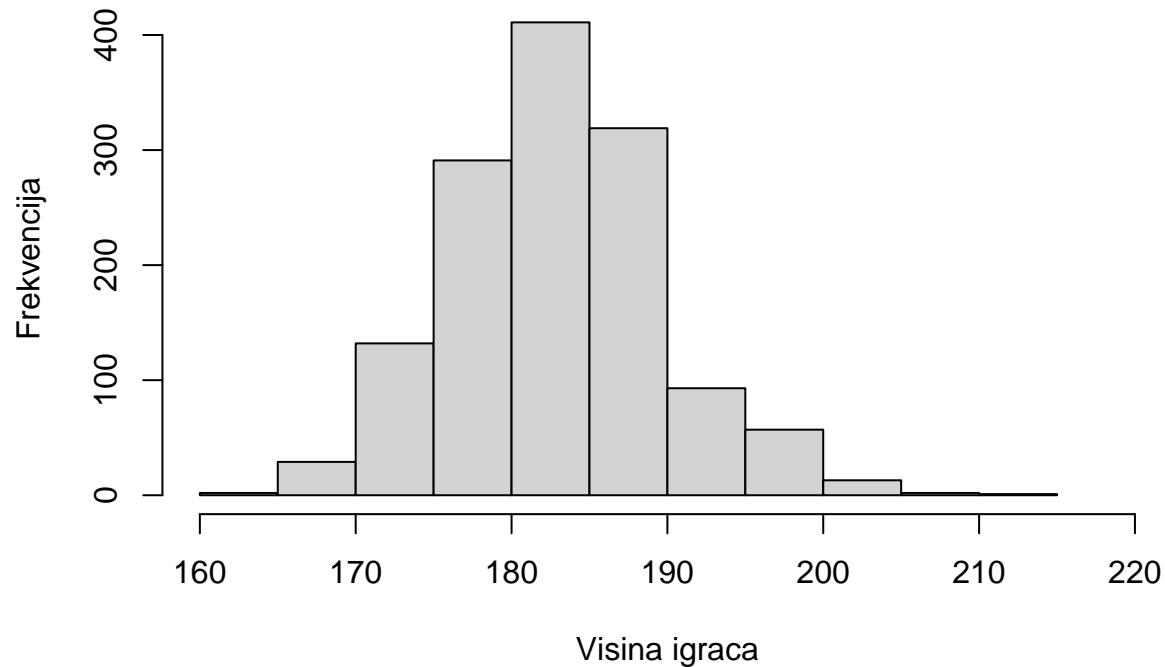
## Vizualizacija podataka

```
hist(topTen$ht,
  xlab="Visina igrača",
  ylab="Frekvencija",
  main="Visina igrača u top deset najboljih",
  xlim = c(160, 220)
)
```



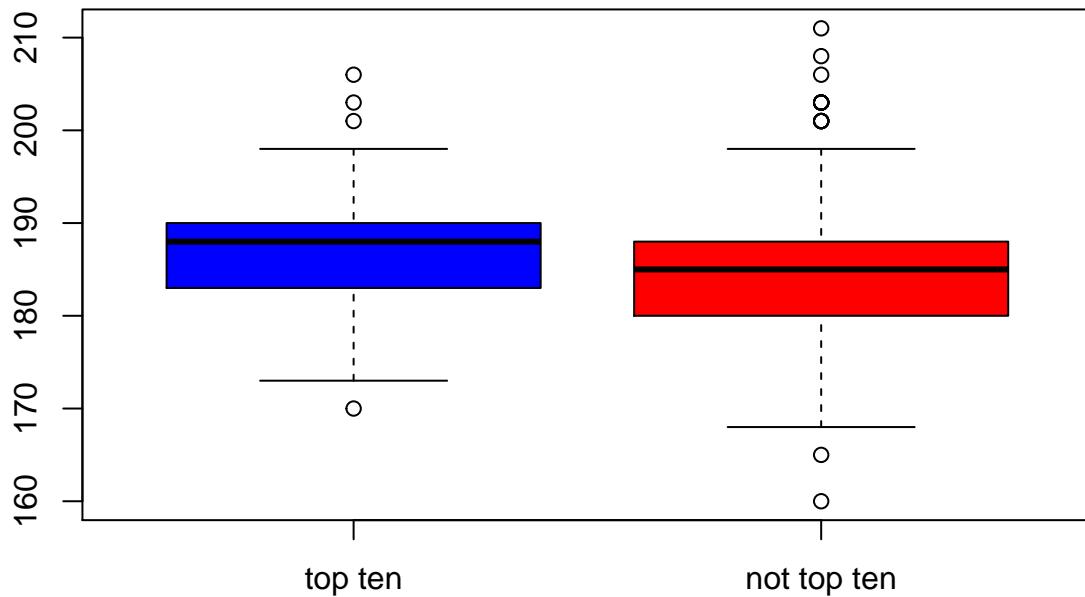
```
hist(notTopTen$ht,
  xlab="Visina igrača",
  ylab="Frekvencija",
  main="Visina igrača koji nisu u top deset najboljih",
  xlim = c(160, 220)
)
```

## Visina igraca koji nisu u top deset najboljih



```
boxplot(topTen$ht, notTopTen$ht,
        col = c("blue", "red"),
        names = c("top ten", "not top ten"),
        main="Visina igrača")
```

## Visina igraca

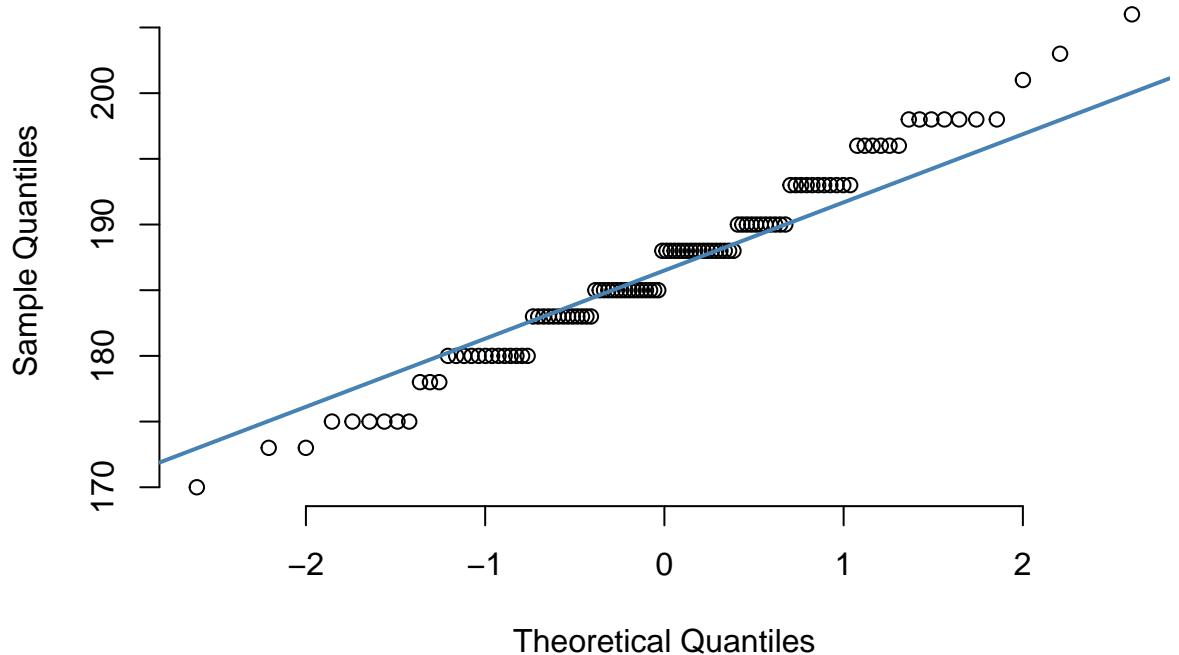


Prije određivanja testa potrebno je provjeriti normalnost podataka.

### QQ dijagram

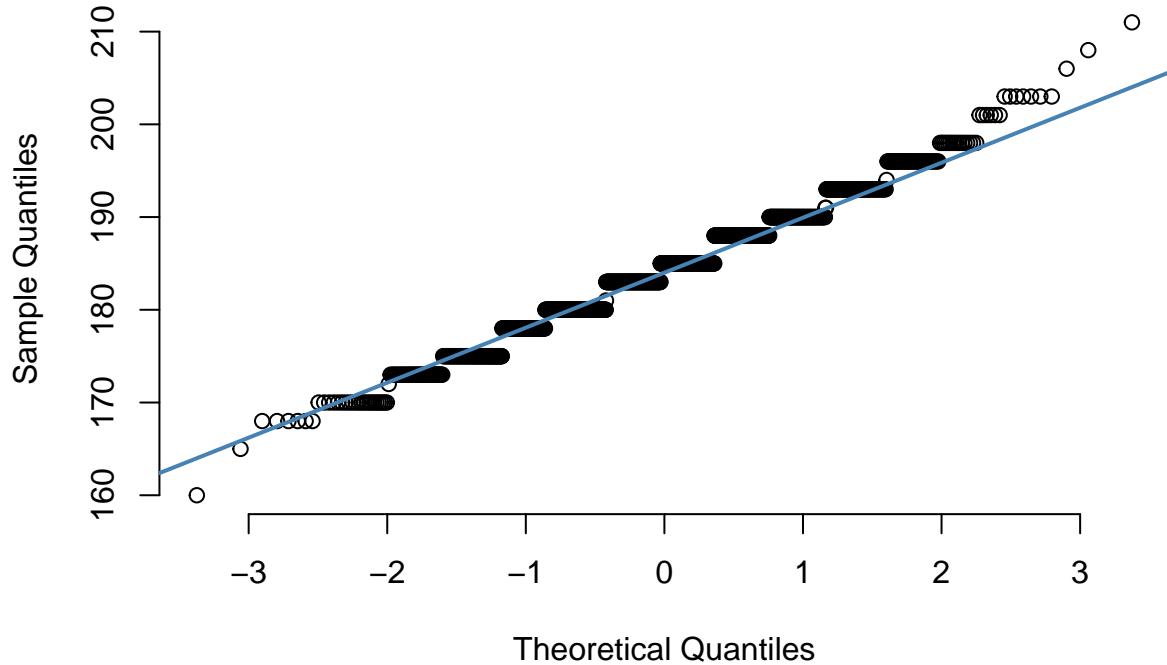
```
# QQ-dijagram
qqnorm(topTen$ht, pch = 1, frame = FALSE, main = "Visina top deset najboljih igrača")
qqline(topTen$ht, col = "steelblue", lwd = 2)
```

## Visina top deset najboljih igraca



```
qqnorm(notTopTen$ht, pch = 1, frame = FALSE,  
       main = "Visina igrača koji nisu u top deset najboljih ")  
qqline(notTopTen$ht, col = "steelblue", lwd = 2)
```

## Visina igraca koji nisu u top deset najboljih



Histogram te QQ-dijagram upućuju na manja odstupanja od normalnosti, no t-test je robustan na ne normalnost kada podaci imaju zvonoliku krivulju.

Odlučujemo se za korištenje t-testa te je potrebno provesti analizu jednakosti varijanci.

### Provjera jednakosti varijanci

```
varTopTen <- var(topTen$ht)
varNotTopTen <- var(notTopTen$ht)
cat("Varijanca top deset najboljih igrača: ", varTopTen, "\n")
```

```
## Varijanca top deset najboljih igrača: 49.09725
```

```
cat("Varijanca igrača izvan top deset najboljih: ", varNotTopTen)
```

```
## Varijanca igrača izvan top deset najboljih: 42.13795
```

### Test o jednakosti varijanci

Ako imamo dva nezavisna slučajna uzorka  $X_1^1, X_1^2, \dots, X_1^{n_1}$  i  $X_2^1, X_2^2, \dots, X_2^{n_2}$  koji dolaze iz normalnih distribucija s varijancama  $\sigma_1^2$  i  $\sigma_2^2$ , tada slučajna varijabla

$$F = \frac{S_{X_1}^2 / \sigma_1^2}{S_{X_2}^2 / \sigma_2^2}$$

ima Fisherovu distribuciju s  $(n_1 - 1, n_2 - 1)$  stupnjeva slobode, pri čemu vrijedi:

$$S_{X_1}^2 = \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (X_1^i - \bar{X}_1)^2, \quad S_{X_2}^2 = \frac{1}{n_2 - 1} \sum_{i=1}^{n_2} (X_2^i - \bar{X}_2)^2.$$

Hipoteze testa jednakosti varijanci glase:

$$\begin{aligned} H_0 : \sigma_1^2 &= \sigma_2^2 \\ H_1 : \sigma_1^2 &\neq \sigma_2^2 \end{aligned}$$

```
var.test(topTen$ht, notTopTen$ht)
```

```
## 
## F test to compare two variances
##
## data: topTen$ht and notTopTen$ht
## F = 1.1652, num df = 109, denom df = 1349, p-value = 0.2508
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.897764 1.562916
## sample estimates:
## ratio of variances
## 1.165155
```

### Zaključak

Ne možemo odbaciti hipotezu  $H_0$  o jednakosti varijanci pri razini značajnosti od 5% s obzirom da je p-vrijednost 0.2508.

### Provđba t-testa

Kod testiranja jednakosti očekivanja dvaju nezavisnih uzorka uz pretpostavku da oni potiču iz normalne distribucije, koristi se testna statistika

$$T = \frac{\mu_1 - \mu_2}{s_p^2 \sqrt{\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

gdje je

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

Odabrana testna statistika ima Studentovu distribuciju sa  $n_1 + n_2 - 2$  stupnjeva slobode.

$H_0$  : visina igrača koji su u posljednjih 30 godina bili u top 10 najboljih je jednaka visini igrača koji u posljednjih 30 godina nisu tako uspješni

$H_1$  : visina igrača koji su u posljednjih 30 godina bili u top 10 najboljih je veća u odnosu na visinu igrača koji u posljednjih 30 godina nisu tako uspješni

odnosno

$$H_0 : \mu_{10} = \mu_{n10}$$

$$H_1 : \mu_{10} > \mu_{n10}$$

```
t.test(topTen$ht, notTopTen$ht, alt = "greater", var.equal = TRUE)

##
## Two Sample t-test
##
## data: topTen$ht and notTopTen$ht
## t = 3.9747, df = 1458, p-value = 3.695e-05
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
## 1.508171      Inf
## sample estimates:
## mean of x mean of y
## 186.8000 184.2259
```

## Zaključak

Na razini značajnosti od 1% odbacujemo hipotezu  $H_0$  premda je p-vrijednost 3.695e-05, postoji značajna razlika visina igrača u top deset najboljih u posljednjih 30 godina i onih izvan top deset najboljih. Visina igrača u top deset najboljih je veća od visine igrača koji nisu u top deset najboljih.

Iz provedenog testa ne možemo zaključiti da visina izravno utječe na uspješnost igrača, već samo korelaciju između visine i uspješnosti igrača.

## Problem 2

### Predviđa li pobjeda prvog seta pobjedu cijelog meča?

Pobjeda prvog meča potencijalno može utjecati na psihološko stanje igrača te njegov stil igre, posebno u količini rizika koje igrač uzima tokom meča, ovisno o tome je li pobijedio ili izgubio prvi set. Postavlja se pitanje da li pobjeda prvog seta predviđa pobjedu cijelog meča.

Kako bi se problem testirao potrebno je izdvojiti podatke o mečevima u kojima je pobijedio onaj tenisač koji je pobijedio i prvi set, te podatke o mečevima u kojima je izbubio onaj tenisač koji je pobijedio prvi set.

Podaci o setovima mogu sadržavati W/O koji označuju predaju ili diskvalifikaciju, takve je podatke potrebno ukloniti iz skupa podataka.

```
# Uklanjanje mečeva kojima je score W/O
tennisMatchesSet <- tennisMatches[!is.na(as.numeric(substr(tennisMatches$score, 1, 1))), ]

## Warning in '[.data.frame'(tennisMatches, !
## is.na(as.numeric(substr(tennisMatches$score, : NAs introduced by coercion

# Izdvajanje rezultata
scores <- tennisMatchesSet$score
# Dohvaćanje prvog seta
firstSet <- substr(scores, 1, 3)
```

Podatke ćemo preslikati u vrijednosti 1 i -1, odnosno, ako je pobjednik dobio prvi set tada rezultat označavamo s 1, a ako je gubitnik dobio prvi set rezultat označavamo s -1. Ako pobjeda prvog seta nije povezana s pobjedom meča, tada očekujemo da srednja vrijednost tako transformiranih podataka neće biti značajno različita od 0.

```

eFunc <- function(strSet){
  wScore = as.numeric(substr(strSet, 1 , 1))
  lScore = as.numeric(substr(strSet, 3,3))
  if(wScore > lScore){
    #W je pobijedio prvi set
    return(1)
  }else if(lScore > wScore){
    #L je dobio prvi set
    return(-1)
  }else{
    return(0)
  }
}
extractedData <- c()
for(i in firstSet){
  extractedData <- c(extractedData, eFunc(i))
}

```

```
cat("Srednja vrijednost transformiranih podataka\n")
```

```
## Srednja vrijednost transformiranih podataka
```

```
mean(extractedData)
```

```
## [1] 0.6139984
```

### Provjeda testa

Koristit ćemo t-test s jednostranom alternativom čija je testna statistika

$$T = \frac{\mu}{s\sqrt{n}}$$

sa  $n - 1$  stupnjeva slobode.

$H_0$  : pobjeda prvog seta predviđa pobjedu cijelog meča  $H_1$  : pobjeda prvog seta ne predviđa pobjedu cijelog meča

odnosno, uz transformaciju podataka,

$$H_0 : \mu = 0$$

$$H_1 : \mu > 0$$

```
t.test(extractedData, alt="greater")
```

```
##
##  One Sample t-test
##
## data:  extractedData
## t = 241.34, df = 96167, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 0
```

```

## 95 percent confidence interval:
## 0.6098136      Inf
## sample estimates:
## mean of x
## 0.6139984

```

## Zaključak

Odbacujemo nultu hipotezu u korist alternativne hipoteze, koja tvrdi da pobjeda prvog seta predviđa pobjedu cijelog meča, na razini značajnosti od 1% premda je p-vrijednost dobivena provedbom testa manje od 2.2e-16.

Iz provedenih testova možemo zaključiti korelaciju podbjede prvog seta u svezi s pobjedom meča, ali ne možemo zaključiti da pobjeda prvog seta uzrokuje pobjedu meča.

## Problem 3

**Možemo li temeljem danih varijabli predvidjeti pobjednika teniskog meča?**

S ciljem predviđanja pobjednika tensikog meča, možemo procijeniti regresijski model s danim podacima kao nezavisnim varijablama. U ovom slučaju ćemo se korisiti logističkom regresijom. Imamo na raspolaganju skup podataka  $D = \{X_1, \dots, X_N\}$  gdje je svaki  $X_i$  vektor vrijednosti prediktorskih varijabli, one mogu biti diskretne (uz prikladno dummy-kodiranje) ili kontinuirane. Imamo i skup očekivanih izlaza  $\{y_1, \dots, y_n\}$  gdje je svaki  $y_i$  binarna varijabla tj. 0 ili 1. Želimo dobiti kao izlaz modela skup izlaza  $\{\hat{y}_1, \dots, \hat{y}_N\}$ . Idealno bismo od dobrog modela očekivali da bude (što je češće moguće)  $\hat{y}_i = y_i$ , tj. da radi dobre predikcije. Također, želimo imati vjerojatnost  $P(\hat{Y}_i = 1|x_i)$  koja bi nam dala mjeru koliko je model "siguran" u svoju odluku i omogućavala da izračunamo predikcije na sljedeći način

$$\hat{y}_i = \begin{cases} 1 & \text{ako } P(\hat{Y}_i = 1|\vec{x}_i) \geq 0.5 \\ 0, & \text{inače} \end{cases}$$

Glavni problem zbog kojeg ne možemo koristiti linearnu regresiju za ovaj zadatak je što  $\beta^T X$  može poprimiti vrijednosti van intervala  $[0, 1]$  pa izlaz linearne regresije ne možemo interpretirati kao vjerojatnost.

Logistička regresija rješava taj problem tako što transformira  $\beta^T X$  koristeći logističku (sigmoidalnu) funkciju:

$$\sigma(\alpha) = \frac{1}{1 + e^{-\alpha}}$$

Model dakle prikazuje gore traženu vjerojatnost na sljedeći način:

$$P(\hat{Y}_i = 1|X_i) = \frac{1}{1 + e^{-\beta^T X_i}}$$

Uz to što za svaki  $x_i$  možemo dobiti vjerojatnost da je pripadni  $y_i$  jednak 1, možemo i donijeti binarne odluke na temelju usporedbe dobivene vjerojatnosti s pragom od 0.5.

## Pripremanje podataka

Za svrhu učenja modela prvo trebamo pripremiti podatke, odnosno promijeniti njihov format u onaj prikladan za učenje modela. U našem slučaju, ovaj korak se sastoji od razdvajanja podataka pobjednika i gubitnika, te naknadnog spajanja podataka na način da svaki redak predstavlja podatke igrača u pojedinom meču. Također izbacujemo nedostajuće vrijednosti te dodajemo dodatni stupac "won" koji predstavlja zavisnu varijablu u treniranju modela. Pobjednik meča u stupcu "won" je označen brojem 1, a gubitnik brojem 0.

```

winners <- tennisMatches[ , grepl( "w" , names( tennisMatches ) ) ]
winners <- winners[,-1]
losers <- tennisMatches[ , grepl( "l" , names( tennisMatches ) ) ]
losers <- losers[,-1]

colnames(winners) <- colnames(losers) <- c("id", "seed", "entry", "name", "hand",
"ht", "ioc", "age", "ace", "df", "svpt", "FirstIn", "FirstWon", "SecondWon",
"SvGms", "bpSaved", "bpFaced", "rank", "rank_points", "ohb")

winners$won <- 1
losers$won <- 0
player = rbind(winners, losers)
player <- na.omit(player)

```

## Učenje modela

Kako bismo naučili dobre vrijednosti za  $\beta$  koristimo postupak procjene najveće izglednosti (vjerojatnosti) (engl. *Maximum Likelihood Estimation*). Za neki fiksni vektor težina  $\beta$  možemo izračunati vjerodostojnost koju model daje našem cijelom skupu podataka. Npr. ako je  $D = \{X_1, X_2, X_3\}$  i skup točnih izlaza je 1, 1, 0 tada je vjerodostojnost podataka uz model logističke regresije koji koristi te konkretne težine jednaka

$$P(D|\beta) = P(Y_1 = 1|X_1)P(Y_2 = 1|X_2)(1 - P(Y_3 = 1|X_3)).$$

Ova veličina se još zove izglednost (vjerojatnost)  $L(\vec{\beta})$  parametara uz dane podatke. Da smo uzeli neki drugi skup težina  $\beta'$ , dobili bismo neku drugu vjerodostojnost  $L(\beta')$ . Algoritam učenja radi tako pronađe onaj skup težina  $\beta$  koji maksimizira ovu veličinu. Upravo taj skup težina najbolje opisuje podatke. Kao kod linearne regresije i ovdje možemo odrediti koje značajke su statistički značajne. U `summary` naredbi modela logističke regresije R će nam također ispisati i devijancu (engl. *deviance*). To je mjera zasnovana na izglednosti i opisuje nam koliko je model dobar, u smislu koliko dobro se prilagodio podacima (veći broj znači da je prilagodba gora). R će nam izbaciti dvije vrste devijance (1) `null deviance` – koja opisuje model koji ima samo slobodni član i (2) `residual deviance` koja uključuje sve prediktorske varijable. Koristeći te dvije veličine, moguće je i izračunati  $R^2$  danog modela kao:

$$R^2 = 1 - \frac{D_{mdl}}{D_0}.$$

Izračunati  $R^2$  može se koristiti kao mjera koja govori koliko je procijenjeni model blizu/daleko od null modela (0-1), tj. kolika je njegova prediktivna moć.

Za prvi model ćemo koristiti sve dostupne dostupne varijable.

```

logreg.mdl.full = glm(won ~ age + ht + seed + hand + ace + df + svpt + FirstIn +
FirstWon + SecondWon + SvGms + bpSaved + bpFaced + rank + rank_points + ohb ,
data = player, family = binomial())
summary(logreg.mdl.full)

```

```

##
## Call:
## glm(formula = won ~ age + ht + seed + hand + ace + df + svpt +
##     FirstIn + FirstWon + SecondWon + SvGms + bpSaved + bpFaced +
##     rank + rank_points + ohb, family = binomial(), data = player)
##
## Deviance Residuals:

```

```

##      Min       1Q    Median       3Q      Max
## -5.5756 -0.6567   0.1830   0.6639   3.9445
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 7.413e+00 2.095e-01 35.390 < 2e-16 ***
## age         -3.970e-02 1.764e-03 -22.503 < 2e-16 ***
## ht          -3.002e-02 1.081e-03 -27.783 < 2e-16 ***
## seed        -1.545e-02 5.860e-04 -26.368 < 2e-16 ***
## handR       9.810e-02 1.850e-02  5.304 1.13e-07 ***
## handU       2.983e-03 5.104e-01  0.006  0.995
## ace         -6.339e-02 1.839e-03 -34.469 < 2e-16 ***
## df          -4.379e-03 3.336e-03 -1.313  0.189
## svpt        -9.457e-02 2.277e-03 -41.527 < 2e-16 ***
## FirstIn     -2.994e-03 2.043e-03 -1.466  0.143
## FirstWon    1.392e-01 3.990e-03 34.889 < 2e-16 ***
## SecondWon   1.434e-01 4.238e-03 33.833 < 2e-16 ***
## SvGms       1.777e-01 8.283e-03 21.453 < 2e-16 ***
## bpSaved     7.533e-01 1.168e-02 64.472 < 2e-16 ***
## bpFaced     -6.932e-01 1.116e-02 -62.128 < 2e-16 ***
## rank        -1.111e-03 8.851e-05 -12.551 < 2e-16 ***
## rank_points 6.863e-05 6.623e-06 10.362 < 2e-16 ***
## ohbTRUE     -1.169e-01 1.551e-02 -7.535 4.88e-14 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 230615  on 166373  degrees of freedom
## Residual deviance: 145677  on 166356  degrees of freedom
## AIC: 145713
##
## Number of Fisher Scoring iterations: 5

```

```
Rsq = 1 - logreg.mdl.full$deviance / logreg.mdl.full>null.deviance
Rsq
```

```
## [1] 0.3683098
```

Bolju informaciju moguće je dobiti iz tzv. matrice zabune (engl. *confusion matrix*), koja je zapravo kontingencijska matrica oznaka iz podataka i modela. Matrica će biti oblika:

	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	$TN$	$FP$
$Y = 1$	$FN$	$TP$

Mjere koje mogu biti od interesa su:

- točnost (eng. accuracy):  $\frac{TP + TN}{TP + FP + TN + FN}$
- preciznost (eng. precision):  $\frac{TP}{TP + FP}$  (udio točnih primjera u svim koji su klasificirani kao TRUE)

- odziv (eng. recall):  $\frac{TP}{TP + FN}$  (udio točnih primjera u skupu svih koji su stvarno TRUE)
- specifičnost (eng. specificity):  $\frac{TN}{TN + FP}$  (udio točnih primjera u svim koji su klasificirani kao FALSE)

```
yHat <- logreg.mdl.full$fitted.values > 0.4
tab <- table(player$won, yHat)
```

```
tab
```

```
##      yHat
##      FALSE  TRUE
## 0 58786 23311
## 1 10359 73918
```

```
accuracy = sum(diag(tab)) / sum(tab)
precision = tab[2,2] / sum(tab[,2])
recall = tab[2,2] / sum(tab[2,])
specificity = tab[1,1] / sum(tab[,1])
```

```
accuracy
```

```
## [1] 0.7976246
```

```
precision
```

```
## [1] 0.7602464
```

```
recall
```

```
## [1] 0.8770839
```

```
specificity
```

```
## [1] 0.8501844
```

### Test omjera izglednosti (likelihood ratio test)

Pokazuje se da za dva modela logističke regresije  $M_1$  sa  $N_1$  prediktorskih varijabli i  $M_2$  sa  $N_2$  prediktorskih varijabli statistika  $-2 \ln \frac{L_1}{L_2}$ , gdje su  $L_1$  i  $L_2$  izglednosti za oba modela, ima  $\chi^2$  distribuciju s  $|N_1 - N_2|$  stupnjeva slobode. Tu statistiku možemo iskoristiti za testiranje postoji li značajna razlika u kvaliteti više alternativnih modela. Ovaj test ima sličnu ulogu kao F-test u slučaju linearne regresije.

Na primjer, možemo testirati postoji li razlika između dva modela – originalnog modela i modela bez statistički značajnih regresora. U tom slučaju ćemo prihvatići smanjeni model ukoliko devijanca nije značajno veća.

```
logreg.mdl.reduced = glm(won ~ age + ht + seed + ace + svpt + FirstWon + SecondWon
+ SvGms + bpSaved + bpFaced + rank + rank_points + ohb + hand,
data = player, family = binomial())
summary(logreg.mdl.reduced)
```

```

## 
## Call:
## glm(formula = won ~ age + ht + seed + ace + svpt + FirstWon +
##      SecondWon + SvGms + bpSaved + bpFaced + rank + rank_points +
##      ohb + hand, family = binomial(), data = player)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.5840 -0.6567  0.1832  0.6639  3.9428
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 7.418e+00 2.091e-01 35.469 < 2e-16 ***
## age         -3.969e-02 1.764e-03 -22.499 < 2e-16 ***
## ht          -3.006e-02 1.078e-03 -27.885 < 2e-16 ***
## seed        -1.544e-02 5.859e-04 -26.351 < 2e-16 ***
## ace         -6.288e-02 1.737e-03 -36.206 < 2e-16 ***
## svpt        -9.640e-02 1.991e-03 -48.412 < 2e-16 ***
## FirstWon   1.379e-01 3.775e-03 36.544 < 2e-16 ***
## SecondWon  1.458e-01 3.958e-03 36.849 < 2e-16 ***
## SvGms       1.770e-01 8.270e-03 21.397 < 2e-16 ***
## bpSaved     7.526e-01 1.168e-02 64.462 < 2e-16 ***
## bpFaced    -6.927e-01 1.115e-02 -62.136 < 2e-16 ***
## rank        -1.114e-03 8.849e-05 -12.592 < 2e-16 ***
## rank_points 6.888e-05 6.620e-06 10.405 < 2e-16 ***
## ohbTRUE    -1.170e-01 1.551e-02 -7.548 4.43e-14 ***
## handR      9.953e-02 1.847e-02  5.388 7.12e-08 ***
## handU     -4.629e-05 5.102e-01  0.000      1
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 230615  on 166373  degrees of freedom
## Residual deviance: 145680  on 166358  degrees of freedom
## AIC: 145712
##
## Number of Fisher Scoring iterations: 5

anova(logreg.mdl.full, logreg.mdl.reduced, test = "LRT")

## Analysis of Deviance Table
##
## Model 1: won ~ age + ht + seed + hand + ace + df + svpt + FirstIn + FirstWon +
##           SecondWon + SvGms + bpSaved + bpFaced + rank + rank_points +
##           ohb
## Model 2: won ~ age + ht + seed + ace + svpt + FirstWon + SecondWon + SvGms +
##           bpSaved + bpFaced + rank + rank_points + ohb + hand
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     166356    145677
## 2     166358    145680 -2    -2.802   0.2464

```

```
Rsq.reduced = 1 - logreg.mdl.reduced$deviance / logreg.mdl.reduced>null.deviance
```

S obzirom na rezultate testa, možemo prihvati model bez varijable "df" zbog toga što devijanca nije značajno veća. Postupak pokušaja redukcije modela ponavljamo s izbacivanjem varijable "hand" te testiranjem razlike devijance.

```
logreg.mdl.reduced2 = glm(won ~ age + ht + seed + ace + svpt + FirstWon + SecondWon + SvGms + bpSaved + bpFaced + rank + rank_points + ohb, data = player, family = binomial())
summary(logreg.mdl.reduced2)
```

```
##  
## Call:  
## glm(formula = won ~ age + ht + seed + ace + svpt + FirstWon +  
##       SecondWon + SvGms + bpSaved + bpFaced + rank + rank_points +  
##       ohb, family = binomial(), data = player)  
##  
## Deviance Residuals:  
##      Min        1Q    Median        3Q       Max  
## -5.5792  -0.6569   0.1832   0.6639   3.9456  
##  
## Coefficients:  
##                 Estimate Std. Error z value Pr(>|z|)  
## (Intercept) 7.513e+00 2.083e-01 36.062 < 2e-16 ***  
## age         -4.019e-02 1.761e-03 -22.821 < 2e-16 ***  
## ht          -3.003e-02 1.078e-03 -27.870 < 2e-16 ***  
## seed        -1.550e-02 5.859e-04 -26.464 < 2e-16 ***  
## ace         -6.284e-02 1.737e-03 -36.183 < 2e-16 ***  
## svpt        -9.637e-02 1.991e-03 -48.399 < 2e-16 ***  
## FirstWon    1.379e-01 3.775e-03 36.544 < 2e-16 ***  
## SecondWon   1.459e-01 3.958e-03 36.863 < 2e-16 ***  
## SvGms       1.766e-01 8.271e-03 21.356 < 2e-16 ***  
## bpSaved     7.516e-01 1.167e-02 64.389 < 2e-16 ***  
## bpFaced     -6.920e-01 1.115e-02 -62.083 < 2e-16 ***  
## rank        -1.118e-03 8.838e-05 -12.654 < 2e-16 ***  
## rank_points 6.904e-05 6.630e-06 10.413 < 2e-16 ***  
## ohbTRUE     -1.187e-01 1.550e-02 -7.657 1.9e-14 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## (Dispersion parameter for binomial family taken to be 1)  
##  
## Null deviance: 230615  on 166373  degrees of freedom  
## Residual deviance: 145709  on 166360  degrees of freedom  
## AIC: 145737  
##  
## Number of Fisher Scoring iterations: 5  
  
anova(logreg.mdl.reduced, logreg.mdl.reduced2, test = "LRT")
```

```
## Analysis of Deviance Table  
##
```

```

## Model 1: won ~ age + ht + seed + ace + svpt + FirstWon + SecondWon + SvGms +
##      bpSaved + bpFaced + rank + rank_points + ohb + hand
## Model 2: won ~ age + ht + seed + ace + svpt + FirstWon + SecondWon + SvGms +
##      bpSaved + bpFaced + rank + rank_points + ohb
##      Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1    166358     145680
## 2    166360     145709 -2    -29.04 4.944e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Kako je devijanca daljnje reducirano modela značajno veća od prijašnjeg, ne možemo ga prihvati.

### Analiza konačnog modela

```

yHat <- logreg.mdl.reduced$fitted.values > 0.4
tab <- table(player$won, yHat)

```

```
tab
```

```

##      yHat
##      FALSE  TRUE
## 0 58770 23327
## 1 10340 73937

```

```

accuracy = sum(diag(tab)) / sum(tab)
precision = tab[2,2] / sum(tab[,2])
recall = tab[2,2] / sum(tab[2,])
specificity = tab[1,1] / sum(tab[,1])

```

```
accuracy
```

```
## [1] 0.7976427
```

```
precision
```

```
## [1] 0.7601682
```

```
recall
```

```
## [1] 0.8773093
```

```
specificity
```

```
## [1] 0.8503834
```

```
Rsq.reduced
```

```
## [1] 0.3682977
```

## Zaključak

Iz provedenih testova te dobivenih metrika, možemo tvrditi da s relativno visokom točnosti od 79.8% možemo predvidjeti pobjednika meča iz danih varijabli.

## Problem 4

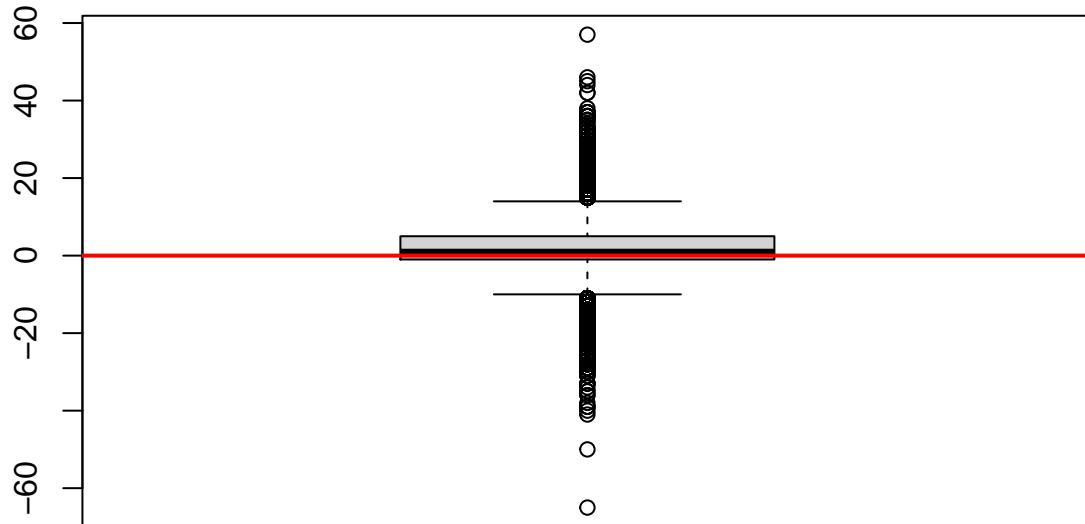
### Utječe li broj aseva na ishod pobjednika?

Zato što u jednom meču podatci broja aseva pobjednika i asevi broja gubitnika ovisi o igračima koji igraju, ne mogu promatrati distribucije odvojeno, stoga zaključujem da trebam provjeriti je li broj aseva koji je ostvario pobjednik statistički značajno veći da bi se zaključila povezanost broja aseva i ishoda meča.

### Pogled na podatke

```
aseviUporeni <- c(tennisMatches$w_ace - tennisMatches$l_ace)

boxplot(aseviUporeni)
abline(h=0, lwd=2, col="red")
```



Iz boxplota vidim da je median iznad 0 i očekujem da broj ostvarenih aseva će utjecati na ishod meča, ali podatci imaju stršećih vrijednosti.

## Korišteni test

Za ovaj problem ću koristiti t test s uparenim vrijednostima. Neka je  $w_i$  realizacija broja aseva pobjednika i  $l_i$  realizacija broja aseva gubitnika, tada definiram realizaciju razlike:  $d_i = w_i - l_i$  za svaki meč u skupu podataka. Ovime dobivam populaciju razlike  $D$  nad kojim ću provesti test.

## Pretpostavke t testa

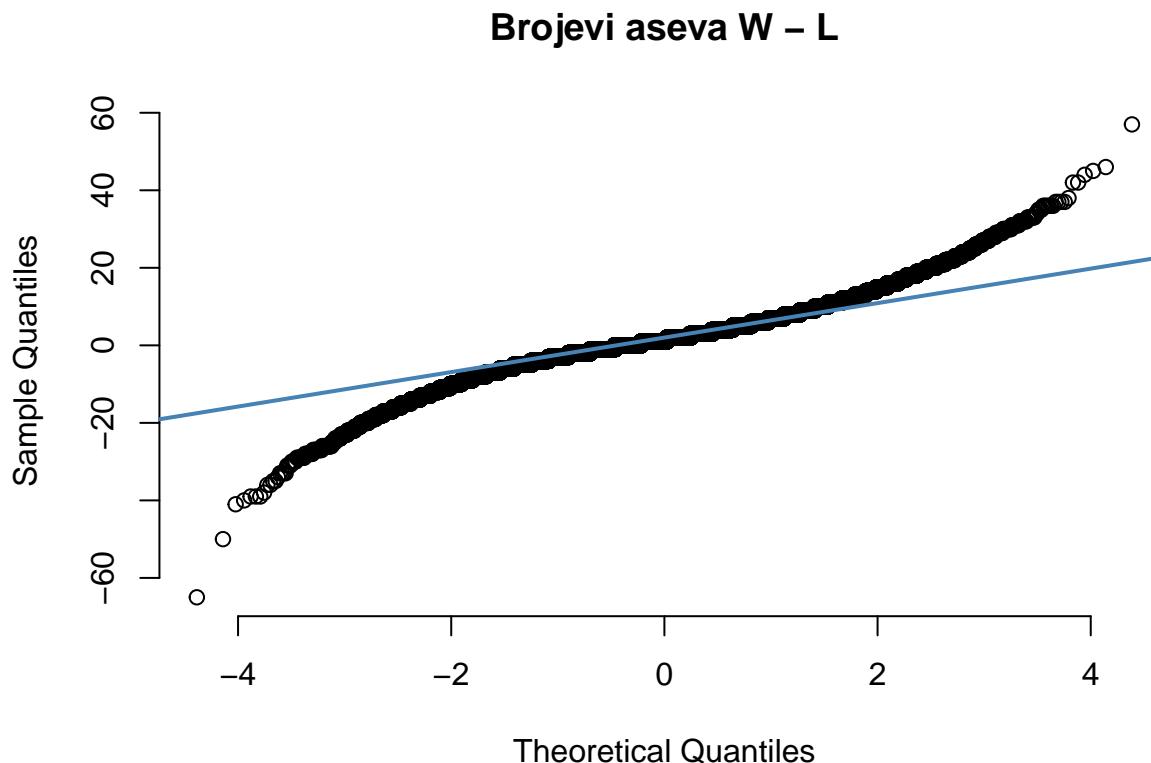
1. Ako znamo da uzorak dolazi iz normalne distribucije t-test je egzaktan
2. Uzorci moraju biti nezavisni

## Provjera normalnosti podataka

Provjeravam jesu li ispunjeni uvjeti za provedbu t testa

Provjeravam normalnost uparenih podataka broja aseva igrača koji je pobijedio (W) umanjene za broj aseva koje je ostvario igrač koji je izgubio (L).

```
qqnorm(aseviUpareni, pch = 1, frame = FALSE, main = "Brojevi aseva W - L")
qqline(aseviUpareni, col = "steelblue", lwd = 2)
```



Iz QQ dijagrama vidim da podatci imaju odstupanja od normalne distribucije.

## Lillieforseov test nad podatcima

Provodim test o normalnosti distribucije

Hipoteze testa su:

$H_0$  : Podatci prate normalnu distribuciju

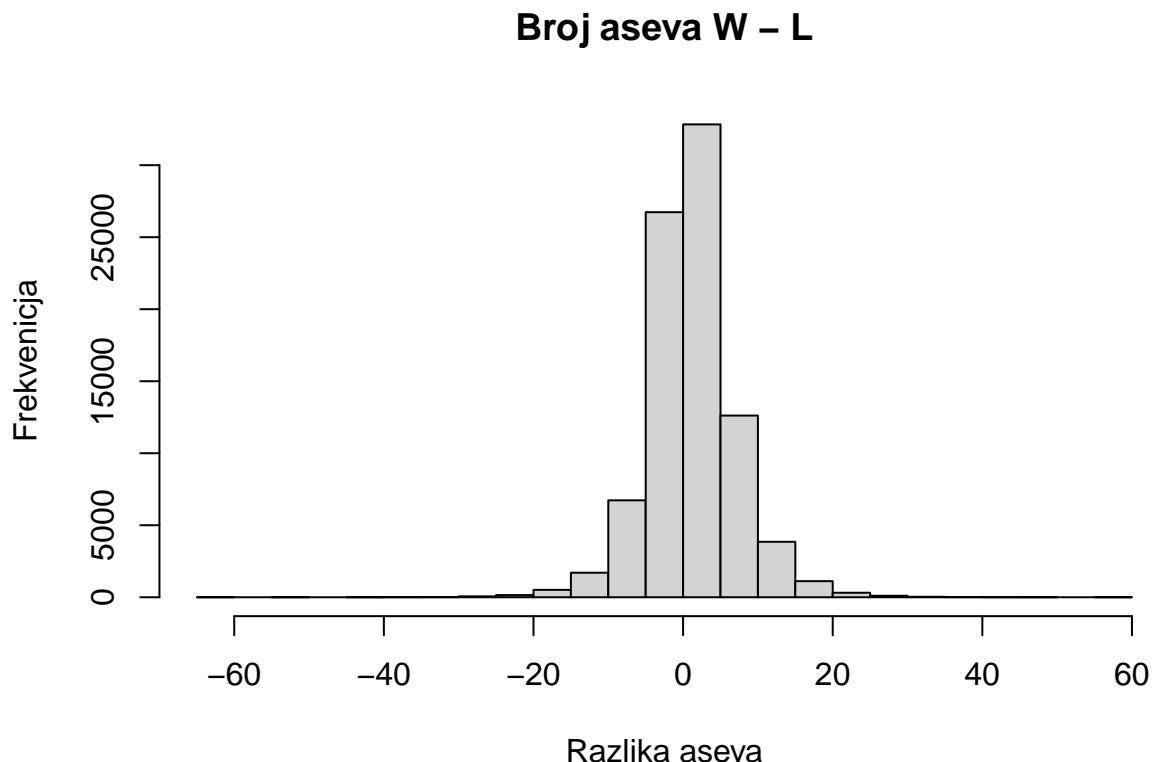
$H_1$  : Podatci ne prate normalnu distribuciju

```
lillie.test(aseviUpareni)
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data: aseviUpareni  
## D = 0.084692, p-value < 2.2e-16
```

Zbog male p-vrijednosti zaključujem da mogu odbaciti  $H_0$  u korist  $H_1$ , a to je da podatci ne prate normalnu distribuciju.

```
hist(aseviUpareni, main = "Broj aseva W - L", xlab = "Razlika aseva", ylab = "Frekvenicja")
```



Podatci prate zvonoliku krivulju, a t-test je robustan na ne normalnost, tj. distribucija uzorka ne mora biti normalna da bi t test dao ispravne rezultate (test je aproksimativan), ali po obliku mora pratiti normalnu krivulju.

## Uvjjeti testa

Zato što distribucija  $D$  populacije nije normalna, ali je zvonolika, zaključujem da je t-test aproksimativan, zbog toga provodim i Jackknife, da utvrdim 99% interval povjerenja za srednju vrijednost.

## Provodim jednostrani T test

Hipoteze:

$$H_0 : \mu_w - \mu_l = 0$$

$$H_1 : \mu_w - \mu_l > 0$$

```
# Provjeda t testa
t.test(aseviUpareni, alt = "greater")
```

```
##
##  One Sample t-test
##
## data: aseviUpareni
## t = 84.944, df = 86808, p-value < 2.2e-16
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##   1.653928      Inf
## sample estimates:
## mean of x
## 1.686588
```

## Zaključak t-testa

Zbog male p-vrijednosti mogu odbaciti hipotezu  $H_0$  i zaključiti da broj ostvarenih aseva igrača koji je pobijedio je statistički značajno veće na dostupnom uzorku nego od broja aseva koje je ostvario igrač koji je izgubio.

## Jackknife

```
aseviUpareni <- na.omit(aseviUpareni)

ps <- numeric(length(aseviUpareni))
n <- length(aseviUpareni)
m <- mean(aseviUpareni)
s <- sum(aseviUpareni)
for(i in 1:length(aseviUpareni)){
  ps[i] = n * m - s + aseviUpareni[i]
}

se <- sd(ps)/sqrt(n)
j <- mean(ps)
cat("Interval povjerenja od 99% za aritmetičku sredinu je [", (j - qt(0.995, n-1)*se), " , ", (j + qt(0.995, n-1)*se), "]")

## Interval povjerenja od 99% za aritmetičku sredinu je [ 1.635443 , 1.737733 ]
```

## Zaključak Jackknifea

Zato što srednja vrijednost  $\mu = 0$  nije unutar 99% intervala procjene uzorka, zaključujem da mogu odbaciti hipotezu  $H_0$ .

## Zaključak

Iz provedenih testova zaključujem da broj ostvarenih aseva utječe na pobjedu u meču, zato što je statistički značajno više igrača koji su pobijedili ostvarilo više aseva od onih koji su izgubili.

## Problem 5

Utječe li postotak osvojenosti poena prvim servisom na ishod pobjednika?

### Opis i postupak

Mnogi tenisači stavlju veliki fokus na povećanje osvajanja poena prvim servisom pa se postavlja pitanje utječe li postotak osvojenosti poena prvim servisom na ishod pobjenika.

Potrebno je izdvojiti postotak osvojenosti poena prvim servisom pobjenika mečeva i postotak osvojenosti poena prvim servisom gubitnika mečeva te ispitati njihov odnos.

Postotak osvojenosti poena prvim servisom računamo kao  $p_1 * q_1$  gdje je  $p_1$  vjerojatnost uspješnosti prvog servisa, a  $q_1$  vjerojatnost osvajanja poena uz uvjet da je prvi servis uspješan.

### Dohvat podataka

```
#vjeratnost uspješnosti prvog servisa
p1stWon.p1.w = tennisMatches$w_1stIn / tennisMatches$w_svpt
p1stWon.p1.l = tennisMatches$l_1stIn / tennisMatches$l_svpt

#vjeratnost osvajanja poena uz uvjet da je prvi servis uspješan
p1stWon.q1.w = tennisMatches$w_1stWon / tennisMatches$w_1stIn
p1stWon.q1.l = tennisMatches$l_1stWon / tennisMatches$l_1stIn

#vjeratnost osvajanja poena prvim servisom
p1stWon.w = p1stWon.p1.w * p1stWon.q1.w
p1stWon.l = p1stWon.p1.l * p1stWon.q1.l

#punjenje nedostajećih vrijednosti očekivanjem
meanW = mean(p1stWon.w, na.rm=TRUE)
meanL = mean(p1stWon.l, na.rm=TRUE)

p1stWon.w[is.na(p1stWon.w)] = meanW
p1stWon.l[is.na(p1stWon.l)] = meanL

cat("Postotak osvojenosti poena prvim servisom pobjednika:\n")
```

```
## Postotak osvojenosti poena prvim servisom pobjednika:
```

```

summary(p1stWon.w)

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.0000  0.4175  0.4639  0.4639  0.5067  1.0000

cat("Postotak osvojenosti poena prvim servisom gubitnika:\n")

```

## Postotak osvojenosti poena prvim servisom gubitnika:

```

summary(p1stWon.1)

##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 0.0000  0.3434  0.3853  0.3853  0.4286  0.8000

```

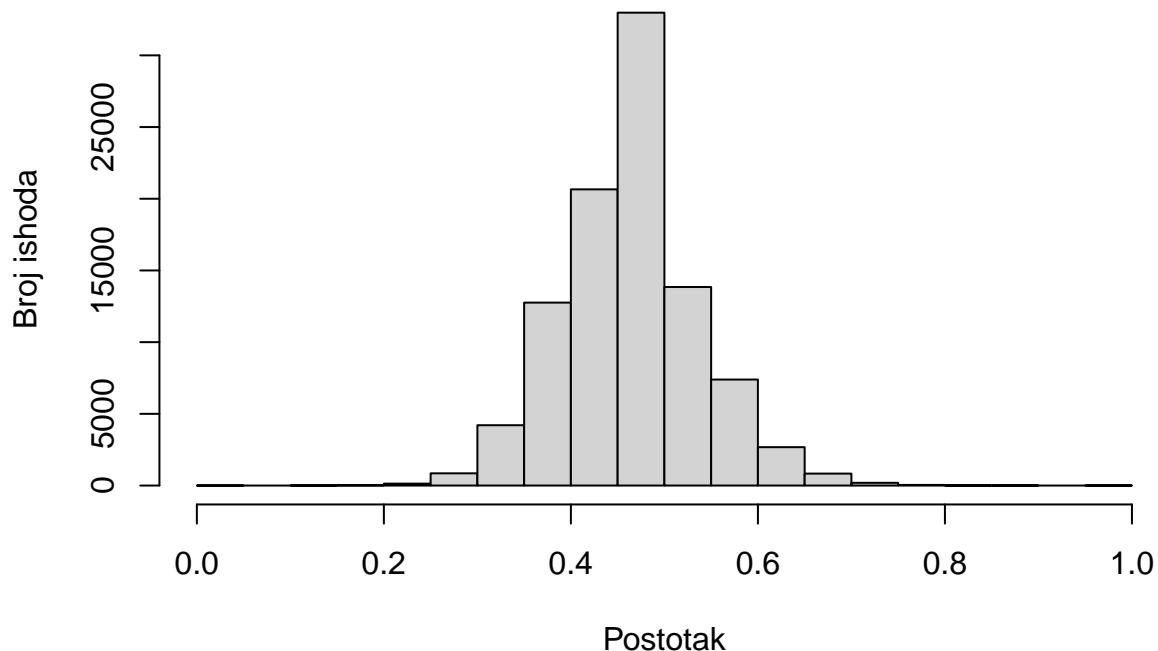
Vizualizacija

```

hist(p1stWon.w,
      xlab="Postotak",
      ylab="Broj ishoda",
      main="Postotak osvojenosti poena prvim servisom pobjednika",
      xlim=c(0,1))

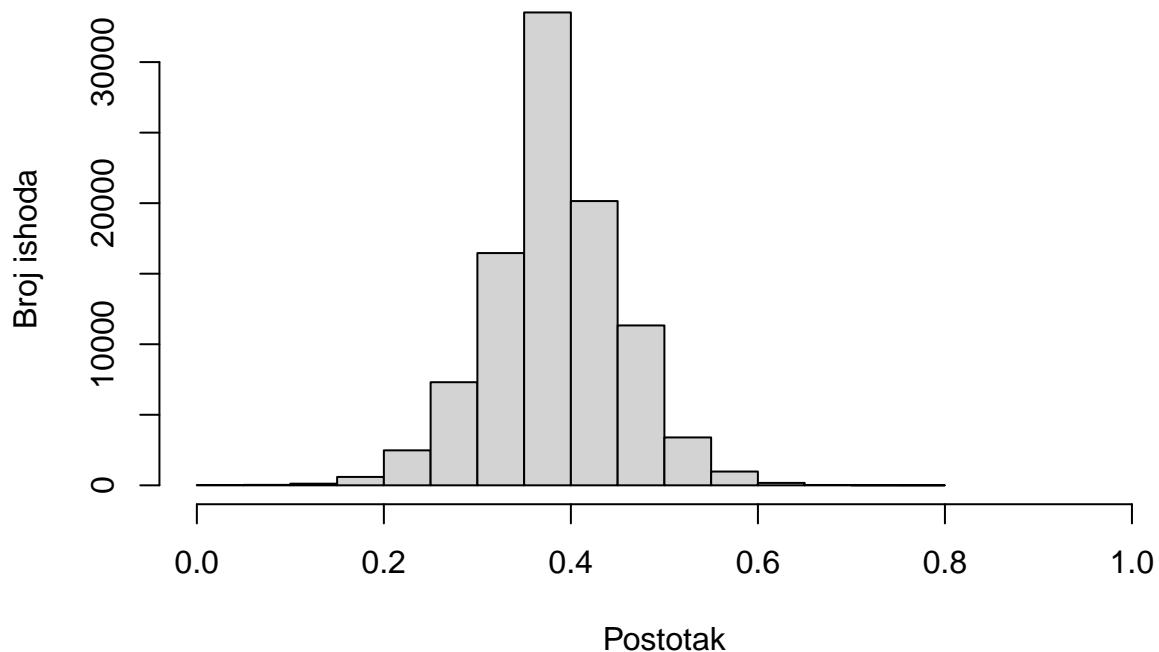
```

### Postotak osvojenosti poena prvim servisom pobjednika

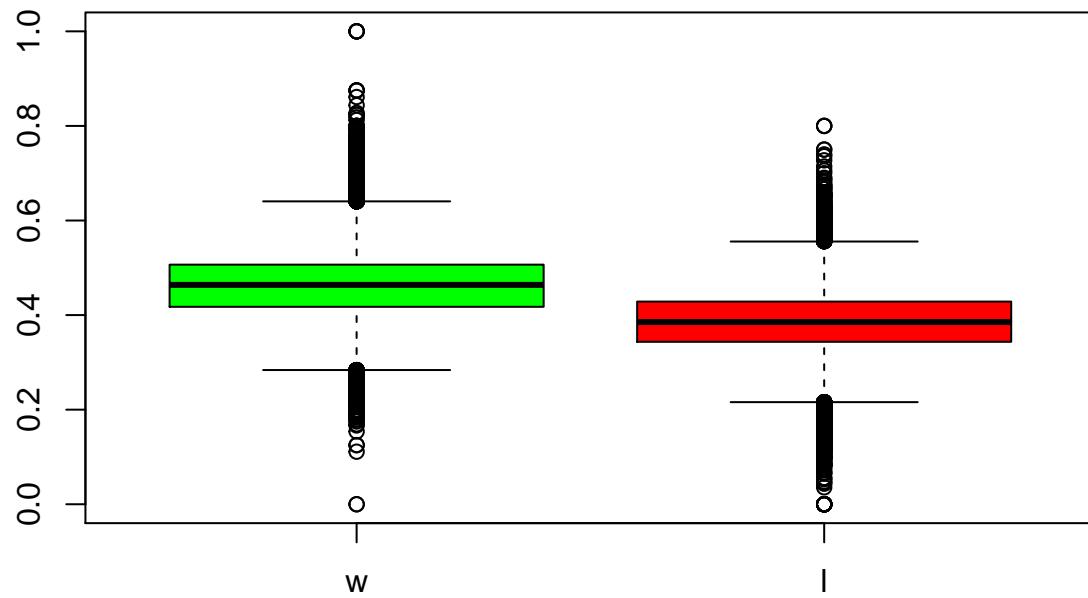


```
hist(p1stWon.l,
     xlab="Postotak",
     ylab="Broj ishoda",
     main="Postotak osvojenosti poena prvim servisom gubitnika",
     xlim=c(0,1))
```

## Postotak osvojenosti poena prvim servisom gubitnika



```
p1Won.frame = data.frame(w = p1stWon.w,l = p1stWon.l)
boxplot(p1Won.frame, col =c("green", "red"))
```

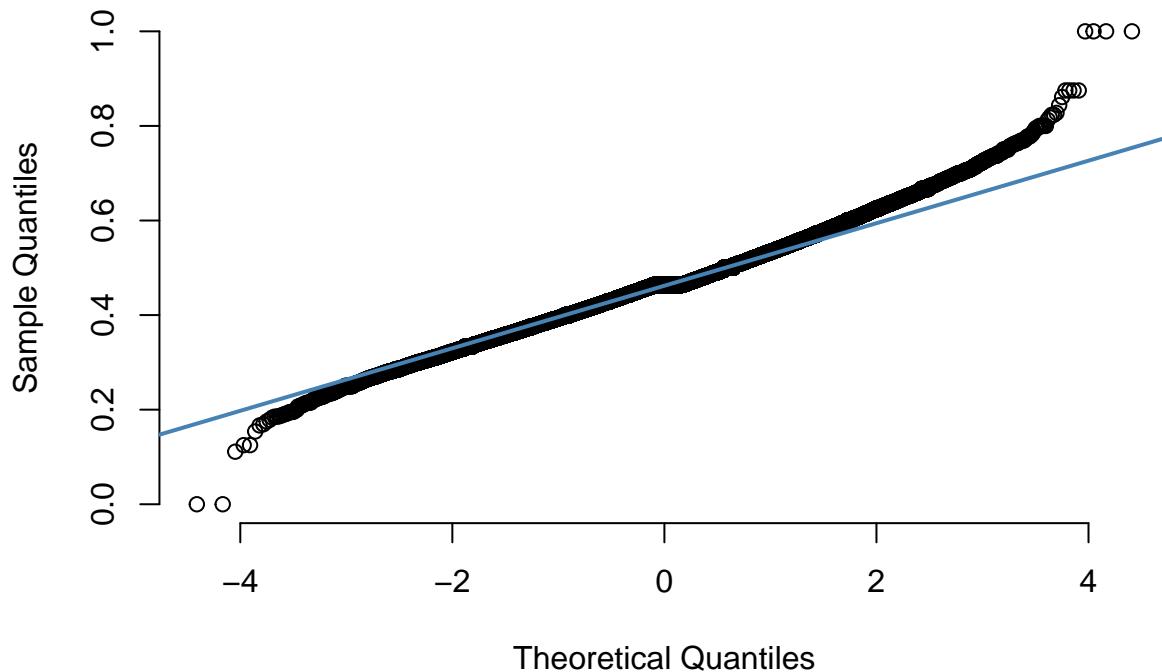


Iz prikazanih dijagrama se može naslutiti da je postotak osvojenosti poena prvim servisom veći kod pobjednika u odnosu na gubitnike. Obje populacije imaju zvonolik oblik, no testirajmo normalnost i qq-plotom.

### Provjera normalnosti

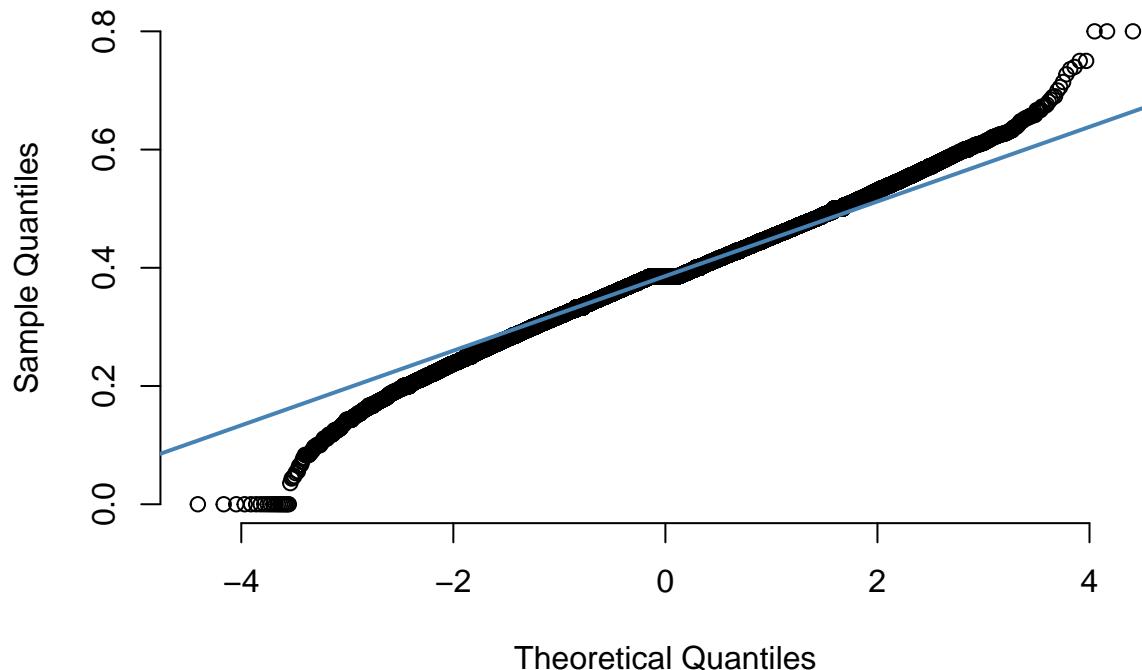
```
qqnorm(p1stWon.w,
  pch = 1,
  frame = FALSE,
  main = "Postotak osvojenosti poena prvim servisom pobjednika"
)
qqline(p1stWon.w, col = "steelblue", lwd = 2)
```

## Postotak osvojenosti poena prvim servisom pobjednika



```
qqnorm(p1stWon.1,  
       pch = 1,  
       frame = FALSE,  
       main = "Postotak osvojenosti poena prvim servisom gubitnika"  
)  
qqline(p1stWon.1, col = "steelblue", lwd = 2)
```

## Postotak osvojenosti poena prvim servisom gubitnika



```
lillie.test(p1stWon.w)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
##  data: p1stWon.w  
##  D = 0.066793, p-value < 2.2e-16
```

```
lillie.test(p1stWon.l)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
##  data: p1stWon.l  
##  D = 0.057009, p-value < 2.2e-16
```

Na temelju Lillieforsovog testa možemo zaključiti da podatci o postotku osvojenosti poena prvim servisom pobjednika i gubitnika nisu iz normalne distribucije, no premda oni prate zvonoliku krivulju te nemaju "duge repove", odlučujemo se za korištenje t-testa premda je on robustan na nenormalnost ukoliko distribucija podataka prati zvonoliku krivulju.

## Testiranje jednakosti varijanci

Kako bi se moglo odlučiti koji će se t-test koristiti potrebno je provesti test o jednakosti varijanci.

```
varPobjednik <- var(p1stWon.w)
varGubitnik <- var(p1stWon.l)

cat("Varijanca postotka dobivenosti poena prvim servisom pobjednika: ", varPobjednik, "\n")

## Varijanca postotka dobivenosti poena prvim servisom gubitnika: 0.005305954

cat("Varijanca postotka dobivenosti poena prvim servisom gubitnika: ", varGubitnik, "\n")

## Varijanca postotka dobivenosti poena prvim servisom gubitnika: 0.004930651
```

Ako imamo dva nezavisna slučajna uzorka  $X_1^1, X_1^2, \dots, X_1^{n_1}$  i  $X_2^1, X_2^2, \dots, X_2^{n_2}$  koji dolaze iz normalnih distribucija s varijancama  $\sigma_1^2$  i  $\sigma_2^2$ , tada slučajna varijabla

$$F = \frac{S_{X_1}^2 / \sigma_1^2}{S_{X_2}^2 / \sigma_2^2}$$

ima Fisherovu distribuciju s  $(n_1 - 1, n_2 - 1)$  stupnjeva slobode, pri čemu vrijedi:

$$S_{X_1}^2 = \frac{1}{n_1 - 1} \sum_{i=1}^{n_1} (X_1^i - \bar{X}_1)^2, \quad S_{X_2}^2 = \frac{1}{n_2 - 1} \sum_{i=1}^{n_2} (X_2^i - \bar{X}_2)^2.$$

Hipoteze testa jednakosti varijanci glase:

$$\begin{aligned} H_0 : \sigma_1^2 &= \sigma_2^2 \\ H_1 : \sigma_1^2 &\neq \sigma_2^2 \end{aligned}$$

```
var.test(p1stWon.w, p1stWon.l)
```

```
##
##  F test to compare two variances
##
##  data: p1stWon.w and p1stWon.l
##  F = 1.0761, num df = 96601, denom df = 96601, p-value < 2.2e-16
##  alternative hypothesis: true ratio of variances is not equal to 1
##  95 percent confidence interval:
##    1.062629 1.089774
##  sample estimates:
##  ratio of variances
##                1.076116
```

Zaključak:

Na razini značajnosti od 1% odbacujemo nullu hipotezu te zaključujemo da varijance postotka dobivenosti poena prvim servisom pobjednika i gubitnika nisu jednake.

## Odabir testa

Iz provednih testova odlučujemo se za testiranje hipoteze uporabom t-testa uz pretpostavku nejednakosti varijanci. Taj test ima sljedeću testnu statistiku:

$$T = \frac{\mu_1 - \mu_2}{\sqrt{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)}}$$

sa

$$v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}}$$

,zaokruženim na manji cijeli broj, stupnja slobode.

## Testiranje hipoteze

Hipoteze:

$H_0$  : postotci dobivenosti poena prvim servisom pobjednika i gubitnika meča su jednaki

$H_1$  : postotak dobivenosti poena prvim servisom je veći kod pobjednika u odnosu na gubitnike meča  
odnosno

$H_0 : \mu_w - \mu_l = 0$

$H_1 : \mu_w - \mu_l > 0$

```
t.test(p1stWon.w, p1stWon.l,
       alternative = "greater",
       var.equal = FALSE
)

## Welch Two Sample t-test
## data: p1stWon.w and p1stWon.l
## t = 241.38, df = 192943, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.07803847      Inf
## sample estimates:
## mean of x mean of y
## 0.4638649 0.3852910
```

Zaključak: Na razini signifikantnosti od 1% odbacujemo hipotezu  $H_0$  u koristi hipoteze  $H_1$  koja tvrdi da je postotak dobivenosti poena prvim servisom pobjednika meča veći od postotka dobivenosti poena prvim servisom gubitnika meča, premda je dobivena p-vrijednost manja od 2.2e-16.

## Provjeda neparametarskog testa

Premda je Lillieforsov test normalnosti dao uvid u odstupanja distribucije podataka o postotcima dobivenosti poena prvim servisom, odlučujem se za provjedu neparametarskog testa kako mi proveli testiranje nad medijanima dviju populacija.

Koristit ćemo Wilcoxon Rank-Sum test. Hipoteze testa glase:

$H_0$  : medijani postotka doivenosti poena prvim servisom pobjednika i gubitnika meča su jednaki

$H_1$  : medijan postotka doivenosti poena prvim servisom je veći kod pobjednika u odnosu na gubitnike meča

$$H_0 : \tilde{x_w} = \tilde{x_l}$$

$$H_1 : \tilde{x_w} > \tilde{x_l}$$

```
wilcox.test(p1stWon.w, p1stWon.l, alternative = "greater")
```

```
##  
##  Wilcoxon rank sum test with continuity correction  
##  
## data: p1stWon.w and p1stWon.l  
## W = 7356769414, p-value < 2.2e-16  
## alternative hypothesis: true location shift is greater than 0
```

## Zaključak

Uz dobivenu p-vrijednost od 2.2e-16. na razini značajnosti od 1% odbacujemo nultu hipotezu, ne možemo tvrditi da su medijani postotaka doivenosti poena prvim servisom jednaki kod pobjednika i gubitnika meča. Imamo razloga vjerovati da pobjednici meča imaju veći medijan postotka doivenosti poena prvim servisom.

Premda su podatci pratili zvonoliku krivulju te su imali kratke repove, mogli smo očekivati da će parametarski i neparametarski test dati slične rezultate, no Wilcoxon test ima veću snagu u odnosu na parametarski t-test u situaciji kada ne možemo tvrditi da podatci dolaze iz normalne distribucije.

## Problem 6

Jesu li ljevaci nezgodniji protivnici dešnjacima koji igraju jednoručni backhand?

### Opis i postupak

U tenisu se ljevac smatraju nezgodnim protivnicima dešnjaka zbog načina na koji dešnjaci moraju igrati protiv njih. Također, jednoručni backhand se smatra superiornom tehnikom u odnosu na klasični backhand ako ga je igrač usavršio. Postavlja se pitanje jesu li ljevaci nezgodniji protivnici i onim dešnjacima koji igraju jednoručni backhand. Potrebno je usporediti odnos broja pobjeda te gubitaka igrača koji je dešnjak i igra jednoručni backhand kada igra protiv ljevaka te kada igra protiv dešnjaka.

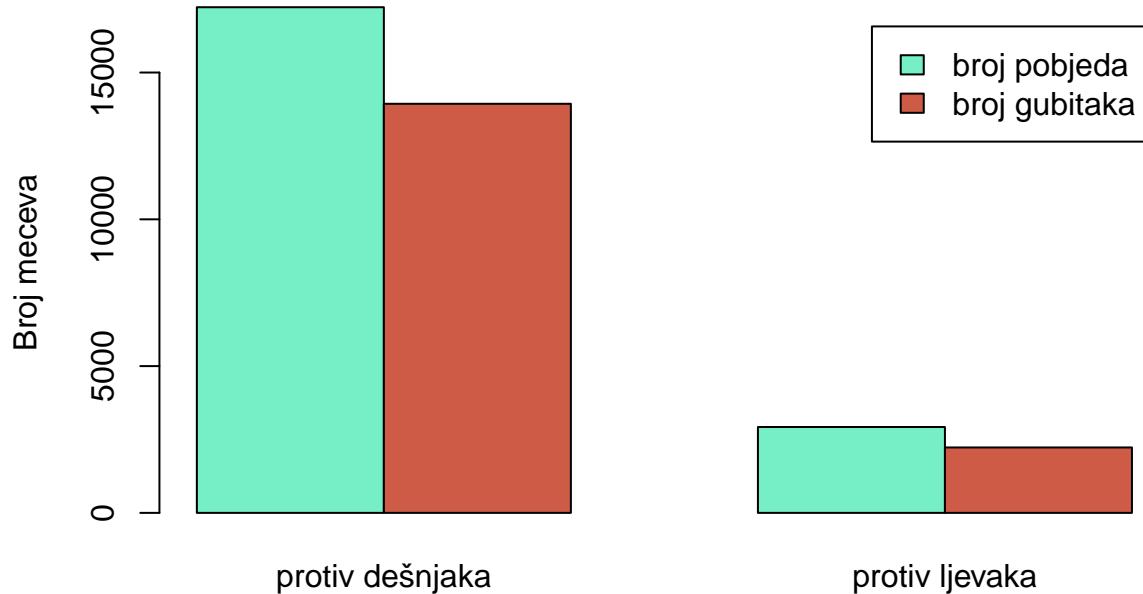
### Korišten test

Hipotezu ćemo testirati Hi-kvadrat testom za jednakost proporcija.

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i}$$

## Izdvajanje podataka i pregled

### Broj pobjeda i gubitaka dešnjaka koji igra jednoruci backhand



```
## [1] "Broj pobjeda protiv dešnjaka: 17225"  
## [1] "Broj gubitaka protiv dešnjaka: 13935"  
## [1] "Udio pobjeda u mečevima protiv dešnjaka (%): 55.2792041078306"  
## [1] "Broj pobjeda protiv ljevaka: 2925"  
## [1] "Broj gubitaka protiv ljevaka: 2228"  
## [1] "Udio pobjeda u mečevima protiv ljevaka (%): 56.7630506501067"
```

Iz prikaza podataka možemo naslutiti da postoji razlika u vjerojatnosti pobjede dešnjaka koji igra jednoruci backhand kada igra protiv dešnjaka te kada igra protiv ljevaka. Moramo provesti testiranje kako bi vidjeli je li opažena razlika statistički značajna.

### Testiranje hipoteze

Provodimo Hi-kvadrat test jednakosti proporcija.

Hipoteze:

$$H_0 : p_r = p_l \quad H_1 : p_r < p_l$$

```

prop.test(x=c(17225, 2925), n=c(17225 + 13935, 2925 + 2228), alternative = "less")

##
## 2-sample test for equality of proportions with continuity correction
##
## data: c(17225, 2925) out of c(17225 + 13935, 2925 + 2228)
## X-squared = 3.882, df = 1, p-value = 0.0244
## alternative hypothesis: less
## 95 percent confidence interval:
## -1.00000000 -0.00246472
## sample estimates:
## prop 1   prop 2
## 0.5527920 0.5676305

```

## Zaključak

Na razini signifikantnosti od 5% odbacujemo nullu hipotezu o jednakosti proporcija u korist alternativne hipoteze koja tvrdi da je udio pobjeda protiv dešnjaka manji od udjela pobjeda protiv ljevaka. To znači da ne možemo tvrditi da su ljevaci nezgodniji protivnici dešnjacima koji igraju jednoručni backhand.

## Problem 7

**Utječe li broj dvostrukih pogrešaka na ishod pobjednika?**

Velik broj dvostrukih pogrešaka u teniskom meču može utjecati na nepotreban gubitak poena. Postavlja se pitanje utječe li broj dvostrukih pogrešaka na ishod pobjednika.

Potrebno je upariti broj dvostrukih pogrešaka pobjednika i gubitnika meča te vidjeti postoji li značajna razlika u broju dvostrukih pogrešaka pobjednika i gubitnika meča.

## Dohvat podataka

```

dfW = tennisMatches$w_df
dfL = tennisMatches$l_df

cat("Podaci o broju dvostrukih pogrešaka pobjenika meča\n")

## Podaci o broju dvostrukih pogrešaka pobjenika meča

summary(dfW)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##      0.000   1.000   2.000   2.745   4.000  26.000    9793

cat("\nPodaci o broju dvostrukih pogrešaka gubitnika meča\n")

##
## Podaci o broju dvostrukih pogrešaka gubitnika meča

```

```

summary(dfL)

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
## 0.000   2.000  3.000  3.502   5.000 26.000    9793

```

### Uklanjanje nedostajećih vrijednosti

Iz podataka uklanjamo one mečeve za koje ne postoje podatci o broju dvostrukih pogrešaka pobjjenika ili gubitnika meča.

```

dfMatches = tennisMatches[names(tennisMatches) %in% c("w_df", "l_df")]

dfMatches = na.omit(dfMatches)

dim(dfMatches)

```

```

## [1] 86809      2

```

```

summary(dfMatches$w_df)

```

```

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.000   1.000  2.000  2.745   4.000 26.000

```

```

summary(dfMatches$l_df)

```

```

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 0.000   2.000  3.000  3.502   5.000 26.000

```

```

dfMatches.upareni = dfMatches$w_df - dfMatches$l_df
summary(dfMatches.upareni)

```

```

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## -20.0000 -3.0000 -1.0000 -0.7567  1.0000 22.0000

```

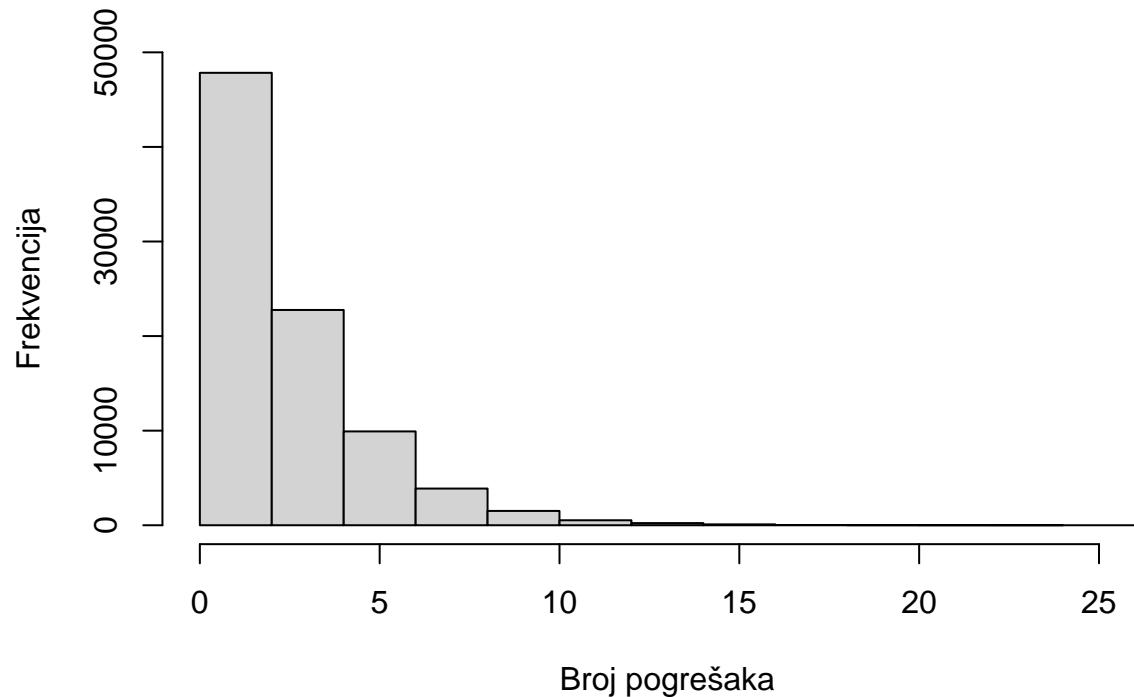
### Vizualizacija podataka

```

hist(dfMatches$w_df,
      xlab="Broj pogrešaka",
      ylab="Frekvencija",
      main="Broj dvostrukih pogrešaka pobjednika",
      ylim = c(0, 50000))

```

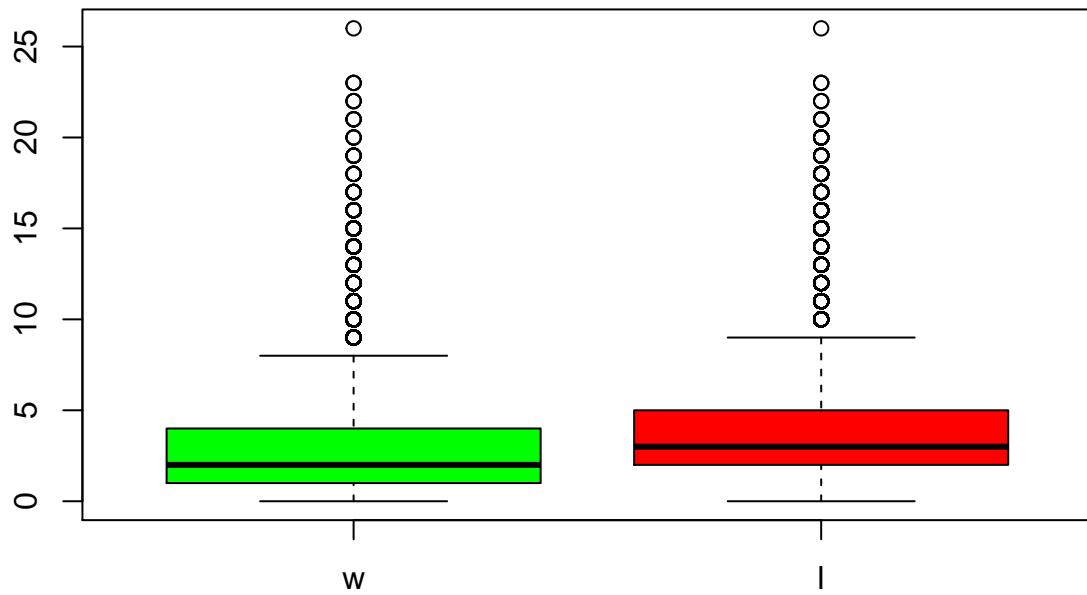
### Broj dvostrukih pogrešaka pobjednika



```
hist(dfMatches$l_df,
      xlab="Broj pogrešaka",
      ylab="Frekvencija",
      main="Broj dvostrukih pogrešaka gubitnika",
      ylim = c(0, 50000))
```

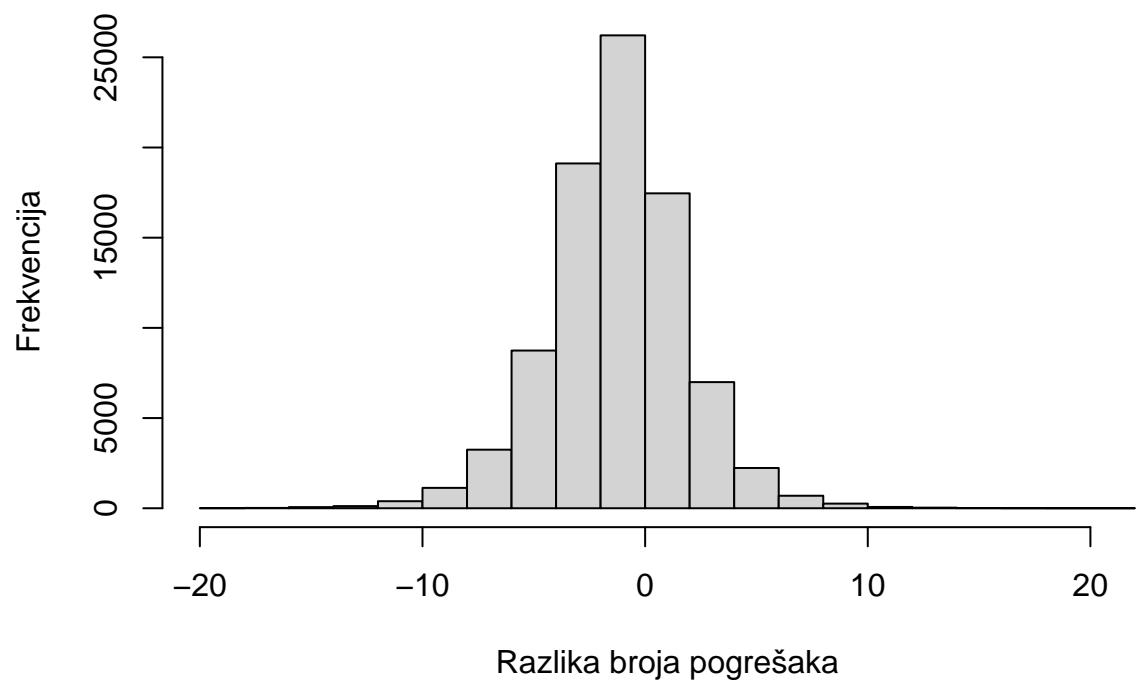


```
dfMatches.frame = data.frame(w = dfMatches$w_df, l = dfMatches$l_df)
boxplot(dfMatches.frame, col = c("green", "red"))
```

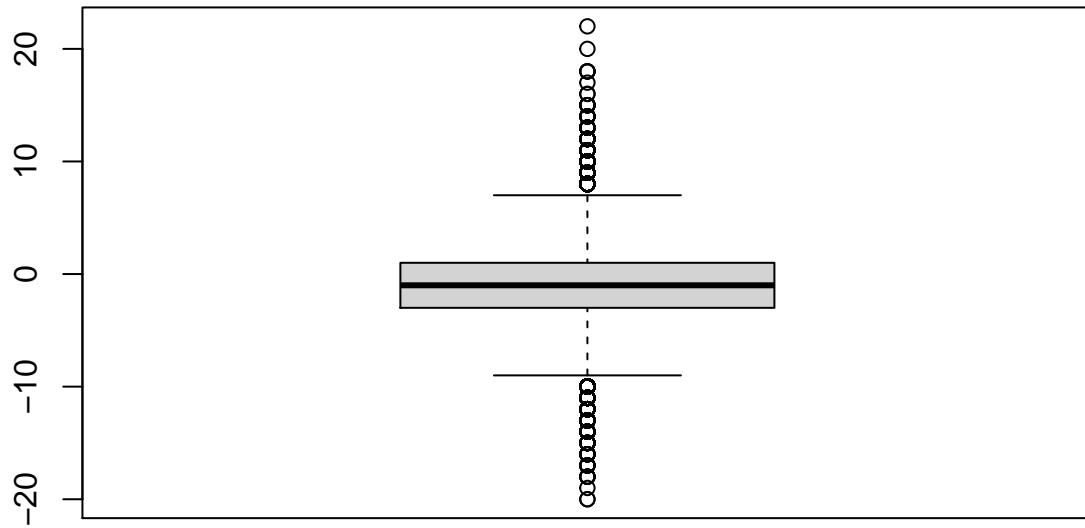


```
hist(dfMatches.upareni,  
      xlab="Razlika broja pogrešaka",  
      ylab="Frekvencija",  
      main="Upareni broj dvostrukih pogrešaka",  
)
```

## Upareni broj dvostrukih pogrešaka



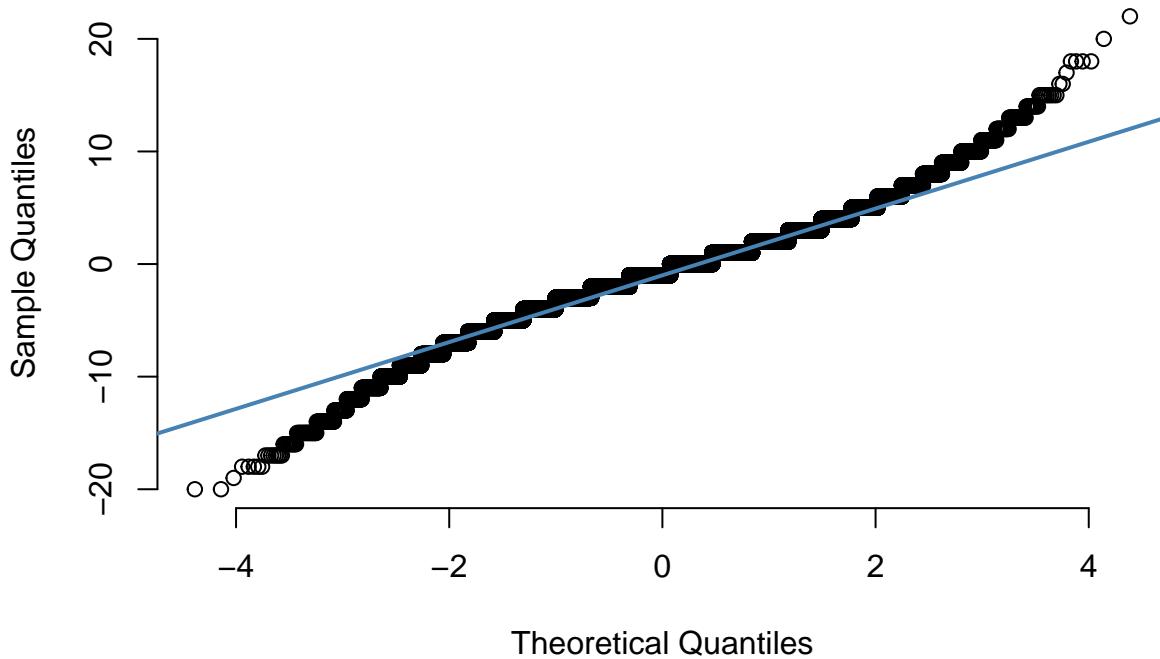
```
boxplot(dfMatches.uparenji)
```



Iz prikazanih dijagrama se može naslutiti kako gubitnici imaju veći broj dvostrukih pogrešaka u usporedbi sa pobjednicima meča. Upareni broj dvostrukih pogrešaka ima zvonolik oblik te njegov histogram nalikuje na normalnu distribuciju, no provjerimo to i qq-plotom.

```
qqnorm(dfMatches.upareni,
  pch = 1,
  frame = FALSE,
  main = "Upareni broj dvostrukih pogrešaka"
)
qqline(dfMatches.upareni, col = "steelblue", lwd = 2)
```

## Upareni broj dvostrukih pogrešaka



### Odabir testa

QQ-plot daje uvid u manja odstupanja uparenog broja dvostrukih pogrešaka od normalne distribucije, no premda distribucija ima zvonolik oblik možemo koristiti t-test premda je on robustan na nenormalnost.

Koristit ćemo t-test čija je testna statistika:

$$T = \frac{\mu - \mu_0}{s/\sqrt{n}}$$

### Hipoteze

$H_0$  : broj dvostrukih pogrešaka pobjednika je jednak broju dvostrukih pogrešaka gubitnika  $H_1$  : broj dvostrukih pogrešaka pobjednika je manji od broja dvostrukih pogrešaka gubitnika

odnosno

$$H_0 : \mu_w - \mu_l = 0$$

$$H_1 : \mu_w - \mu_l < 0$$

premda provodimo test nad uparenim podacima.

## Provjeda testa

```
t.test(dfMatches.upareni, alt = "less")  
  
##  
## One Sample t-test  
##  
## data: dfMatches.upareni  
## t = -73.026, df = 86808, p-value < 2.2e-16  
## alternative hypothesis: true mean is less than 0  
## 95 percent confidence interval:  
##       -Inf -0.7396403  
## sample estimates:  
##   mean of x  
## -0.7566842
```

## Zaključak

Na razini značajnosti od 1% odbacujemo nultu hipotezu u korist alternativne hipoteze koja govori da je broj dvostrukih pogrešaka pobjednika manji od broja dvostukih pogrešaka gubitnika meča. Ne možemo tvrditi da je broj dvostrukih pogrešaka pobjednika i gubitnika meča jednak.