SAP - Projekt

Analiza UFC borbi

Patrik Kukić, Filip Penzar, Željko Antunović, Noa Margeta

2023-01-06

Početna analiza podataka

```
total_fight_data = read.csv("../total_fight_data.csv", sep = ";")
dim(total_fight_data)
```

```
## [1] 6012   41
```

```
fighter_details = read.csv("../fighter_details.csv", sep = ",")
dim(fighter_details)
```

```
## [1] 3596   14
```

```
all <- merge(total_fight_data, fighter_details, by.x = "R_fighter", by.y = "fighter_name",
    all.x = TRUE)
all <- merge(all, fighter_details, by.x = "B_fighter", by.y = "fighter_name", all.x = TRUE,
    suffixes = c(".r", ".b"))
```

```
dim(all)
```

```
## [1] 6012   67
```

Zadatak 1: Možemo li očekivati završetak borbe knockout-om ovisno o razlici u dužini ruku između boraca?

Početni korak u rješavanju ovog zadatka bila je pretvorba težine, visine i dosega oba borca iz imperijalnog sustava u metrički sustav. Ovdje je prikazana jedna pretvorba, na isti način su napravljene i ostalih 5 pretvorbi. Ignorirali smo sve datapoint-ove sa NA vrijednostima.

```
# Pretvaranje in u cm
all$Height_cm.b = sapply(strsplit(as.character(all$Height.b), "'|\""), function(x) {
    30.48 * as.numeric(x[1]) + 2.54 * as.numeric(x[2])
})
```

```
# Micanje redaka koji imaju NA reach
all_without_na_in_reach <- subset(all, !is.na(Reach_cm.b))
all_without_na_in_reach <- subset(all_without_na_in_reach, !is.na(Reach_cm.r))
```

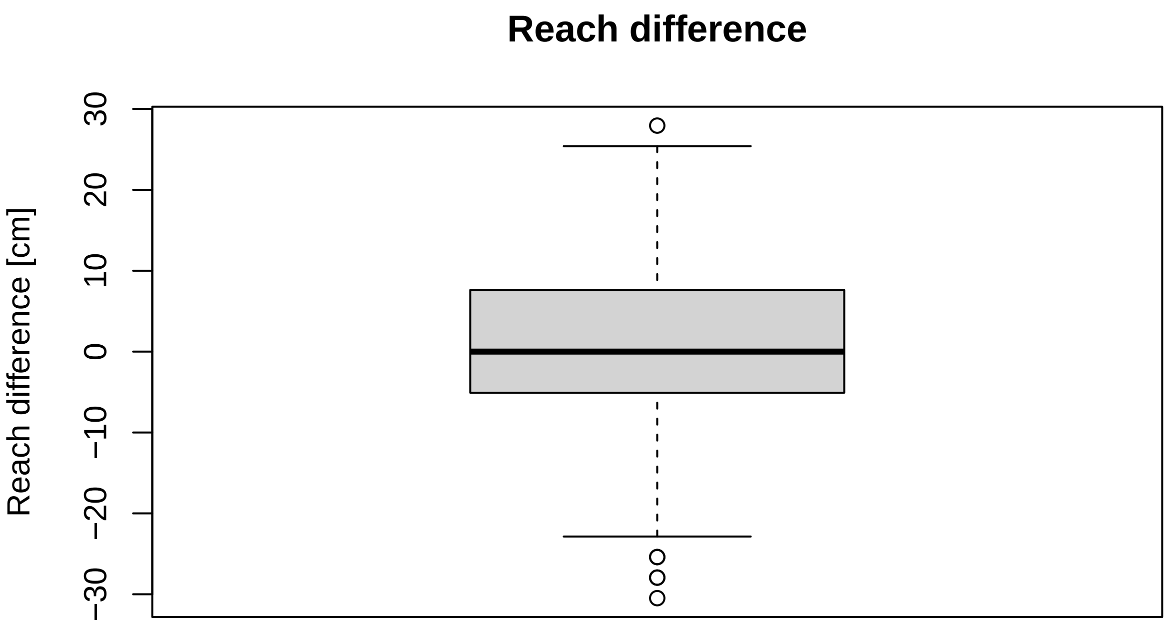
```
# Samo borbe koje su završile knockout-om
all_only_knockouts = subset(all_without_na_in_reach, all_without_na_in_reach$win_by ==
    "KO/TKO")
```

```
# Računanje razlike dosega pobjednika i gubitnika
d = c()
for (i in 1:nrow(all_only_knockouts)) {
```

```r
    row = all_only_knockouts[i, ]
    diff = row$Reach_cm.r - row$Reach_cm.b
    if (row$Winner == row$R_fighter) {
        d = append(d, diff)
    } else {
        d = append(d, -diff)
    }
}

summary(d)
```

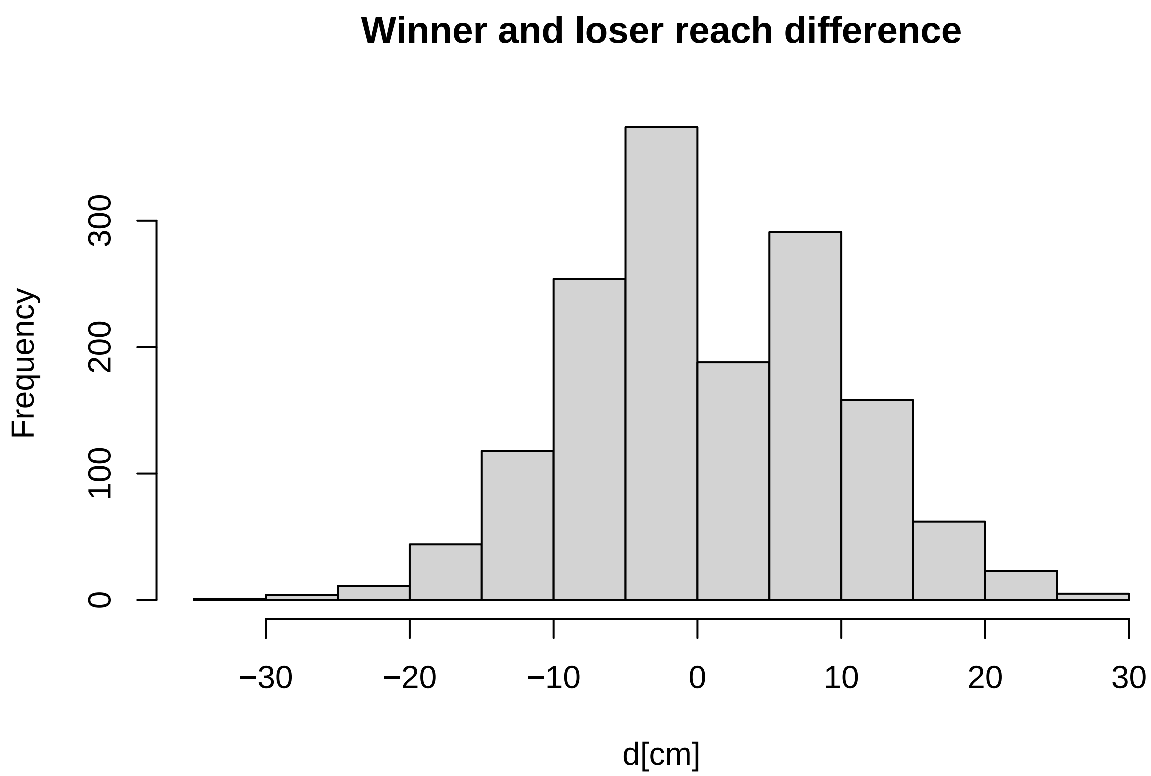
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -30.4800  -5.0800   0.0000   0.8251   7.6200  27.9400
```

```r
boxplot(d, ylab = "Reach difference [cm]", main = "Reach difference")
```

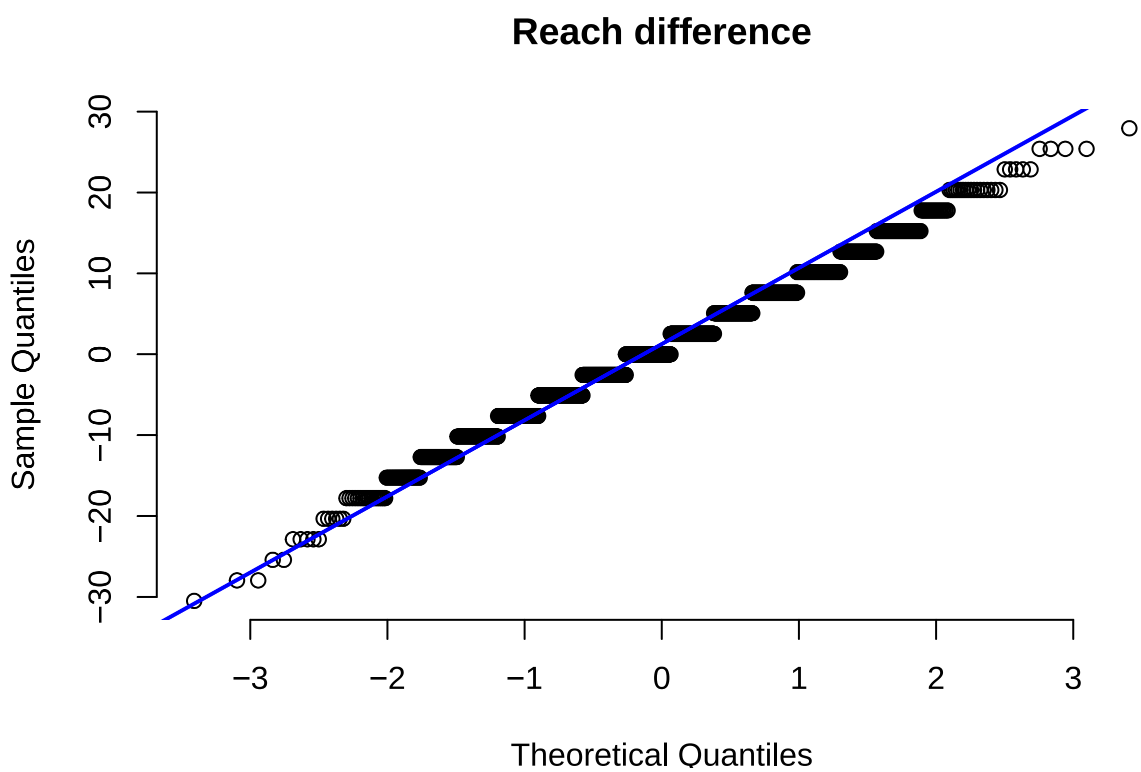


Kako bi mogli primjeniti t-test, prvo je potrebno provjeriti normalnost razdiobe podataka.

```r
hist(d, main = "Winner and loser reach difference", xlab = "d[cm]")
```




```
qqnorm(d, pch = 1, frame = FALSE, main = "Reach difference")
qqline(d, col = "blue", lwd = 2)
```



Iz histograma i Q-Q plota, možemo zaključiti da su podaci normalno distribuirani, te primjenjujemo t-test.

- H0: Razlika u dosegu između pobjednika i gubitnika jednaka je nuli.

- H1: Pobjednici imaju veći doseg od gubitnika.

```
t.test(d, alternative = "greater", mu = 0, conf.level = 0.95)
```

```
##
##  One Sample t-test
##
## data:  d
## t = 3.8558, df = 1532, p-value = 6.006e-05
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  0.4729235       Inf
## sample estimates:
## mean of x
## 0.8251272
```

S razinom značajnosti $\alpha = 0.05$ možemo odbaciti hipotezu H0 u korist hipoteze H1.

Zadatak 2: Razlikuje li se trajanje borbi (s) između pojednih kategorija?

Najprije smo iz zapisa formata borbe i trajanja zadnje runde izračunali sveukupno trajanje borbe.

```
# Računanje ukupnog trajanja borbe
fight_length <- function(parsed_format, last_round, last_round_time) {
    if (parsed_format[1] == "No Time Limit") {
        return(convert_string_time_to_seconds(last_round_time))
    }
    if (last_round == 1) {
        return(convert_string_time_to_seconds(last_round_time))
    }
    total_time = 0
    for (i in 1:(last_round - 1)) {
        total_time = total_time + parsed_format[i] * 60
    }

    total_time = total_time + convert_string_time_to_seconds(last_round_time)
    return(total_time)
}

# Na temelju retka računanje ukupnog trajanja borbe
time_from_row <- function(row) {
    parsed_format = parse_format(row$Format)
    last_round = row$last_round
    last_round_time = row$last_round_time
    return(fight_length(parsed_format, last_round, last_round_time))
}

# Računanje vektora trajanja borbe za svaki redak tablice
dur = c()
for (i in 1:nrow(all)) {
    dur = append(dur, time_from_row(all[i, ]))
}

# Dodavanje stupca ukupnog trajanja borbe u sekundama
all$Fight_duration_s <- dur
```

```r
# Grupiranje po kategorijama (odvojeno po spolu)
men_classes = c("Light Heavyweight", "Open Weight", "Lightweight", "Heavyweight",
    "Featherweight", "Bantamweight", "Welterweight", "Middleweight", "Flyweight")
women_classes = c("Women's Bantamweight", "Women's Strawweight", "Women's Featherweight",
    "Women's Flyweight")

# Funkcija za string s vraća TRUE ako sadrži neku od prije navedenih klasa
# (men_classes, women_classes)
filter_not_in_classes <- function(s) {
    for (w in women_classes) {
        if (grepl(w, s)) {
            return(TRUE)
        }
    }
    for (m in men_classes) {
        if (grepl(m, s)) {
            return(TRUE)
        }
    }
    return(FALSE)
}

# Funkcija za string s vraća kategoriju iz men_classes ili women_classes koju
# sadrži
check_which_class <- function(s) {
    for (w in women_classes) {
        if (grepl(w, s)) {
            return(w)
        }
    }
    for (m in men_classes) {
        if (grepl(m, s)) {
            return(m)
        }
    }
}

# Svi tipovi borbi koje ne znamo grupirati u kategorije po težini i spolu
ignore_fight_types = c()
categories = unique(all$Fight_type)
for (category in categories) {
    if (!filter_not_in_classes(category)) {
        ignore_fight_types = append(ignore_fight_types, category)
    }
}

ignore_fight_types
```

```
## [1] "Catch Weight Bout"
## [2] "UFC 4 Tournament Title Bout"
## [3] "UFC Superfight Championship Bout"
## [4] "UFC 5 Tournament Title Bout"
## [5] "UFC 6 Tournament Title Bout"
## [6] "Ultimate Ultimate '96 Tournament Title Bout"
```

```
##  [7] "UFC 10 Tournament Title Bout"
##  [8] "UFC 8 Tournament Title Bout"
##  [9] "UFC 3 Tournament Title Bout"
## [10] "Ultimate Ultimate '95 Tournament Title Bout"
## [11] "UFC 2 Tournament Title Bout"
## [12] "UFC 7 Tournament Title Bout"
```

Pojedine kategorije ne sadržavaju informaciju o spolu i težini te ih stoga ne uzimamo u obzir tokom daljnje analize.

```r
# Iz cijelog skupa podataka mičemo borbe čiji je fight_type unutar vektora
# ignore_fight_types
all_without_unknown_weight_classes = subset(all, !(Fight_type %in% ignore_fight_types))
```

Pretpostavke parametarske ANOVA metode su:

- nezavisnost pojedinih podataka u uzorcima
- normalna razdioba podataka
- homogenost varijanci među populacijama

1) Pretpostavljamo nezavisnost podataka u uzorcima, jer su borbe međusobno nezavisne.
2) Nastavaljamo sa testiranjem normalnosti razdiobe podataka. Koristimo Lillieforsov test normalnosti.

- H0: Podaci pripadaju normalnoj razdiobi.
- H1: Podaci ne pripadaju normalnoj razdiobi.

3) Ako razdioba podataka nije normalna, nema smisla provjeravati homoskedastičnost. U drugom slučaju, homoskedastičnost moramo provjeriti Bartlettovim testom.

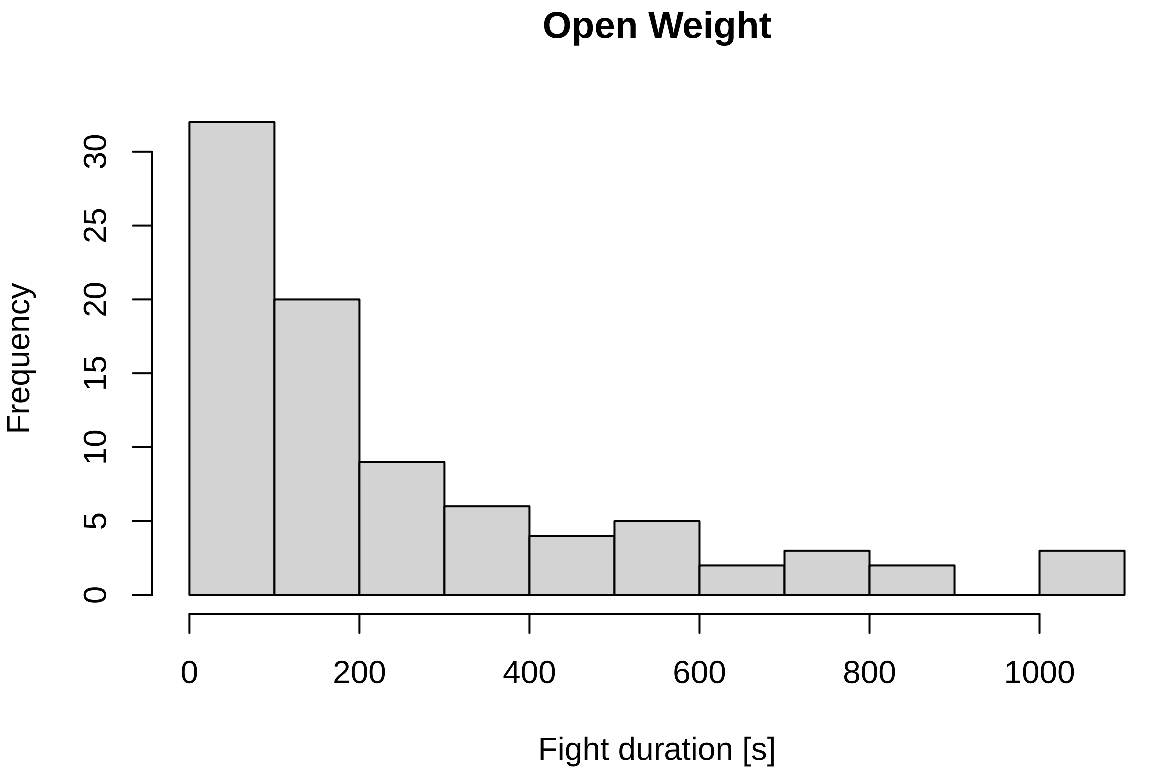
```
## Loading required package: nortest
```

```r
lillie.test(all_without_unknown_weight_classes$Fight_duration_s[weight_class == "Open Weight"])
```

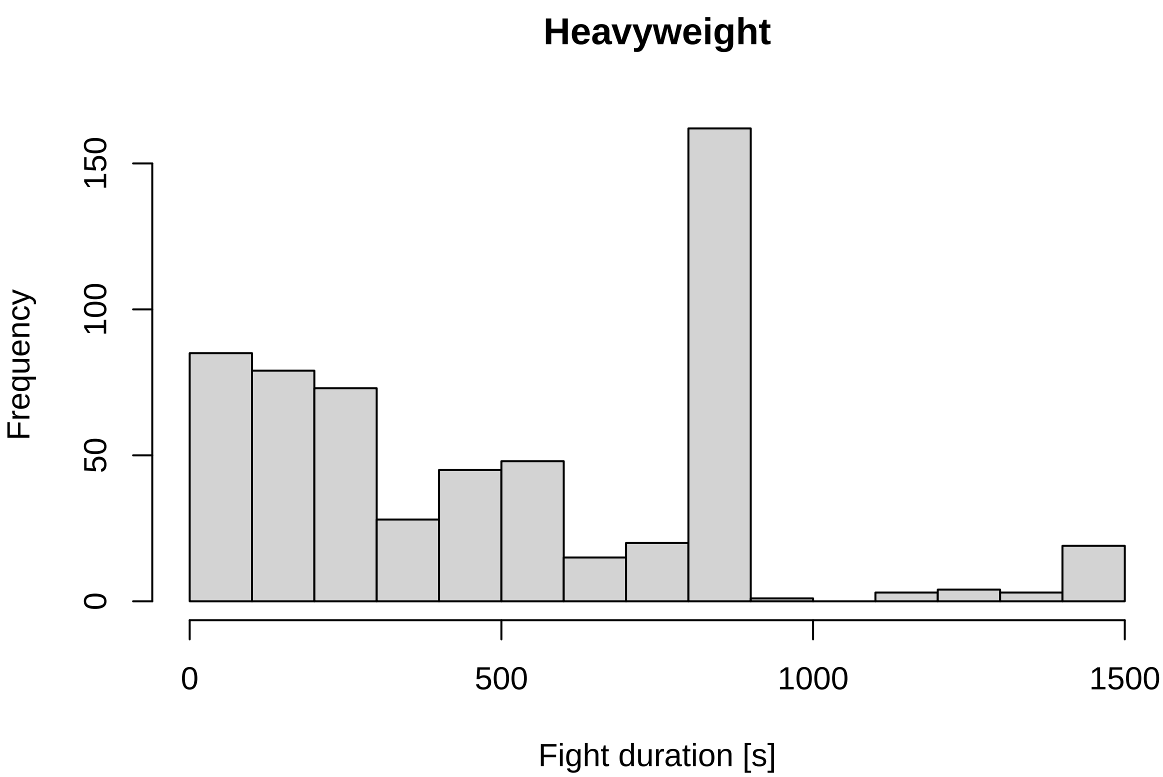
```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  all_without_unknown_weight_classes$Fight_duration_s[weight_class ==     "Open Weight"]
## D = 0.19826, p-value = 6.363e-09
```

Zbog vrlo male p vrijednosti odbacujemo H0 u korist H1 i zaključujemo da podaci nisu normalno distribuirani. Zato moramo koristiti neparametarsku verziju ANOVA testa, Kruskal-Wallis χ^2-test. Stoga ne testiramo homogenost varijanci među kategorijama.

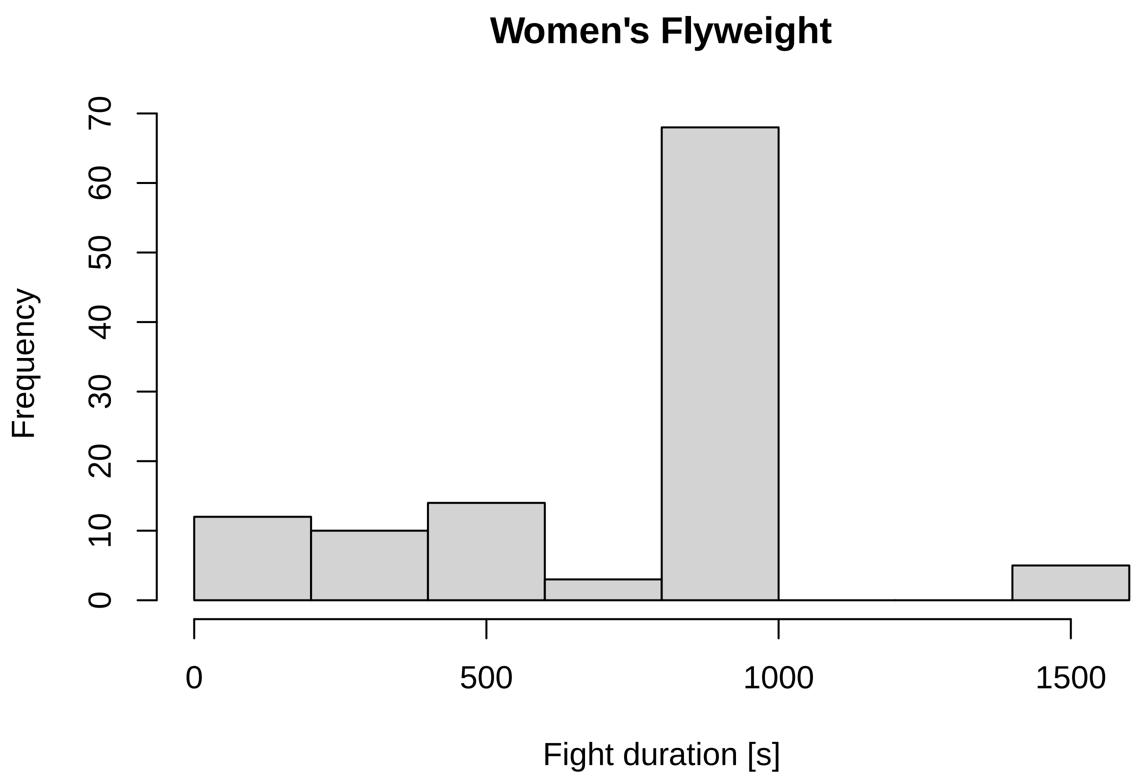
```r
# weight_classes = c(men_classes, women_classes)
hist(all_without_unknown_weight_classes$Fight_duration_s[all_without_unknown_weight_classes$weight_class
    "Open Weight"], xlab = "Fight duration [s]", main = "Open Weight")
```




```r
hist(all_without_unknown_weight_classes$Fight_duration_s[all_without_unknown_weight_classes$weight_clas
    "Heavyweight"], xlab = "Fight duration [s]", main = "Heavyweight")
```




```r
hist(all_without_unknown_weight_classes$Fight_duration_s[all_without_unknown_weight_classes$weight_clas
    "Women's Flyweight"], xlab = "Fight duration [s]", main = "Women's Flyweight")
```



Iz prikazanih histograma uočavamo da su vremena trajanja borbi sukladna formatima borbi (većina borbi završava u 15. minuti jer su formata 5+5+5 minuta).

Kako bi proveli Kruskal-Wallisov test moramo imati minimalno 5 opservacija u svakoj od kategorija, što možemo i potvrditi iz sljedeće tablice:

```
table(all_without_unknown_weight_classes$weight_class)
```

```
##
##          Bantamweight         Featherweight             Flyweight
##                   475                   551                   230
##           Heavyweight     Light Heavyweight           Lightweight
##                   585                   573                  1091
##          Middleweight           Open Weight          Welterweight
##                   813                    86                  1083
##  Women's Bantamweight Women's Featherweight     Women's Flyweight
##                   151                    16                   112
##   Women's Strawweight
##                   192
```

Postavaljamo hipoteze:

- H0: Trajanje borbi se ne razlikuje između kategorija.
- H1: Trajanje borbi se razlikuje između barem dvije kategorije.

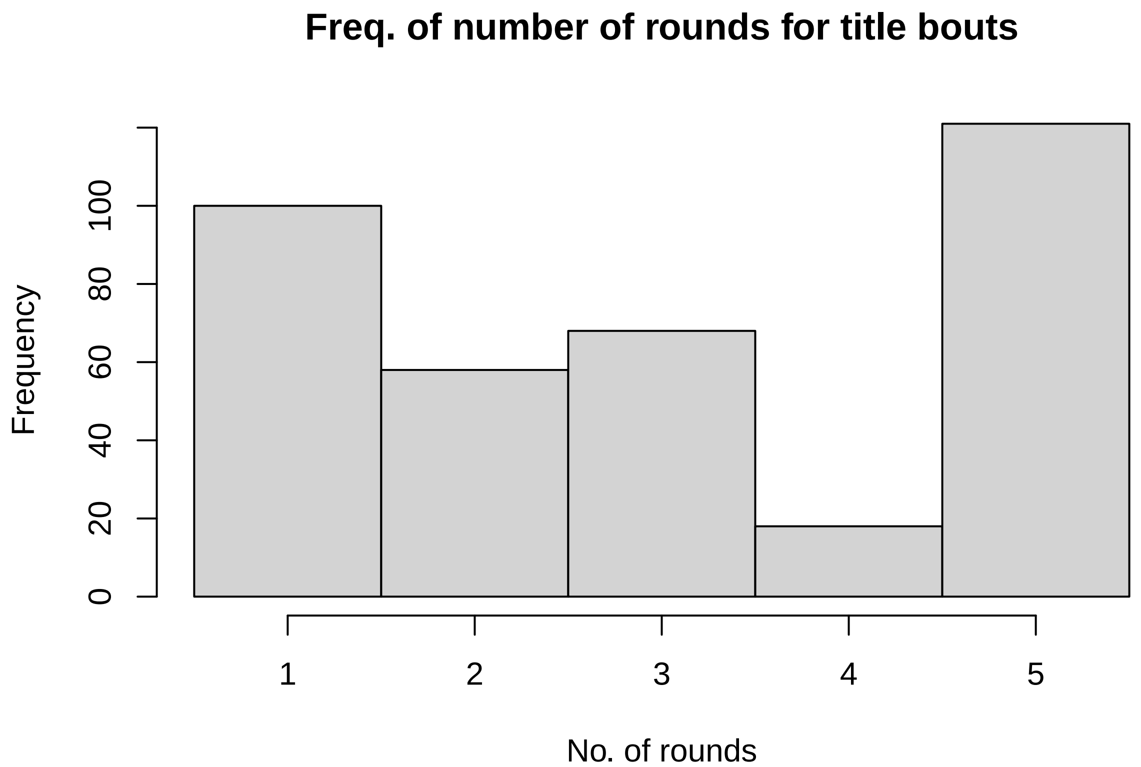
```
kruskal.test(Fight_duration_s ~ weight_class, data = all_without_unknown_weight_classes)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Fight_duration_s by weight_class
## Kruskal-Wallis chi-squared = 283.65, df = 12, p-value < 2.2e-16
```

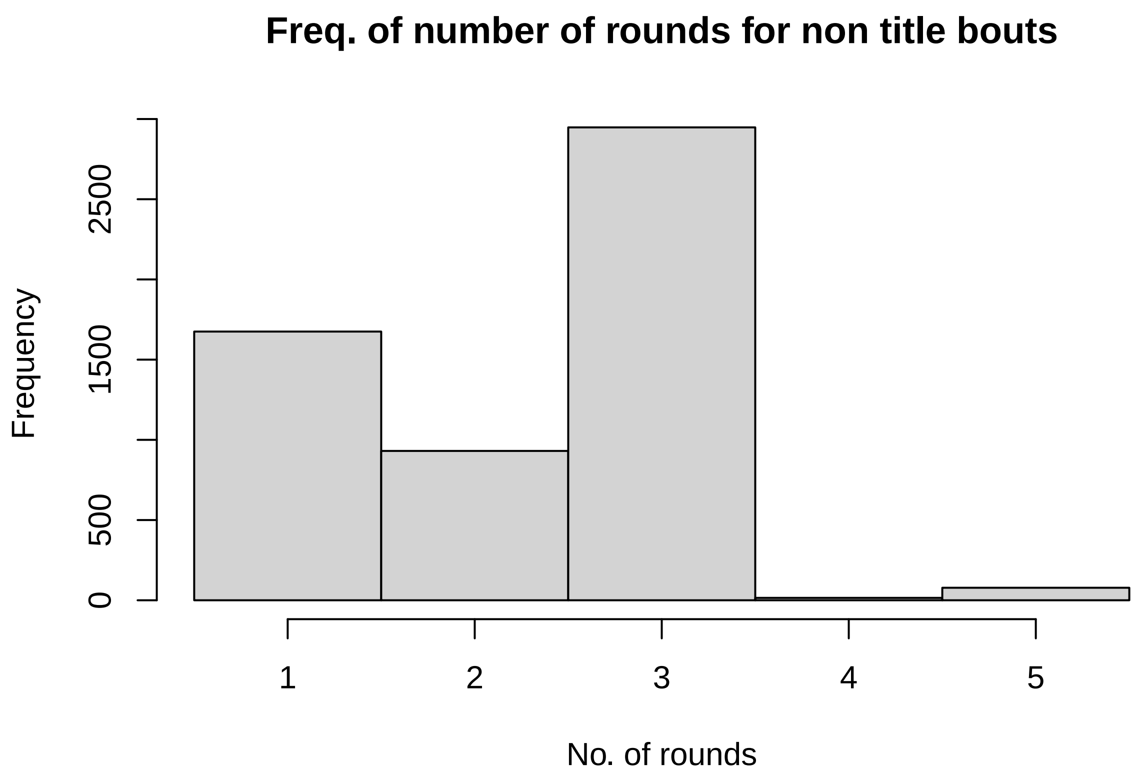
Zbog male p-vrijednosti odbacujemo H0 u korist H1 i zaključujemo da se trajanje borbi statistički značajno razlikuje između barem dvije težinske kategorije.

Zadatak 3: Traju li (u rundama) borbe za titulu duže od ostalih borbi u natjecanju?

```
hist(title_bouts_last_round, breaks = seq(min(title_bouts_last_round) - 0.5, max(title_bouts_last_round)
    0.5, by = 1), main = "Freq. of number of rounds for title bouts", xlab = "No. of rounds")
```




```
hist(non_title_bouts_last_round, breaks = seq(min(non_title_bouts_last_round) - 0.5,
    max(non_title_bouts_last_round) + 0.5, by = 1), main = "Freq. of number of rounds for non title bou
    xlab = "No. of rounds")
```




```r
lillie.test(non_title_bouts_last_round)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  non_title_bouts_last_round
## D = 0.31964, p-value < 2.2e-16
```

```r
lillie.test(title_bouts_last_round)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  title_bouts_last_round
## D = 0.22181, p-value < 2.2e-16
```

Iz histograma i Lillieforsovog testa vidimo da podaci nisu normalno distribuirani te stoga primjenjujemo neparametarsku verziju t-testa, Wilcoxonov signed rank test. Postavljamo hipoteze: - H0: Borbe za titulu ne traju duže (u rundama) od ostalih borbi u natjecanju. - H1: Borbe za titulu traju duže (u rundama) od ostalih borbi u natjecanju.

```r
wilcox.test(title_bouts_last_round, non_title_bouts_last_round, alternative = "greater",
    conf.level = 0.9)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  title_bouts_last_round and non_title_bouts_last_round
## W = 1264014, p-value = 1.3e-15
## alternative hypothesis: true location shift is greater than 0
```

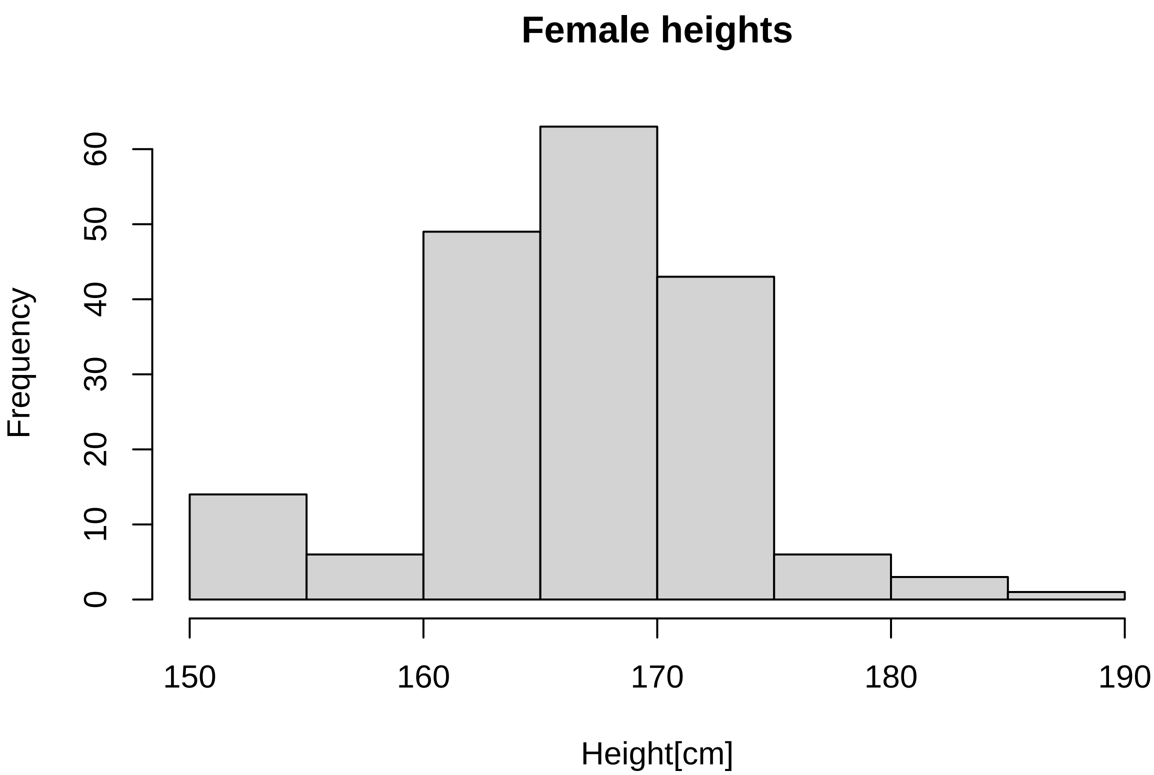
Odabrali smo razinu značajnosti $\alpha = 0.1$ jer želimo veću robustnost testa. Zbog izračunate p-vrijednosti odbacujemo H0 u korist H1 i zaključujemo da borbe za titulu traju duže (u rundama) od ostalih borbi u natjecanju.

Dodatni zadatak 1. - Pobjeđuju li niži borci češće preko submissiona (predaje)?

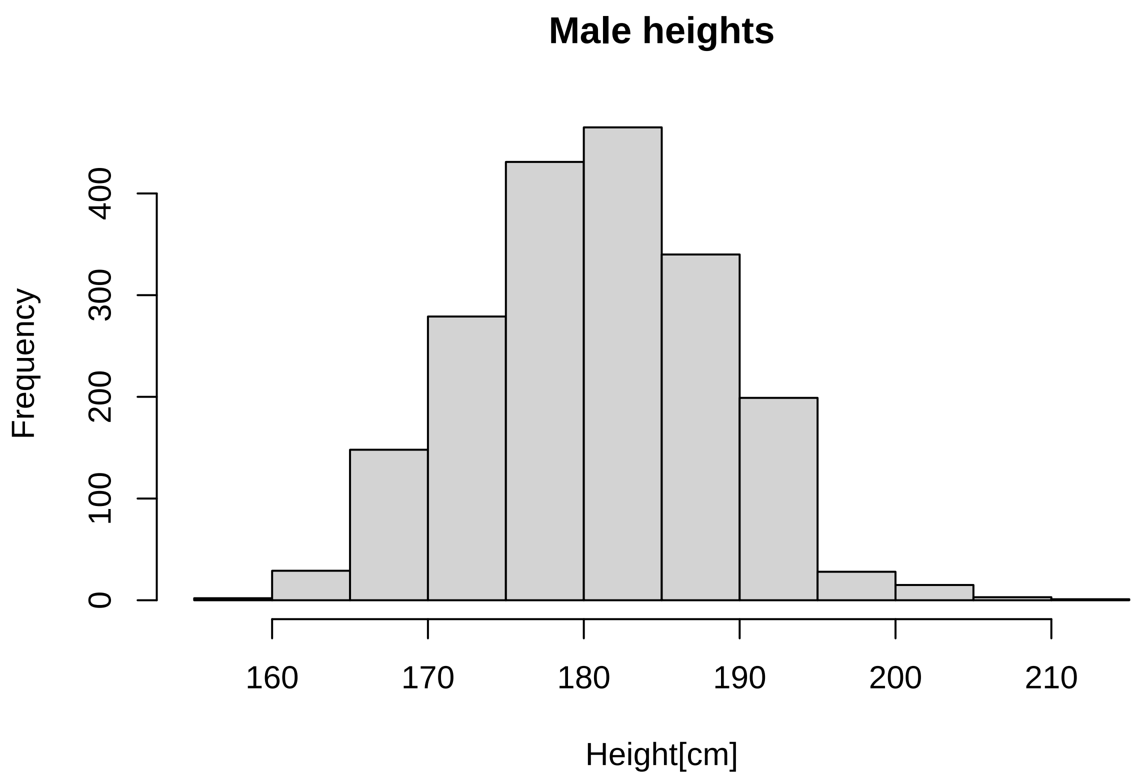
Svim borcima dodali smo obilježje spola koje smo odredili putem imena kategorija borbi u kojima se taj borac borio.

Zatim smo iznos svih visina boraca pretvorili iz imperijalnog sustava mjernih jedinica u metrički.

```
female_heights = subset(fighter_details, gender == "female")$Height_cm
male_heights = subset(fighter_details, gender == "male")$Height_cm
hist(female_heights, main = "Female heights", xlab = "Height[cm]")
```




```
hist(male_heights, main = "Male heights", xlab = "Height[cm]")
```



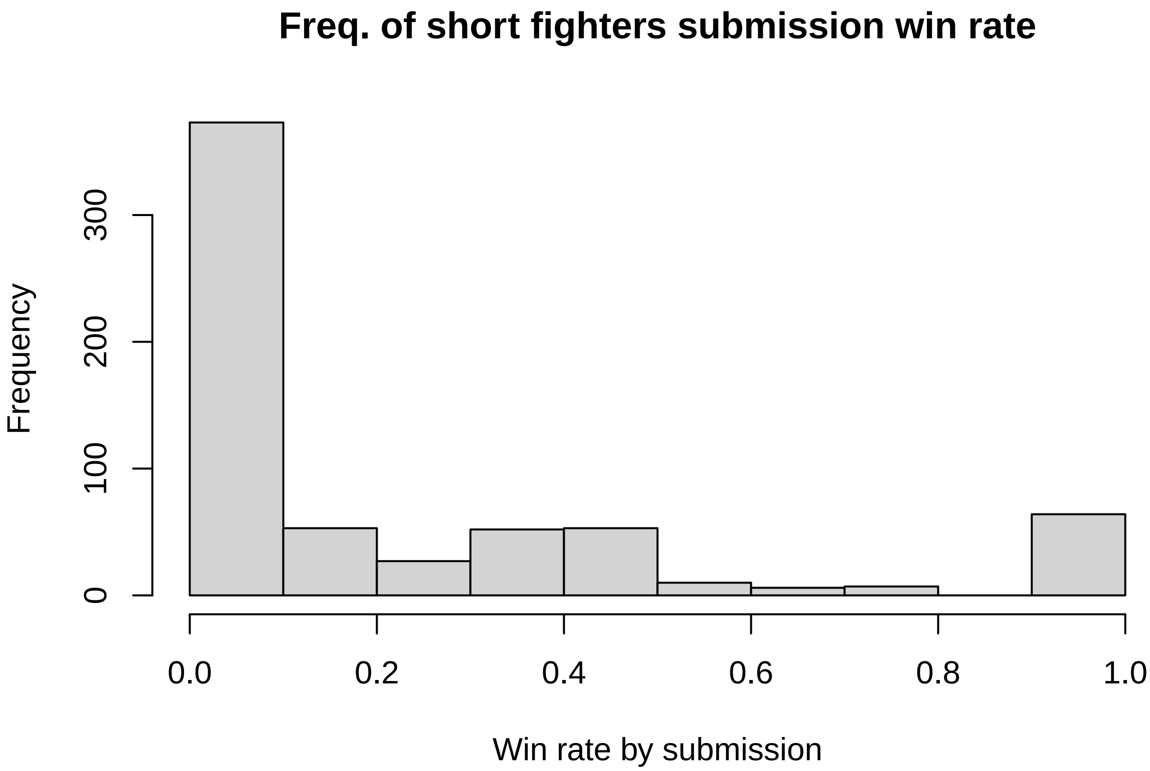
Na temelju medijana svih muških i ženskih visina napravili smo podjelu boraca na niže i više borce, s obzirom na spol.

```
# Određivanje medijana visine za mušku i žensku populaciju
male_median_height = median(male_heights, na.rm = TRUE)
female_median_height = median(female_heights, na.rm = TRUE)

# Određivanje kategorije visine po spolu (short za visine ispod mediana, tall
# za visine iznad mediana)
height_category = c()
for (i in 1:nrow(fighter_details)) {
    if (is.na(fighter_details[i, ]$Height_cm) | is.na(fighter_details[i, ]$gender)) {
        height_category = append(height_category, NA)
        next
    } else {
        if (fighter_details[i, ]$gender == "male") {
            if (fighter_details[i, ]$Height_cm >= male_median_height) {
                height_category = append(height_category, "tall")
            } else {
                height_category = append(height_category, "short")
            }
        } else {
            if (fighter_details[i, ]$Height_cm >= female_median_height) {
                height_category = append(height_category, "tall")
            } else {
                height_category = append(height_category, "short")
            }
        }
    }
}
```

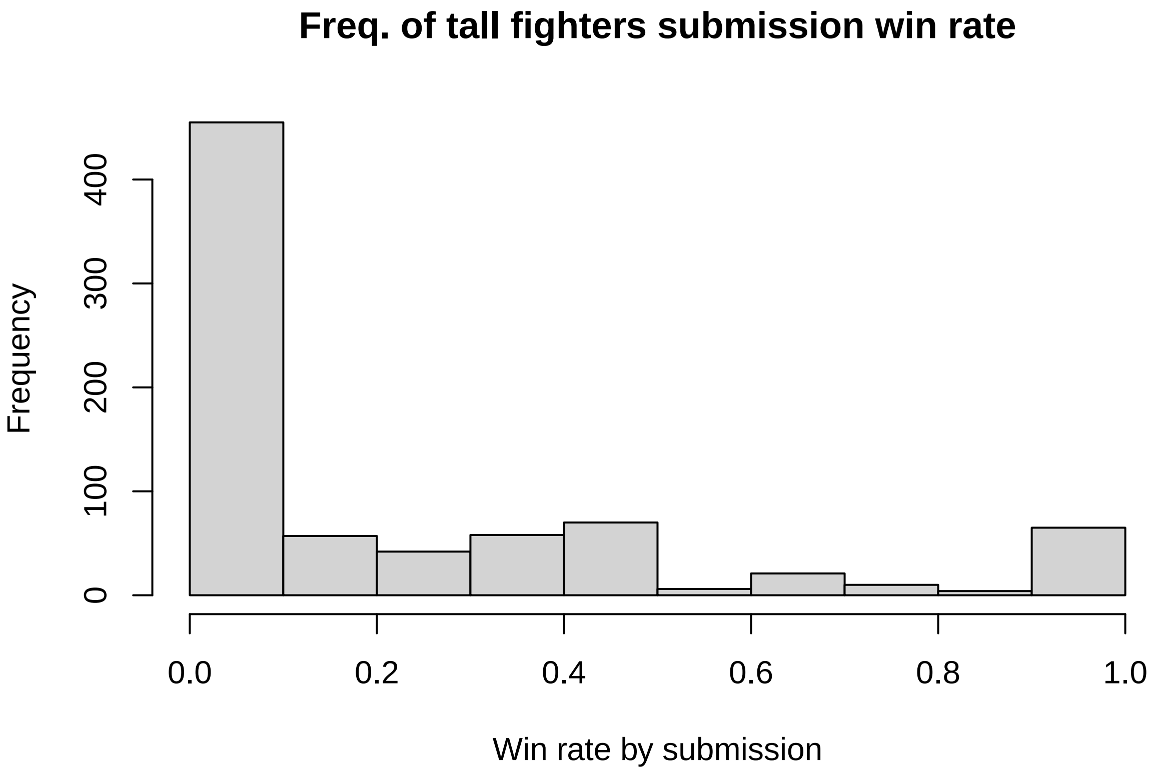
```r
# Dodavanje stupca kategorije visine
fighter_details$height_category = height_category
```

Za svakog borca odredili smo postotak njegovih pobjeda putem predaje protivničkog borca. Ukoliko borac nije imao niti jednu pobjedu, postotak pobjeda putem predaje protivnika označili smo sa NA.

```r
# Vektor postotaka pobjede putem submissiona za niske borce
short_winners = subset(fighter_details, height_category == "short" & !is.na(win_rate_by_submission))$wi
# Vektori postotaka pobjede putem submissiona za visoke borce
tall_winners = subset(fighter_details, height_category == "tall" & !is.na(win_rate_by_submission))$win_

hist(short_winners, main = "Freq. of short fighters submission win rate", xlab = "Win rate by submissio
```




```r
hist(tall_winners, main = "Freq. of tall fighters submission win rate", xlab = "Win rate by submission")
```



Postavljamo sljedeće hipoteze:

- H0: Postotci pobjeda putem submissiona jednaki su za visoke i niske borce.
- H1: Postotci pobjeda putem submissiona manji su za visoke borce.

Razinu značajnosti α postavljamo na 0.1 zbog toga što želimo biti manje osjetljivi na ne odbacivanje H0.

```
wilcox.test(tall_winners, short_winners, alternative = "less", conf.level = 0.9)
```

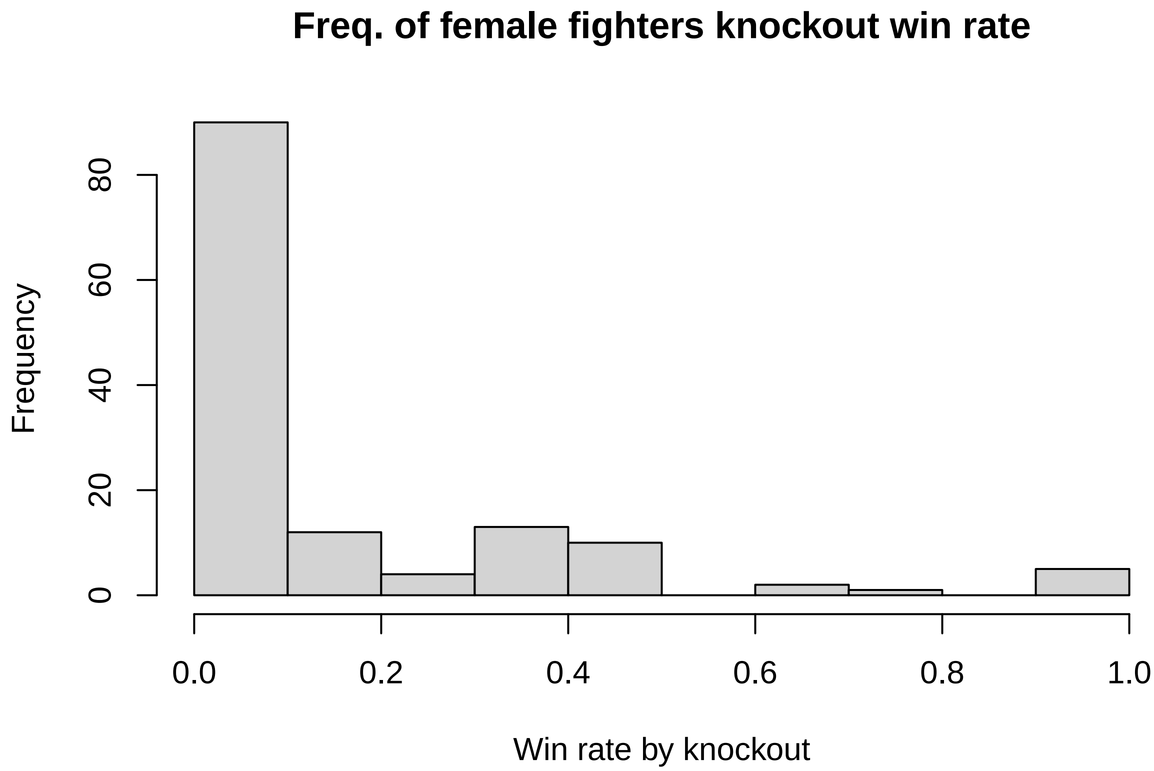
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  tall_winners and short_winners
## W = 254820, p-value = 0.539
## alternative hypothesis: true location shift is less than 0
```

Na razini značajnosti $\alpha = 0.1$ i dobivene p vrijednosti iz Wilcoxonovog testa sume rangova zaključujemo da ne možemo odbaciti H0 u korist H1 (ne možemo odbaciti hipotezu da su postotci pobjeda putem submissiona jednaki za visoke i niske borce).

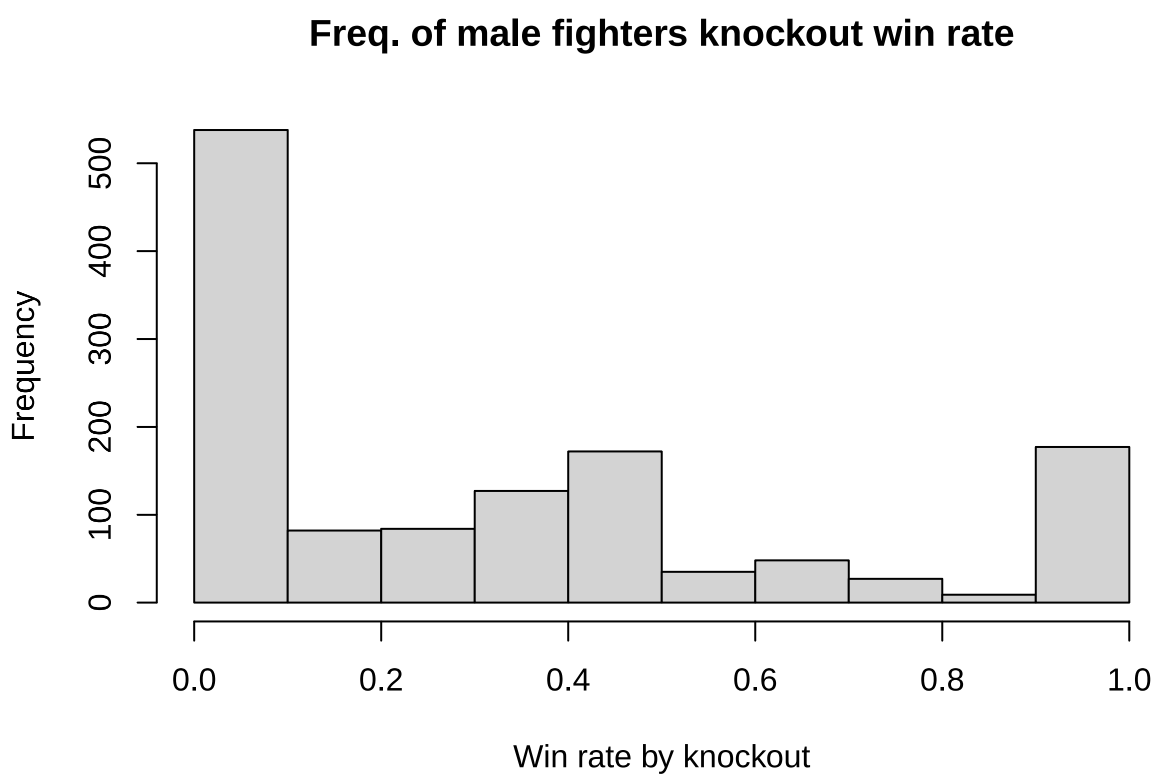
Dodatni zadatak 2. - Završavaju li muške borbe češće nokautom?

Kao i za prethodni zadatak, najprije smo odredili postotak pobjeda svakog borca putem nokauta. Za borca koji nije imao pobjeda, zabilježili smo postotak pobjeda putem nokauta sa NA.

```
female_ko_winners = subset(fighter_details, gender == "female" & !is.na(win_rate_by_ko))$win_rate_by_ko
male_ko_winners = subset(fighter_details, gender == "male" & !is.na(win_rate_by_ko))$win_rate_by_ko

hist(female_ko_winners, main = "Freq. of female fighters knockout win rate", xlab = "Win rate by knocko
```




```
hist(male_ko_winners, main = "Freq. of male fighters knockout win rate", xlab = "Win rate by knockout")
```



Postavljamo hipoteze:

- H0: Postotci pobjeda putem nokauta jednaki su za muškarce i žene.
- H1: Postotci pobjeda putem nokauta veći su za muškarce.

Razinu značajnosti α postavljamo na 0.1 kao i u prethodnim testovima.

```
wilcox.test(male_ko_winners, female_ko_winners, alternative = "greater", conf.level = 0.9)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  male_ko_winners and female_ko_winners
## W = 116214, p-value = 3.734e-10
## alternative hypothesis: true location shift is greater than 0
```

Na razini značajnosti $\alpha = 0.1$ možemo odbaciti H0 u korist H1 (postotak pobjeda putem nokauta veći je za muškarce).

Dodatni zadatak 3. - Razlikuje li se broj pobjeda i pobjeda putem nokauta ovisno o stavu borca (stance)?

```
# Odredivanje broja pobjeda i broja pobjeda putem knockout-a za borce
total_wins = c()
total_wins_by_ko = c()
for (i in (1:nrow(fighter_details))) {
    fn = fighter_details[i, ]$fighter_name
    wins = subset(all, Winner == fn)
    wins_by_ko = subset(wins, win_by == "KO/TKO")
    total_wins = append(total_wins, nrow(wins))
    total_wins_by_ko = append(total_wins_by_ko, nrow(wins_by_ko))
}

# Dodavanje stupaca ukupnih pobjeda, ukupnih pobjeda putem knockout-a i ukupnih
# pobjeda bez knockout-a
fighter_details$total_wins = total_wins
fighter_details$total_wins_by_ko = total_wins_by_ko
fighter_details$total_wins_without_ko = total_wins - total_wins_by_ko

table(fighter_details$Stance)
```

```
##
##             Open Stance    Orthodox    Sideways    Southpaw      Switch
##         804           7        2163           3         493         126
```

Ignoriramo borce s nepoznatim stavom. Također ignoriramo borce sa stavom "Open Stance" i "Sideways" zbog male frekvencije. Ako je borac stava "Orthodox", onda je dešnjak. Ako je stava "Southpaw", onda je ljevak. Ako je "Switch", onda je ambidekstar.

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
stance_table
```

```
##      Stance total_wins_by_ko total_wins_without_ko
## 1: Orthodox             1408                  2999
## 2: Southpaw              384                   856
## 3:   Switch              100                   125
```

```r
# Moramo maknuti Stance jer je u tablici to predstavljeno kao zavisna
# varijabla, a zapravo je nezavisna
stance_table = select(stance_table, -Stance)
```

Očekivane frekvencije su veće od 5 u svakoj ćeliji tablice. Stoga smijemo primjeniti test homogenosti. Postavljamo hipoteze:

- H0: Postotak pobjeda putem nokauta jednak je za svaku od kategorija boraca prema stavu (ljevaci, dešnjaci i ambidekstri).
- H1: Postotak pobjeda putem nokauta nije jednak za barem dvije od kategorija boraca prema stavu (ljevaci, dešnjaci i ambidekstri).

Za chisq.test nije dostupan argument conf_level, tako da ne postavljamo nikakvu razinu značajnosti kao argument testa. Ipak, odabiremo razinu značajnosti $\alpha = 0.05$.

```r
chisq.test(stance_table, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  stance_table
## X-squared = 16.434, df = 2, p-value = 0.00027
```

Na odabranoj razini značajnosti možemo odbaciti H0 u korist H1 (udio pobjeda putem KO i pobjeda drugim načinima nije isti za sve kategorije Stance). Iz tablice *stance_table* možemo naslutiti da borci koji su ambidekstri imaju veći udio pobjeda putem KO.

Zadatak 4: Možemo li iz zadanih obilježja predvidjeti pobjednika?

Za svaku borbu smo izračunali dob oba borca (Red i Blue) na dan borbe.

```
## Loading required package: timechange
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Određeni stupci unutar tablice svih borbi su u obliku "x of y" jer govore o tome koliko je udaraca borac obranio, primio i slično. Za podskup tih stupaca smo uzimali u obzir samo prvi broj x, jer nam on daje informaciju o razmijenjenim udarcima tijekom borbe. Drugi podskup tih stupaca opisuje općenitu preciznost borca, i za taj podskup stupaca smo izračunali omjer x/y (postotak).

Nakon toga smo odredili regresorske varijable. Zavisna varijabla je indikatorska varijabla u obliku vektora (označava pobjedu crvenog borca).

```
# Odabrane regresorske varijable i zavisna varijabla
selected_columns = c("R_KD", "B_KD", "R_SUB_ATT", "B_SUB_ATT", "R_REV", "B_REV",
    "TD_Avg.r", "SLpM.r", "SApM.r", "Sub_Avg.r", "TD_Avg.b", "SLpM.b", "SApM.b",
    "Sub_Avg.b", "Height_cm.b", "Height_cm.r", "Reach_cm.b", "Reach_cm.r", "Weight_kg.b",
    "Weight_kg.r", "red_age", "blue_age", "r_sig_str", "b_sig_str", "r_total_str",
    "b_total_str", "r_td", "b_td", "r_head", "b_head", "r_body", "b_body", "r_leg",
    "b_leg", "r_distance", "b_distance", "r_clinch", "b_clinch", "r_ground", "b_ground",
    "str_def.r", "str_acc.r", "td_acc.r", "td_def.r", "str_def.b", "str_acc.b", "td_acc.b",
    "td_def.b", "red_is_winner", "is_b_southpaw", "is_b_orthodox", "is_r_southpaw",
    "is_r_orthodox")
variables = selected_columns[selected_columns != "red_is_winner"]
```

```
library(tidyr)
# Iz seta podataka uzimamo samo odabrane regresorske varijable i zavisnu
# varijablu
logreg_data = subset(all_for_logreg, select = selected_columns)
# Uzimamo samo retke koji nemaju NA vrijednosti unutar odabranih varijabli
logreg_data = logreg_data %>%
    drop_na()
```

Koristimo model logističke regresije jer je zavisna varijabla indikatorska.

```
require(caret)
```

```
## Loading required package: caret

## Loading required package: ggplot2

## Loading required package: lattice
```

```
# b je formula varijabla_1 + varijabla_2 + ..., pri čemu je varijabla_i unutar
# skupa odabranih regresorskih varijabli
b <- paste(variables, collapse = " + ")
logreg_mdl = glm(as.formula(paste("red_is_winner ~ ", b)), data = logreg_data, family = binomial())
summary(logreg_mdl)
```

```
##
## Call:
## glm(formula = as.formula(paste("red_is_winner ~ ", b)), family = binomial(),
##     data = logreg_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7502  -0.4115   0.1432   0.4793   3.9856
##
## Coefficients: (4 not defined because of singularities)
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.315747   2.083947  -1.111 0.266469
## R_KD           1.615162   0.129801  12.443  < 2e-16 ***
## B_KD          -1.497188   0.116559 -12.845  < 2e-16 ***
## R_SUB_ATT      0.846789   0.074988  11.292  < 2e-16 ***
## B_SUB_ATT     -0.570666   0.068959  -8.275  < 2e-16 ***
## R_REV          0.332944   0.120130   2.772 0.005579 **
## B_REV         -0.584883   0.120491  -4.854 1.21e-06 ***
## TD_Avg.r      -0.165656   0.050094  -3.307 0.000943 ***
```

```
## SLpM.r        -0.242986   0.059511  -4.083 4.45e-05 ***
## SApM.r         0.109205   0.059600   1.832 0.066906 .
## Sub_Avg.r      0.042456   0.076657   0.554 0.579686
## TD_Avg.b       0.059693   0.051272   1.164 0.244326
## SLpM.b         0.080281   0.058936   1.362 0.173141
## SApM.b        -0.039535   0.056360  -0.701 0.483002
## Sub_Avg.b      0.118123   0.075815   1.558 0.119220
## Height_cm.b   -0.011844   0.013229  -0.895 0.370655
## Height_cm.r    0.008308   0.012877   0.645 0.518828
## Reach_cm.b     0.001010   0.010314   0.098 0.922021
## Reach_cm.r     0.012004   0.010020   1.198 0.230929
## Weight_kg.b    0.010779   0.010546   1.022 0.306725
## Weight_kg.r   -0.006530   0.010316  -0.633 0.526739
## red_age       -0.042532   0.012143  -3.503 0.000461 ***
## blue_age       0.017983   0.012749   1.410 0.158396
## r_sig_str      0.086270   0.013324   6.475 9.51e-11 ***
## b_sig_str     -0.093908   0.013507  -6.952 3.59e-12 ***
## r_total_str    0.013798   0.002766   4.988 6.11e-07 ***
## b_total_str   -0.002868   0.002696  -1.064 0.287503
## r_td           0.350693   0.042225   8.305  < 2e-16 ***
## b_td          -0.388208   0.041210  -9.420  < 2e-16 ***
## r_head         0.014486   0.007965   1.819 0.068955 .
## b_head        -0.025500   0.008057  -3.165 0.001551 **
## r_body        -0.015307   0.011333  -1.351 0.176814
## b_body        -0.014322   0.011892  -1.204 0.228449
## r_leg                NA         NA      NA       NA
## b_leg                NA         NA      NA       NA
## r_distance    -0.039667   0.010012  -3.962 7.44e-05 ***
## b_distance     0.050576   0.010486   4.823 1.41e-06 ***
## r_clinch      -0.035021   0.012530  -2.795 0.005189 **
## b_clinch       0.055289   0.013084   4.226 2.38e-05 ***
## r_ground             NA         NA      NA       NA
## b_ground             NA         NA      NA       NA
## str_def.r      1.026442   0.884551   1.160 0.245882
## str_acc.r      1.025007   0.822912   1.246 0.212917
## td_acc.r       0.436551   0.278385   1.568 0.116845
## td_def.r      -0.135485   0.276350  -0.490 0.623945
## str_def.b      0.898371   0.805421   1.115 0.264677
## str_acc.b     -0.525559   0.770376  -0.682 0.495106
## td_acc.b       0.527283   0.270049   1.953 0.050873 .
## td_def.b      -0.352784   0.240572  -1.466 0.142529
## is_b_southpaw  0.110691   0.246610   0.449 0.653539
## is_b_orthodox  0.011043   0.231438   0.048 0.961944
## is_r_southpaw  0.215440   0.277156   0.777 0.436968
## is_r_orthodox  0.064542   0.262283   0.246 0.805621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6505.0  on 4894  degrees of freedom
## Residual deviance: 3154.8  on 4846  degrees of freedom
## AIC: 3252.8
##
```

```
## Number of Fisher Scoring iterations: 6
```

Iz ispisa uočavamo da su neki od regresora međusobno zavisni (NA vrijednosti). U ispisu su označeni statistički signifikantni regresori.

Na tri različita načina evaluirat ćemo kvalitetu dobivenog modela.

Računamo R^2 koji govori o tome koliko je procjenjeni model blizu ili daleko od nul-modela (što je R^2 bliži 1, to je model bolji).

```
# Računanje Rsq
Rsq = 1 - logreg_mdl$deviance/logreg_mdl$null.deviance
Rsq
```

```
## [1] 0.5150178
```

Izrađujemo matricu zabune.

```
# Izrada confusion matrix-a
yhat <- logreg_mdl$fitted.values >= 0.5
tab <- table(logreg_data$red_is_winner, yhat)

tab
```

```
##    yhat
##     FALSE TRUE
##   0  1527  337
##   1   273 2758
```

Iz matrice zabune možemo zaključiti da model dobro predviđa ishod borbe (borbe u kojima crveni borac nije pobjednik su označene kao takve, i obrnuto).

```
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] 0.875383
```

```
precision
```

```
## [1] 0.8911147
```

```
recall
```

```
## [1] 0.9099307
```

```
specificity
```

```
## [1] 0.8483333
```

Zbog visokih vrijednosti izračunatih varijabli (točnost, preciznost, odziv i specifičnost) zaključujemo da je model kvalitetan.

Model bez linearno zavisnih i neznačajnih regresora

```
# Izbacivanje nesignifikantnih varijabli
significant_variables = c("R_KD", "B_KD", "R_SUB_ATT", "B_SUB_ATT", "R_REV", "B_REV",
    "TD_Avg.r", "red_age", "r_sig_str", "b_sig_str", "r_total_str", "r_td", "b_td",
    "r_head", "b_head", "r_distance", "b_distance", "r_clinch", "b_clinch", "td_acc.b")
```

```r
b <- paste(significant_variables, collapse = " + ")
logreg_mdl_reduced = glm(as.formula(paste("red_is_winner ~ ", b)), data = logreg_data,
    family = binomial())
summary(logreg_mdl_reduced)
```

```
##
## Call:
## glm(formula = as.formula(paste("red_is_winner ~ ", b)), family = binomial(),
##     data = logreg_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.6433  -0.4280   0.1465   0.4861   4.0328
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.962231   0.368571   2.611  0.00904 **
## R_KD         1.607436   0.128391  12.520  < 2e-16 ***
## B_KD        -1.508489   0.112876 -13.364  < 2e-16 ***
## R_SUB_ATT    0.862154   0.068966  12.501  < 2e-16 ***
## B_SUB_ATT   -0.542844   0.061487  -8.829  < 2e-16 ***
## R_REV        0.345064   0.118854   2.903  0.00369 **
## B_REV       -0.604762   0.118980  -5.083 3.72e-07 ***
## TD_Avg.r    -0.120331   0.043399  -2.773  0.00556 **
## red_age     -0.026947   0.011107  -2.426  0.01526 *
## r_sig_str    0.072030   0.011583   6.219 5.01e-10 ***
## b_sig_str   -0.101973   0.010755  -9.482  < 2e-16 ***
## r_total_str  0.012599   0.002618   4.812 1.49e-06 ***
## r_td         0.340061   0.040379   8.422  < 2e-16 ***
## b_td        -0.376136   0.035837 -10.496  < 2e-16 ***
## r_head       0.025316   0.006281   4.031 5.56e-05 ***
## b_head      -0.020417   0.006350  -3.215  0.00130 **
## r_distance  -0.038536   0.009728  -3.961 7.46e-05 ***
## b_distance   0.051120   0.009804   5.214 1.85e-07 ***
## r_clinch    -0.037243   0.012331  -3.020  0.00253 **
## b_clinch     0.053164   0.012694   4.188 2.81e-05 ***
## td_acc.b     0.680129   0.242015   2.810  0.00495 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6505  on 4894  degrees of freedom
## Residual deviance: 3210  on 4874  degrees of freedom
## AIC: 3252
##
## Number of Fisher Scoring iterations: 6
```

Kao i za prethodni model, računamo iste mjere kvalitete (R^2, točnost, preciznost, odziv i specifičnost).

```r
Rsq = 1 - logreg_mdl_reduced$deviance/logreg_mdl_reduced$null.deviance
Rsq
```

```
## [1] 0.5065386
```

```r
yhat <- logreg_mdl_reduced$fitted.values >= 0.5
tab <- table(logreg_data$red_is_winner, yhat)

tab
```

```
##    yhat
##     FALSE TRUE
##   0  1519  345
##   1   268 2763
```

```r
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] 0.8747702
```

```r
precision
```

```
## [1] 0.8889961
```

```r
recall
```

```
## [1] 0.9115803
```

```r
specificity
```

```
## [1] 0.850028
```

Usporedba originalnog i reduciranog modela

Za usporedbu modela koristit ćemo ANOVA-u. Postavljamo hipoteze:

- H0: Modeli su jednake kvalitete
- H1: Originalni model je bolji od reduciranog

```r
# Usporedba dva modela
anova(logreg_mdl, logreg_mdl_reduced, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: red_is_winner ~ R_KD + B_KD + R_SUB_ATT + B_SUB_ATT + R_REV +
##     B_REV + TD_Avg.r + SLpM.r + SApM.r + Sub_Avg.r + TD_Avg.b +
##     SLpM.b + SApM.b + Sub_Avg.b + Height_cm.b + Height_cm.r +
##     Reach_cm.b + Reach_cm.r + Weight_kg.b + Weight_kg.r + red_age +
##     blue_age + r_sig_str + b_sig_str + r_total_str + b_total_str +
##     r_td + b_td + r_head + b_head + r_body + b_body + r_leg +
##     b_leg + r_distance + b_distance + r_clinch + b_clinch + r_ground +
##     b_ground + str_def.r + str_acc.r + td_acc.r + td_def.r +
##     str_def.b + str_acc.b + td_acc.b + td_def.b + is_b_southpaw +
##     is_b_orthodox + is_r_southpaw + is_r_orthodox
## Model 2: red_is_winner ~ R_KD + B_KD + R_SUB_ATT + B_SUB_ATT + R_REV +
##     B_REV + TD_Avg.r + red_age + r_sig_str + b_sig_str + r_total_str +
##     r_td + b_td + r_head + b_head + r_distance + b_distance +
##     r_clinch + b_clinch + td_acc.b
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      4846     3154.8
```

```
## 2      4874     3210.0 -28  -55.157 0.001627 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sa razinom značajnosti $\alpha = 0.05$ zaključujemo da možemo odbaciti H0 u korist H1 (originalni model bolji je od reduciranog).

Model s apriornim podacima

Postavlja se zanimljivo pitanje možemo li samo na temelju značajki dostupnih prije borbe odrediti pobjednika (prijašnja statistika svakog borca).

Odabiremo samo varijable dostupne prije borbe za svakog borca, te njih koristimo kao regresore u novom logističkom modelu.

```r
fighter_details_variables = c("TD_Avg.r", "SLpM.r", "SApM.r", "Sub_Avg.r", "TD_Avg.b",
    "SLpM.b", "SApM.b", "Sub_Avg.b", "Height_cm.b", "Height_cm.r", "Reach_cm.b",
    "Reach_cm.r", "Weight_kg.b", "Weight_kg.r", "red_age", "blue_age", "str_def.r",
    "str_acc.r", "td_acc.r", "td_def.r", "str_def.b", "str_acc.b", "td_acc.b", "td_def.b",
    "red_is_winner", "is_b_southpaw", "is_b_orthodox", "is_r_southpaw", "is_r_orthodox")
logreg_fighters_data = subset(logreg_data, select = fighter_details_variables)
fighter_details_variables = fighter_details_variables[fighter_details_variables !=
    "red_is_winner"]

b <- paste(fighter_details_variables, collapse = " + ")
logreg_mdl_fighter_details = glm(as.formula(paste("red_is_winner ~ ", b)), data = logreg_fighters_data,
    family = binomial())
summary(logreg_mdl_fighter_details)
```

```
##
## Call:
## glm(formula = as.formula(paste("red_is_winner ~ ", b)), family = binomial(),
##     data = logreg_fighters_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5911  -1.1276   0.6372   0.9376   2.5427
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.362263   1.470762   0.246 0.805442
## TD_Avg.r       0.082014   0.031043   2.642 0.008243 **
## SLpM.r         0.193547   0.041759   4.635 3.57e-06 ***
## SApM.r        -0.236117   0.046166  -5.115 3.15e-07 ***
## Sub_Avg.r      0.228914   0.053854   4.251 2.13e-05 ***
## TD_Avg.b      -0.135603   0.030649  -4.424 9.67e-06 ***
## SLpM.b        -0.393796   0.040807  -9.650  < 2e-16 ***
## SApM.b         0.290459   0.042341   6.860 6.89e-12 ***
## Sub_Avg.b     -0.020913   0.050196  -0.417 0.676955
## Height_cm.b    0.003686   0.009000   0.410 0.682136
## Height_cm.r   -0.018386   0.008944  -2.056 0.039817 *
## Reach_cm.b    -0.010076   0.006999  -1.440 0.149995
## Reach_cm.r     0.015135   0.007032   2.152 0.031382 *
## Weight_kg.b   -0.006954   0.007143  -0.973 0.330337
## Weight_kg.r    0.018584   0.007034   2.642 0.008240 **
## red_age       -0.070837   0.008392  -8.441  < 2e-16 ***
```

```
## blue_age        0.035646   0.008532   4.178 2.94e-05 ***
## str_def.r       2.573977   0.627867   4.100 4.14e-05 ***
## str_acc.r       1.311565   0.568228   2.308 0.020990 *
## td_acc.r        0.246586   0.199762   1.234 0.217052
## td_def.r        0.642828   0.193815   3.317 0.000911 ***
## str_def.b       0.009585   0.573514   0.017 0.986666
## str_acc.b      -0.089157   0.533090  -0.167 0.867177
## td_acc.b        0.727728   0.185291   3.927 8.58e-05 ***
## td_def.b       -0.958857   0.174279  -5.502 3.76e-08 ***
## is_b_southpaw  0.131983   0.173821   0.759 0.447669
## is_b_orthodox  0.201698   0.163164   1.236 0.216397
## is_r_southpaw  0.214007   0.196056   1.092 0.275027
## is_r_orthodox  0.035178   0.186356   0.189 0.850274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6505.0  on 4894  degrees of freedom
## Residual deviance: 5812.6  on 4866  degrees of freedom
## AIC: 5870.6
##
## Number of Fisher Scoring iterations: 4
```

Izračunavamo mjere kvalitete modela.

```r
# Računanje Rsq
Rsq = 1 - logreg_mdl_fighter_details$deviance/logreg_mdl_fighter_details$null.deviance
Rsq
```

```
## [1] 0.1064381
```

```r
yhat <- logreg_mdl_fighter_details$fitted.values >= 0.5
tab <- table(logreg_fighters_data$red_is_winner, yhat)

tab
```

```
##    yhat
##     FALSE TRUE
##   0   754 1110
##   1   437 2594
```

```r
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] 0.6839632
```

```r
precision
```

```
## [1] 0.700324
```

```r
recall
```

```
## [1] 0.8558232
```

```
specificity
```

```
## [1] 0.6330814
```

Iz izračunatih mjera kvalitete naslućujemo da je model lošiji od prijašnjih, ali također i da je bolji od običnog pogađanja.