TATINI SINOVI

SAP PROJEKT

Matija Pintarić, Filip Šimićević, Luka Vukelić, Fran Žužić

SAP projekt

2022-12-19

Prije pokretanja koda savjetujemo instalaciju potrebnih paketa.

```
# Instalacija paketa
#install.packages("dplyr")
#install.packages("xlsx")
#install.packages("readxl")
#install.packages("stringr")
#install.packages("base")
#install.packages("broom")
#install.packages("tidyverse")
#install.packages("readr")
#install.packages("purrr")
#install.packages("nortest")
#install.packages("magrittr")
#install.packages("ggplot2", dependencies = TRUE)
#install.packages("gplots")
```

Učitavamo potrebne biblioteke.

```
library(dplyr)
library(stringr)
library(base)
library(magrittr)
library(xlsx)
library(readxl)
library(tidyverse)
library(gplots)
library(ggplot2)
library(broom)
library(readr)
library(purrr)
library(nortest)
```

Pogledajmo podatke kojima raspolažemo. Stvaramo odgovarajuće dataframes čitajući iz datoteka fighter_details.csv i total_fight_data.csv.

```
fighter_data = read.csv("fighter_details.csv")
dim(fighter_data)
```

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

```
fight_data = read.csv("total_fight_data.csv" ,header = T,sep = ";")
```

Ukupno imamo 3596 registriranih boraca. Za njih je zabilježeno 14 atributa, a oni su:

```
names(fighter_data)
```

```
##  [1] "fighter_name" "Height"       "Weight"       "Reach"        "Stance"
##  [6] "DOB"          "SLpM"         "Str_Acc"      "SApM"         "Str_Def"
```

```
## [11] "TD_Avg"       "TD_Acc"       "TD_Def"       "Sub_Avg"
```

```
View(fighter_data)
```

Raspolažemo i podatcima održanih borbi. Pogledajmo koje atribute sadrži ta tablica.

```
names(fight_data)
```

```
##  [1] "R_fighter"       "B_fighter"       "R_KD"            "B_KD"
##  [5] "R_SIG_STR."      "B_SIG_STR."      "R_SIG_STR_pct"   "B_SIG_STR_pct"
##  [9] "R_TOTAL_STR."    "B_TOTAL_STR."    "R_TD"            "B_TD"
## [13] "R_TD_pct"        "B_TD_pct"        "R_SUB_ATT"       "B_SUB_ATT"
## [17] "R_REV"           "B_REV"           "R_CTRL"          "B_CTRL"
## [21] "R_HEAD"          "B_HEAD"          "R_BODY"          "B_BODY"
## [25] "R_LEG"           "B_LEG"           "R_DISTANCE"      "B_DISTANCE"
## [29] "R_CLINCH"        "B_CLINCH"        "R_GROUND"        "B_GROUND"
## [33] "win_by"          "last_round"      "last_round_time" "Format"
## [37] "Referee"         "date"            "location"        "Fight_type"
## [41] "Winner..."
```

```
View(fighter_data)
```

Zadatak 1. Možemo li očekivati završetak borbe nokautom ovisno o razlici u dužini ruku između boraca?

Kada gledamo UFC borbe uvijek postoji rasprava hoće li određeni atributi boraca ići njima u korist da dobiju protivnika i kakva bi ta pobjeda mogla biti. Jedna od najočitijih razlika uz visinu je doseg ruku, a možda i jedna od bitnijih jer upravo one borcima služe kao primarni izvor napada i obrane. To razmišljanje je motivacija za sljedeće statističko istraživanje.

Prvi korak je micanje nepotrebnih atributa za naše istraživanje.

```
fighter_data = fighter_data[,c('fighter_name','Reach')]
fight_data = fight_data[,c('R_fighter','B_fighter','win_by','Winner...')]
```

Zanimaju nas borci koji imaju zabilježenu duljinu ruku.

```
fighter_data[fighter_data == ''] <- NA
for (col_name in names(fighter_data)){
  if (sum(is.na(fighter_data[,col_name])) > 0){
    cat('Ukupno nedostajućih vrijednosti za varijablu ',col_name, ': ',
        sum(is.na(fighter_data[,col_name])),'\n')
  }
}
```

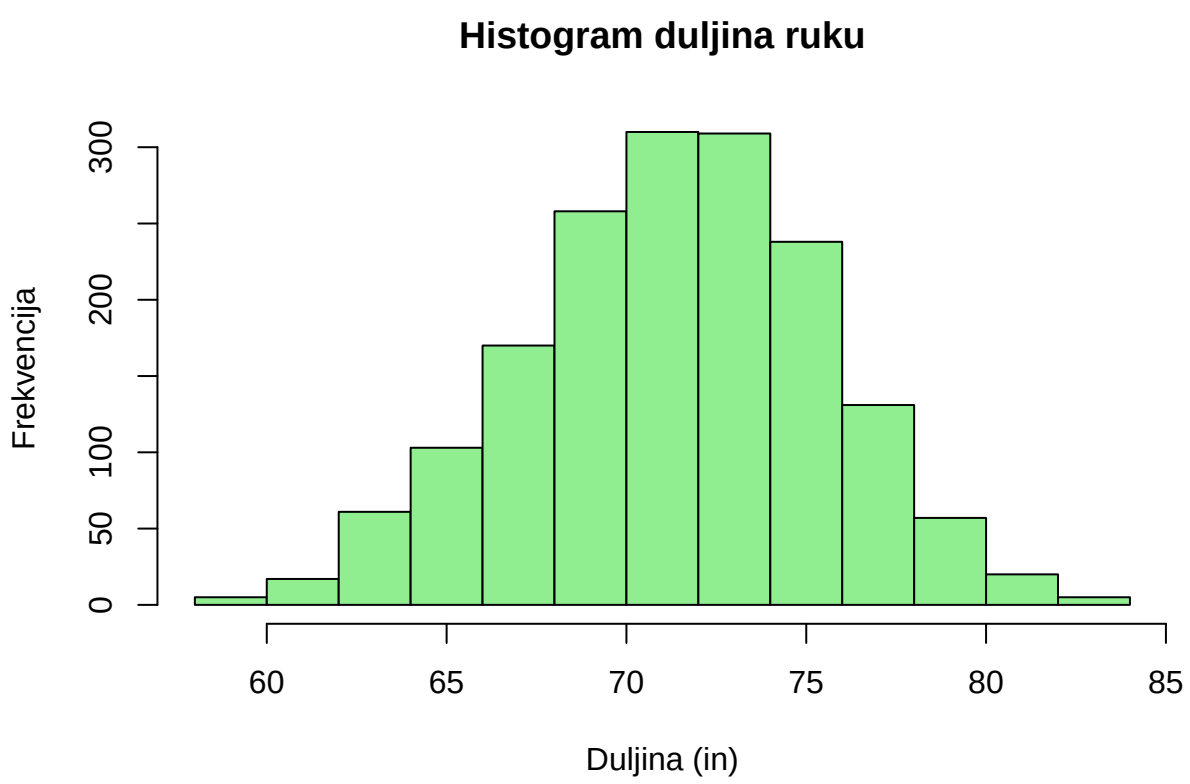
```
## Ukupno nedostajucih vrijednosti za varijablu  Reach :  1912
```

```
#fighter_reach_data <- na.omit(fighter_data$Reach)
fighter_reach_data = fighter_data[!is.na(fighter_data$Reach),]
```

Za početak nas zanima je li duljina ruku normalno distribuirana varijabla, što očekujemo da je.Potvrđujemo sa histogramom.

```
fighter_reach_data <- fighter_reach_data %>%
  mutate_at("Reach", str_replace, "\"", "")

fighter_reach_data[,"Reach"] <- sapply(fighter_reach_data[,"Reach"], as.numeric)

h = hist(fighter_reach_data$Reach,
```

```
        main="Histogram duljina ruku",
        xlab="Duljina (in)",
        ylab="Frekvencija",
        col="lightgreen")
```



Ovako izgledaju varijable objekta s kojim radimo.

```
fighter_reach_data = fighter_reach_data[c("fighter_name", "Reach")]
summary(fighter_reach_data)
```

```
##  fighter_name           Reach
##  Length:1684        Min.   :58.00
##  Class :character   1st Qu.:69.00
##  Mode  :character   Median :72.00
##                     Mean   :71.83
##                     3rd Qu.:75.00
##                     Max.   :84.00
```

Podatke o borbama provjeravamo za vrijednost win_by i redove s praznim vrijednostima izbacujemo. Zatim uzimamo samo atribute koji su potrebni za naš test (2 borca i tip pobjede)

```
fight_data[fight_data == ''] <- NA
for (col_name in names(fight_data)){
  if (sum(is.na(fight_data[,col_name])) > 0){
    cat('Ukupno nedostajućih vrijednosti za varijablu ',col_name, ': ',
        sum(is.na(fight_data[,col_name])),'\n')
  }
}
```

```
## Ukupno nedostajucih vrijednosti za varijablu  Winner... :  104
```

```
fight_data = fight_data[!is.na(fight_data$win_by),]
fight_data = fight_data[!is.na(fight_data$Winner),]
win_data <- fight_data[c("B_fighter", "R_fighter", "win_by")]
```

Moramo za svaku borbu u kojoj znamo raspon ruku oba borca grupirati razliku raspona u 2 grupe. Onu u kojoj je borba završila sa KO/TKO i u onu koju nije.

```
dif_KO1 = merge(win_data, fighter_reach_data, by.x = "B_fighter", by.y="fighter_name")
dif_KO = merge(dif_KO1, fighter_reach_data, by.x = "R_fighter", by.y="fighter_name")
dif_KO = dif_KO %>% filter(win_by == "KO/TKO")
dif_KO$dif = dif_KO$Reach.x - dif_KO$Reach.y

dif_else1 = merge(win_data, fighter_reach_data, by.x = "B_fighter", by.y="fighter_name")
dif_else = merge(dif_else1, fighter_reach_data, by.x = "R_fighter", by.y="fighter_name")
dif_else = dif_else %>% filter(win_by != "KO/TKO")
dif_else$dif = dif_else$Reach.x - dif_else$Reach.y
```

```
mean_KO <- mean(dif_KO$dif)
mean_else <- mean(dif_else$dif)
cat("Srednja vrijednost razlike raspona ruku ako borba","\n",
    "nije završila nokautom na temelju uzorka:", mean_else)
```

```
## Srednja vrijednost razlike raspona ruku ako borba
##  nije završila nokautom na temelju uzorka: 0.0005979073
```

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

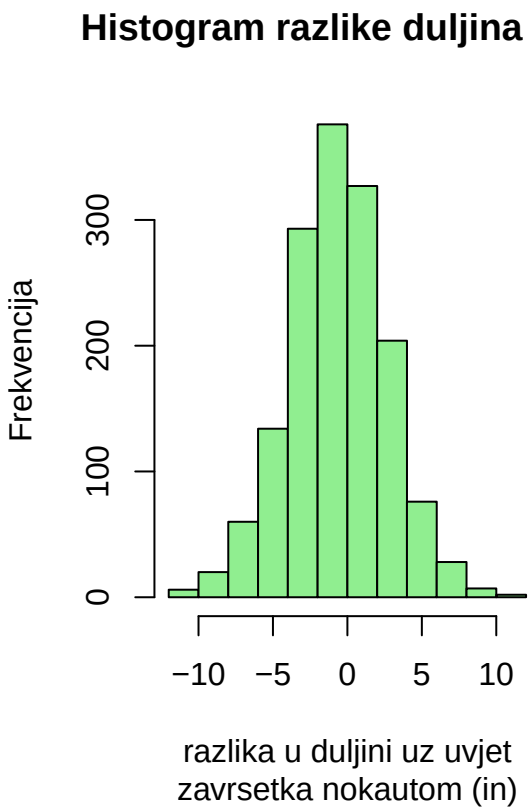
```
cat("Srednja vrijednost razlike raspona ruku ako je borba","\n",
    "završila nokautom na temelju uzorka:",mean_KO)
```

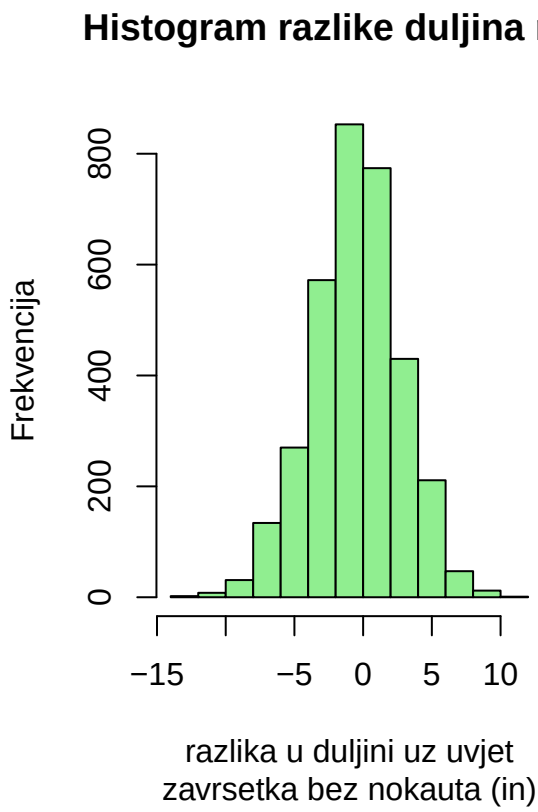
```
## Srednja vrijednost razlike raspona ruku ako je borba
##  završila nokautom na temelju uzorka: -0.1435095
```

Da bi smo mogli koristiti t-test kako bi analizirali razliku u srednjim vrijednostima naše dvije skupine prvo moramo provjeriti jesu li normalno distribuirane i je li F test zadovoljava jednakost o varijancama. Dokazivanjem da je raspon ruku normalno distribuiran možemo zaključiti da će i razlika dva raspona ruku ponovo biti slučajna varijabla koja podliježe normalnoj distribuciji.

To se i vidi na histogramima u nastavku.

```
par(mfrow=c(1, 2))
h = hist(dif_KO$dif,
         main="Histogram razlike duljina ruku",
         xlab=strwrap("razlika u duljini uz uvjet zavrsetka nokautom (in)", width=35),
         ylab="Frekvencija",
         col="lightgreen")

h = hist(dif_else$dif,
         main="Histogram razlike duljina ruku",
         xlab=strwrap("razlika u duljini uz uvjet zavrsetka bez nokauta (in)", width=35),
         ylab="Frekvencija",
         col="lightgreen")
```





Sada postavljamo hipoteze o jednakosti varijanci za provođenje F-testa. H0 - varijance su jednake H1 - varijance nisu jednake

Provodimo test o jednakosti varijance kako bi znali koju vrstu t testa koristiti i na temelju dobivenih podataka zaključujemo da na razini značajnosti od 5% ne možemo odbaciti hipotezu da su varijance ova dva skupa jednake.

```
var.test(dif_KO$dif, dif_else$dif)
```

```
##
##  F test to compare two variances
##
## data:  dif_KO$dif and dif_else$dif
## F = 1.0521, num df = 1532, denom df = 3344, p-value = 0.241
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9664972 1.1468285
## sample estimates:
## ratio of variances
##           1.052084
```

Sada možemo provesti t-test. Formiramo hipoteze. H0 - nema razlike u srednjim vrijednostima H1 - postoji razlika u srednjim vrijednostima

```
t.test(dif_KO$dif, dif_else$dif,var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
```

```
## data:  dif_KO$dif and dif_else$dif
## t = -1.4355, df = 4876, p-value = 0.1512
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.34091321  0.05269848
## sample estimates:
##     mean of x     mean of y
## -0.1435094586  0.0005979073
```

Vidimo da je p vrijednost veća od 5% što znači da ovaj slučaj nema statističku važnost. Na razini 5% zaključujemo da ne možemo odbaciti hipotezu H0 te ju prihvaćamo da nema razlika u srednjim vrijednostima. Iz toga naknadno zaključujemo da ne možemo očekivati nokaut ovisno o razlici duljina ruku.

Zadatak 2. Razlikuje li se trajanje mečeva (u sekundama) izmedu pojedinih kategorija?

Kategorije UFC boraca dijele se, ovisno o kilaži, na 9 osnovnih (Strawweight, Flyweight, Bantamweight, Featherweight, Lightweight, Welterweight, Middleweight, Light Heavyweight i Heavyweight). Super-weight kategorije i Cruiserweight kategoriju smo izostavili da uštedimo na kompleksnosti samih podataka koje promatramo. Ako uzmemo za usporedbu Strawweight kategoriju u kojoj se borci natječu do 52.2 kg i Heavyweight koja je do 120.2 kg, možemo si postaviti pitanje traju li borbe tih dviju kategorija u prosjeku jednako. Razlika u fizičkim sposobnostima nam tu izgleda poprilično utjecajna. Traju li općenito svi mečevi približno jednako ili se neki razlikuju? Na takva ćemo pitanja pokušati odgovoriti ANOVA testom te analizama koje su samom testu usko vezane.

#1.Učitavanje podataka

```
total_fight_data_xlsx = suppressWarnings(read_excel("total_fight_data.xlsx"))

df = data.frame(total_fight_data_xlsx)
```

Fight_type stupac je procesiran stupac koji opisuje koje je vrste borba te ćemo iz njega izvući kategoriju borbe. Ukupnu sumu sekundi meča izvesti ćemo zbrajanjem sekundi iz svake runde.

#Sređivanje podataka

```
#Promotrit ćemo kako su strukturirane borbe po rundama
df$extracted <- str_extract(df$Format, "\\(([^)]+)\\)")
```

Trajanje runde je uvijek ograničeno na pet minuta.

```
sum(is.na(df[,"last_round_time"]))
```

```
## [1] 0
```

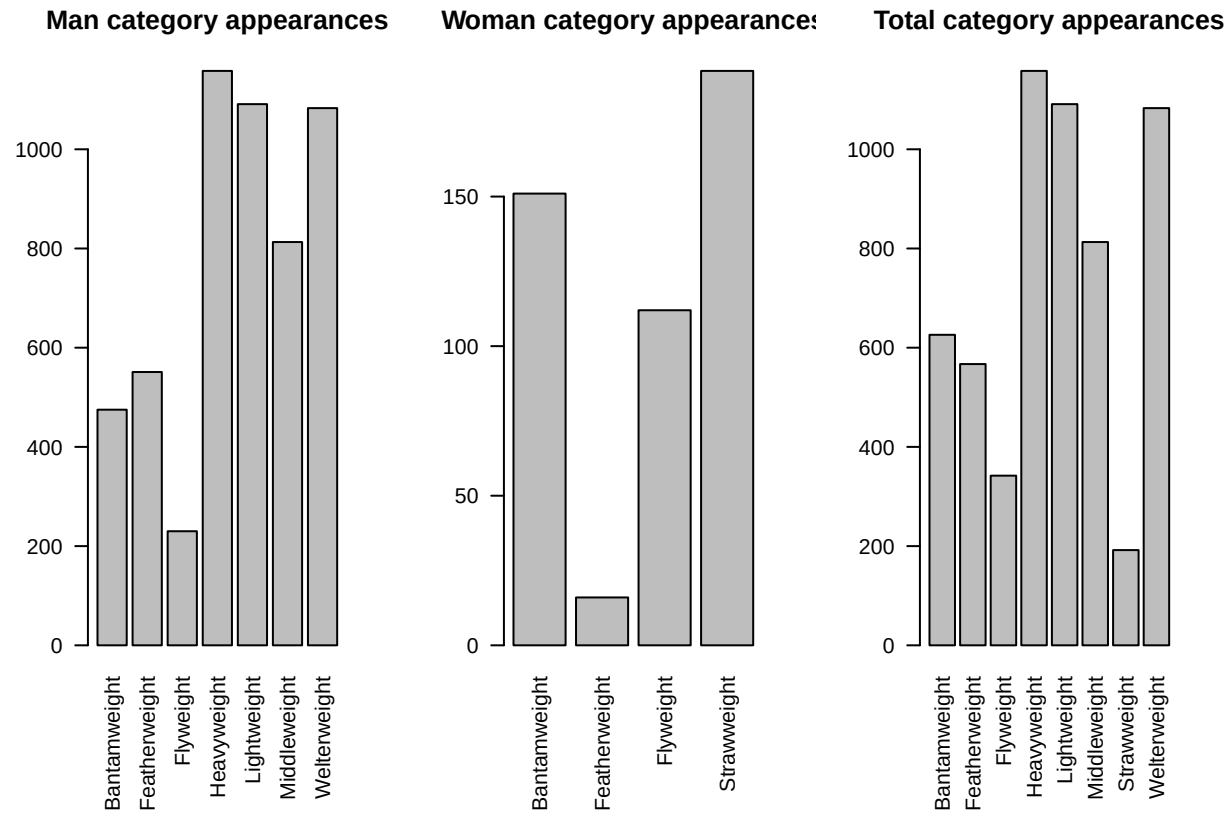
```
last_round_seconds <- rep(0, nrow(df))
df <- cbind(df, last_round_seconds)

for(index in seq_along(df)){
time <- strptime(df[index, "last_round_time"], "%M:%S")
seconds <- time$min * 60 + time$sec
df[index, "last_round_seconds"] = seconds
}
df$last_round_seconds <- as.double(df$last_round_seconds)
df <- df %>%
  mutate(seconds_in_fight = (last_round - 1) * 5 * 60 + last_round_seconds)
```

Stvorili smo stupac koji opisuje ukupan broj sekundi trajanja borbe. Sljedeće šta sređujemo je stupac Fight_type.

```r
# u arr polje stavljamo sve moguće kategorije u UFC-u,
# osim Super-weight kategorija i Cruiserweighta

arr <- c("Strawweight", "Flyweight", "Bantamweight", "Featherweight",
         "Lightweight", "Welterweight", "Middleweight", "Light Heavyweight", "Heavyweight")

df$Gender[str_detect(df$Fight_type, "Women's")]  <- "Woman"
df$Gender[!str_detect(df$Fight_type, "Women's")]  <- "Man"

for (category in arr) {
df$Fight_type[str_detect(df$Fight_type, category)] <- category
}

df = df %>% filter(!str_detect(Fight_type, "Catch"))
df = df %>% filter(!str_detect(Fight_type, "Open"))
df = df %>% filter(!str_detect(Fight_type, "Tournament"))
df = df %>% filter(!str_detect(Fight_type, "Championship"))
```

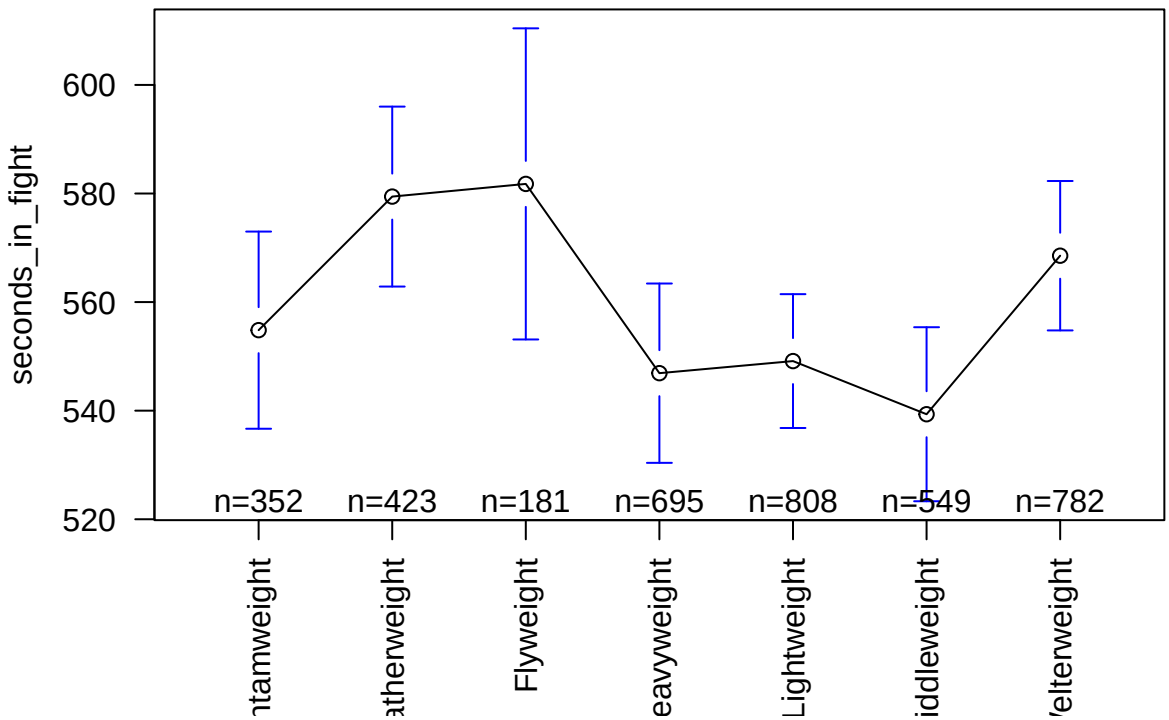
#3.Generalna vizualizacija podataka

```r
par(mar = c(7,4,4,2) + 0.1)
par(mfrow = c(1, 3))

barplot(table(df$Fight_type[df$Gender=="Man"]),
        las=2,cex.names=1,
        main='Man category appearances')
barplot(table(df$Fight_type[df$Gender=="Woman"]),
        las=2,cex.names=1,
        main='Woman category appearances')
barplot(table(df$Fight_type),
        las=2,cex.names=1,
        main='Total category appearances')
```



Puno je manje ženskih borbi te u ženskom UFC-u postoje samo 4 kategorije.

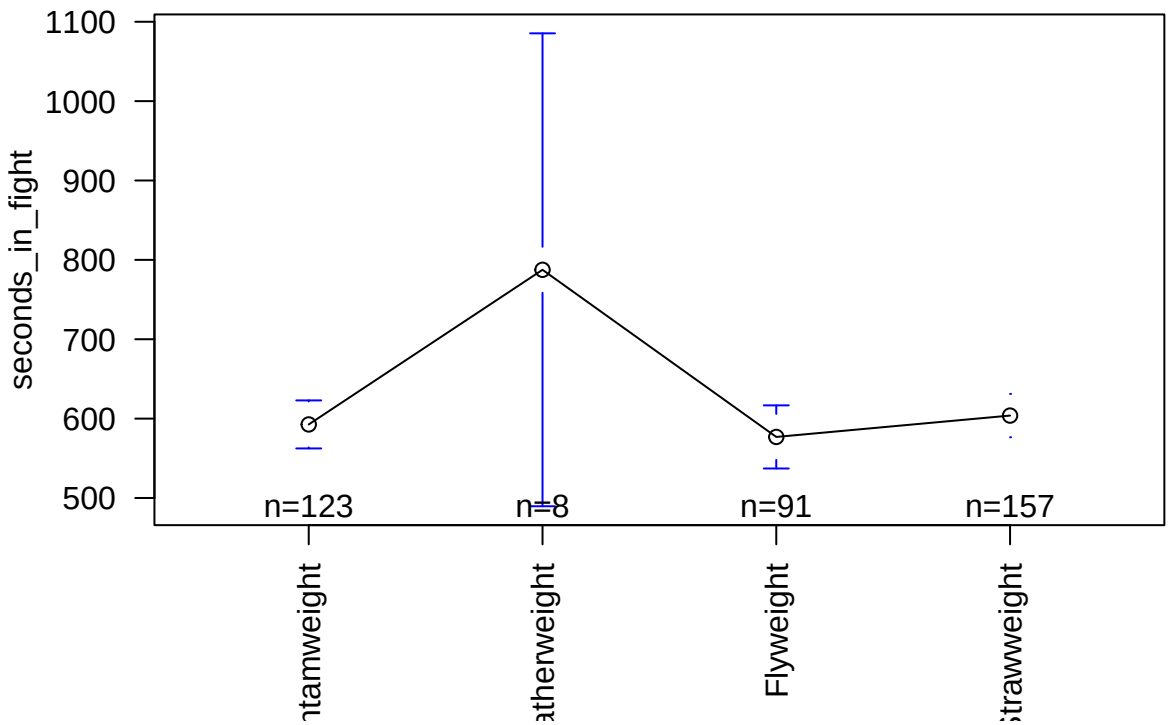
```
#Brisanje borbi bez zapisa o vremenu trajanja

df.man <- filter(df, Gender == "Man")
df.man <- filter(df.man, seconds_in_fight != 0)
row_indexes <- complete.cases(df.man[, c("Fight_type", "seconds_in_fight")])
df.man <- df.man[row_indexes, ]

df.woman <- filter(df, Gender == "Woman")
df.woman <- filter(df.woman, seconds_in_fight != 0)
row_indexes <- complete.cases(df.woman[, c("Fight_type", "seconds_in_fight")])
df.woman <- df.woman[row_indexes, ]

plotmeans(seconds_in_fight~Fight_type, data = df.man, las = 2, xlab = '',
          main = "Average seconds per man fight compared for every category")
```



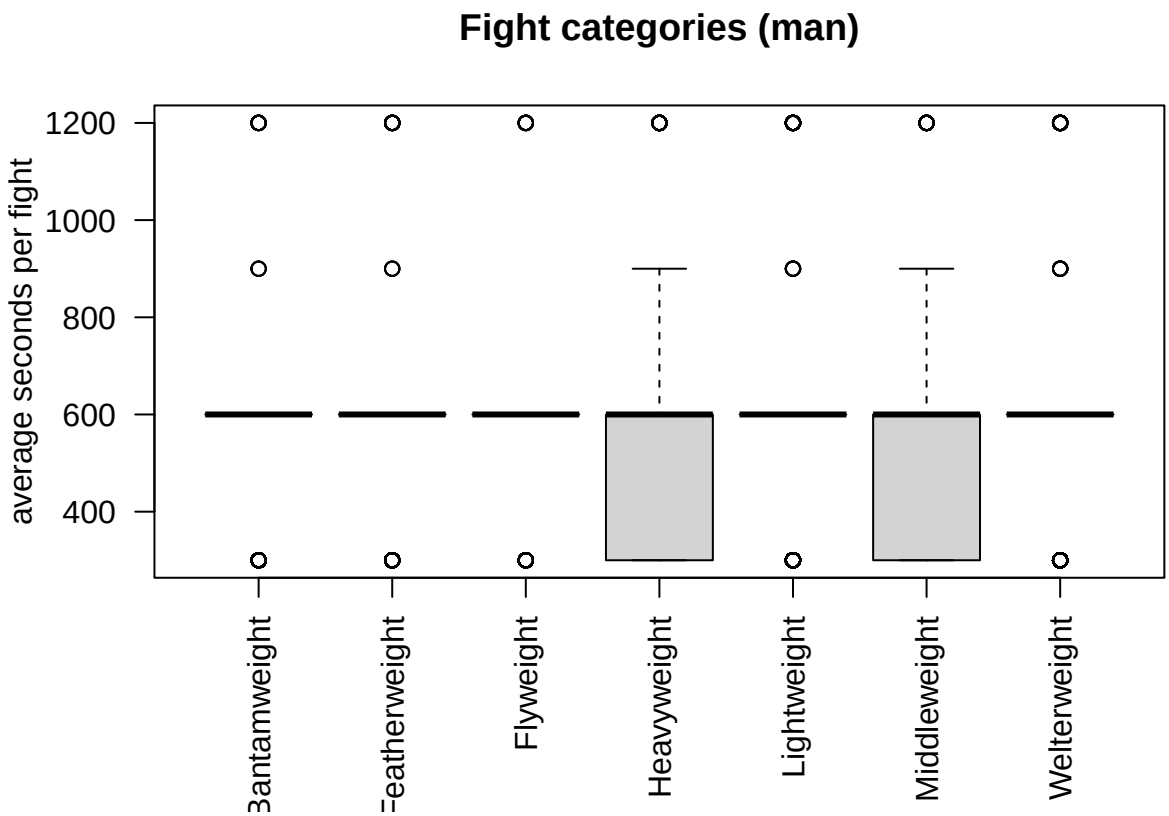

```
plotmeans(seconds_in_fight~Fight_type, data = df.woman, las = 2, xlab = '',
          main = "Average seconds per woman fight compared for every category")
```

```
## Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped
```

```
## Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped
```




```
par(mar = c(6,4,4,2) + 0.1)
boxplot(seconds_in_fight ~ Fight_type,data=df.man,
        las = 2, cex.names = 0.5,
        xlab="", ylab="average seconds per fight",
        main = "Fight categories (man)")
```



Ovakav box plot nije priželjkivan, ali ćemo ga ostaviti da prikazuje podatke koje smo dobili. Iz njega možemo iščitati da je median svake grupe 600 te je najčešći slučaj da je borba završena nakon druge runde.

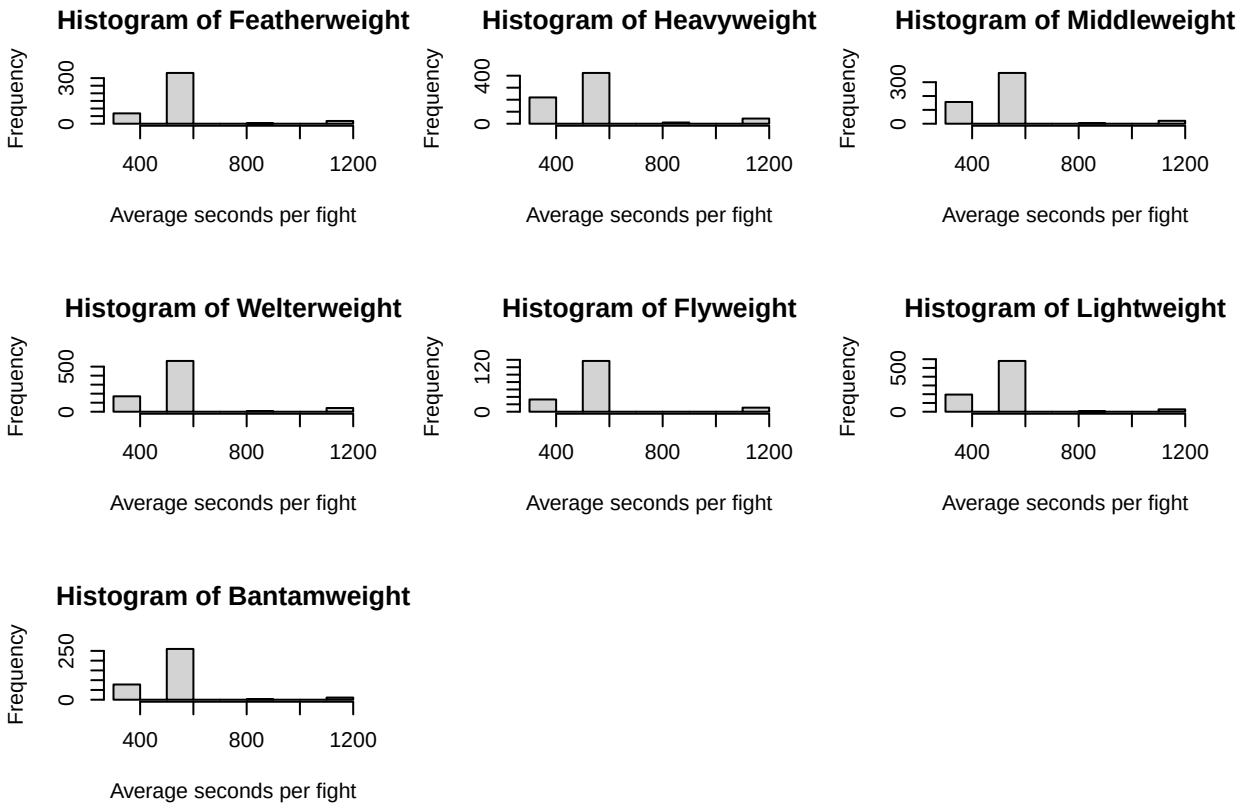
#4.Provedba testova

#4.1 Trajanje borbi kod muškaraca ovisno o kategoriji

```
#Provjera normalnosti zbog pretpostavke
require(nortest)
lillie.test(df.man$seconds_in_fight)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  df.man$seconds_in_fight
## D = 0.35744, p-value < 2.2e-16
```

```
par(mfrow = c(3, 3)) # Specify layout with 1 row and 3 columns
for (category in unique(df.man$Fight_type)){
  hist(df.man$seconds_in_fight[df.man$Fight_type==category],
       main = paste("Histogram of" , category),
       xlab = "Average seconds per fight")
}
```




```
#Provjera homogenosti varijance zbog pretpostavke
bartlett.test(df.man$seconds_in_fight ~ df.man$Fight_type)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  df.man$seconds_in_fight by df.man$Fight_type
## Bartlett's K-squared = 55.825, df = 6, p-value = 3.157e-10
```

```
var((df.man$seconds_in_fight[df.man$Fight_type=='Heavyweight']))
```

```
## [1] 49179.9
```

Ne možemo provesti ANOVA test jer nam kategorije nemaju normalnu distribuciju niti homogenost varijance (p-value testova je značajno manji od 0.05). Provjerom normalnosti koja odbacuje samu normalnost zaključujemo da moramo korisiti neparametarske testove. Koristit ćemo Kruskal-Wallisov test. Unatoč neuspješnom dokazivanju normalnosti provest ćemo ANOVA test kako bismo razmotrili rezultat.

```
anova = aov(seconds_in_fight~Fight_type, data = df.man)
summary(anova)
```

```
##               Df    Sum Sq Mean Sq F value  Pr(>F)
## Fight_type     6    721496  120249    3.25 0.00345 **
## Residuals   3783 139979232   37002
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
kruskal.test(df.man$seconds_in_fight, df.man$Fight_type)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  df.man$seconds_in_fight and df.man$Fight_type
## Kruskal-Wallis chi-squared = 32.39, df = 6, p-value = 1.373e-05
```

#4.2 Trajanje borbi kod žena ovisno o kategoriji

Featherweight borbi kod žena ima samo 9 pa ćemo shodno tome njih isključiti iz analize.

```
df.woman <- filter(df.woman, Fight_type != "Featherweight")
lillie.test(df.woman$seconds_in_fight[df.woman$Fight_type == "Strawweight"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  df.woman$seconds_in_fight[df.woman$Fight_type == "Strawweight"]
## D = 0.45148, p-value < 2.2e-16
```

```
lillie.test(df.woman$seconds_in_fight[df.woman$Fight_type == "Bantamweight"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  df.woman$seconds_in_fight[df.woman$Fight_type == "Bantamweight"]
## D = 0.434, p-value < 2.2e-16
```

```
lillie.test(df.woman$seconds_in_fight[df.woman$Fight_type == "Flyweight"])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  df.woman$seconds_in_fight[df.woman$Fight_type == "Flyweight"]
## D = 0.39696, p-value < 2.2e-16
```

Distribucija, kao i kod muškaraca, nije normalna te koristimo neparametarski test.

```
kruskal.test(df.woman$seconds_in_fight, df.woman$Fight_type)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  df.woman$seconds_in_fight and df.woman$Fight_type
## Kruskal-Wallis chi-squared = 2.5917, df = 2, p-value = 0.2737
```

Test pokazuje da je razlika u trajanju borbi ovisno o kategoriji jednaka tj. da ne možemo odbaciti nultu hipotezu (p-value » 0.05).

#5.Dodatna provedba testa i zaključak Promatranjem podataka zaključuje se da se borbe dijele na one koje traju 3 runde,jedne koje traju 5 rundi te još par vrsti koje imaju kombinaciju u strukturi samih rundi npr. “1 Rnd + 2OT (15-3-3)” “1 Rnd (12)”. Taj faktor nam može biti utjecajan pri provedbi testa te ispitivanja distribucije tako da ćemo dodatno promatrati samo one borbe koje traju 3 runde.

```
df.additional <- filter(df.man, Format == "3 Rnd (5-5-5)")
lillie.test(df.additional$seconds_in_fight)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  df.additional$seconds_in_fight
```

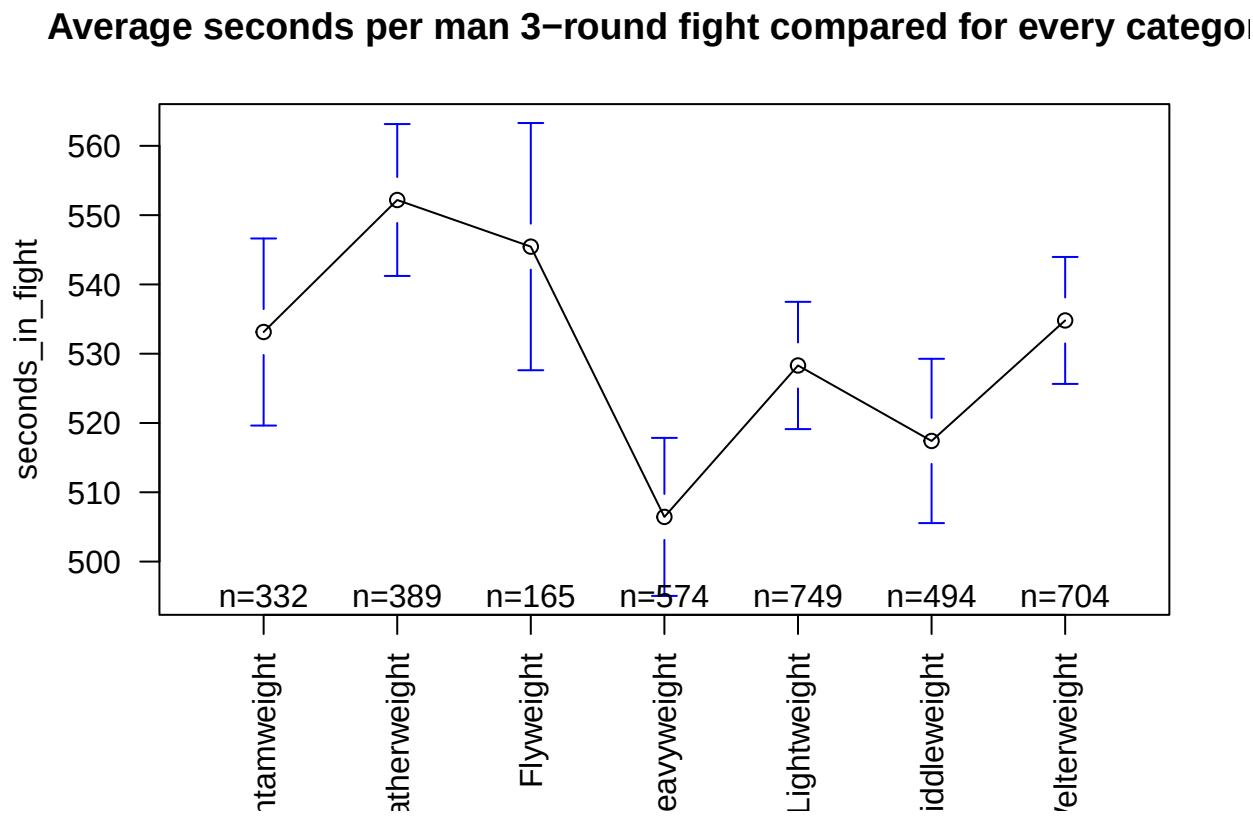
```
## D = 0.47355, p-value < 2.2e-16
```

```
kruskal.test(df.additional$seconds_in_fight, df.additional$Fight_type)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  df.additional$seconds_in_fight and df.additional$Fight_type
## Kruskal-Wallis chi-squared = 39.17, df = 6, p-value = 6.628e-07
```

Neparametarski test nam govori da postoji razlika u trajanju borbi između kategorija.

```
plotmeans(seconds_in_fight~Fight_type, data = df.additional, las = 2, xlab = '',
          main = "Average seconds per man 3-round fight compared for every category")
```




```
df.additional <- filter(df.additional, Fight_type != "Featherweight")
df.additional <- filter(df.additional, Fight_type != "Flyweight")
df.additional <- filter(df.additional, Fight_type != "Heavyweight")
kruskal.test(df.additional$seconds_in_fight, df.additional$Fight_type)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  df.additional$seconds_in_fight and df.additional$Fight_type
## Kruskal-Wallis chi-squared = 5.871, df = 3, p-value = 0.1181
```

Kada izbacimo kategorije koje nam 'iskaču' na grafu te usporedimo četiri kategorije (Bantamweight, Lightweight, Middleweight i Welterweight) i provedemo Kruskal-Wallisov test dobivamo p-vrijednost veću od 0.05 te dokazujemo da su nam razlike trajanja u sekundama približno jednake. Iz grafa vidimo da

Featherweight i Flyweight kod muškaraca traju duže dok Heavyweight borbe traju najkraće.

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

Moderni UFC razlikuje dva formata borbi: 3-Rnd (5-5-5) format od tri runde koji se koristi za obične borbe te 5-Rnd (5-5-5-5-5) format od 5 rundi koji se koristi za borbe za titulu. Naravno, bilo je situacija kada se 5-Rnd format koristio za obične borbe npr. UFC 202 borba Conor McGregora i Nate Diaza. Činjenica da borbe za titulu imaju prostora trajati dulje zbog formata nas motivira ispitati je li to istina. S jedne strane moramo biti svjesni činjenice da možda nećemo dobiti realnu sliku odnosa trajanja borbi jer su se mnoge 3-Rnd borbe koje su završile na odluku vjerojatno mogle produljiti. Iz tog razloga prvo ćemo usporediti trajanja u rundama neovisno o formatu, a zatim ćemo uspoređivati isključivo 5-Rnd borbe za titulu i 5-Rnd obične borbe jer ih ima podjednako.

Učitavamo podatke za sve borbe. Potom iz tih podataka stvaramo dva nova okvira od kojih jedan sadrži podatke za sve borbe za titulu, a drugi sadrži podatke za borbe koje nisu za titulu.

```
fight_data_task3 = suppressWarnings(read_excel("total_fight_data.xlsx"))

df_fight_data_task3 = data.frame(fight_data_task3);
df_fight_data_task3 <- subset(df_fight_data_task3,
                              select = c(R_fighter, B_fighter,
                                         last_round, Format, Fight_type))

# izdvajamo veliki dataset u dva manja radi lakšeg snalaženja
df_task3_titleFights = df_fight_data_task3 %>% filter(str_detect(Fight_type, "Title"))
df_task3_nonTitleFights = df_fight_data_task3 %>% filter(!str_detect(Fight_type, "Title"))

df_fight_data_task3$Fight_type[
  str_detect(df_fight_data_task3$Fight_type, "Title")] <- "Title Bout"

df_fight_data_task3$Fight_type[
  !str_detect(df_fight_data_task3$Fight_type, "Title")] <- "Normal Bout"
```

Vidimo da borbi za titulu ima značajno manje nego običnih borbi.

```
print(paste("Broj normalnih non-title borbi je: ", nrow(df_task3_nonTitleFights)))
```

```
## [1] "Broj normalnih non-title borbi je:  5647"
```

```
print(paste("Broj borbi za titulu je: ", nrow(df_task3_titleFights)))
```

```
## [1] "Broj borbi za titulu je:  365"
```

Cilj nam je provjeriti traju li borbe za titulu duže od ostalih borbi u natjecanju. Za navedeno istraživanje postavljamo kontingencijsku tablicu koja za stupce ima kategorijsku varijablu "Title bout" tj. "Bout".

Provjerimo prije svega koje formate borbi imamo u našem datasetu.

```
table(df_task3_nonTitleFights$Format)
```

```
##
##           1 Rnd (10)           1 Rnd (12)           1 Rnd (15)
##                    6                    4                    8
##           1 Rnd (18)           1 Rnd (20) 1 Rnd + 2OT (15-3-3)
##                    2                   21                    4
## 1 Rnd + 2OT (24-3-3)    1 Rnd + OT (12-3)    1 Rnd + OT (15-3)
##                    1                   74                    2
```

```
##    1 Rnd + OT (30-3)    1 Rnd + OT (30-5)    1 Rnd + OT (31-5)
##                    1                    1                    1
##          2 Rnd (5-5)        3 Rnd (5-5-5) 3 Rnd + OT (5-5-5-5)
##                   14                 5252                    2
##    5 Rnd (5-5-5-5-5)        No Time Limit
##                  228                   26
```

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

```
table(df_task3_titleFights$Format)
```

```
##
##           1 Rnd (30) 1 Rnd + 2OT (15-3-3)    1 Rnd + OT (12-3)
##                    1                   16                    6
##    1 Rnd + OT (27-3)    1 Rnd + OT (30-5)        3 Rnd (5-5-5)
##                    1                    2                   37
## 3 Rnd + OT (5-5-5-5)    5 Rnd (5-5-5-5-5)        No Time Limit
##                   20                  279                    3
```

Uočavamo da postoji dosta formata s izrazito malo održanih borbi. To su prastari formati koji se više ne koriste npr. 1 Rnd (12) format koji je korišten za samo 4 obične borbe. Takvi formati mogli bi utjecati na naše rezultate s obzirom da su limitirani na trajanje od samo jedne runde, a ona je trajala npr. 10 ili 20 minuta. Takve borbe mogle bi nam dati krivu sliku jer imaju određenu težinu u ovakvom istraživanju. Dodatna motivacija za eliminaciju borbi s prastarim formatima je ta što želimo da ovo istraživanje bude primjenjivo za budućnost UFC-a koji danas koristi isključivo 5-Rnd i 3-Rnd format. Imamo sreću što nećemo eliminirati veliki broj podataka. Potaknuti ovim zaključcima, modificiramo okvire da sadrže samo 3-Rnd i 5-Rnd formate koje ćemo potom uspoređivati.

```
df_fight_data_task3['Format'] <- sapply(df_fight_data_task3['Format'], as.character);

df_fight_data_task3 = df_fight_data_task3 %>%
  filter(grepl("5 Rnd (5-5-5-5-5)", Format, fixed = TRUE) |
         grepl("3 Rnd (5-5-5)",Format, fixed = TRUE))
```

Naši podaci su sada smanjeni s 6012 redaka na 5796, maknuli smo samo 3.5% podataka.

Pogledajmo sada koja sve trajanja natjecanja (u rundama) imamo za obje kategorije borbi.

```
levels(factor(df_task3_titleFights$last_round))
```

```
## [1] "1" "2" "3" "4" "5"
```

```
levels(factor(df_task3_nonTitleFights$last_round))
```

```
## [1] "1" "2" "3" "4" "5"
```

Vidimo da su u oba dataseta prisutna identična trajanja koja će u našem slučaju odgovarati redcima kontingencijske tablice. Kako bi si olakšali rad s podacima, pretvaramo podatke "last_round" u character format.

```
df_task3_titleFights['last_round'] <- sapply(df_task3_titleFights['last_round'],
                                             as.character)

df_task3_nonTitleFights['last_round'] <- sapply(df_task3_nonTitleFights['last_round'],
                                             as.character)

df_fight_data_task3['last_round'] <- sapply(df_fight_data_task3['last_round'],
                                            as.character)
```

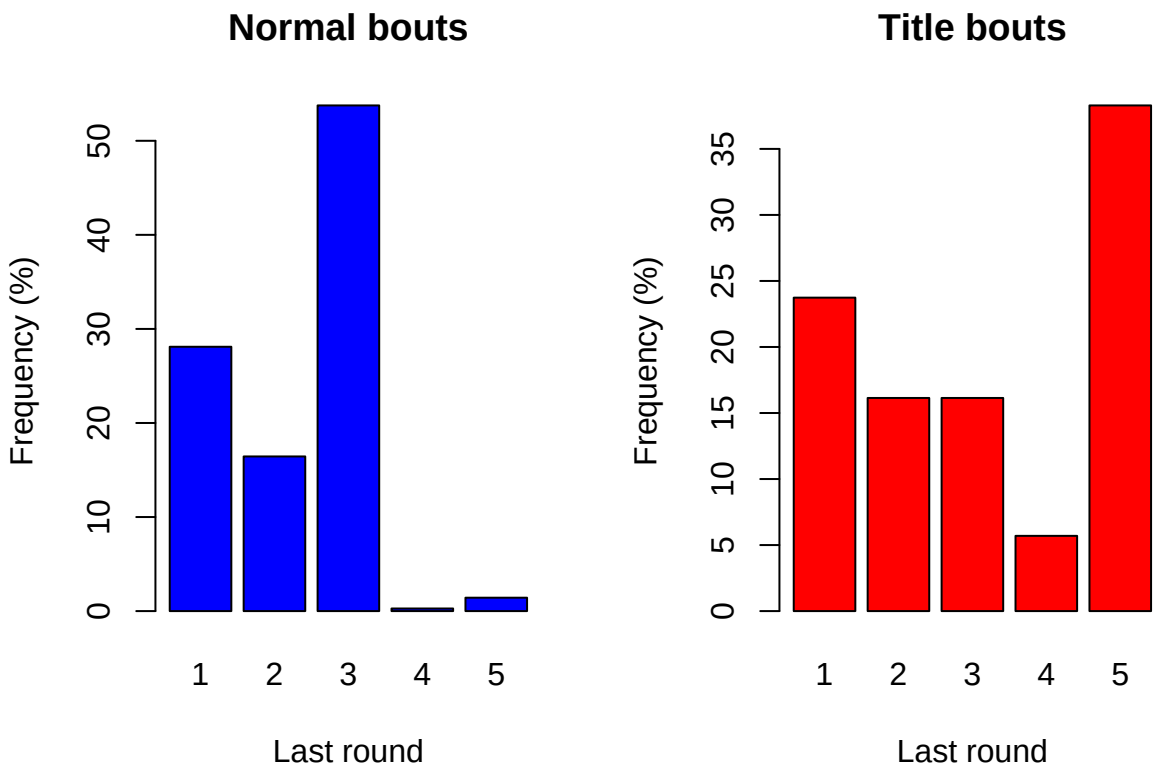
Kreiramo kontingencijsku tablicu na temelju podataka.

```
tbl = table(df_fight_data_task3[df_fight_data_task3$last_round == "1" |
                                df_fight_data_task3$last_round == "2" |
                                df_fight_data_task3$last_round == "3" |
                                df_fight_data_task3$last_round == "4" |
                                df_fight_data_task3$last_round == "5",
                               ]$last_round,

            df_fight_data_task3[df_fight_data_task3$last_round == "1" |
                                df_fight_data_task3$last_round == "2" |
                                df_fight_data_task3$last_round == "3" |
                                df_fight_data_task3$last_round == "4" |
                                df_fight_data_task3$last_round == "5",
                               ]$Fight_type)
tbl
```

```
##
##     Normal Bout Title Bout
##   1        1540         75
##   2         901         51
##   3        2946         51
##   4          15         18
##   5          78        121
```

Dobra praksa je vizualizirati kategorijske podatke. Jedan on načina je da prikažemo odnose trajanja pomoću bar plota.

```
counts1 <- df_fight_data_task3 %>%
  filter(Fight_type == "Normal Bout") %>% count(last_round)

counts2 <- df_fight_data_task3 %>%
  filter(Fight_type == "Title Bout") %>% count(last_round)

counts1$pct <- counts1$n / sum(counts1$n) * 100
counts2$pct <- counts2$n / sum(counts2$n) * 100
par(mfrow=c(1, 2))

barplot(counts1$pct,
        names.arg=counts1$last_round,
        col="blue",
        main="Normal bouts",
        xlab="Last round",
        ylab="Frequency (%)",
        #legend=unique(counts$Fight_type)
        )

barplot(counts2$pct,
        names.arg=counts2$last_round,
        col="red",
        main="Title bouts",
        xlab="Last round",
        ylab="Frequency (%)",
        #legend=unique(counts$Fight_type)
        )
```



Na prvi pogled uočavamo da je u normalnim borbama najčešći slučaj da borba traje 3 runde. U slučajevima borbe za titulu najviše borbi završi u petoj rundi.

Tablici na kraju dodajemo margine.

```
tbl_margins = addmargins(tbl)
tbl_margins
```

```
##
##       Normal Bout Title Bout  Sum
##   1          1540         75 1615
##   2           901         51  952
##   3          2946         51 2997
##   4            15         18   33
##   5            78        121  199
##   Sum        5480        316 5796
```

Test nezavisnosti χ^2 test u programskom paketu R implementiran je u funkciji `chisq.test()` koja kao ulaz prima kontingencijsku tablicu podataka koje testiramo na nezavisnost. Ispitujemo nezavisnost trajanja borbe (u rundama) o vrsti borbe.

Pretpostavka testa je da očekivana frekvencija pojedinog razreda mora biti veća ili jednaka 5 (`chisq.test()` pretpostavlja da je ovaj uvjet zadovoljen stoga je prije provođenja testa potrebno to provjeriti).

```
for (col_names in colnames(tbl_margins)){
  for (row_names in rownames(tbl_margins)){
    if (!(row_names == 'Sum' | col_names == 'Sum') ){
      cat('Očekivane frekvencije za razred ',col_names,'-',row_names,': ',
          (tbl_margins[row_names,'Sum'] *
```

```
          tbl_margins['Sum',col_names]) /
          tbl_margins['Sum','Sum'],'\n')
    }
  }
}
```

```
## Ocekivane frekvencije za razred  Normal Bout - 1 :  1526.95
## Ocekivane frekvencije za razred  Normal Bout - 2 :  900.0966
## Ocekivane frekvencije za razred  Normal Bout - 3 :  2833.602
## Ocekivane frekvencije za razred  Normal Bout - 4 :  31.20083
## Ocekivane frekvencije za razred  Normal Bout - 5 :  188.1504
## Ocekivane frekvencije za razred  Title Bout - 1 :  88.05038
## Ocekivane frekvencije za razred  Title Bout - 2 :  51.90338
## Ocekivane frekvencije za razred  Title Bout - 3 :  163.3975
## Ocekivane frekvencije za razred  Title Bout - 4 :  1.799172
## Ocekivane frekvencije za razred  Title Bout - 5 :  10.84955
```

Uočavamo da nažalost pretpostavka testa nije ispunjena. Za slučaj "Title bout - 4" uočavamo da je očekivana frekvencija 1.799172 što je manje od 5. U ovakvoj situaciji ne možemo primjeniti χ^2 test nezavisnosti, ali nude nam se dva rješenja.

1. rješenje - Možemo spajati dvije kategorije, u ovom slučaju pametan odabir bi bio spojiti borbe koje traju 4 i 5 rundi u jedno trajanje zvano "4-5". Motivacija za takvo spajanje je činjenica da su formati borbe ili 3-round ili 5-round, stoga ako je borba trajala dulje od 3 runde, spada u format 5 round.
2. rješenje - Možemo umjesto χ^2 testa nezavisnosti koristiti Fisherov egzaktni test, koji je neparametarski i služi istoj svrsi kao i χ^2 test nezavisnosti.

Prvo ćemo pokazati rješenje pod 1. Moramo prvo modificirati vrijednosti gdje borbe traju 4 ili 5 rundi u "4-5"

```
dfCombined_fight_data_task3 = data.frame(df_fight_data_task3)

dfCombined_fight_data_task3$last_round[
  str_detect(dfCombined_fight_data_task3$last_round, "4") |
  str_detect(dfCombined_fight_data_task3$last_round, "5")] <- "4-5"
```

Ponovo stvaramo tablicu.

```
tbl_combined = table(dfCombined_fight_data_task3[
                      dfCombined_fight_data_task3$last_round == "1" |
                      dfCombined_fight_data_task3$last_round == "2" |
                      dfCombined_fight_data_task3$last_round == "3" |
                      dfCombined_fight_data_task3$last_round == "4-5",
                      ]$last_round,

                     dfCombined_fight_data_task3[
                      dfCombined_fight_data_task3$last_round == "1" |
                      dfCombined_fight_data_task3$last_round == "2" |
                      dfCombined_fight_data_task3$last_round == "3" |
                      dfCombined_fight_data_task3$last_round == "4-5",
                      ]$Fight_type)

tbl_combined_margins = addmargins(tbl_combined)
tbl_combined_margins
```

```
##
```

```
##       Normal Bout Title Bout  Sum
##   1          1540         75 1615
##   2           901         51  952
##   3          2946         51 2997
##   4-5          93        139  232
##   Sum        5480        316 5796
```

Ponavljamo provjeru pretpostavke testa.

```
for (col_names in colnames(tbl_combined_margins)){
  for (row_names in rownames(tbl_combined_margins)){
    if (!(row_names == 'Sum' | col_names == 'Sum') ){
      cat('Očekivane frekvencije za razred ',col_names,'-',row_names,': ',
          (tbl_combined_margins[row_names,'Sum'] *
           tbl_combined_margins['Sum',col_names]) /
           tbl_combined_margins['Sum','Sum'],'\n')
    }
  }
}
```

```
## Ocekivane frekvencije za razred  Normal Bout - 1 :  1526.95
## Ocekivane frekvencije za razred  Normal Bout - 2 :  900.0966
## Ocekivane frekvencije za razred  Normal Bout - 3 :  2833.602
## Ocekivane frekvencije za razred  Normal Bout - 4-5 :  219.3513
## Ocekivane frekvencije za razred  Title Bout - 1 :  88.05038
## Ocekivane frekvencije za razred  Title Bout - 2 :  51.90338
## Ocekivane frekvencije za razred  Title Bout - 3 :  163.3975
## Ocekivane frekvencije za razred  Title Bout - 4-5 :  12.64872
```

Sve su frekvencije veće od 5 pa konačno možemo provesti test. Formirajmo sada hipoteze: H0 - trajanja borbe (u rundama) su jednaka za obje vrste borbi H1 - trajanja borbe (u rundama) nisu jednaka za obje vrste borbi

```
chisq.test(tbl_combined,correct=F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_combined
## X-squared = 1418.8, df = 3, p-value < 2.2e-16
```

Odbacujemo pretpostavku H0 - "trajanja borbe (u rundama) su jednaka za obje vrste borbi" u svrhu H1 - "trajanja borbe (u rundama) nisu jednaka za obje vrste borbi". ## Drugo rješenje je korištenje Fisherovog egzaktnog testa, te koristimo tbl koji smo ranije definirali.

```
fisher.test(tbl, simulate.p.value=TRUE)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
##  2000 replicates)
##
## data:  tbl
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

Odbacujemo pretpostavku H0 - "trajanja borbe (u rundama) su nezavisna o vrsti borbe" u svrhu H1 - "trajanja borbe (u rundama) su zavisna o vrsti borbe".

Ponovimo sada statističko istraživanje tako da uspoređujemo isključivo 5-Rnd borbe u obje kategorije borbi.

Format “5 Rnd (5-5-5-5-5)” je popularan u borbama za titulu (279 borbi), a dovoljno popularan u normalnim borbama (228 borbi). S obzirom da je broj borbi tog formata za obje kategorije velik i ne previše različit, filtrirat ćemo podatke koje imamo i ponoviti istraživanje samo nad borbama formata “5 Rnd (5-5-5-5-5)”.

```
df_5rndFormat_fight_data = data.frame(df_fight_data_task3)
df_5rndFormat_fight_data['Format'] <-
  sapply(df_5rndFormat_fight_data['Format'], as.character);

df_5rndFormat_fight_data = df_5rndFormat_fight_data %>%
  filter(grepl("5 Rnd (5-5-5-5-5)", Format, fixed = TRUE))
```

Kreiramo tablicu za borbe formata “5 Rnd (5-5-5-5-5)”.

```
tbl_5rnd = table(df_5rndFormat_fight_data[
                  df_5rndFormat_fight_data$last_round == "1" |
                  df_5rndFormat_fight_data$last_round == "2" |
                  df_5rndFormat_fight_data$last_round == "3" |
                  df_5rndFormat_fight_data$last_round == "4" |
                  df_5rndFormat_fight_data$last_round == "5",]$last_round,

                  df_5rndFormat_fight_data[
                    df_5rndFormat_fight_data$last_round == "1" |
                    df_5rndFormat_fight_data$last_round == "2" |
                    df_5rndFormat_fight_data$last_round == "3" |
                    df_5rndFormat_fight_data$last_round == "4" |
                    df_5rndFormat_fight_data$last_round == "5",]$Fight_type)
tbl_5rnd_margins = addmargins(tbl_5rnd)
tbl_5rnd_margins
```

```
##
##       Normal Bout Title Bout Sum
##   1           70         63 133
##   2           41         43  84
##   3           24         34  58
##   4           15         18  33
##   5           78        121 199
##   Sum        228        279 507
```

Provjeravamo očekivane frekvencije.

```
for (col_names in colnames(tbl_5rnd_margins)){
  for (row_names in rownames(tbl_5rnd_margins)){
    if (!(row_names == 'Sum' | col_names == 'Sum') ){
      cat('Očekivane frekvencije za razred ',col_names,'-',row_names,': ',
          (tbl_5rnd_margins[row_names,'Sum'] *
           tbl_5rnd_margins['Sum',col_names]) /
           tbl_5rnd_margins['Sum','Sum'],'\n')
    }
  }
}
```

```
## Ocekivane frekvencije za razred  Normal Bout - 1 :  59.81065
## Ocekivane frekvencije za razred  Normal Bout - 2 :  37.77515
## Ocekivane frekvencije za razred  Normal Bout - 3 :  26.08284
## Ocekivane frekvencije za razred  Normal Bout - 4 :  14.84024
## Ocekivane frekvencije za razred  Normal Bout - 5 :  89.49112
## Ocekivane frekvencije za razred  Title Bout - 1 :  73.18935
```

```
## Ocekivane frekvencije za razred  Title Bout - 2 :  46.22485
## Ocekivane frekvencije za razred  Title Bout - 3 :  31.91716
## Ocekivane frekvencije za razred  Title Bout - 4 :  18.15976
## Ocekivane frekvencije za razred  Title Bout - 5 :  109.5089
```

Pretpostavka testa je ispunjena pa ga provodimo.

```
chisq.test(tbl_5rnd, correct=F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl_5rnd
## X-squared = 6.6414, df = 4, p-value = 0.1561
```

Zanimljivo, ovakvo istraživanje nam je dalo drugačiji zaključak. Ne možemo odbaciti hipotezu H0 - "trajanja borbe (u rundama) su jednaka za obje vrste borbi" te ju prihvaćamo s obzirom na p-vrijednost i vrijednost statistike manje od kritične vrijednosti. Mogli smo pretpostaviti da ćemo doći do ovakvog zaključka gledajući kontingencijsku tablicu za "5 Rnd" format borbe jer je za isto trajanje borbe u obje kategorije podjednak broj opaženih borbi.

Zadatak 4. Mogu li dostupne značajke predvidjeti pobjednika??

Zadatak je odrediti može li se predvidjeti pobjednik UFC borbe pomoću dostupnih značajki.To ćemo napraviti pomoću modela logističke regresije. Za regresore ćemo samo koristiti značajke dostupne prije borbe, npr: tip borbe, težine borca, postotak obaranja u borbama borca itd. a nećemo koristiti značajke dobivene tijekom borbe: značajni udarci itd. Napraviti ćemo nekoliko modela, svaki sa različitim pristupom, a svaki pristup ćemo objasniti prije stvaranja modela. Idemo prvo urediti naš dataset. Prvo učitavamo podatke o svim borbama koje imamo:

Iz te tablice uzimamo samo 4 stupca: B_Fighter, R_Fighter, Fight_type i Winner. Ostale značajke su dobivene tijekom borbe ili ne bi smjele utjecati na pobjednika(datum, lokacija..). Ako je pobjednik crveni, Winner mijenjamo u 0, a ako je plavi u 1. Stupci Winner i Fight_Type ćemo transformirati u tip varijable faktor

```
suppressWarnings({
  fightdata=read_excel("total_fight_data.xlsx")
})
#izbriši redove gdje je NA
fightdata <- filter(fightdata, !is.na(Winner) & !is.na(R_fighter) & !is.na(B_fighter) )
fightdataprocessed=select(fightdata,R_fighter,B_fighter,Fight_type,Winner)
fightdataprocessed$Winner=factor(
  ifelse(fightdataprocessed$Winner == fightdataprocessed$R_fighter, 0, 1))
#crveni u 0 plavi u 1
fightdataprocessed$Winner=factor(fightdataprocessed$Winner)
fightdataprocessed$Fight_type=factor(fightdataprocessed$Fight_type)
summary(fightdataprocessed)
```

```
##   R_fighter          B_fighter                          Fight_type   Winner
##  Length:5902        Length:5902        Lightweight Bout      :1030   0:3979
##  Class :character   Class :character   Welterweight Bout     :1010   1:1923
##  Mode  :character   Mode  :character   Middleweight Bout     : 759
##                                        Heavyweight Bout      : 518
##                                        Featherweight Bout    : 517
##                                        Light Heavyweight Bout: 507
##                                        (Other)               :1561
```

Koristiti ćemo nekoliko različitih modela, koje ćemo pri kraju usporediti.

Sad ćemo izvaditi podatke o borcima , te odmah urediti(postotke iz stringa u decimalne brojeve, visinu,težinu i doseg u brojeve itd.,Stance u faktor tip varijable)

```
fightersdata=read.csv("fighter_details.csv")

fightersdata=select(fightersdata,-DOB)#mičemo datum rođenja
#makni redove gdje nema Reach
fightersdata <- filter(fightersdata,  nchar(as.character(fightersdata$Reach)) > 0)
 #makni redove gdje je Height 0
fightersdata <- filter(fightersdata,  nchar(as.character(fightersdata$Height)) > 0)
 #nmakni redove gdje nema Stance
fightersdata <- filter(fightersdata,  nchar(as.character(fightersdata$Stance)) > 0)
#pretvori postotke u decimale
fightersdata$Str_Acc=parse_number(fightersdata$Str_Acc)/100

fightersdata$Str_Def=parse_number(fightersdata$Str_Def)/100

fightersdata$Weight=parse_number(fightersdata$Weight)

fightersdata$Reach=parse_number(fightersdata$Reach)

fightersdata$TD_Acc=parse_number(fightersdata$TD_Acc)/100

fightersdata$TD_Def=parse_number(fightersdata$TD_Def)/100

# kod pronađen na internetu, za konverziju visine iz incha u centimetre
# Split height column into feet and inches columns
fightersdata <- fightersdata %>%
  mutate(feet = str_split(Height, " ") %>% map_chr(1),
         inches = str_split(Height, ' ') %>% map_chr(2))

fightersdata$feet=parse_number(fightersdata$feet)
fightersdata$inches=parse_number(fightersdata$inches)

# Convert feet to inches and add to inches column
fightersdata$total_inches <- fightersdata$feet * 12 + fightersdata$inches

# Convert total inches to centimeters
fightersdata$height_cm <- fightersdata$total_inches * 2.54

# Remove unnecessary columns
fightersdata <- fightersdata %>%
  select(-feet, -inches, -total_inches, -Height)

fightersdata$Stance=factor(fightersdata$Stance)
summary(fightersdata)
```

```
##  fighter_name          Weight          Reach              Stance
##  Length:1664        Min.   :115.0   Min.   :58.00   Open Stance:   2
##  Class :character   1st Qu.:135.0   1st Qu.:69.00   Orthodox   :1247
##  Mode  :character   Median :155.0   Median :72.00   Southpaw   : 316
##                     Mean   :164.8   Mean   :71.84   Switch     :  99
##                     3rd Qu.:185.0   3rd Qu.:75.00
##                     Max.   :265.0   Max.   :84.00
##       SLpM           Str_Acc            SApM           Str_Def
```

```
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 2.270   1st Qu.:0.3800   1st Qu.: 2.470   1st Qu.:0.4800
##  Median : 3.135   Median :0.4400   Median : 3.190   Median :0.5500
##  Mean   : 3.292   Mean   :0.4402   Mean   : 3.499   Mean   :0.5343
##  3rd Qu.: 4.053   3rd Qu.:0.5000   3rd Qu.: 4.150   3rd Qu.:0.6000
##  Max.   :19.910   Max.   :0.8800   Max.   :23.330   Max.   :0.8600
##      TD_Avg           TD_Acc           TD_Def         Sub_Avg
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.000   Min.   : 0.0000
##  1st Qu.: 0.430   1st Qu.:0.2000   1st Qu.:0.380   1st Qu.: 0.0000
##  Median : 1.180   Median :0.3500   Median :0.600   Median : 0.4000
##  Mean   : 1.534   Mean   :0.3572   Mean   :0.546   Mean   : 0.6534
##  3rd Qu.: 2.300   3rd Qu.:0.5000   3rd Qu.:0.750   3rd Qu.: 0.9000
##  Max.   :18.000   Max.   :1.0000   Max.   :1.000   Max.   :20.4000
##    height_cm
##  Min.   :152.4
##  1st Qu.:172.7
##  Median :177.8
##  Mean   :178.0
##  3rd Qu.:185.4
##  Max.   :210.8
```

Vrijeme je za izradu prvog modela logističke regresije. Tablicu s podacima za model ćemo dobiti tako što spojimo fighterdata sa fightdataprocessed pomoću funkcije merge, prvo spajajući po crvenom kutu, zatim plavom, te za oba borca i u plavom i u crvenom kutu gledamo pojedinačno njihove značajke, svaku kao stupac. X varijabla će predstavljati crveni kut, dok će y varijabla predstavljati plavi kut. Ovaj model će koristiti sve moguće značajke pri predviđanju(osim imena)

```
privremena1=merge(fightdataprocessed,fightersdata,by.x="R_fighter",by.y="fighter_name")
final_data1=merge(privremena1,fightersdata,by.x="B_fighter",by.y="fighter_name")

summary(final_data1)
```

```
##   B_fighter          R_fighter                          Fight_type   Winner
##  Length:4864        Length:4864        Lightweight Bout      : 854   0:3071
##  Class :character   Class :character   Welterweight Bout     : 836   1:1793
##  Mode  :character   Mode  :character   Middleweight Bout     : 608
##                                        Featherweight Bout    : 468
##                                        Light Heavyweight Bout: 438
##                                        Bantamweight Bout     : 398
##                                        (Other)               :1262
##     Weight.x      Reach.x            Stance.x        SLpM.x
##  Min.   :115   Min.   :60.0   Open Stance:  11   Min.   : 0.000
##  1st Qu.:145   1st Qu.:70.0   Orthodox   :3663   1st Qu.: 2.590
##  Median :170   Median :72.0   Southpaw   :1039   Median : 3.260
##  Mean   :168   Mean   :72.2   Switch     : 151   Mean   : 3.389
##  3rd Qu.:185   3rd Qu.:75.0                      3rd Qu.: 4.070
##  Max.   :265   Max.   :84.0                      Max.   :19.910
##
##    Str_Acc.x          SApM.x         Str_Def.x         TD_Avg.x
##  Min.   :0.0000   Min.   : 0.100   Min.   :0.1800   Min.   :0.000
##  1st Qu.:0.3900   1st Qu.: 2.440   1st Qu.:0.5300   1st Qu.:0.680
##  Median :0.4400   Median : 3.010   Median :0.5700   Median :1.420
##  Mean   :0.4409   Mean   : 3.168   Mean   :0.5648   Mean   :1.661
##  3rd Qu.:0.4900   3rd Qu.: 3.780   3rd Qu.:0.6100   3rd Qu.:2.390
##  Max.   :0.7700   Max.   :23.330   Max.   :0.8600   Max.   :8.330
```

```
##
##     TD_Acc.x         TD_Def.x        Sub_Avg.x       height_cm.x
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :152.4
##  1st Qu.:0.3000   1st Qu.:0.5300   1st Qu.:0.2000   1st Qu.:172.7
##  Median :0.4000   Median :0.6300   Median :0.5000   Median :177.8
##  Mean   :0.4005   Mean   :0.6193   Mean   :0.6707   Mean   :178.6
##  3rd Qu.:0.5000   3rd Qu.:0.7500   3rd Qu.:1.0000   3rd Qu.:185.4
##  Max.   :1.0000   Max.   :1.0000   Max.   :7.4000   Max.   :210.8
##
##     Weight.y        Reach.y             Stance.y        SLpM.y
##  Min.   :115.0   Min.   :58.00   Open Stance:   4   Min.   : 0.180
##  1st Qu.:145.0   1st Qu.:70.00   Orthodox   :3669   1st Qu.: 2.498
##  Median :155.0   Median :72.00   Southpaw   : 986   Median : 3.220
##  Mean   :167.5   Mean   :72.16   Switch     : 205   Mean   : 3.324
##  3rd Qu.:185.0   3rd Qu.:75.00                      3rd Qu.: 4.032
##  Max.   :265.0   Max.   :84.00                      Max.   :15.070
##
##    Str_Acc.y          SApM.y         Str_Def.y         TD_Avg.y
##  Min.   :0.1000   Min.   : 0.100   Min.   :0.1900   Min.   : 0.000
##  1st Qu.:0.3900   1st Qu.: 2.530   1st Qu.:0.5100   1st Qu.: 0.610
##  Median :0.4400   Median : 3.095   Median :0.5600   Median : 1.270
##  Mean   :0.4359   Mean   : 3.327   Mean   :0.5522   Mean   : 1.536
##  3rd Qu.:0.4800   3rd Qu.: 3.900   3rd Qu.:0.6000   3rd Qu.: 2.220
##  Max.   :0.8800   Max.   :21.180   Max.   :0.7800   Max.   :10.860
##
##     TD_Acc.y         TD_Def.y       Sub_Avg.y       height_cm.y
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :152.4
##  1st Qu.:0.2700   1st Qu.:0.500   1st Qu.:0.1000   1st Qu.:172.7
##  Median :0.3800   Median :0.620   Median :0.5000   Median :177.8
##  Mean   :0.3819   Mean   :0.597   Mean   :0.6464   Mean   :178.7
##  3rd Qu.:0.5000   3rd Qu.:0.750   3rd Qu.:0.9000   3rd Qu.:185.4
##  Max.   :1.0000   Max.   :1.000   Max.   :7.4000   Max.   :210.8
##
```

Vrijeme je za regresiju, jedino ćemo prije nje za faktorske varijable Stance.x i Stance.y, odlučiti da je referentni level Stance Orthodox, pošto je najfrekventniji, jer bi model trebao bit precizniji ako za faktorsku varijablu odaberemo referentni level koji je najfrekventniji.

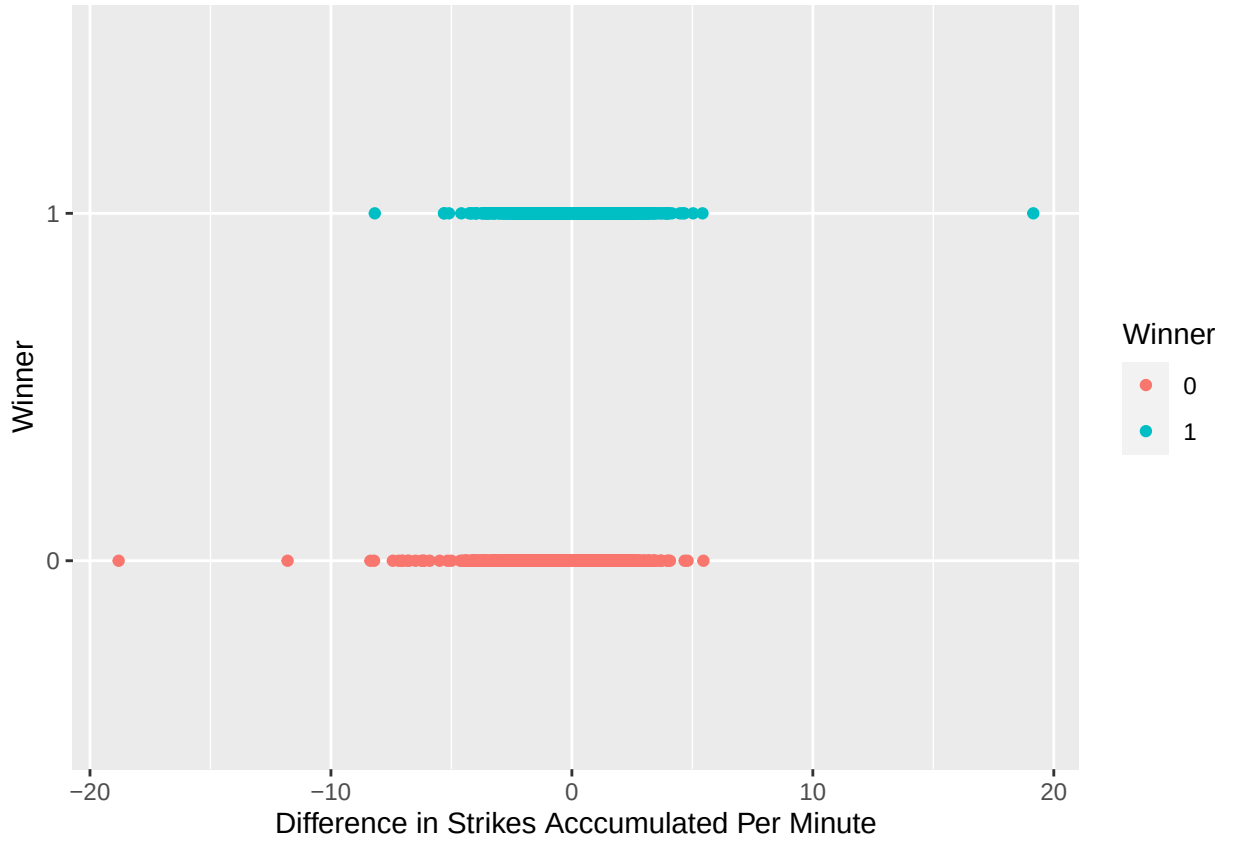
```
final_data1$Stance.x = relevel(final_data1$Stance.x, ref = "Orthodox")
final_data1$Stance.y = relevel(final_data1$Stance.y, ref = "Orthodox")
logreg.mdl1 = glm(Winner ~Fight_type + Weight.x + Reach.x +Stance.x+
                    SLpM.x+Str_Acc.x + SApM.x + Str_Def.x + TD_Avg.x +
                    TD_Acc.x + TD_Def.x + Sub_Avg.x+ height_cm.x + Weight.y +
                    Reach.y +Stance.y+ SLpM.y+Str_Acc.y + SApM.y + Str_Def.y +
                    TD_Avg.y + TD_Acc.y + TD_Def.y +
                    Sub_Avg.y+ height_cm.y, data = final_data1, family = binomial())
summary(logreg.mdl1)
#output ovog summary ne izbacujemo u pdf jer zauzima 10+ stranica
```

```
Rsq1 = 1 - logreg.mdl1$deviance/logreg.mdl1$null.deviance
cat("Rsq1: ",Rsq1)
```

```
## Rsq1:  0.1141735
```

Možemo primijetiti da su neki regresori znatno bitniji od drugih, napravimo graf gdje ćemo vidjeti razliku između pojedinih regresora i utjecaj na pobjednika

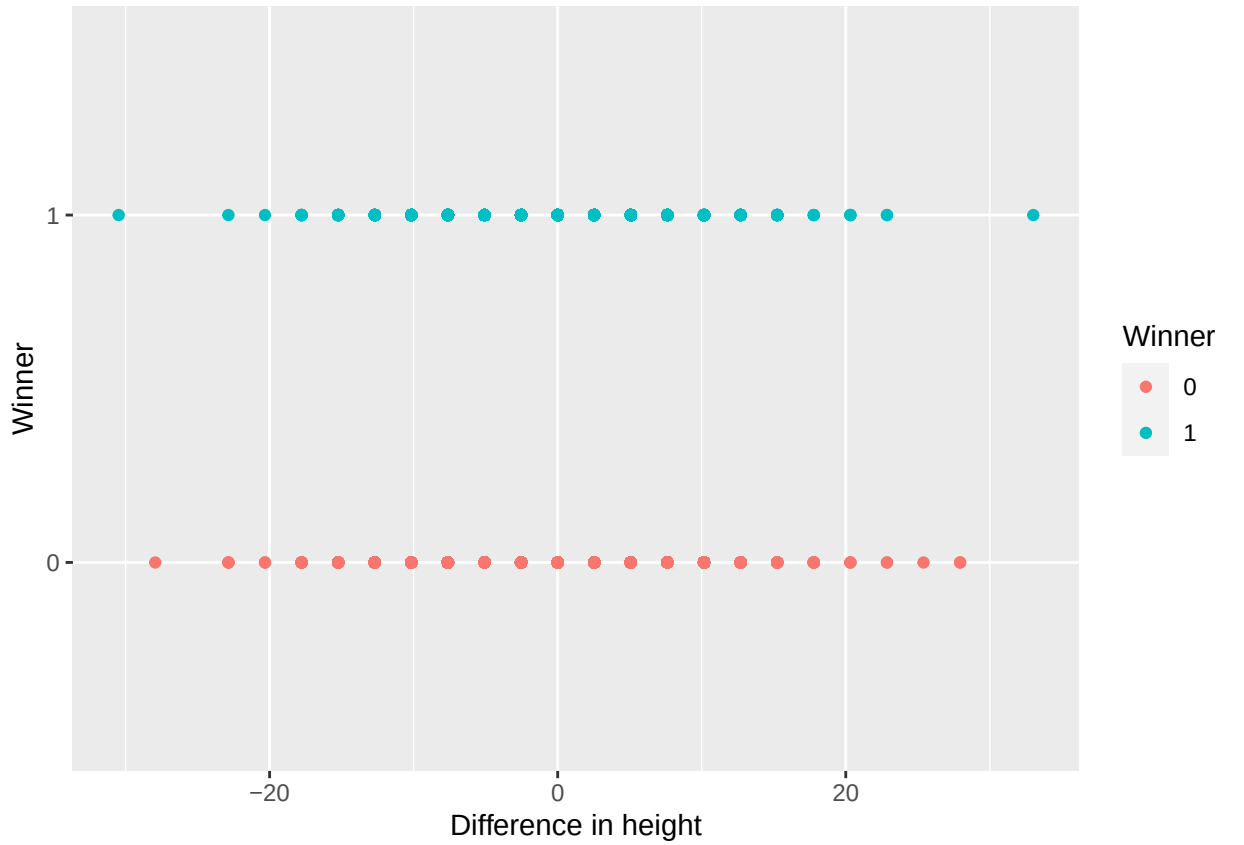
```
ggplot(data = final_data1, aes(x = SApM.x-SApM.y, y = Winner, color = Winner)) +
  geom_point() +
  labs(x = "Difference in Strikes Acccumulated Per Minute", y = "Winner")
```



Kada je razlika broja primljenih udaraca po minuti(regresor s značajnom p vrijednosti za oba borca) negativan, znači da je plavi borac primio više udaraca, i vidimo na grafu da onda imamo više crvenih pobjednika, dok pozitivna razlika znači da je crveni primio više udaraca, te imamo više plavih pobjednika.

Usporedimo to s visinom, regresorom koji nema značajnu p vrijednost

```
ggplot(data = final_data1, aes(x = height_cm.x-height_cm.y, y = Winner, color = Winner)) +
  geom_point() +
  labs(x = "Difference in height", y = "Winner")
```



Vidimo da nema neke razlike u broju pobjednika.

Gledajući grafove, možemo zaključiti da neki regresori nisu značajni. Preko funkcije step s parametrom backward ćemo sada napraviti reducirani model sa značajnijim regresorima

```
logreg.mdl2=step(logreg.mdl1,direction = "backward",trace=0)
summary(logreg.mdl2)
```

```
##
## Call:
## glm(formula = Winner ~ Weight.x + Reach.x + SLpM.x + Str_Acc.x +
##     SApM.x + Str_Def.x + TD_Avg.x + TD_Def.x + Sub_Avg.x + height_cm.x +
##     Weight.y + Reach.y + SLpM.y + SApM.y + TD_Avg.y + TD_Acc.y +
##     TD_Def.y, family = binomial(), data = final_data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7873  -0.9304  -0.6597   1.1465   2.3854
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.187695   1.197423  -0.157 0.875443
## Weight.x    -0.008810   0.003193  -2.759 0.005803 **
## Reach.x     -0.039967   0.017800  -2.245 0.024748 *
## SLpM.x      -0.252709   0.041166  -6.139 8.32e-10 ***
## Str_Acc.x   -1.145054   0.552902  -2.071 0.038360 *
## SApM.x       0.277273   0.045664   6.072 1.26e-09 ***
## Str_Def.x   -2.118148   0.614129  -3.449 0.000563 ***
```

```
## TD_Avg.x    -0.094105   0.029775  -3.161 0.001575 **
## TD_Def.x    -0.790281   0.191079  -4.136 3.54e-05 ***
## Sub_Avg.x   -0.246962   0.053658  -4.603 4.17e-06 ***
## height_cm.x  0.012694   0.008875   1.430 0.152650
## Weight.y     0.004705   0.003190   1.475 0.140186
## Reach.y      0.031658   0.013030   2.430 0.015118 *
## SLpM.y       0.435848   0.033636  12.958  < 2e-16 ***
## SApM.y      -0.289144   0.035662  -8.108 5.15e-16 ***
## TD_Avg.y     0.157506   0.029909   5.266 1.39e-07 ***
## TD_Acc.y    -0.753340   0.184924  -4.074 4.63e-05 ***
## TD_Def.y     1.085076   0.168117   6.454 1.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6403.2  on 4863  degrees of freedom
## Residual deviance: 5781.1  on 4846  degrees of freedom
## AIC: 5817.1
##
## Number of Fisher Scoring iterations: 4
```

```
Rsq2 = 1 - logreg.mdl2$deviance/logreg.mdl2$null.deviance
cat("Rsq2: ",Rsq2)
```

```
## Rsq2:  0.097148
```

Dobijamo model s većom rezidualnom devijancom nego naš početni model. Provesti ćemo Likelihood Ratio Test, i ako se ispostavi da se rezidualna devijanca nije previše povećala(provjeravamo preko p vrijednosti), uzet ćemo reducirani model kao bolji

```
anova(logreg.mdl1, logreg.mdl2, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Winner ~ Fight_type + Weight.x + Reach.x + Stance.x + SLpM.x +
##     Str_Acc.x + SApM.x + Str_Def.x + TD_Avg.x + TD_Acc.x + TD_Def.x +
##     Sub_Avg.x + height_cm.x + Weight.y + Reach.y + Stance.y +
##     SLpM.y + Str_Acc.y + SApM.y + Str_Def.y + TD_Avg.y + TD_Acc.y +
##     TD_Def.y + Sub_Avg.y + height_cm.y
## Model 2: Winner ~ Weight.x + Reach.x + SLpM.x + Str_Acc.x + SApM.x + Str_Def.x +
##     TD_Avg.x + TD_Def.x + Sub_Avg.x + height_cm.x + Weight.y +
##     Reach.y + SLpM.y + SApM.y + TD_Avg.y + TD_Acc.y + TD_Def.y
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      4765     5672.1
## 2      4846     5781.1 -81  -109.02  0.02068 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No, provedbom LRT vidimo da se rezidualna devijanca povećala, i to na razini značajnosti manjoj od 0.05, pa ne možemo koristiti reducirani model umjesto opširnog. Tu smo probali i neke druge modele, gdje bi micali samo jedan ili 2 regresora, no ni oni nisu zadovoljili na LRT, pa ih nismo ostavili u projektu.

Provjerimo sada koliko je model uspješan u predviđanju. Koristićemo "matricu zabune". Njezini parametri: - accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$ - precision: $\frac{TP}{TP+FP}$ - recall: $\frac{TP}{TP+FN}$ - specificity: $\frac{TN}{TN+FP}$

Što su parametri veći, model uspješnije predviđa

```
yHat <- logreg.mdl1$fitted.values >= 0.5
tab <- table(final_data1$Winner, yHat)

tab
```

```
##    yHat
##     FALSE TRUE
##   0  2679  392
##   1  1110  683
```

```
accuracy1 = sum(diag(tab)) / sum(tab)
precision1 = tab[2,2] / sum(tab[,2])
recall1 = tab[2,2] / sum(tab[2,])
specificity1 = tab[1,1] / sum(tab[,1])

cat("Accuracy: ",accuracy1,"\n")
```

```
## Accuracy:  0.6912007
```

```
cat("Precision: ",precision1,"\n")
```

```
## Precision:  0.6353488
```

```
cat("Recall: ",recall1,"\n")
```

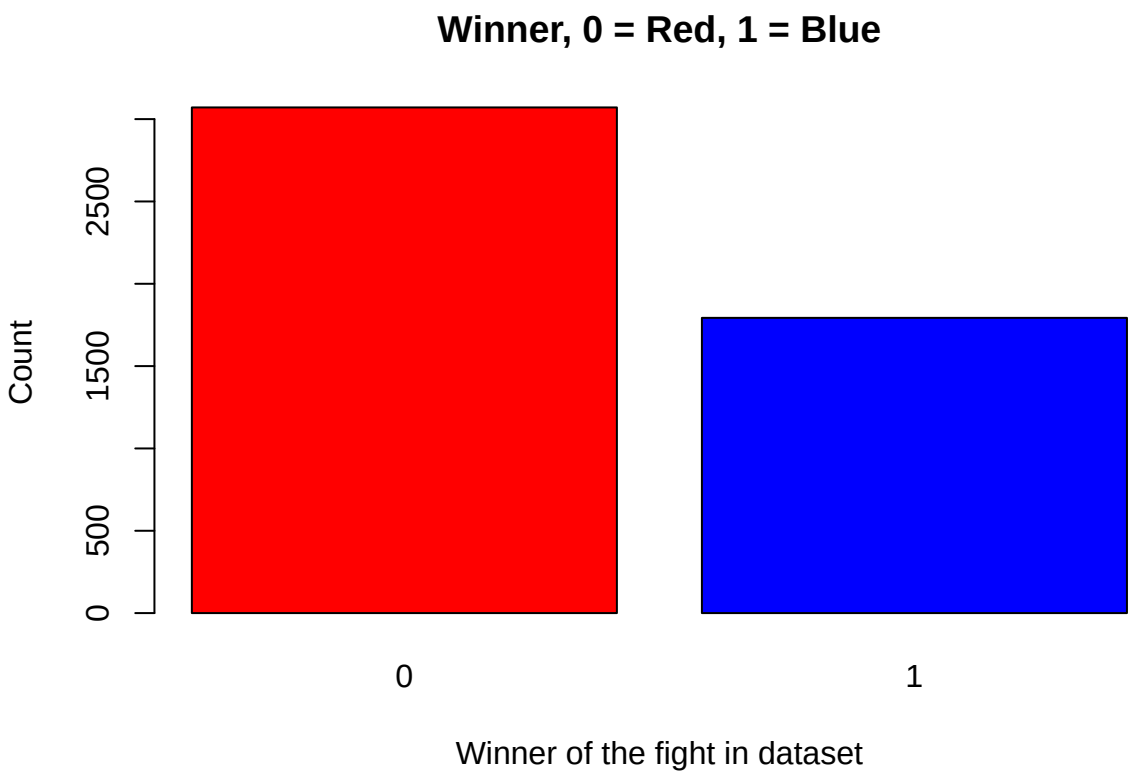
```
## Recall:  0.3809258
```

```
cat("Specificity: ",specificity1,"\n")
```

```
## Specificity:  0.7070467
```

Od naših borbi u datasetu, oko 3001 je pobjedio crveni, a 1793 je pobijedio plavi. Naš model od 3071 pobjede crvenog njih oko 2670(oko 2670) zato jer ponovnim pokretanjem bilježnice se dobivaju blago promijenjenje vrijednosti) klasificira točno, dok od 1793 pobjede plavih samo oko 680 klasificira kao pobjedu plavih. To je vidljivo i u niskoj vrijednosti recall varijabli.

```
barplot(table(final_data1$Winner),
        col = c("red","blue"),
        main = "Winner, 0 = Red, 1 = Blue",
        xlab = "Winner of the fight in dataset",
        ylab = "Count")
```



Tu se može doći do zanimljive pretpostavke, a to je da naš model u većini slučajeva samo pretpostavi da je crveni pobjednik, tj. da ima "bias" prema crvenom. S obzirom da je praksa UFC da stavlja svoje favorite u crveni kut, ovu pretpostavku treba provjeriti. Kako to testirati, i kako poboljšati model ako je to istina. Naša je ideja bila da prođemo kroz cijeli dataset, i onda za svaku slučajnu borbu zamijenimo kuteve boraca, i njihovu statistiku. Npr, ako pomoću random funkcije odlučimo da se mijenjanju, B_Fighter postaje R_Fighter, te sve varijable koje su imale .y na kraju postaju .x. Provedimo to.

```
fightdataprocessed2=fightdataprocessed
for(i in 1:nrow(fightdataprocessed2)){
  random_number <- sample(0:1,1)
  if(random_number==1){
    row=fightdataprocessed2[i,]
    storage=row$R_fighter
    row$R_fighter=row$B_fighter
    row$B_fighter=storage
    row$Winner=factor(ifelse(row$Winner==0,1,0))
    fightdataprocessed2[i,]=row
  }

}
summary(fightdataprocessed2)
```

```
##   R_fighter          B_fighter                          Fight_type   Winner
##  Length:5902        Length:5902        Lightweight Bout      :1030   0:2895
##  Class :character   Class :character   Welterweight Bout     :1010   1:3007
##  Mode  :character   Mode  :character   Middleweight Bout     : 759
##                                        Heavyweight Bout      : 518
```

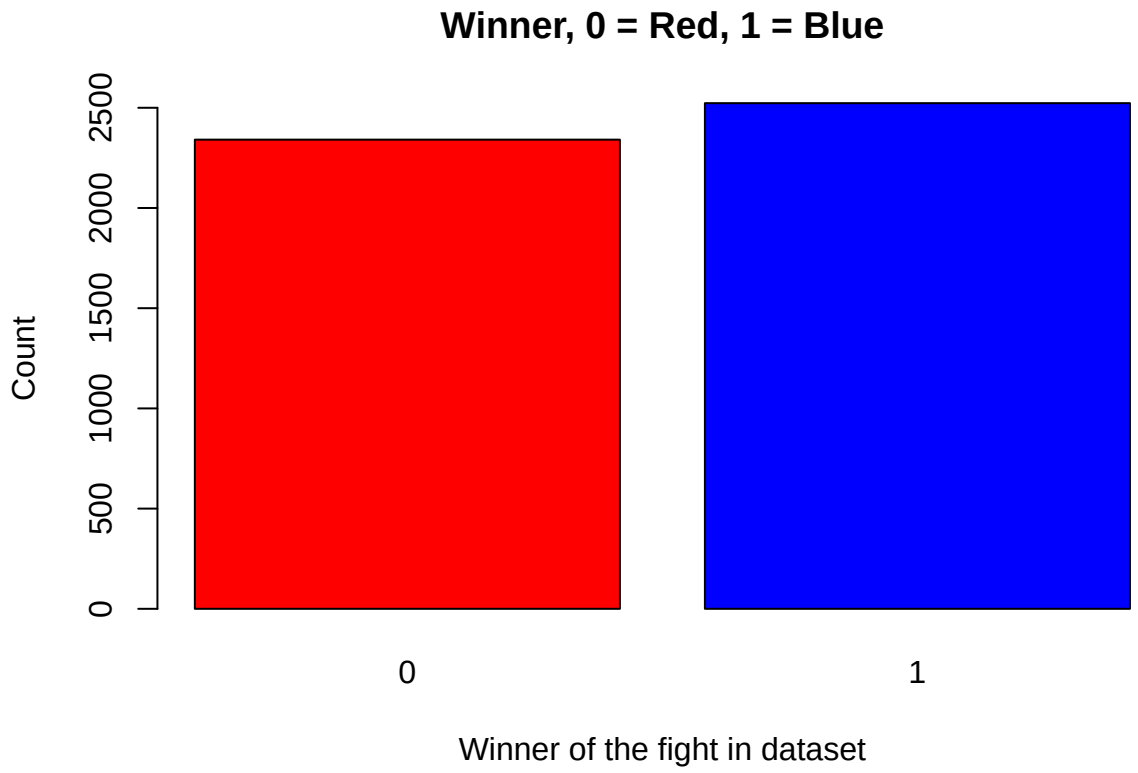
```
##                                       Featherweight Bout    : 517
##                                       Light Heavyweight Bout: 507
##                                       (Other)               :1561
```

```
privremena2=merge(fightdataprocessed2,fightersdata,by.x="R_fighter",by.y="fighter_name")
final_data3=merge(privremena2,fightersdata,by.x="B_fighter",by.y="fighter_name")


summary(final_data3)
```

```
##   B_fighter          R_fighter                          Fight_type   Winner
##  Length:4864        Length:4864        Lightweight Bout      : 854   0:2341
##  Class :character   Class :character   Welterweight Bout     : 836   1:2523
##  Mode  :character   Mode  :character   Middleweight Bout     : 608
##                                        Featherweight Bout    : 468
##                                        Light Heavyweight Bout: 438
##                                        Bantamweight Bout     : 398
##                                        (Other)               :1262
##     Weight.x        Reach.x             Stance.x        SLpM.x
##  Min.   :115.0   Min.   :58.00   Open Stance:  10   Min.   : 0.180
##  1st Qu.:145.0   1st Qu.:70.00   Orthodox   :3648   1st Qu.: 2.570
##  Median :170.0   Median :72.00   Southpaw   :1041   Median : 3.250
##  Mean   :167.9   Mean   :72.16   Switch     : 165   Mean   : 3.367
##  3rd Qu.:185.0   3rd Qu.:75.00                      3rd Qu.: 4.060
##  Max.   :265.0   Max.   :84.00                      Max.   :15.070
##
##    Str_Acc.x          SApM.x         Str_Def.x         TD_Avg.x
##  Min.   :0.0800   Min.   : 0.100   Min.   :0.2100   Min.   : 0.000
##  1st Qu.:0.3900   1st Qu.: 2.500   1st Qu.:0.5100   1st Qu.: 0.650
##  Median :0.4400   Median : 3.070   Median :0.5600   Median : 1.360
##  Mean   :0.4383   Mean   : 3.257   Mean   :0.5581   Mean   : 1.598
##  3rd Qu.:0.4900   3rd Qu.: 3.850   3rd Qu.:0.6100   3rd Qu.: 2.292
##  Max.   :0.8800   Max.   :11.900   Max.   :0.8100   Max.   :10.860
##
##     TD_Acc.x         TD_Def.x        Sub_Avg.x       height_cm.x
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :152.4
##  1st Qu.:0.2900   1st Qu.:0.5000   1st Qu.:0.1000   1st Qu.:172.7
##  Median :0.3900   Median :0.6300   Median :0.5000   Median :177.8
##  Mean   :0.3931   Mean   :0.6055   Mean   :0.6612   Mean   :178.7
##  3rd Qu.:0.5000   3rd Qu.:0.7500   3rd Qu.:0.9000   3rd Qu.:185.4
##  Max.   :1.0000   Max.   :1.0000   Max.   :7.4000   Max.   :210.8
##
##     Weight.y        Reach.y            Stance.y        SLpM.y
##  Min.   :115.0   Min.   :60.0   Open Stance:   5   Min.   : 0.000
##  1st Qu.:145.0   1st Qu.:70.0   Orthodox   :3684   1st Qu.: 2.540
##  Median :155.0   Median :72.0   Southpaw   : 984   Median : 3.230
##  Mean   :167.6   Mean   :72.2   Switch     : 191   Mean   : 3.346
##  3rd Qu.:185.0   3rd Qu.:75.0                      3rd Qu.: 4.050
##  Max.   :265.0   Max.   :84.0                      Max.   :19.910
##
##    Str_Acc.y          SApM.y         Str_Def.y         TD_Avg.y
##  Min.   :0.0000   Min.   : 0.520   Min.   :0.1800   Min.   :0.000
##  1st Qu.:0.3900   1st Qu.: 2.480   1st Qu.:0.5200   1st Qu.:0.670
##  Median :0.4400   Median : 3.040   Median :0.5600   Median :1.320
##  Mean   :0.4385   Mean   : 3.238   Mean   :0.5589   Mean   :1.599
```

```
##  3rd Qu.:0.4800   3rd Qu.: 3.810   3rd Qu.:0.6100   3rd Qu.:2.360
##  Max.   :0.8800   Max.   :23.330   Max.   :0.8600   Max.   :8.330
##
##     TD_Acc.y         TD_Def.y        Sub_Avg.y       height_cm.y
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :152.4
##  1st Qu.:0.2800   1st Qu.:0.5175   1st Qu.:0.1000   1st Qu.:172.7
##  Median :0.3900   Median :0.6300   Median :0.5000   Median :180.3
##  Mean   :0.3893   Mean   :0.6108   Mean   :0.6559   Mean   :178.7
##  3rd Qu.:0.5000   3rd Qu.:0.7500   3rd Qu.:0.9000   3rd Qu.:185.4
##  Max.   :1.0000   Max.   :1.0000   Max.   :7.4000   Max.   :210.8
##
```

```
barplot(table(final_data3$Winner),
        col = c("red","blue"),
        main = "Winner, 0 = Red, 1 = Blue",
        xlab = "Winner of the fight in dataset",
        ylab = "Count")
```



Primjećujemo da sada imamo podjednak broj pobjednika u plavom i crvenom kutu. Idemo sada napraviti model logističke regresije.

```
final_data3$Stance.x = relevel(final_data3$Stance.x, ref = "Orthodox")
final_data3$Stance.y = relevel(final_data3$Stance.y, ref = "Orthodox")
logreg.mdl5 = glm(Winner ~Fight_type + Weight.x + Reach.x +Stance.x+ SLpM.x+
                    Str_Acc.x + SApM.x + Str_Def.x + TD_Avg.x + TD_Acc.x +
                    TD_Def.x + Sub_Avg.x+ height_cm.x + Weight.y + Reach.y +Stance.y+
                    SLpM.y+Str_Acc.y + SApM.y + Str_Def.y + TD_Avg.y + TD_Acc.y + TD_Def.y
```

```
                  + Sub_Avg.y+ height_cm.y, data = final_data3, family = binomial())
summary(logreg.mdl5)
#output ovog summary ne izbacujemo u pdf jer zauzima 10+ stranica
```

```
Rsq5 = 1 - logreg.mdl5$deviance/logreg.mdl5$null.deviance
cat("Rsq5: ",Rsq5)
```

```
## Rsq5:  0.1080883
```

Primjećujemo da ovakav model ima lošije parametre od početnog, kao npr. AIC, Residual deviance, Rsq, no moramo uzeti u obzir da je i dataset drukčiji, te da nema "biasa" prema crvenom.

Provjerimo je li ovim pristupom reducirani model moguć.

```
logreg.mdl6=step(logreg.mdl5,direction = "backward",trace=0)
summary(logreg.mdl6)
```

```
##
## Call:
## glm(formula = Winner ~ Weight.x + Reach.x + SLpM.x + SApM.x +
##     Str_Def.x + TD_Avg.x + TD_Def.x + Sub_Avg.x + height_cm.x +
##     Weight.y + Reach.y + SLpM.y + Str_Acc.y + SApM.y + Str_Def.y +
##     TD_Avg.y + TD_Acc.y + TD_Def.y + Sub_Avg.y, family = binomial(),
##     data = final_data3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2746  -1.0879   0.5281   1.0614   2.2418
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.527274   1.229855  -1.242 0.214298
## Weight.x    -0.009047   0.003070  -2.947 0.003208 **
## Reach.x     -0.055305   0.017251  -3.206 0.001347 **
## SLpM.x      -0.352084   0.034267 -10.275  < 2e-16 ***
## SApM.x       0.273709   0.039295   6.965 3.27e-12 ***
## Str_Def.x   -0.839525   0.541834  -1.549 0.121282
## TD_Avg.x    -0.125509   0.028068  -4.472 7.76e-06 ***
## TD_Def.x    -0.976065   0.176233  -5.539 3.05e-08 ***
## Sub_Avg.x   -0.192680   0.050203  -3.838 0.000124 ***
## height_cm.x  0.018850   0.008574   2.199 0.027912 *
## Weight.y     0.008352   0.003055   2.734 0.006248 **
## Reach.y      0.026683   0.012646   2.110 0.034866 *
## SLpM.y       0.336589   0.039695   8.479  < 2e-16 ***
## Str_Acc.y    0.822364   0.527270   1.560 0.118839
## SApM.y      -0.287392   0.042903  -6.699 2.10e-11 ***
## Str_Def.y    1.526359   0.567419   2.690 0.007145 **
## TD_Avg.y     0.120692   0.029520   4.089 4.34e-05 ***
## TD_Acc.y    -0.386946   0.187138  -2.068 0.038668 *
## TD_Def.y     0.957174   0.175697   5.448 5.10e-08 ***
## Sub_Avg.y    0.111564   0.047268   2.360 0.018264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 6736.1  on 4863  degrees of freedom
## Residual deviance: 6090.6  on 4844  degrees of freedom
## AIC: 6130.6
##
## Number of Fisher Scoring iterations: 4
```

```
Rsq6 = 1 - logreg.mdl6$deviance/logreg.mdl6$null.deviance
Rsq6
```

```
## [1] 0.09583031
```

Reducirani model ima manji AIC, no ima manji Rsq i veću rezidualnu devijancu. Provedimo LRT da vidimo možemo li ga zadržati.

```
anova(logreg.mdl5, logreg.mdl6, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Winner ~ Fight_type + Weight.x + Reach.x + Stance.x + SLpM.x +
##     Str_Acc.x + SApM.x + Str_Def.x + TD_Avg.x + TD_Acc.x + TD_Def.x +
##     Sub_Avg.x + height_cm.x + Weight.y + Reach.y + Stance.y +
##     SLpM.y + Str_Acc.y + SApM.y + Str_Def.y + TD_Avg.y + TD_Acc.y +
##     TD_Def.y + Sub_Avg.y + height_cm.y
## Model 2: Winner ~ Weight.x + Reach.x + SLpM.x + SApM.x + Str_Def.x + TD_Avg.x +
##     TD_Def.x + Sub_Avg.x + height_cm.x + Weight.y + Reach.y +
##     SLpM.y + Str_Acc.y + SApM.y + Str_Def.y + TD_Avg.y + TD_Acc.y +
##     TD_Def.y + Sub_Avg.y
##   Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
## 1      4765     6008.0
## 2      4844     6090.6 -79  -82.571   0.3696
```

Vidimo da ako koristimo razinu značajnosti 0.05, možemo koristiti reducirani model. Provjerimo sada kakav je naš model u predviđanju

```
yHat <- logreg.mdl6$fitted.values >= 0.5
tab <- table(final_data3$Winner, yHat)

tab
```

```
##    yHat
##     FALSE TRUE
##   0  1417  924
##   1   802 1721
```

```
accuracy6 = sum(diag(tab)) / sum(tab)
precision6 = tab[2,2] / sum(tab[,2])
recall6 = tab[2,2] / sum(tab[2,])
specificity6 = tab[1,1] / sum(tab[,1])

cat("Accuracy: ",accuracy6,"\n")
```

```
## Accuracy:  0.645148
```

```
cat("Precision: ",precision6,"\n")
```

```
## Precision:  0.6506616
```

```
cat("Recall: ",recall6,"\n")
```

```
## Recall:  0.6821245
```

```
cat("Specificity: ",specificity6,"\n")
```

```
## Specificity:  0.6385759
```

```
ggplot(data = final_data3, aes(x = SApM.x-SApM.y, y = Winner, color = Winner)) +
  geom_point() +
  labs(x = "Difference in Strikes Acccumulated Per Minute, New Model", y = "Winner")
```

PRIMJER ZNAČAJNOG REGRESORA

Primjećujemo da sada podjednako dobro pogađa i za crvene i za plave kada je tko pobijedio. Reccall je dosta bolji za ovakav model, dok su ostali parametri slični kao prije. Za pogađanje pobjednika mi bi koristili model koji ne zna tko je u kojem kutu, jer ima puno bolji recall.

Sada dolazimo do glavnog pitanja, možemo li pomoću ovakvog modela i dostupnih značajki pogoditi pobjednika UFC borbe? Mi ne bi rekli da možemo pogoditi sa 100% sigurnošću (a ne bi ni model, naprotiv, on kaže 64.47%), jer UFC borba je puno više od same statistike i nama dostupnih značajki, no, ovaj model nam može dati čvrstog favorita što je bolje od pogađanja.

Probati ćemo model na nekim borbama koje nikad nije vidio, npr., borbe iz 2022, ili nadolazeće borbe. Model ispisuje vjerojatnost pobjede plavog(y) borca

```
fighterx<-filter(fightersdata,fighter_name=="Sean Strickland")
fightery<-filter(fightersdata,fighter_name=="Nassourdine Imavov")
fighterx$Merge=1
fightery$Merge=1
fight=merge(fighterx,fightery,by ="Merge")
#x=Strickland vs y=Imavov
#Nadolazeća borba, u nedjelju
```

```
data=data.frame(fight)
predict(logreg.mdl6,data,type="response")
```

```
##        1
## 0.581967
```

```
print("Model predviđa pobjedu Imavova")
```

```
## [1] "Model predvida pobjedu Imavova"
```

```
#x=Makhachev vs y=Oliveira
fighterx<-filter(fightersdata,fighter_name=="Islam Makhachev")
fightery<-filter(fightersdata,fighter_name=="Charles Oliveira")
fighterx$Merge=1
fightery$Merge=1
fight=merge(fighterx,fightery,by ="Merge")
#Borba iz 2022 koju je pobijedio Makhachev
data=data.frame(fight)
predict(logreg.mdl6,data,type="response")
```

```
##         1
## 0.4257936
```

```
print("Model predviđa pobjedu Makhacheva")
```

```
## [1] "Model predvida pobjedu Makhacheva"
```

```
#x=Chandler vs y=Ferguson
#Borba koju je pobijedio Chandler
fighterx<-filter(fightersdata,fighter_name=="Michael Chandler")
fightery<-filter(fightersdata,fighter_name=="Tony Ferguson")
fighterx$Merge=1
fightery$Merge=1
fight=merge(fighterx,fightery,by ="Merge")
data=data.frame(fight)
predict(logreg.mdl6,data,type="response")
```

```
##         1
## 0.5291267
```

```
print("Model predviđa pobjedu Fergusona")
```

```
## [1] "Model predvida pobjedu Fergusona"
```

Kao što vidimo, model nekad pogodi, ali nekad i ne pogodi, što je i za očekivati. ## Bonus zadatak 5. - Postoji li veza između postotka obranjenih udaraca u karijeri borca (Str_Def) i značajnih primljenih udaraca po minuti u karijeri borca (SApM)?

Zanima nas možemo li ustanoviti da borci s velikim Strike Defensom često imaju manji broj primljenih udaraca po minuti u karijeri. Za ispitivanje ovog zadatka koristit ćemo model linearne regresije s kojim ćemo pokušati vidjeti je li Str_Def značajan u predviđanju SApM.

Za početak učitavamo podatke i formatiramo dataframe da sadrži stupce imena borca, njegovog SApM i Str_Def. Filtriramo one borce za koje ti podatci ne postoje.

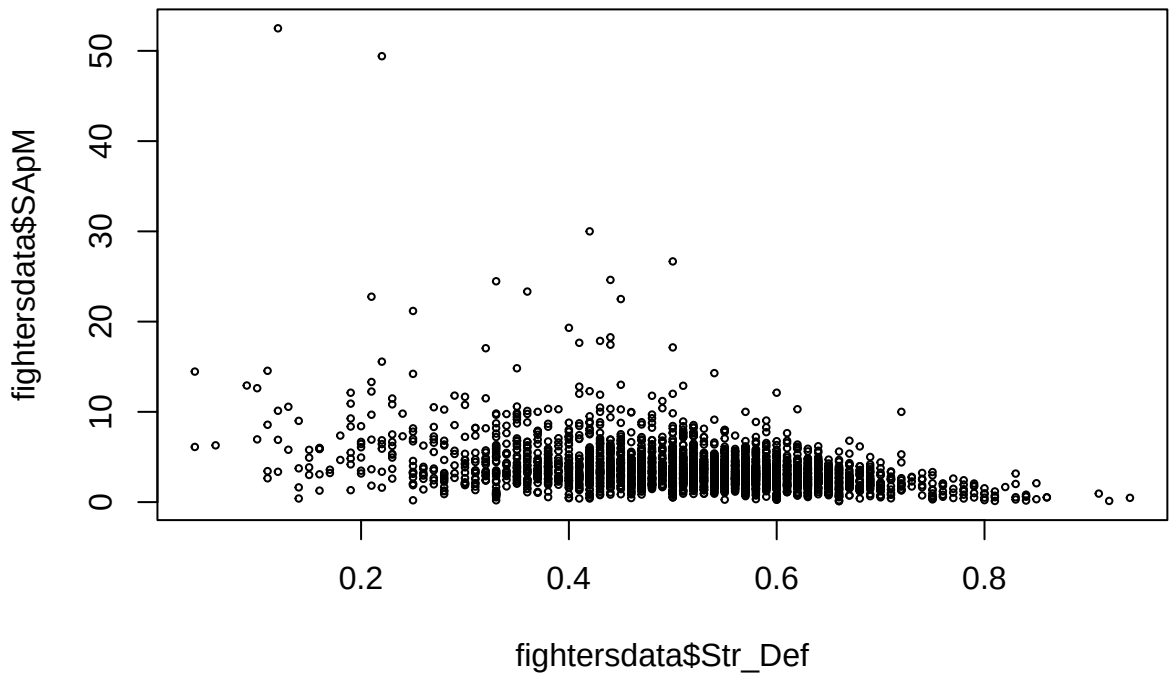
```
fightersdata=read.csv("fighter_details.csv")

fightersdata=fightersdata[c("fighter_name","SApM","Str_Def")]
fightersdata$Str_Def=parse_number(fightersdata$Str_Def)/100#postotak u decimale
```

```
fightersdata <- filter(fightersdata,  fightersdata$SApM > 0)#mičemo fightere koji nemaju
fightersdata <- filter(fightersdata,  fightersdata$Str_Def > 0)

summary(fightersdata)
```

```
##  fighter_name            SApM           Str_Def
##  Length:2915        Min.   : 0.100   Min.   :0.0400
##  Class :character   1st Qu.: 2.250   1st Qu.:0.4500
##  Mode  :character   Median : 3.160   Median :0.5300
##                     Mean   : 3.659   Mean   :0.5176
##                     3rd Qu.: 4.320   3rd Qu.:0.6000
##                     Max.   :52.500   Max.   :0.9400
```

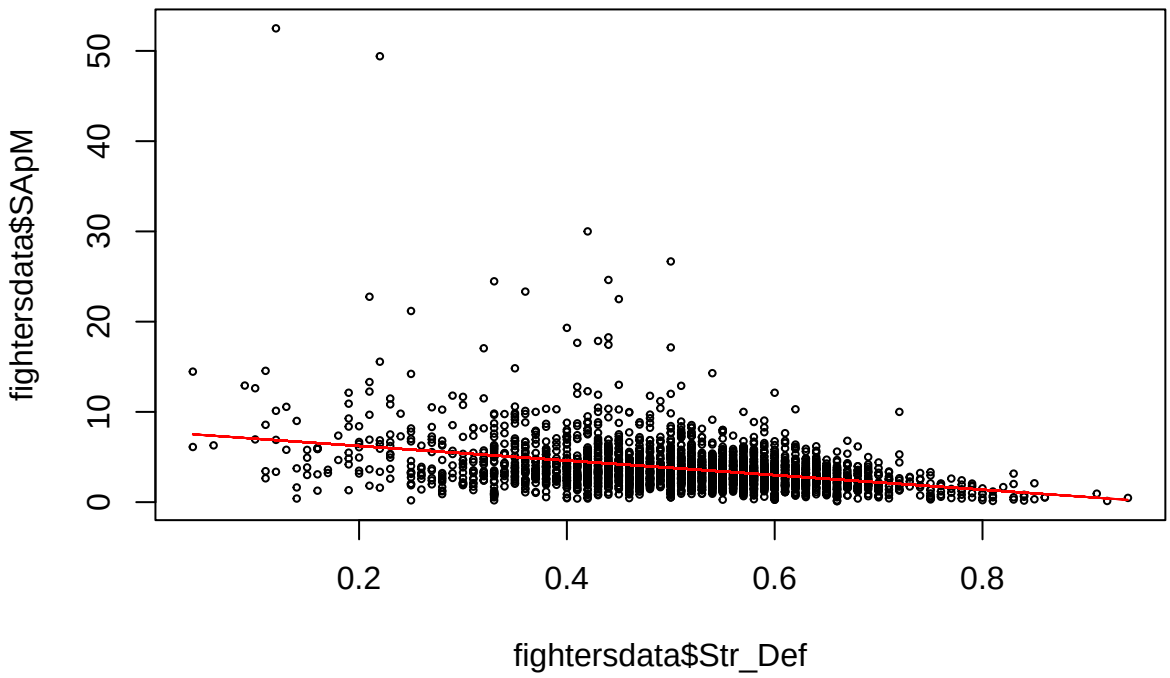
Pogledajmo odnos varijabli pomoću scatter plota. Nezavisna varijabla u našem slučaju je Str_Def. Iz priloženog grafa vidimo da postoji odnos između dvije varijable i nagađamo da je linearna. Što je veći Str_Def pretpostavljamo da će biti manji SApM. Također vidimo da postoje outlieri koji imaju izrazito veliki SApM za određeni Str_Def.

```
plot(fightersdata$Str_Def,fightersdata$SApM,cex=0.45)
```



Namjestimo sada linearni model. Prema grafu možemo pretpostaviti da pravac dobro opisuje odnos.

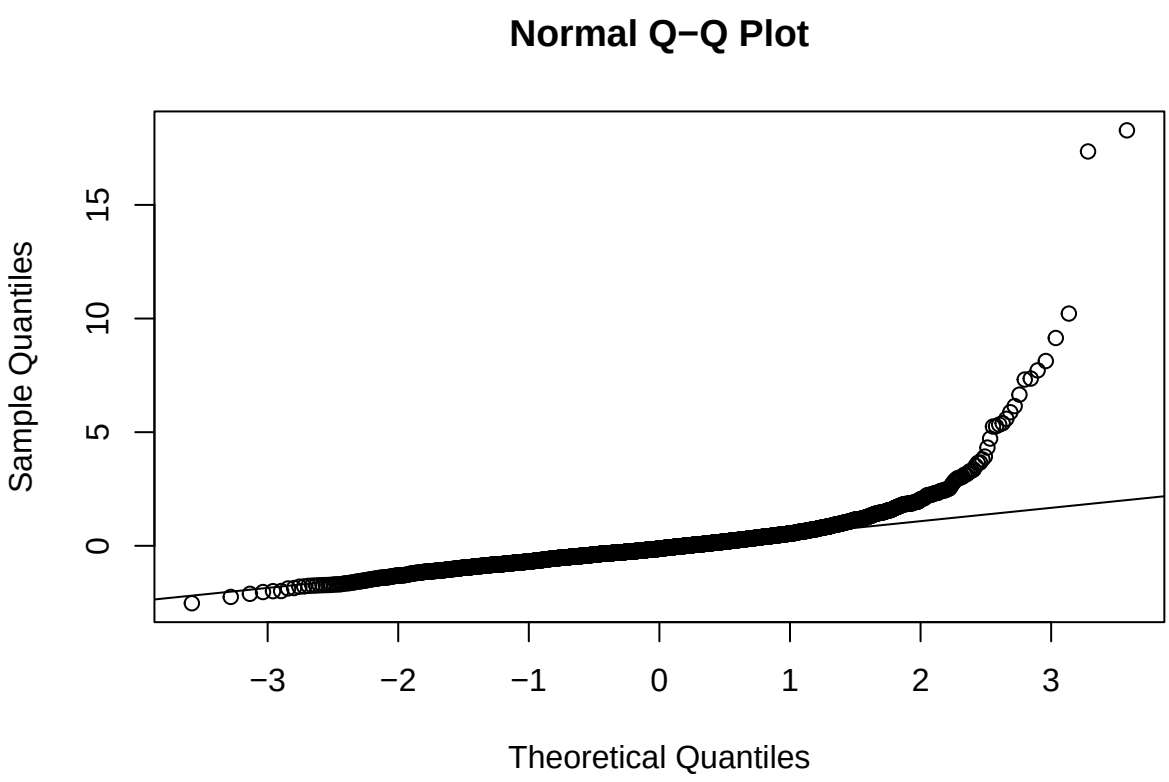
```
fit.St = lm(SApM~Str_Def,data=fightersdata)

plot(fightersdata$Str_Def,fightersdata$SApM,cex=0.45) #grafički prikaz podataka
lines(fightersdata$Str_Def,fit.St$fitted.values,col='red')
```




```
#graficki prikaz procijenjenih vrijednosti iz modela
```

Primjećujemo da postoji odnos, no da je varijanca velika i da postoji određen broj outliera. Provjeravamo normalnost reziduala.

```
selected.model = fit.St


#q-q plot reziduala s linijom normalne distribucije
qqnorm(rstandard(selected.model))
qqline(rstandard(selected.model))
```




```
lillie.test(rstandard(fit.St))
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(fit.St)
## D = 0.13905, p-value < 2.2e-16
```

Lilliefors testom dolazimo do zaključka da naši reziduali nisu normalno distribuirani.

```
summary(fit.St)
```

```
##
## Call:
## lm(formula = SApM ~ Str_Def, data = fightersdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.308 -1.214 -0.315  0.766 45.631
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.8378     0.2059   38.06   <2e-16 ***
## Str_Def      -8.0729     0.3876  -20.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.501 on 2913 degrees of freedom
```

```
## Multiple R-squared:  0.1296, Adjusted R-squared:  0.1293
## F-statistic: 433.8 on 1 and 2913 DF,  p-value: < 2.2e-16
```

Nakon obavljenog summary-a, imamo sljedeće činjenice. Reziduali nam nisu normalno distribuirani, nemamo veliki R-squared, no Str_Def je jako značajan po p-vrijednosti i imamo jako malu p-vrijednost u F statistici. Ovaj model nije najbolji za predviđanje borčevog SApM, no to nije ni bio cilj istraživanja. Cilj je bio pokazati da postoji veza između Str_Def i SApM, i sudeći po rezultatima ove linearne regresije vidimo da je Str_Def vrlo značajan za vrijednost SApM(<2e-16). Po tome zaključujemo da Str_Def igra ulogu u određivanju SApM. Naša pretpostavka zašto model za predviđanje nije još dovoljno dobar je da je potrebno još varijabli koje bi objasnile varijancu, što ostavljamo za buduća istraživanja.