Analiziranje rezultata teniskih mečeva

Luka Raić, Anđelko Prskalo, Martin-Ante Rogošić, Katarina Zec

2024-01-21

Sadržaj

1 Definicija zajednički korištenih podataka i funkcija:

Funkcije za dohvat podataka i željenog perioda:

```
fetch_data <- function(years) {

  df_list <- list()
  for (year in years) {
    file_path <- paste0("./ATP-Matches/atp_matches_", year, ".csv")
    if (file.exists(file_path)) {
      df <-
        read_csv(file_path, col_types =
                   "cccnccnnncccncnnncccncncncnnnnnnnnnnnnnnnnnnnnnnn",
                 show_col_types = FALSE)
      df_list[[as.character(year)]] <- df
    } else {
      warning(paste("File not found for year", year))
    }
  }

  matches <- bind_rows(df_list)
  rm(df_list, file_path, df, year, years)

  matches
}
```

2 1. istrazivačko pitanje: Kakva je distribucija mečeva na specifičnim podlogama u različitim godišnjim dobima?

U ovom istraživačkom pitanju ćemo koristiti podatke iz perioda 1968-2023. Za početak, dohvatimo potrebne podatke, uklonimo neregularne obzervacije (one mečeve za koje ne znamo na kojoj podlozi su odigrani) te pretvorimo datume u objekte tipa "date":

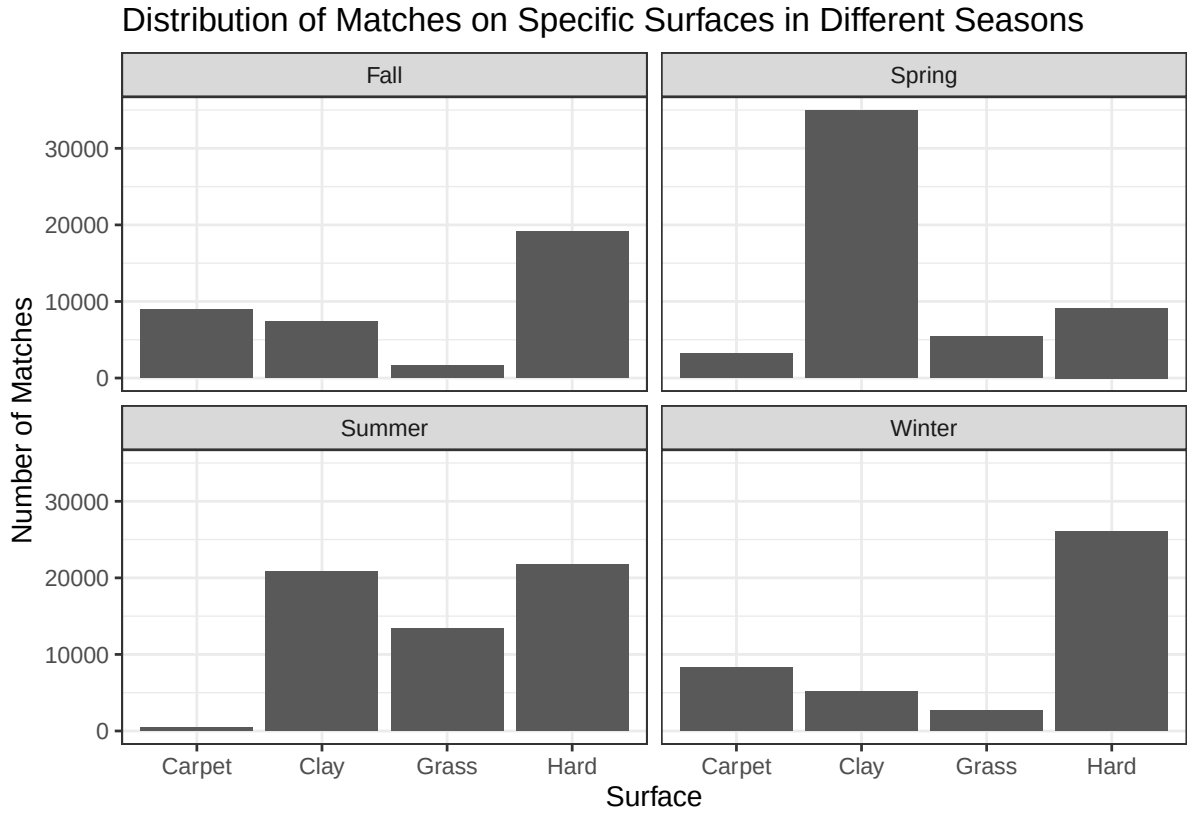
```
matches <- fetch_data(1968:2023)

matches <- filter(matches, !is.na(surface))
matches$tourney_date <- ymd(matches$tourney_date)
```

Kako bismo odredili u kojem godišnjem dobu je odigran meč, za to kreiramo specifičnu funkciju "getSeason()":

```
getSeason <- function(DATES) {
    WS <- as.Date("2012-12-15", format = "%Y-%m-%d") # Winter Solstice
    SE <- as.Date("2012-3-15",  format = "%Y-%m-%d") # Spring Equinox
    SS <- as.Date("2012-6-15",  format = "%Y-%m-%d") # Summer Solstice
    FE <- as.Date("2012-9-15",  format = "%Y-%m-%d") # Fall Equinox

    d <- as.Date(strftime(DATES, format="2012-%m-%d"))

    ifelse (d >= WS | d < SE, "Winter",
      ifelse (d >= SE & d < SS, "Spring",
        ifelse (d >= SS & d < FE, "Summer", "Fall")))
}

matches$season <- getSeason(matches$tourney_date)
```

Sada kada imamo sve potrebne informacije za svaki meč, još nam jedino ostaje vizualizacija i pregled rezultata:

```
ggplot(matches, aes(x = surface)) +
  geom_bar() + theme_bw() +
   facet_wrap(season ~ ., nrow = 2, ncol = 2) +
    labs(title = "Distribution of Matches on Specific Surfaces in Different Seasons",
     x = "Surface", y = "Number of Matches")
```



Zbog hladnog vremena zimi se većina mečeva igra u zatvorenim prostorima na tvrdoj podlozi i tepihu, u proljeće dominira šljaka na kojoj se između ostalog između sredine svibnja i ranog lipnja u Parizu održava French Open, ljeti su podjednako zastupljene šljaka i tvrda podloga s tim da se naglo povećava i broj mečeva na travi na kojoj se između ostalog u to vrijeme igra i famozni Wimbeldon, dok u jesen opet dominira tvrda podloga na kojoj se krajem kolovoza igra US Open.

3 2. istraživačko pitanje: Postoji li značajna razlika u prosječnom broju dvostrukih pogrešaka između mečeva odigranih na otvorenom u odnosu na mečeve odigrane na zatvorenom terenu?

3.1 Postavljanje problema

Prvi problem u riješavanju ovog zadatka bila je kategorizacija mečeva na one igrane na otvorenom i one igrane na zatvorenom terenu, budući da dataset nije sadržavao tu specifičnu kategoriju podataka. Odlučili smo se za pristup koji svrstava mečeve u one na otvorenom ili zatvorenom terenu, na temelju podloge na kojoj su odigrani. Nakon toga moramo analizirati svojstva ovih dviju distribucija. Dobijemo li da se distribucije ponašaju jednako zaključujemo da ne postoji značajna razlika u broju dvostrukih pogrešaka među mečevima odigranima na zatvorenim i otvorenim terenima. U suprotnom pokazat cemo da razlika postoji.

3.2 Priprema podataka

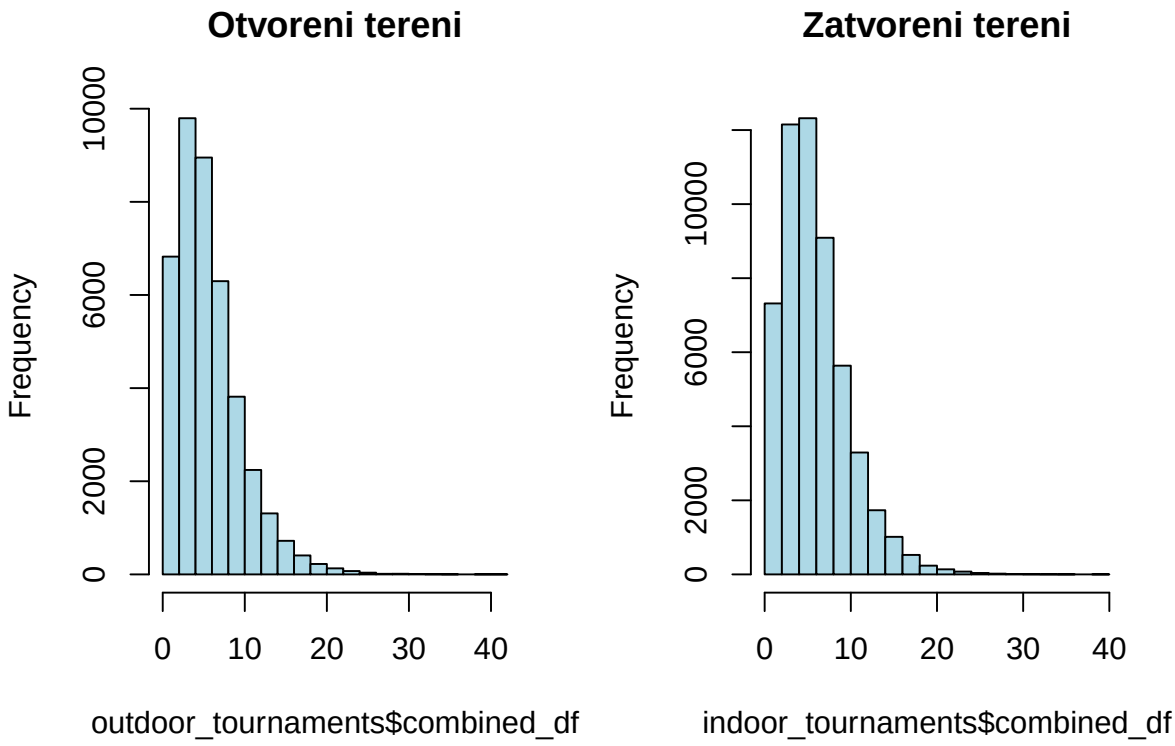
Podatke smo pripremili tako da smo čitav skup podataka spojili u podatkovni okvir, iz tog okvira izbacili smo sve stupce koji nas ne zanimaju u kontekstu ovog zadatke (sve osim podloge i broja dvostrukih pogrešaka za oba igrača). U sljedećem koraku izbacili smo svaki meč za kojeg svi relevantni podatci nisu definirani. Konačno dodali smo novi stupac u podatkovni okvir koji predstavlja ukupan broj dvostrukih grešaka za pojedini meč. Upravo je ukupan broj dvostrukih pogrešaka podatak kojeg želimo analizirati. Originalni podatkovni okvir sada razdvajamo na 4 podatkovna okvira, jedan za svaku vrstu podloge. Zaključili smo da je najbolji način za odrediti jeli meč igran na otvorenom ili zatvorenom terenu jest gledati na kojoj je podlozi igran. Procjenili smo da su podloge ilovine i trave karakteritične za vanjske terene dok tu tepisi i tvrde podloge karakteristični za zatvorene. Tako odredivši kategorizaciju spojili smo podatkovne okvire, nakon čega su nam preostala dva okvira, jedan za mečeve odigrane na otvorenom i jedan za one odigrane na zatvorenom terenu.

```
fileList <- list.files(path = "./ATP-Matches/", pattern = "\\.csv", full.names = TRUE)
combined_data <- data.frame()
for(file in fileList){
  temp <- read.csv(file, header = TRUE)
  combined_data <- rbind(combined_data, temp)

}

unique_tournaments <- unique(combined_data$tourney_name)

combined_data <- subset(combined_data, select = c("surface", "l_df", "w_df"))

combined_data <- na.omit(combined_data)

combined_data$combined_df <- combined_data$l_df + combined_data$w_df

hard_surface <- subset(combined_data, surface == "Hard")
carpet_surface <- subset(combined_data, surface == "Carpet")
clay_surface <-subset(combined_data, surface == "Clay")
grass_surface <- subset(combined_data, surface == "Grass")

outdoor_tournaments <- rbind(clay_surface,grass_surface)
indoor_tournaments <- rbind(carpet_surface, hard_surface)
```

3.3 Deskriptivna statistika

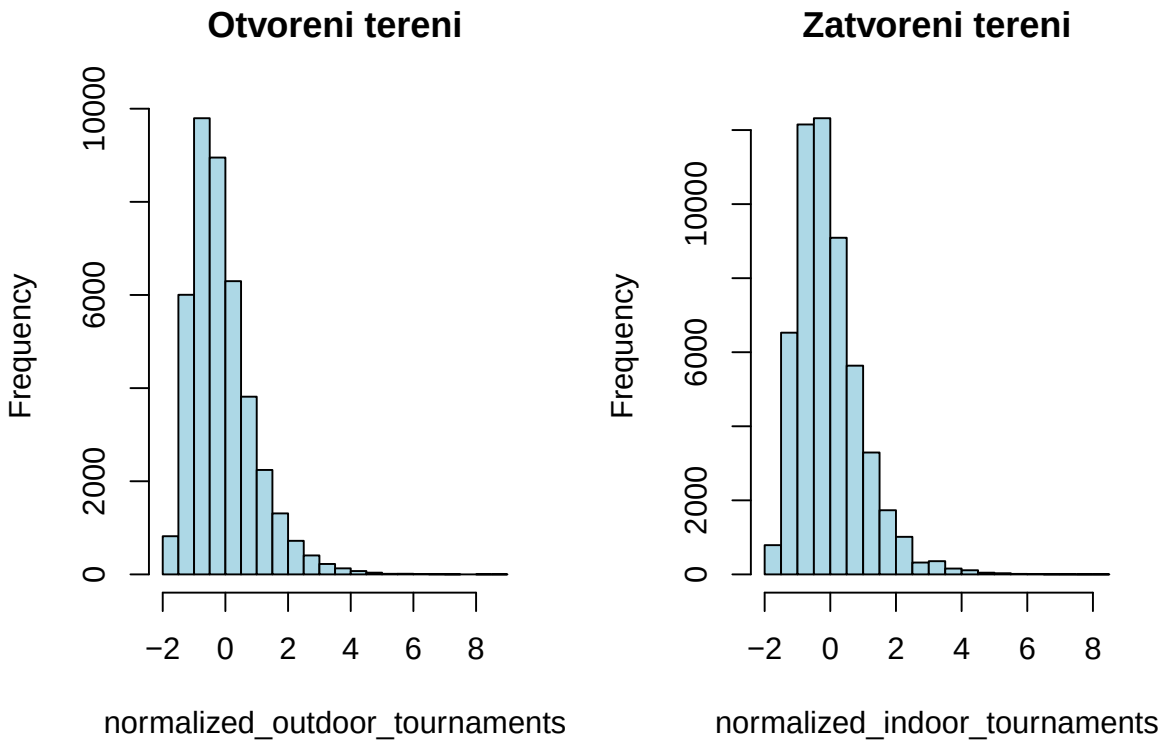
Kako bismo dobili bolju intuiciju za distribucije varijabli, pogledajmo histograme mečeva

```
par(mfrow = c(1, 2))

hist(outdoor_tournaments$combined_df, border = "black", col = "lightblue",
     main = "Otvoreni tereni")

hist(indoor_tournaments$combined_df, border = "black", col = "lightblue",
     main = "Zatvoreni tereni")
```



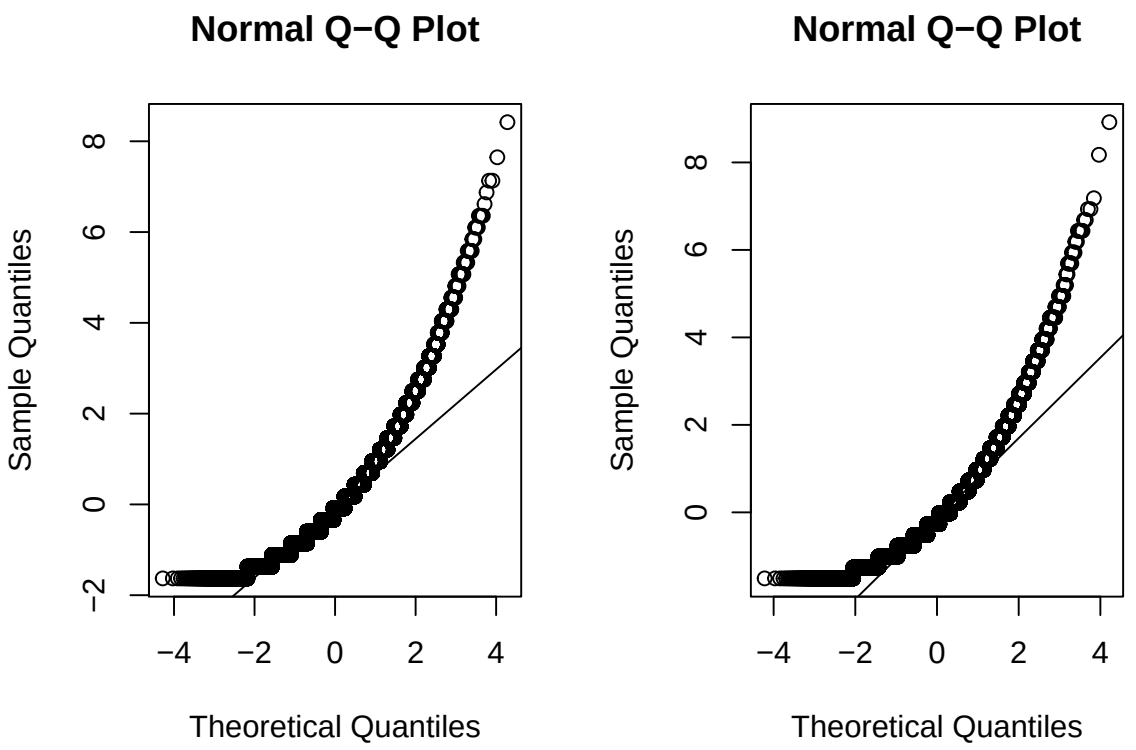
Uočimo da je izgled dvaju histograma dosta sličan. Distribucija na prvi pogled izgleda kao eksponencijalna, međutim prvi stupac histograma kvari taj obrazac, pogledajmo izgled histograma jednom kad normaliziramo varijable kako bismo potencijalno dobili nešto smislenije.

```
normalized_outdoor_tournaments <- scale(outdoor_tournaments$combined_df)
normalized_indoor_tournaments <- scale(indoor_tournaments$combined_df)

par(mfrow = c(1, 2))

hist(normalized_outdoor_tournaments, border = "black", col = "lightblue",
     main = "Otvoreni tereni", )

hist(normalized_indoor_tournaments, border = "black", col = "lightblue",
     main = "Zatvoreni tereni")
```



Kao što vidimo distribucija vrijednosti nije normalna niti odgovara nekoj opće poznatoj distribuciji. Distribucija je donekle simetrična i vidimo da je nagnuta na lijevu stranu. Ako nas zanima koliko blisko su ove dvije disribucije normalnoj možemo provjeriti upotrebom qq plota,

```
par(mfrow = c(1, 2))
qqnorm(normalized_indoor_tournaments)
qqline(normalized_indoor_tournaments)


qqnorm(normalized_outdoor_tournaments)
qqline(normalized_outdoor_tournaments)
```



3.4 Prediktivna statistika

Jasno je da su podatci izričito nenormalni stoga će se za testiranje morati koristiti neparametarska metoda koja nije osjetljiva na normalnost distribucija koje promatramo. Dodatno s obzirom da proučavamo sredine uzoraka čije distribucije ne znamo, znamo da naš neparametarski test mora biti pandan parametarskom t-testu. Jedan od testova kojeg možemo koristiti je Mann-Whitney-Wilcoxonov. Mann-Whitney-Wilcoxonov općenito testira tendenciju da jedna distribucija daje veće vrijednosti od druge distribucije,stoga je prikladan u ovom slučaju. Kao razinu značajnosti za naš statistički test uzeti ćemo standardnu razinu 0.05. Kao nultu hipotezu postaviti ćemo tezu: ne postoji razlika između broja dvostrukih grešaka između mečeva odigranih na otvorenom odnosno zatvorenom terenu. Kao alternativnu hipotezi postaviti ćemo da razlika postoji.

```
wilcox.test(normalized_indoor_tournaments, normalized_outdoor_tournaments)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  normalized_indoor_tournaments and normalized_outdoor_tournaments
## W = 1062254763, p-value = 1.341e-15
## alternative hypothesis: true location shift is not equal to 0
```

S obzirom da je p-vrijednost vrlo mala, na razini značajnosti od 0.05 možemo odbaciti nultu hipotezu u korist alternative te zaključiti da postoji značajna razlika u broju dvostrukih grešaka odigranih na otvorenom terenu u odnosu na one odigrane u zatvorenom terenu.

4 3. istraživačko pitanje: Ima li razlike u broju serviranih asova na različitim podlogama?

4.1 Učitavanje podataka

Podatke ćemo učitati iz CSV datoteka u listu podatkovnih okvira. Svaki podatkovni okvir sadržava podatke mečeva jedne godine. Nakon toga spajamo sve podatkove okvire u jedan te time imamo informacije

o svim odigranim mečevima od 1968. do 2023. godine dostupne u jednom podatkovnom okviru. Iz tog okvira ćemo naknadno izvlačiti potrebne podatke za svrhe analize.

Proces učitavanja podataka je prikazan u sljedećem bloku koda:

```
matches <- fetch_data(1968:2023)
```

4.2 Prilagodba tipova podataka

Nakon učitavanja podataka, potrebno je ispraviti tip podataka za stupac `surface` koji je potreban kod istraživanja.

Prilagodba je prikazana u sljedećem kodu:

```
matches$surface <- factor(
  matches$surface,
  levels = c("Grass", "Clay", "Hard", "Carpet")
)
```

4.3 Statistička analiza

4.3.1 Priprema podataka

Za provođenje statističke analize, potrebno je pripremiti podatke. U ovom slučaju, potrebno je izvući podatke o broju serviranih asova po meču i podlogama na kojima se igrao meč. Također, potrebno je ukloniti sve mečeve koji nemaju podatak o broju serviranih asova ili podlogama na kojima se igrao meč.

Podatke spremamo u nove podatkovne okvire od kojih svaki predstavlja aseve na jednoj podlozi:

```
# izdvajanje broja serviranih asova na travnatoj podlozi
aces_grass <- matches %>%
  filter(!is.na(w_ace) & !is.na(l_ace) & surface == "Grass") %>%
  mutate(aces = w_ace + l_ace) %>%
  dplyr::select(aces)

# izdvajanje broja serviranih asova na zemljanoj podlozi
aces_clay <- matches %>%
  filter(!is.na(w_ace) & !is.na(l_ace) & surface == "Clay") %>%
  mutate(aces = w_ace + l_ace) %>%
  dplyr::select(aces)

# izdvajanje broja serviranih asova na tvrdoj podlozi
aces_hard <- matches %>%
  filter(!is.na(w_ace) & !is.na(l_ace) & surface == "Hard") %>%
  mutate(aces = w_ace + l_ace) %>%
  dplyr::select(aces)

# izdvajanje broja serviranih asova na tepihu
aces_carpet <- matches %>%
  filter(!is.na(w_ace) & !is.na(l_ace) & surface == "Carpet") %>%
  mutate(aces = w_ace + l_ace) %>%
  dplyr::select(aces)

# izdvajanje broja serviranih asova na svim podlogama
aces_all <- matches %>%
  filter(!is.na(w_ace) & !is.na(l_ace) & !is.na(surface)) %>%
  mutate(aces = w_ace + l_ace) %>%
  dplyr::select(aces)
```

4.3.2 Deskriptivna statistika

Sada kada smo pripremili podatke, možemo izračunati deskriptivnu statistiku za broj serviranih asova po podlogama. Za svaku podlogu, izračunat ćemo sljedeće vrijednosti za svaku od podloga:

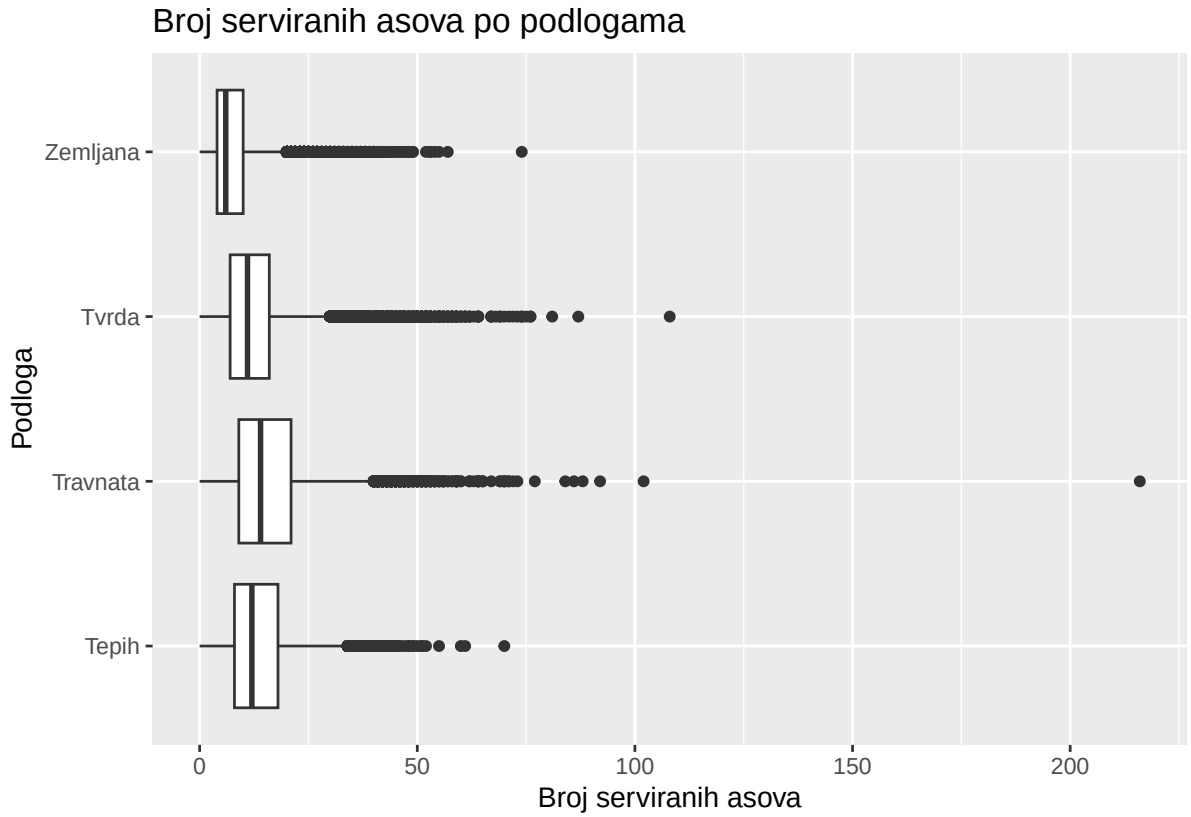
- prosjecni broj asova
- standardnu devijaciju broja asova
- medijan broja asova
- minimalni broj asova
- maksimalni broj asova
- broj mečeva

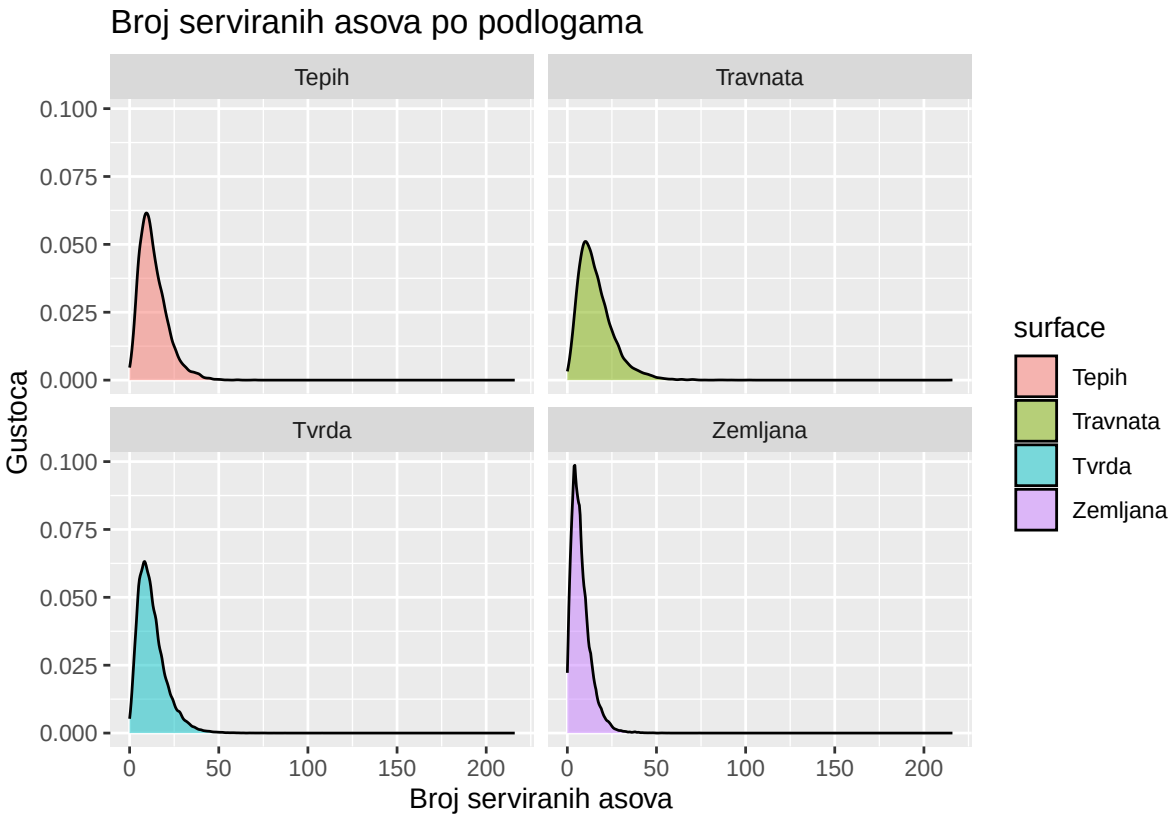
Table 1: Servirani asovi po podlogama

surface	mean	sd	median	min	max	count
Tepih	13.36	7.95	12	0	70	5878
Travnata	16.16	10.22	14	0	216	9783
Zemljana	7.62	5.65	6	0	74	31085
Tvrda	12.58	8.08	11	0	108	47729
Ukupno	11.37	8.14	10	0	216	94475

Iz prikazanih vizualizacija, možemo vidjeti kako je, u prosijeku, najviše asova servirano na travi, a najmanje na zemlji. Također, možemo vidjeti kako je na travi servirano najviše asova u jednom meču, a na tepihu najmanje. Najveću varijancu broja serviranih asova po meču imamo na travi.

Ove podatke možemo vizualizirati pomoću boxplot dijagrama i dijagrama gustoće.



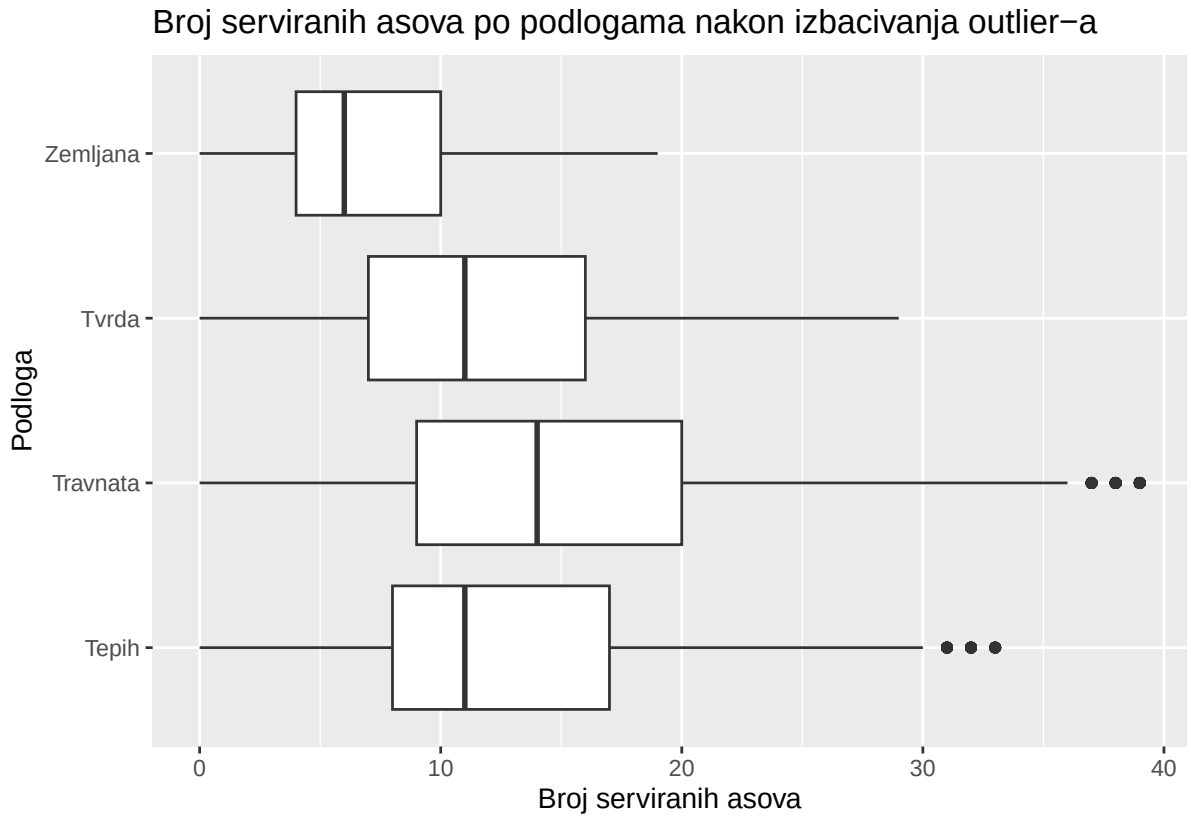


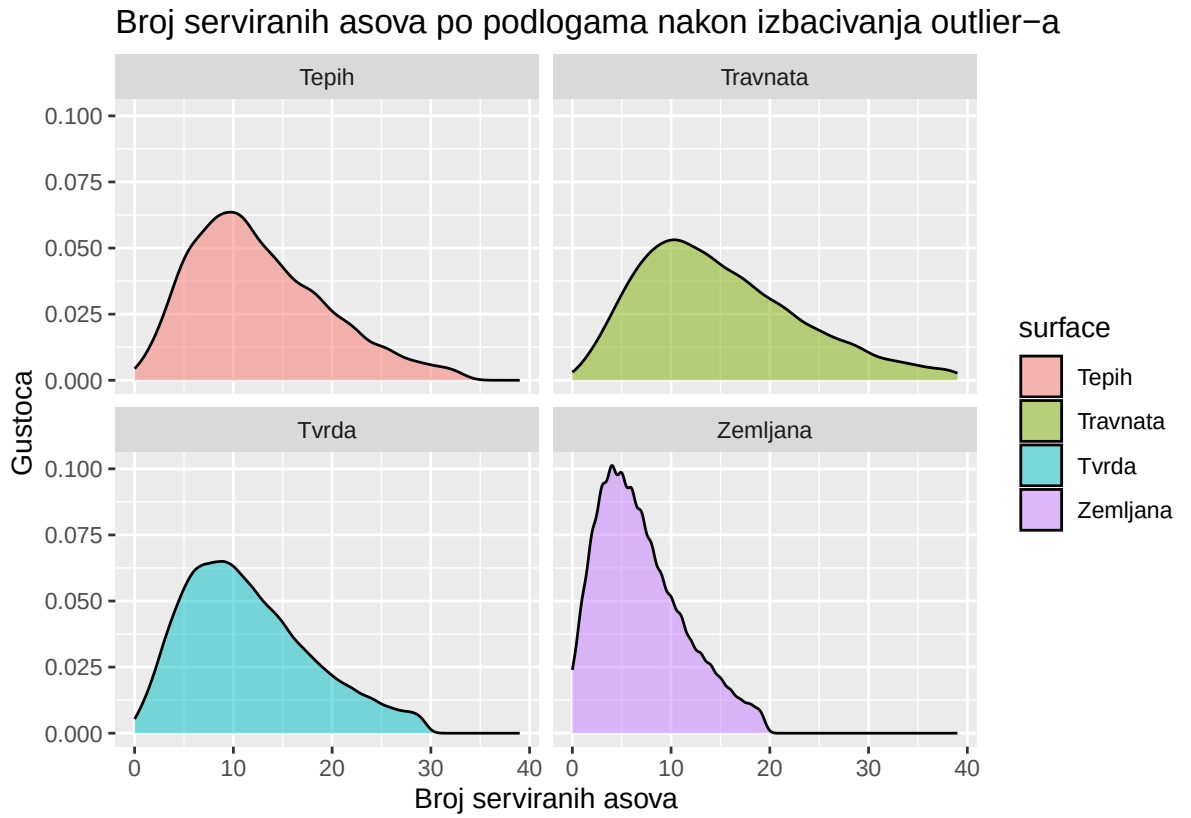
Iz prikazanih boxplot dijagrama i dijagrama gustoće možemo vidjeti kako je distribucija svih uzoraka jako nakošena u lijevo što znači da je većina uzoraka blizu minimalne vrijednosti.

Kako bismo pokušali normalizirati distribucije, možemo izbaciti sve observacije s vrijednostima izvan 1.5 interkvartilnog raspona (outlier-e).

Table 2: Servirani asovi po podlogama nakon izbacivanja outlier-a

surface	mean	sd	median	min	max	count
Tepih	12.68	6.79	11	0	33	5729
Travnata	15.08	8.13	14	0	39	9470
Tvrda	11.59	6.39	11	0	29	45839
Zemljana	6.91	4.35	6	0	19	29866





Nakon izbacivanja outlier-a, distribucije su i dalje nakošene u lijevo, ali je to manje izraženo nego prije. Također možemo vidjeti kako je sada varijanca svake podloge manja nego prije.

4.3.3 Statističko testiranje

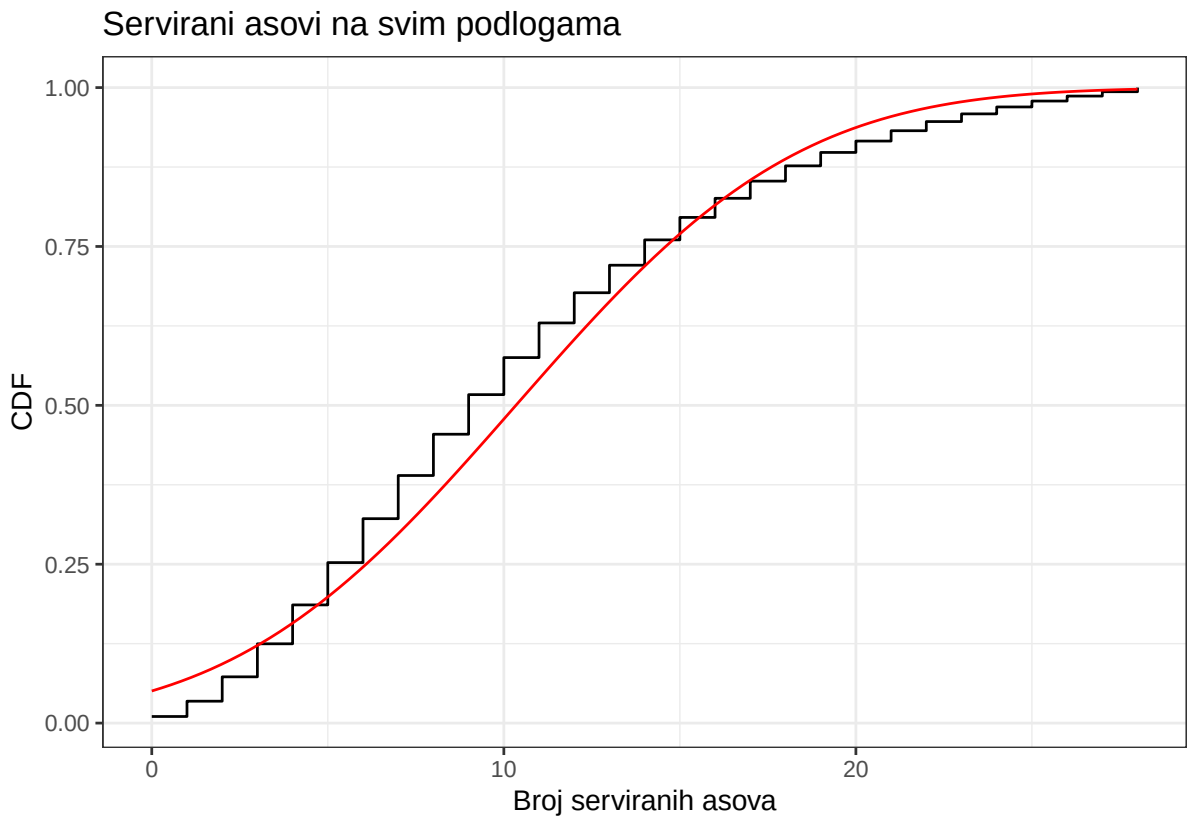
Nakon vizualiziranja podataka i izbacivanja outlier-a, možemo provesti statističko testiranje. Budući da distribucije uzoraka ne izgledaju kao da potječu iz normalne distribucije, prvo ćemo provjeriti njihovu normalnost, a zatim na temelju ishoda testa odlučiti koji test ćemo koristiti za provođenje statističkog testiranja.

4.3.3.1 Testiranje normalnosti Za testiranje normalnosti koristit ćemo Kolmogorov-Smirnovljev test. Testirat ćemo normalnost svih uzoraka zajedno, a zatim svakog uzorka posebno.

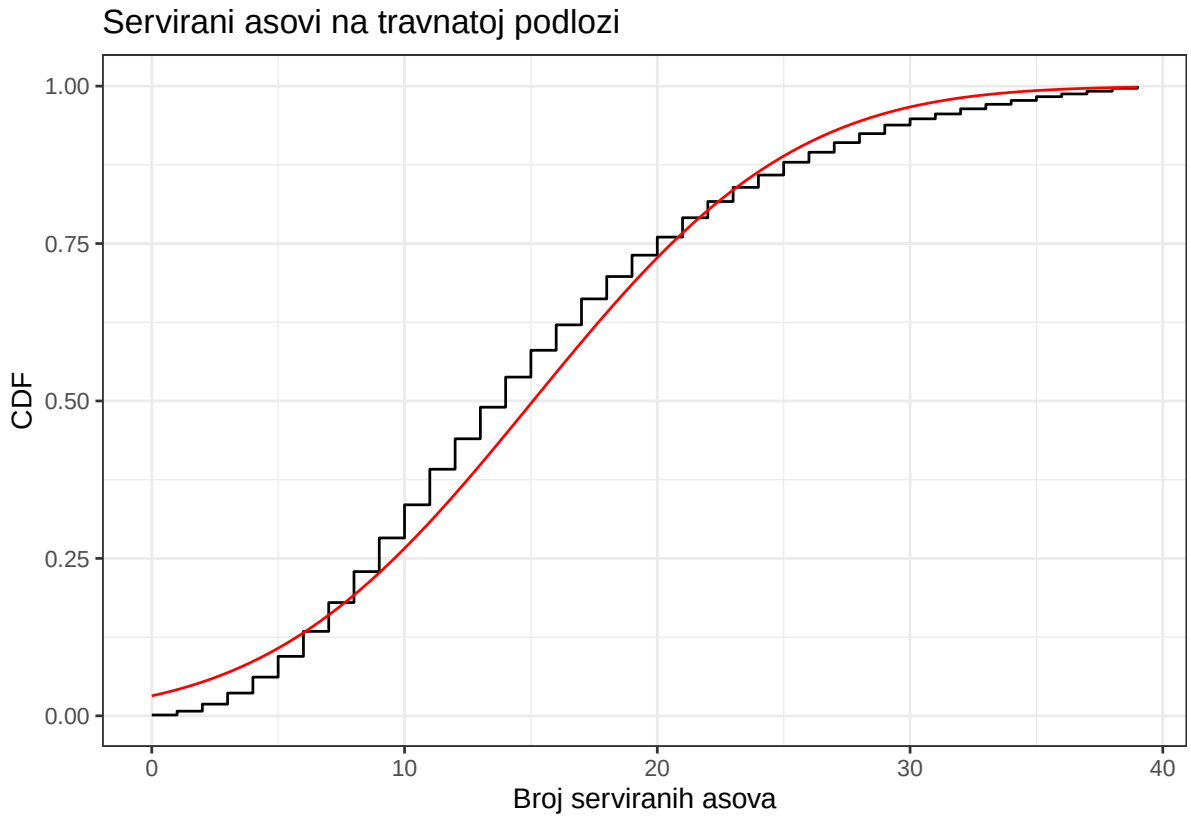
Prije testiranja, postavljamo hipoteze:

- H0: distribucija uzorka je normalna
- H1: distribucija uzorka nije normalna

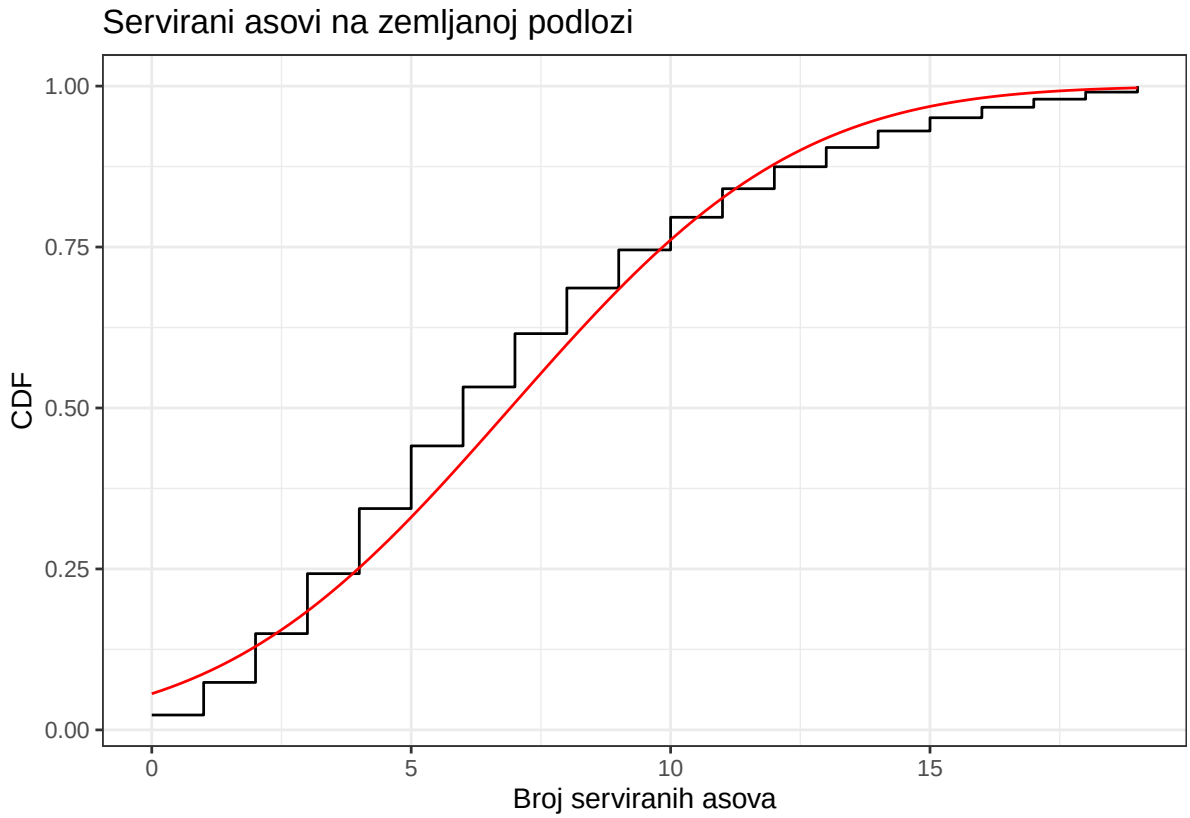
Za razinu značajnosti uzimamo 0.05.



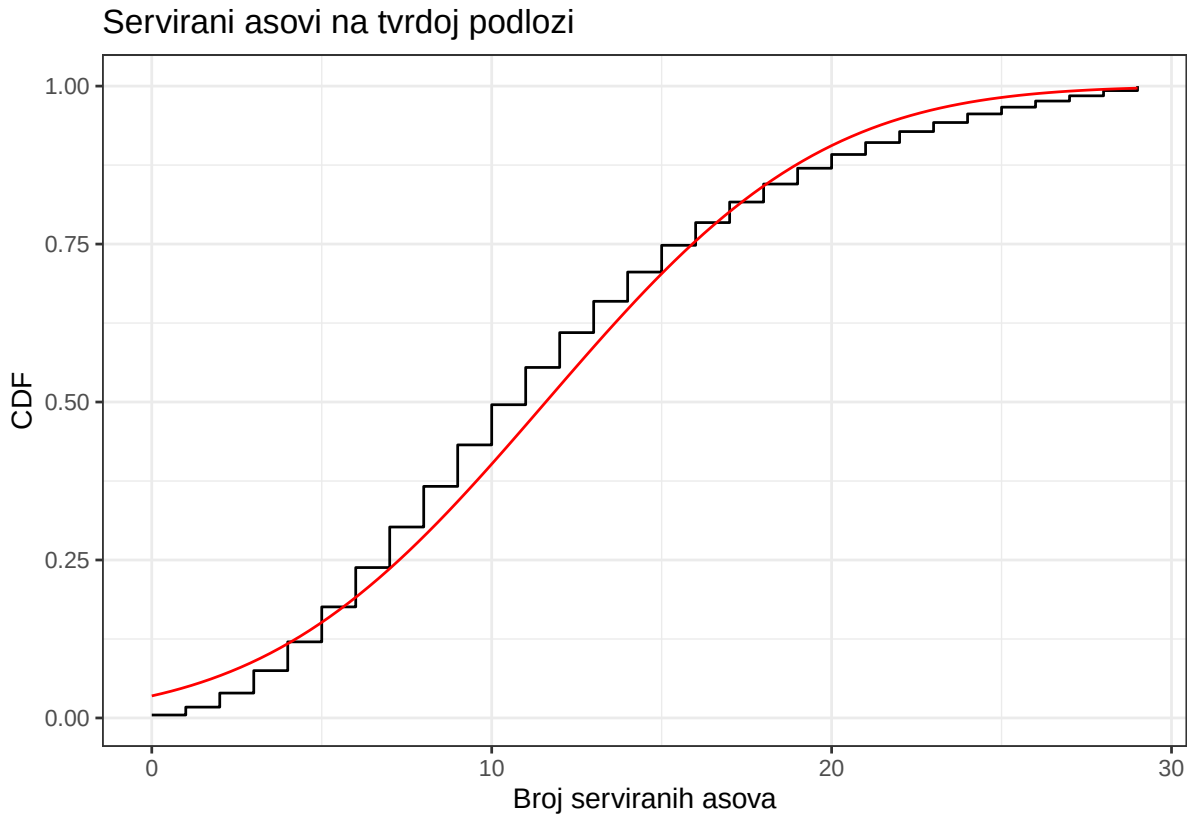

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  aces_all$aces
## D = 0.10114, p-value < 2.2e-16
## alternative hypothesis: two-sided
```



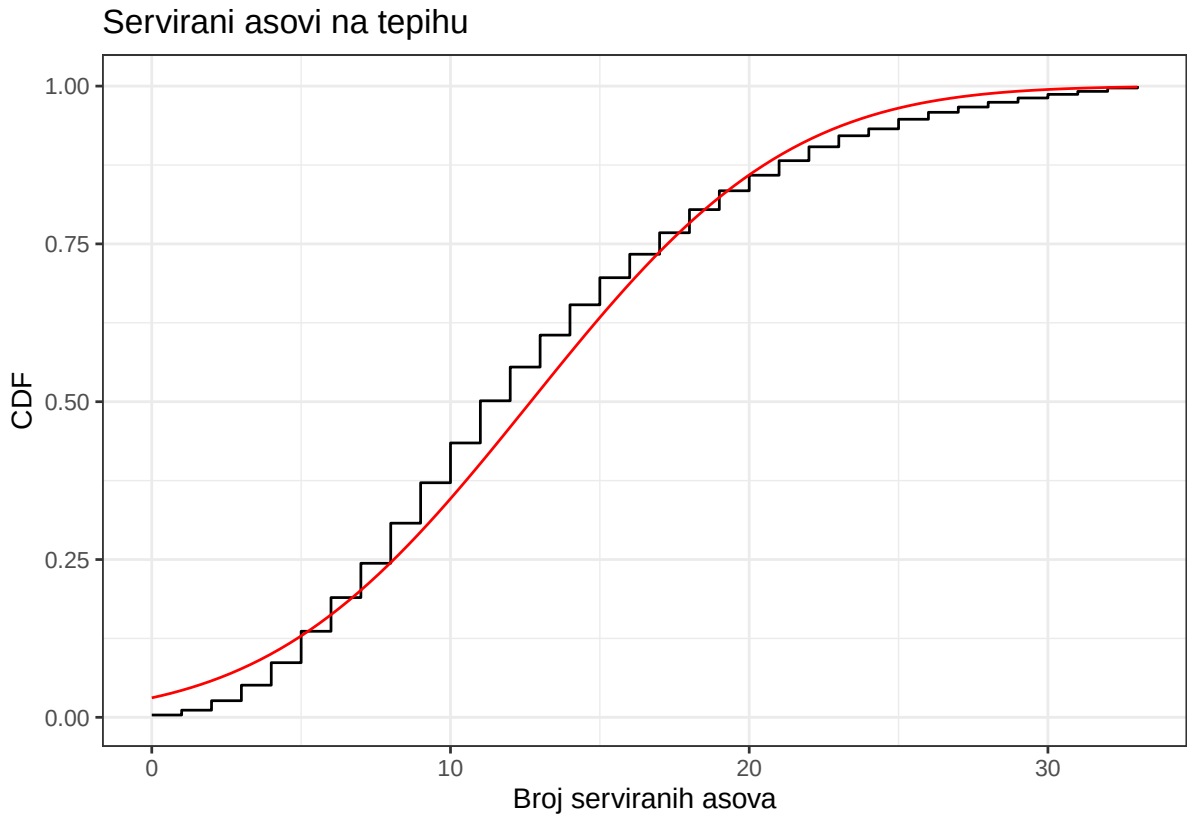

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  aces_grass$aces
## D = 0.091126, p-value < 2.2e-16
## alternative hypothesis: two-sided
```




```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  aces_clay$aces
## D = 0.11556, p-value < 2.2e-16
## alternative hypothesis: two-sided
```




```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  aces_hard$aces
## D = 0.093996, p-value < 2.2e-16
## alternative hypothesis: two-sided
```




```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  aces_carpet$aces
## D = 0.099377, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Iz prikazanih dijagrama i rezultata testiranja možemo vidjeti kako distribucije uzoraka nisu normalne. Stoga ćemo za statističko testiranje koristiti neparametarske testove.

4.3.3.2 Neparametarsko testiranje Za neparametarsko testiranje koristit ćemo Kruskal-Wallisov test. Prije testiranja, postavljamo hipoteze:

- H0: nema razlike u broju serviranih asova na različitim podlogama
- H1: postoji razlika u broju serviranih asova na različitim podlogama

Za razinu značajnosti uzimamo 0.05.

```
##
##  Kruskal-Wallis rank sum test
##
## data:  aces by surface
## Kruskal-Wallis chi-squared = 15060, df = 3, p-value < 2.2e-16
```

Iz rezultata testiranja možemo vidjeti kako je p-vrijednost manja od 0.05, stoga odbacujemo nultu hipotezu i zaključujemo kako postoji razlika u broju serviranih asova na različitim podlogama.

Kako bismo saznali na kojim podlogama postoji razlika u broju serviranih asova, provest ćemo post-hoc testiranje. Za post-hoc testiranje koristit ćemo Dunnov test.

```
##
##                             Comparison of x by group
##                                   (Bonferroni)
## Col Mean-|
## Row Mean |      Tepih   Travnata      Tvrda
```

```
## ---------+---------------------------------
## Travnata |  -14.68551
##          |    0.0000*
##          |
##    Tvrda |   10.50965   34.82310
##          |    0.0000*    0.0000*
##          |
## Zemljana |   62.56852   97.36533   101.5528
##          |    0.0000*    0.0000*    0.0000*
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

Iz rezultata testiranja možemo vidjeti kako je p-vrijednost manja od 0.05 za sve kombinacije podloga, stoga odbacujemo nultu hipotezu za sve kombinacije podloga i zaključujemo kako postoji razlika u broju serviranih asova na svim kombinacijama podloga.

4.4 Zaključak

Na temelju provedene statističke analize, možemo zaključiti kako postoji razlika u broju serviranih asova na različitim podlogama. Štoviše, postoji razlika u broju serviranih asova na svim kombinacijama podloga. Najviše asova servirano je na travi, a najmanje na zemlji. Također, najveća varijanca broja serviranih asova po meču je na travi, a najmanja na tepihu.

5 4. istraživačko pitanje: Kakva je veza između vrste terena i vjerojatnosti da će mečevi otići u 5. set?

Ovo pitanje možemo riješiti pomoću hi-kvadrat testa homogenosti, s tim da moramo paziti da određene pretpostavke budu zadovoljene. Kako bismo to mogli provjeriti, prvo moramo izolirati željene podatke u obliku pogodnom za obradu. Za početak, idemo učitati sve potrebne podatke:

```
matches <- fetch_data(1968:2023)
matches <- matches[, c("surface", "score", "best_of")]
```

Zatim moramo eliminirati neregularne obzervacije: mečeve za koje ne znamo podlogu na kojoj su odigrani i/ili mečevi u kojima teoretski nije bilo moguće odigrati 5 setova (zbog samog formata natjecanja):

```
matches %>% filter(!is.na(surface), best_of == 5) -> matches
matches$best_of <- NULL
```

Sada dijelimo mečeve u 2 skupa: oni u kojima je odigrano svih 5 setova i oni u kojima nije (NAPOMENA: u ovom 2. slučaju pritom ignoriramo mečeve koji su otkazani zbog neprimjerenog ponašanja igrača, pogreške sudca u organizacijskom smislu, zbog loših vremenskih uvjeta i raznih drugih okolnosti koje "nisu vezane uz sam sport"):

```
fiveSetMatchesIndices <- which(
  !str_detect(matches$score, "[^0-9- ]") & str_count(matches$score, "-") == 5)
fiveSetsMatches <- matches[fiveSetMatchesIndices, ]

lessThanFiveSetsMatches <- matches[-fiveSetMatchesIndices, ]
lessThanFiveSetsMatches <-
  filter(lessThanFiveSetsMatches,
  !str_detect(score,
              "(Played and unfinished|Played and abandoned|W/O|UNK|DEF|Default)"
              ))
```

Sada gradimo kontingencijsku tablicu:

```
fiveSetsMatches %>%
  group_by(surface) %>%
  summarise(count = n()) -> fiveSetsMatches
```

```
lessThanFiveSetsMatches %>%
  group_by(surface) %>%
  summarise(count = n()) -> lessThanFiveSetsMatches

data <- as.table(
  rbind(pull(fiveSetsMatches, count), pull(lessThanFiveSetsMatches, count))
)
colnames(data) <- c("Carpet", "Clay", "Grass", "Hard")
rownames(data) <- c("5 sets played", "Less than 5 sets played")
probabilities <- data[1, ] / data[2, ]
```

Sada provodimo test i gledamo ispise:

```
test <- chisq.test(data)
test$observed
```

```
##                           Carpet  Clay Grass  Hard
## 5 sets played                186  1714  1377  1498
## Less than 5 sets played     1402 12052  8879 13483
```

```
test$expected
```

```
##                              Carpet     Clay    Grass      Hard
## 5 sets played              186.8074  1619.39 1206.484  1762.319
## Less than 5 sets played 1401.1926 12146.61 9049.516 13218.681
```

```
test$residuals
```

```
##                               Carpet        Clay       Grass        Hard
## 5 sets played            -0.05907485  2.35105300  4.90912192 -6.29630184
## Less than 5 sets played   0.02157003 -0.85844123 -1.79247029  2.29897203
```

```
test$stdres
```

```
##                               Carpet        Clay       Grass        Hard
## 5 sets played            -0.06415712  3.07881622  6.04537198 -8.43863029
## Less than 5 sets played   0.06415712 -3.07881622 -6.04537198  8.43863029
```

```
test
```

```
##
##  Pearson's Chi-squared test
##
## data:  data
## X-squared = 78.509, df = 3, p-value < 2.2e-16
```

S obzirom na to da je vrijednost svake ćelije frekvencija, da svaka obzervacija pripada jednoj ćeliji te da je očekivana vrijednost svake ćelije >= 5, možemo zaključiti da su rezultati ovog testa validni te da vjerojatnost da će meč otići u 5. set ne ovisi o vrsti terena (jako mala P-vrijednost: 2.2e-16)

6 5. istraživačko pitanje: Možemo li procijeniti broj asova koje će igrač odservirati u tekućoj (zadnjoj dostupnoj sezoni) na temelju njegovih rezultata iz prethodnih sezona?

6.1 Previđanje broja aseva u trenutnoj sezoni na temelju podataka iz prethodnih sezona

Statistika je jedna od glavnih znanosti koje se vežu uz sport. Jedan od najpoznatijih pojmova koji se vežu uz tenis je zasigurno pojam asa. Pojam označava uspješan servis pri kojem protivnik nije dotaknuo serviranu lopticu. Asevi su najsigurniji i gotovo najlakše dobiveni poeni u tenisu, gotovo poput zakucavanja u košarci.

Postavlja se pitanje: možemo li uz pomoć programskog jezika R i statistika o ATP mečevima od 1968. do 2022. godine predvidjeti koliko će aseva određeni igrač imati u tekućoj sezoni?

Pogledamo li dataframe s podacima nakon uvoza svih .csv datoteka, primjećujemo da se statistika o broju aseva po meču za pobjednika i gubitnika počela bilježiti tek 1991. godine. Stoga, kada uvozimo podatke, izuzimamo sve retke čija je vrijednost varijable w_ace ili l_ace jednaka NA.

```
# Dohvaćanje popisa .csv datoteka u direktoriju
tennis_files <- list.files(path = "./ATP-Matches/",
                           pattern = "\\.csv$", full.names = TRUE)

# Inicijalizacija praznog dataframe-a za čuvanje podataka
tennis_dataset <- data.frame()

# Iteracija kroz svaku .csv datoteku
for (datoteka in tennis_files) {
  # Učitavanje .csv datoteke
  podaci <- read.csv(datoteka)

  # Uklanjanje redova s NA vrijednostima u w_ace i l_ace
  podaci_bez_na <- podaci[complete.cases(podaci$w_ace, podaci$l_ace), ]

  # Dodavanje učitanih podataka u dataset
  tennis_dataset <- rbind(tennis_dataset, podaci_bez_na)
}
```

6.2 Odabir potrebnih parametara

Da bismo mogli kreirati model višestruke linearne regresije, moramo znati koje od dostupnih varijabli utječu na traženu zavisnu varijablu.

Pogledamo li dostupne varijable iz zadanog dataseta, vidimo da možemo eliminirati podatke o identifikaciji meča, kao što su podaci o rundi turnira, rezultat, država iz koje igrač dolazi, način na koji je igrač došao na turnir...

Zadržat ćemo podatke o performansu igrača u meču:

- broj aseva
- dvostruke pogreške
- uspješnih prvih i drugih servisa
- dobivenih prvih servisa
- serviranih gejmova
- odigranih *breakova* i spašenih *breakova*

Informacije o visini i dominantnoj ruci igrača nećemo uzimati u obzir budući da, iako imaju utjecaj na servis igrača i samim time na broj aseva koje igrač postigne u karijeri, ostaju iste kroz karijeru pojedinca.

```
# Kreirajte dataframe za pobjednike
winners <- tennis_dataset[, c("winner_id", "winner_name", "tourney_id",
                              "surface","winner_age",  "winner_rank",  "w_ace",
                              "w_df", "w_svpt", "w_1stIn", "w_1stWon", "w_2ndWon",
                              "w_SvGms", "w_bpSaved", "w_bpFaced", "minutes" )]

colnames(winners) <- c("player_id", "name", "tour_id", "surface",  "age",
                       "rank", "ace", "double_fault", "saved points",
                       "1st serves made", "1st serves won", "2nd serves won",
                       "served games", "saved breaks", "faced breaks", "minutes")

# Kreirajte dataframe za gubitnike
```

```
losers <- tennis_dataset[, c("loser_id", "loser_name", "tourney_id", "surface",
                             "loser_age",  "loser_rank",  "l_ace", "l_df",
                             "l_svpt", "l_1stIn", "l_1stWon", "l_2ndWon",
                             "l_SvGms", "l_bpSaved", "l_bpFaced", "minutes" )]

colnames(losers) <- c("player_id", "name", "tour_id", "surface", "age", "rank",
                      "ace", "double_fault", "saved points", "1st serves made",
                      "1st serves won", "2nd serves won", "served games",
                      "saved breaks", "faced breaks", "minutes")

all_players_stats <- rbind(winners, losers)

stats_per_player <- split.data.frame(all_players_stats, all_players_stats[["name"]])
```

6.3 Izbor igrača na temelju čijih se podata izrađuje model

Nakon što smo napravili potrebne redukcije i transformacije na originalnom datasetu, dobili smo popis sa 2425 elemenata, tj. popis od 2425 igrača. Za svakog od njih imamo podatke o svim teniskim mečevima koje su odigrali u sklopu ATP turnira. Budući da ne možemo napraviti linearnu regresiju s malim brojem mečeva, izbacujemo sve igrače za koje imamo manje od 30 mečeva.

```
reduced_stats_per_player <-
  Filter(function(dataset) nrow(dataset) >= 30, stats_per_player)
```

Nakon odabira igrača na temelju čijih podataka će se napraviti model linearne regresije, uvodimo novu varijablu "year" na temelju koje ćemo razdvojiti podatke u set za treniranje modela i set za testiranje modela.

Kao primjer uzet ćemo podatke za Davida Ferrera. David Ferrer je odabran primarno zbog velikog broja dostupnih podataka.

```
player_stats <- subset(reduced_stats_per_player$`David Ferrer`)
```

Nakon podijele dostupnog dataseta na dva podseta koji će poslužiti za "trening" modela i kasnije testiranje istog, prelazimo na identifikaciju vrijednosti koje uistinu utječu na servis i broj aseva.

6.4 Obrada parametara

Započnimo s uvođenjem dummy varijabli za kategorijsku varijablu surface:

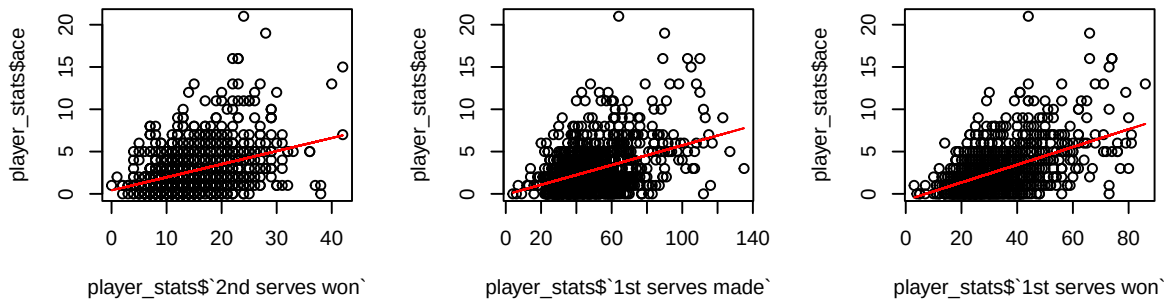
```
#Dummy varijable za surface
dummy_df <- data.frame(model.matrix(~ surface - 1, data = player_stats))
```

Prelazimo na ispitivanje koje od dostupnih varijabli uistinu utječu na broj aseva u sezoni. Pretpostavka je da će nam bitne biti isključivo varijable vezane uz uspješnost servisa, učestalost servisa, te uz vrstu podloge na kojoj se igra. Provjerimo također imaju li iskustvo i rang na ljestvici na kojem se igrač nalazi koorelaciju s brojem aseva u meču.

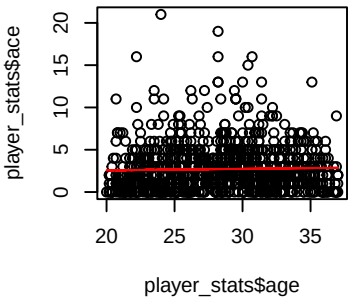
```
linearnost_2serv <- lm(ace~train_data$`2nd serves won`, data = as.data.frame(train_data))
```

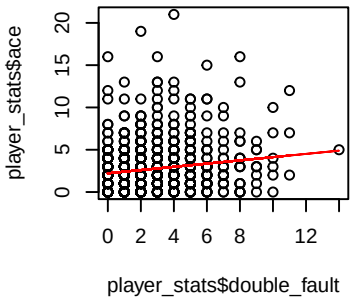
```
par(mfrow = c(2, 3))

plot(player_stats$`2nd serves won`, player_stats$ace)
lines(player_stats$`2nd serves won`, linearnost_2serv$fitted.values, col='red')

plot(player_stats$`1st serves made`, player_stats$ace)
lines(player_stats$`1st serves made`, linearnost_1serv$fitted.values, col='red')

plot(player_stats$`1st serves won`, player_stats$ace)
lines(player_stats$`1st serves won`, linearnost_1won$fitted.values, col='red')
```

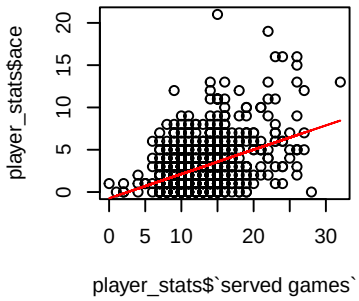
```r
par(mfrow = c(2, 3))
```



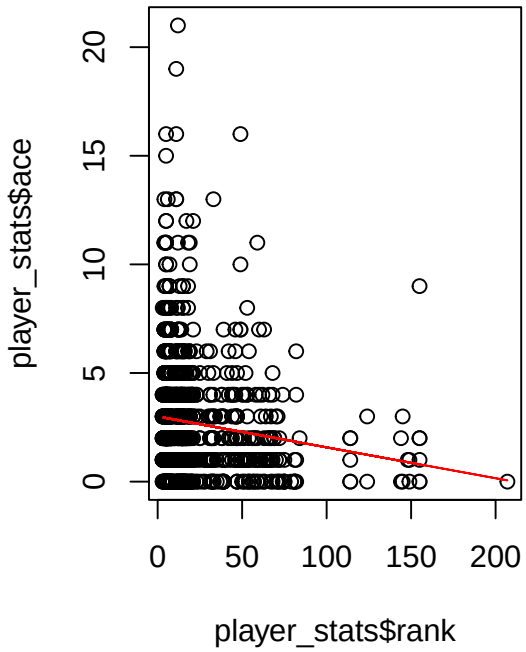

```r
plot(player_stats$age, player_stats$ace)
lines(player_stats$age, linearnost_godina$fitted.values, col='red')


plot(player_stats$double_fault, player_stats$ace)
lines(player_stats$double_fault, linearnost_dvostrukih$fitted.values, col='red')

plot(player_stats$`served games`, player_stats$ace)
lines(player_stats$`served games`, linearnost_svg$fitted.values, col='red')

par(mfrow = c(1, 2))
```








```
plot(player_stats$rank, player_stats$ace)
lines(player_stats$rank, linearnost_ranga$fitted.values, col='red')
```



Vidimo da, kao što je pretpostavljeno, podaci kao što su broj serviranih gejmova, broj uspješnih prvih servisa, broj osvojenih prvih servisa, broj drugih servisa, te dvostrukih pogrešaka imaju linearnu razdiobu

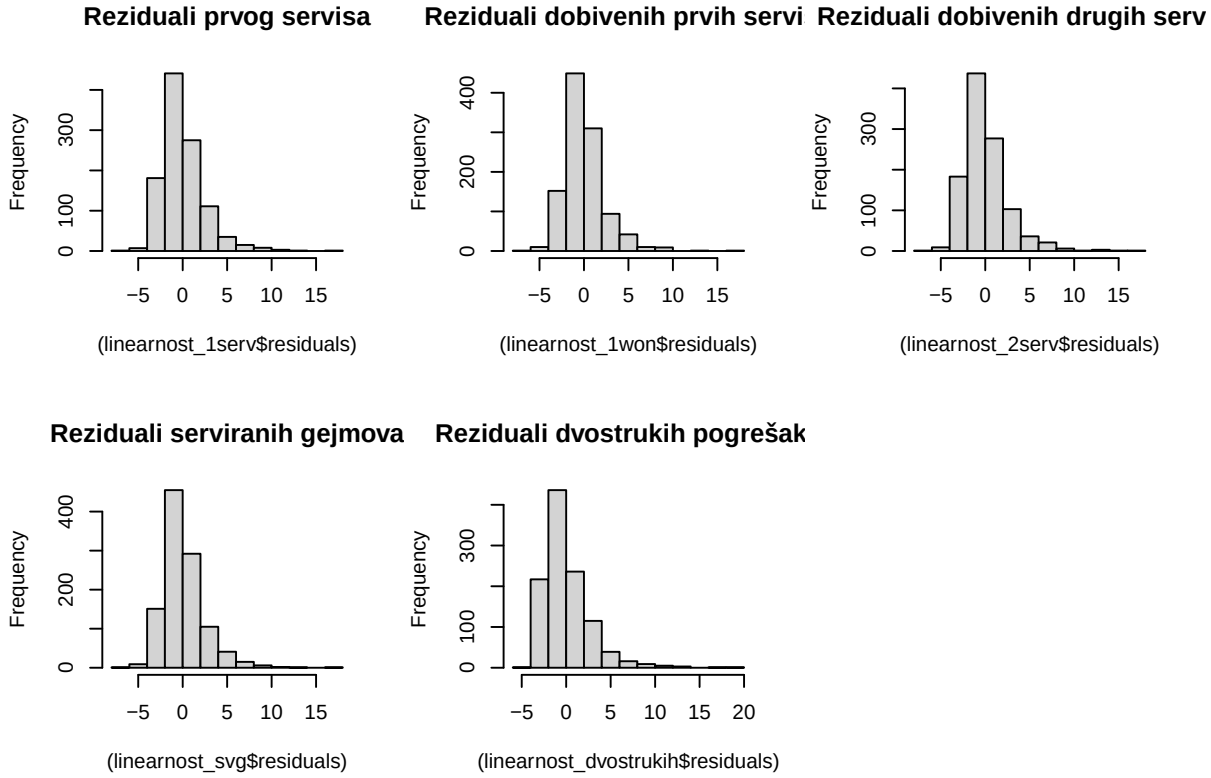
u ovisnosti o broju aseva u meču.

Na dijagramima za rang i godine nije vidljiva značajna koorelacija, tako da te dvije varijable izbacujemo iz modela.

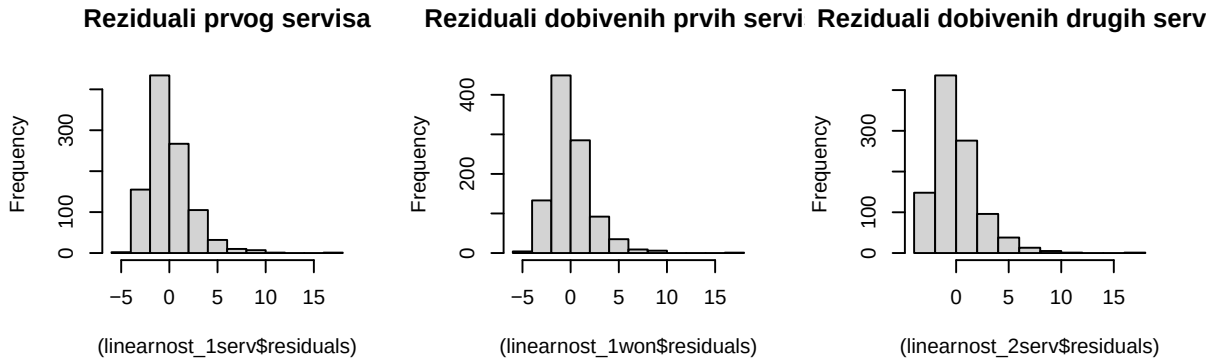
6.5 Ispitivanje normalnosti reziduala

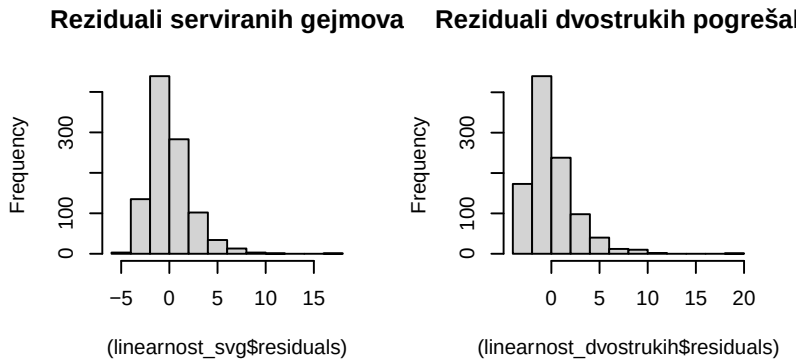
Budući da je jedan od uvjeta linearne regresije normalnost reziduala, provjeravamo reziduale jednostavnih linearnih regresija za svaku od potrebnih varijabli. Idealno, to se može napraviti korištenjem Kolmogorov-Smirnovljevog testa, no on je osjetljiv na velike uzorke. Ovdje su već na početku eliminirani svi igrači koji imaju manje od 30 odigranih mečeva, tako da normalnost nikako nećemo testirati Kolmogorov-Smirnovljevim testom.

Radi praktičnosti, koristimo histograme i provjeravamo oblikuju li reziduali zvonoliku krivulju:



Vidimo da sve varijable daju zvonoliku krivulju uz poneke "repiće". Njih tretiramo izbacivanjem outliera iz seta podataka. To će nam omogućiti dobivanje boljeg modela višestruke regresije.





6.6 Kreiranje modela višestruke linearne regresije

Na temelju selektiranih podataka izrađujemo model višestruke regresije. Nakon dobivanja modela, procjenjujemo kvalitetu istog. Osim uobičajenih parametara kvalitete testa koje dobivamo korištenjem funkcije summary(), za procjenu modela linearne regresije koristimo i metodu najmanjih kvadrata (SSE) i srednju kvadratnu pogrešku (MSE).

```
# Kreiranje modela na temelju podataka iz trening-seta
model_igraca <- lm(ace ~ train_data$surface + train_data$`1st serves made` +
                     train_data$`1st serves won` + train_data$double_fault +
                     train_data$`2nd serves won`+ train_data$`served games` , data = as.data.frame(t

summary(model_igraca)
```

```
##
## Call:
## lm(formula = ace ~ train_data$surface + train_data$`1st serves made` +
##     train_data$`1st serves won` + train_data$double_fault + train_data$`2nd serves won` +
##     train_data$`served games`, data = as.data.frame(train_data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8531 -1.3115 -0.2536  0.9890 17.5934
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.614052   0.574605   2.809 0.005066 **
## train_data$surfaceClay        -2.084018   0.532853  -3.911 9.81e-05 ***
## train_data$surfaceGrass       -1.104258   0.589565  -1.873 0.061358 .
## train_data$surfaceHard        -1.422974   0.530763  -2.681 0.007461 **
## train_data$`1st serves made` -0.085657   0.012093  -7.083 2.64e-12 ***
## train_data$`1st serves won`   0.194977   0.016880  11.551  < 2e-16 ***
```

```
## train_data$double_fault       0.001901   0.037273   0.051 0.959342
## train_data$`2nd serves won`   0.058763   0.017207   3.415 0.000663 ***
## train_data$`served games`    -0.042552   0.050414  -0.844 0.398841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.021 on 1005 degrees of freedom
## Multiple R-squared:  0.2622, Adjusted R-squared:  0.2563
## F-statistic: 44.64 on 8 and 1005 DF,  p-value: < 2.2e-16
```

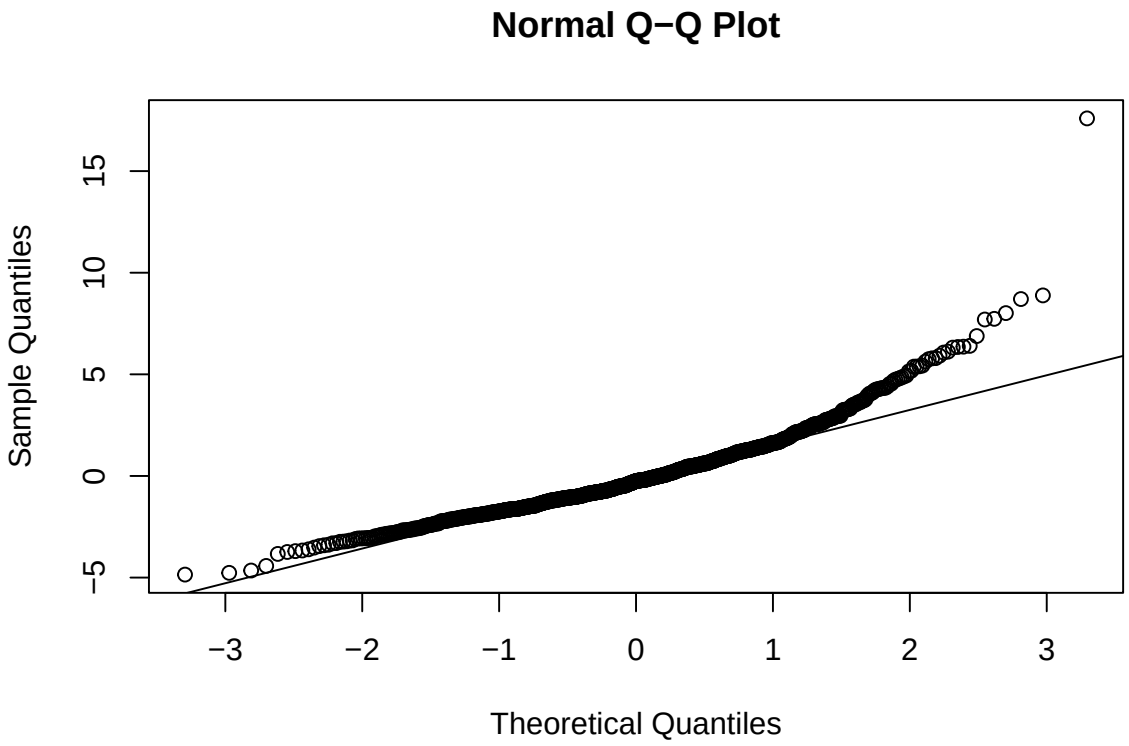
```r
# Procjena performansi modela na test setu
predicted_aces_test <- predict(model_igraca, predicted_data = as.data.frame(test_data))
mse <- mean((test_data$ace - predicted_aces_test)^2)
sse <- ((test_data$ace - predicted_aces_test)^2)
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 7.32700906349629"
```

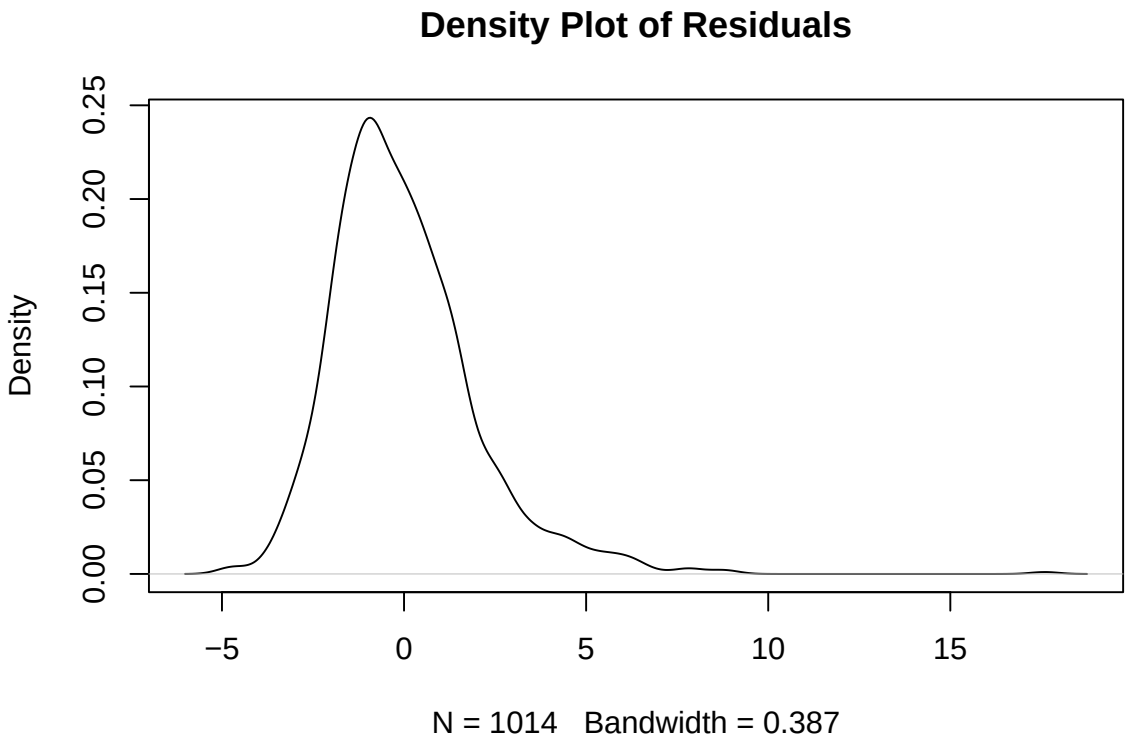
```r
tail(paste("Squared Sum Error (SSE): ", sse))
```

```
## [1] "Squared Sum Error (SSE):  0.67878550606576"
## [2] "Squared Sum Error (SSE):  3.11274849359583"
## [3] "Squared Sum Error (SSE):  3.03312841903801"
## [4] "Squared Sum Error (SSE):  28.9298423931807"
## [5] "Squared Sum Error (SSE):  0.995343507361965"
## [6] "Squared Sum Error (SSE):  0.00143673497260423"
```

```r
qqnorm(residuals(model_igraca))
qqline(residuals(model_igraca))
```



```r
plot(density(residuals(model_igraca)), main = "Density Plot of Residuals")
```



6.6.1 Pokazatelji kvalitete modela

Da bi model bio dobar, vrijednosti SSE-a i MSE-a moraju biti što bliže nuli. Vidljivo je da, iako SSE uvelike varira u svojim vrijednostima, MSE ima relativno prihvatljiv rezultat od 7.71. Premda je ta brojka zadovoljavajuća, pogled na druge parametre modela ukazuje na činjenicu da odabrane nezavisne varijable nisu idealan pokazatelj kvalitete servisa igrača. Pogledamo li p-vrijednosti koeficijenata, vidimo da u slučaju Davida Ferrera možemo izbaciti dvostruke pogreške i broj serviranih gejmova iz modela.

6.6.2 Prilagođavanje odabira parametara

Napravimo li model bez te dvije varijable, vidmo da se vrijednost MSE-a poveća na 9.16.

Još jedan pokazatelj kvalitete modela je višestruki R kvadrat i prilagođeni R kvadrat. Oni ukazuju na kvalitetu izabranih prediktora u modelu. Raspon im je od 0 do 1, gdje 1 ukazuje na bolje odabrane prediktore i model koji dobro predviđa tražene vrijednosti. Prilagođeni R kvadrat kažnjava nepotrebno dodane prediktore. Uistinu, nakon uklanjanja dvostrukih pogrešaka i serviranih gejmova iz modela, dobivamo bolju vrijednost R kvadrata.Ta vrijednost niti nakon uklanjanja nepotrebnih prediktora nije dobra, što ukazuje na činjenicu da postoje faktori koje nismo uključili u model.

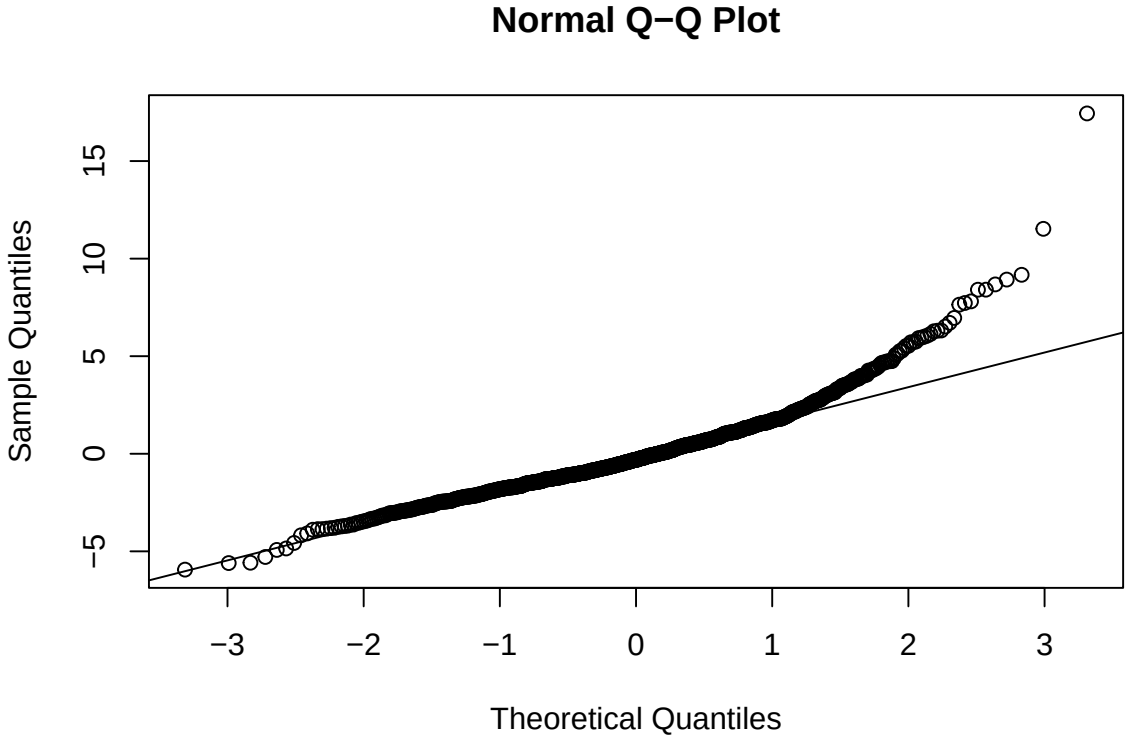
```r
# Kreiranje modela na trening setu
model_igraca <- lm(ace ~ train_data$surface + train_data$`1st serves made` +
                     train_data$`1st serves won` +
                     train_data$`2nd serves won`, data = as.data.frame(train_data))

summary(model_igraca)
```

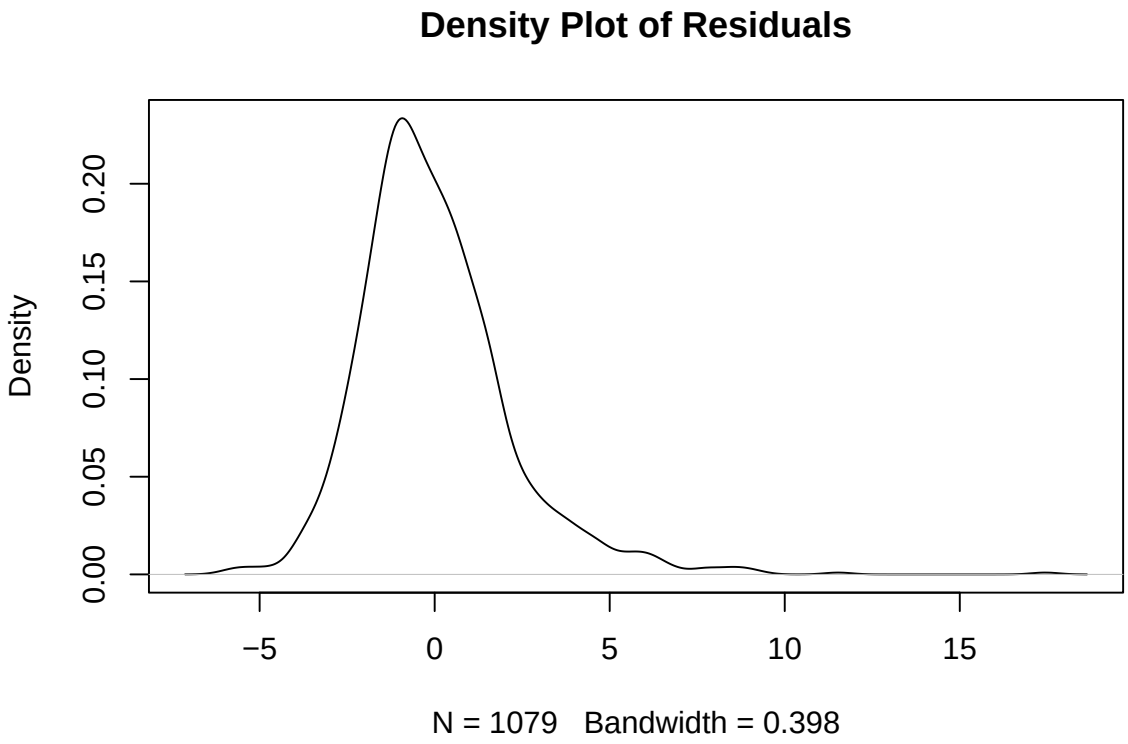
```
##
## Call:
## lm(formula = ace ~ train_data$surface + train_data$`1st serves made` +
##     train_data$`1st serves won` + train_data$`2nd serves won`,
##     data = as.data.frame(train_data))
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9373 -1.3351 -0.3041  1.0603 17.4414
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.03249    0.57121   1.808 0.070959 .
## train_data$surfaceClay        -1.97981    0.55297  -3.580 0.000359 ***
## train_data$surfaceGrass       -0.45758    0.60587  -0.755 0.450263
## train_data$surfaceHard        -1.35655    0.55128  -2.461 0.014022 *
## train_data$`1st serves made` -0.08671    0.01107  -7.836 1.12e-14 ***
## train_data$`1st serves won`   0.20254    0.01631  12.418  < 2e-16 ***
## train_data$`2nd serves won`   0.04765    0.01391   3.426 0.000635 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.171 on 1072 degrees of freedom
## Multiple R-squared:  0.335,  Adjusted R-squared:  0.3313
## F-statistic: 90.02 on 6 and 1072 DF,  p-value: < 2.2e-16
```

```
## [1] "Mean Squared Error (MSE) na test setu: 9.01624924380286"
```

```
## [1] "Squared Sum Error (SSE) na test setu:  3.98587663916027"
## [2] "Squared Sum Error (SSE) na test setu:  3.47281783109507"
## [3] "Squared Sum Error (SSE) na test setu:  2.69492554632227"
## [4] "Squared Sum Error (SSE) na test setu:  26.3153692175162"
## [5] "Squared Sum Error (SSE) na test setu:  1.22934330577954"
## [6] "Squared Sum Error (SSE) na test setu:  0.00514896526447333"
```





Osim parametara modela, gledamo i normalnost reziduala. Pogledamo li dijagram gustoće reziduala, vidimo da dobivamo zvonoliku krivulju, što ukazuje da su reziduali normalno distribuirani.

6.7 Možemo li zaključiti da je linearnom regresijom moguće predvidjeti broj aseva u sezoni?

Premda bismo mogli na temelju dosadašnjih saznanja mogli zaključiti da je moguće predvidjeti broj aseva koje će igrač odigrati u tekućoj sezoni, provođenjem izračuna na temelju podataka nekog drugog igrača, na primjer Novaka Đokovića, dolazimo do saznanja da do sada korišteni model ne funkcionira za sve igrače jednako:

```
player_stats <- subset(reduced_stats_per_player$`Novak Djokovic`)
```

```
# Kreiranje modela na temelju podataka iz trening-seta
model_igraca <- lm(ace ~ train_data$surface + train_data$`1st serves made` +
                     train_data$`1st serves won` + train_data$double_fault +
                     train_data$`2nd serves won`+ train_data$`served games` , data = as.data.frame(t

summary(model_igraca)
```

```
##
## Call:
## lm(formula = ace ~ train_data$surface + train_data$`1st serves made` +
##     train_data$`1st serves won` + train_data$double_fault + train_data$`2nd serves won` +
##     train_data$`served games`, data = as.data.frame(train_data))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.060 -1.925 -0.243  1.630 14.827
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                   2.91566    0.98588   2.957 0.003167 **
## train_data$surfaceClay       -3.84568    0.95594  -4.023 6.13e-05 ***
## train_data$surfaceGrass      -1.45839    0.98473  -1.481 0.138888
## train_data$surfaceHard       -2.39604    0.94599  -2.533 0.011450 *
## train_data$`1st serves made` -0.18851    0.01789 -10.535  < 2e-16 ***
## train_data$`1st serves won`   0.32875    0.02625  12.523  < 2e-16 ***
## train_data$double_fault      -0.02705    0.05689  -0.476 0.634486
## train_data$`2nd serves won`  -0.05431    0.02638  -2.059 0.039730 *
## train_data$`served games`     0.27727    0.07795   3.557 0.000391 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.811 on 1117 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3552
## F-statistic: 78.47 on 8 and 1117 DF,  p-value: < 2.2e-16
```

```
# Procjena performansi modela na test setu
predicted_aces_test <- predict(model_igraca,
                               predicted_data = as.data.frame(test_data))
mse <- mean((test_data$ace - predicted_aces_test)^2)
```

```
## Warning in test_data$ace - predicted_aces_test: longer object length is not a
## multiple of shorter object length
```

```
sse <- ((test_data$ace - predicted_aces_test)^2)
```

```
## Warning in test_data$ace - predicted_aces_test: longer object length is not a
## multiple of shorter object length
```

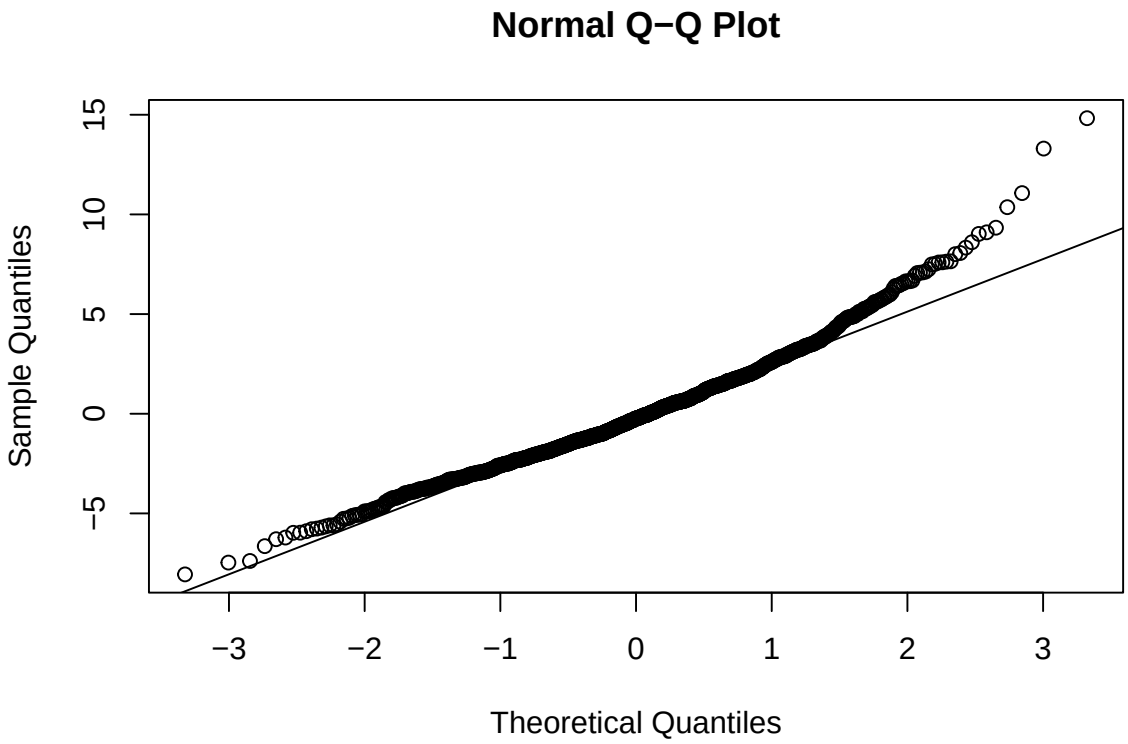
```
print(paste("Mean Squared Error (MSE):", mse))
```

```
## [1] "Mean Squared Error (MSE): 18.4153895671121"
```

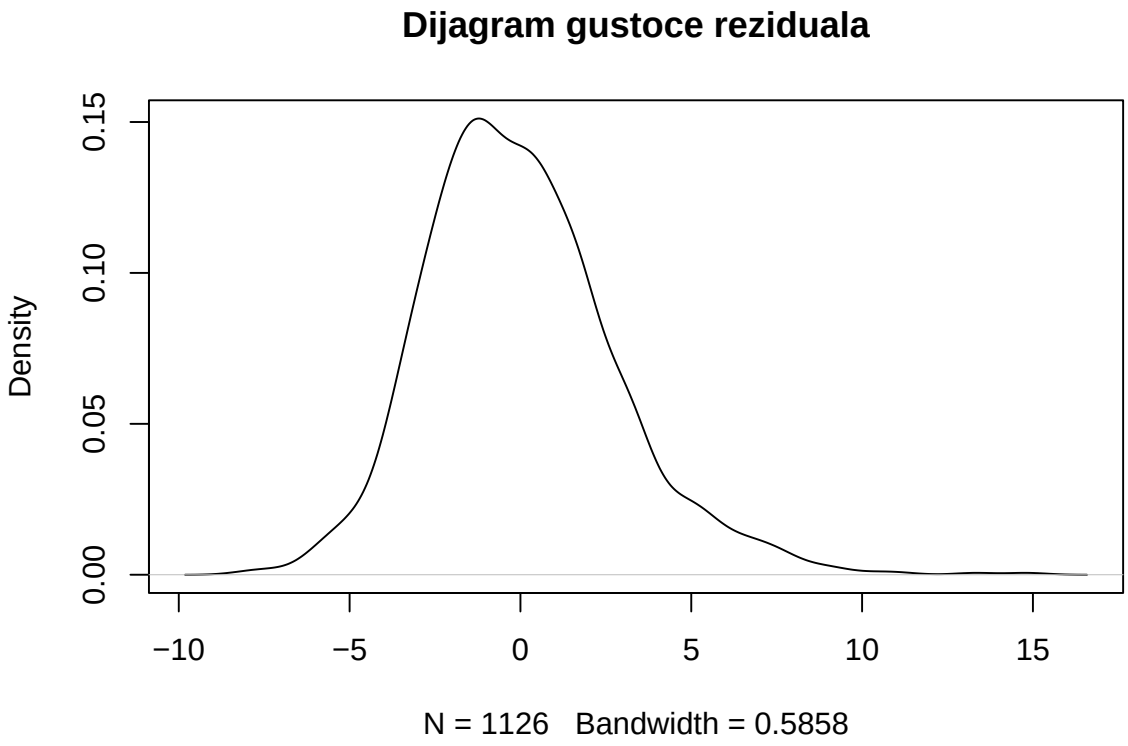
```
tail(paste("Squared Sum Error (SSE): ", sse))
```

```
## [1] "Squared Sum Error (SSE):  1.81947705437879"
## [2] "Squared Sum Error (SSE):  9.34803345867482"
## [3] "Squared Sum Error (SSE):  9.58189319076072"
## [4] "Squared Sum Error (SSE):  0.000416787451568511"
## [5] "Squared Sum Error (SSE):  2.73285168602346"
## [6] "Squared Sum Error (SSE):  20.1912737934176"
```

```
qqnorm(residuals(model_igraca))
qqline(residuals(model_igraca))
```




```
plot(density(residuals(model_igraca)), main = "Dijagram gustoće reziduala")
```



Pogledamo li sažetak modela za Novaka Đokovića, vidimo da kod njega možemo odbaciti dvostruke pogreške iz modela, dok je broj serviranih gejmova vrlo značajan za broj aseva. Ukanjanjem tog prediktora blago povećamo vrijednost R kvadrata i MSE-a.

```r
# Kreiranje modela na temelju podataka iz trening-seta
model_igraca <- lm(ace ~ train_data$surface + train_data$`1st serves made`+
                     train_data$`1st serves won` + train_data$`2nd serves won`+
                     train_data$`served games` , data = as.data.frame(train_data))

summary(model_igraca)
```

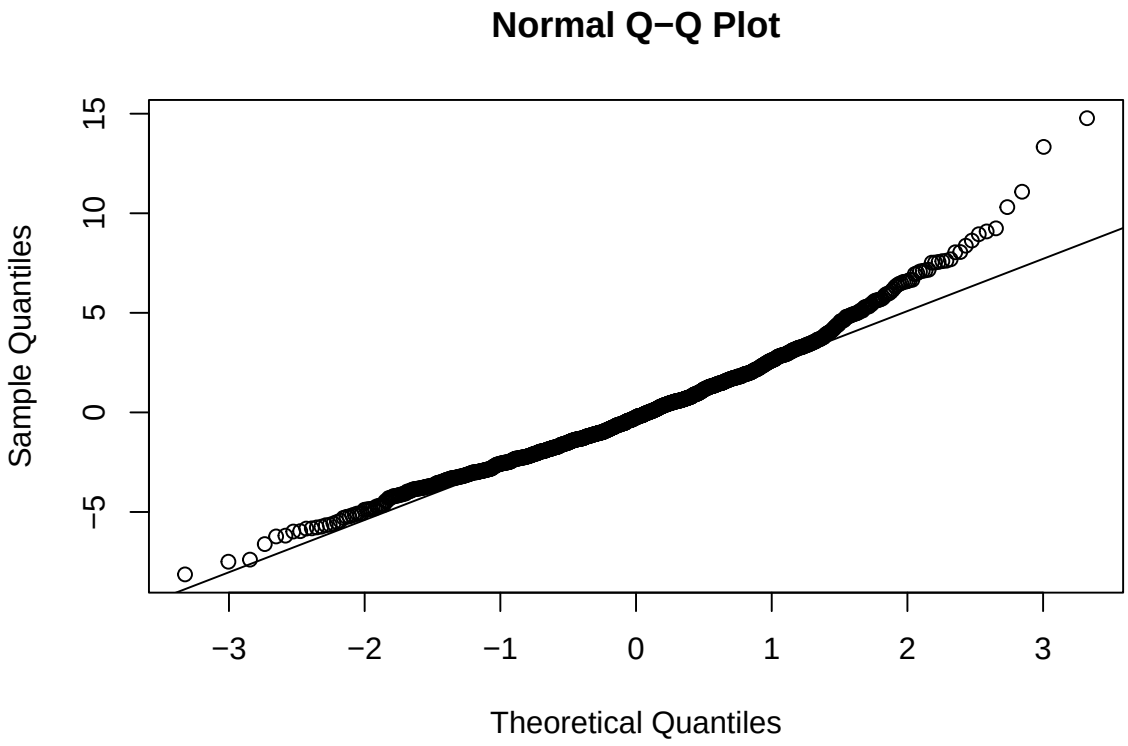
```
##
## Call:
## lm(formula = ace ~ train_data$surface + train_data$`1st serves made` +
##     train_data$`1st serves won` + train_data$`2nd serves won` +
##     train_data$`served games`, data = as.data.frame(train_data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1340 -1.9264 -0.2597  1.6122 14.7735
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    2.89891    0.98491   2.943 0.003314 **
## train_data$surfaceClay        -3.83361    0.95527  -4.013 6.39e-05 ***
## train_data$surfaceGrass       -1.45242    0.98431  -1.476 0.140340
## train_data$surfaceHard        -2.39224    0.94563  -2.530 0.011550 *
## train_data$`1st serves made` -0.18956    0.01775 -10.679  < 2e-16 ***
## train_data$`1st serves won`   0.33013    0.02608  12.658  < 2e-16 ***
## train_data$`2nd serves won`  -0.05507    0.02632  -2.092 0.036656 *
## train_data$`served games`     0.27459    0.07772   3.533 0.000428 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.81 on 1118 degrees of freedom
## Multiple R-squared:  0.3597, Adjusted R-squared:  0.3556
## F-statistic: 89.71 on 7 and 1118 DF,  p-value: < 2.2e-16
```

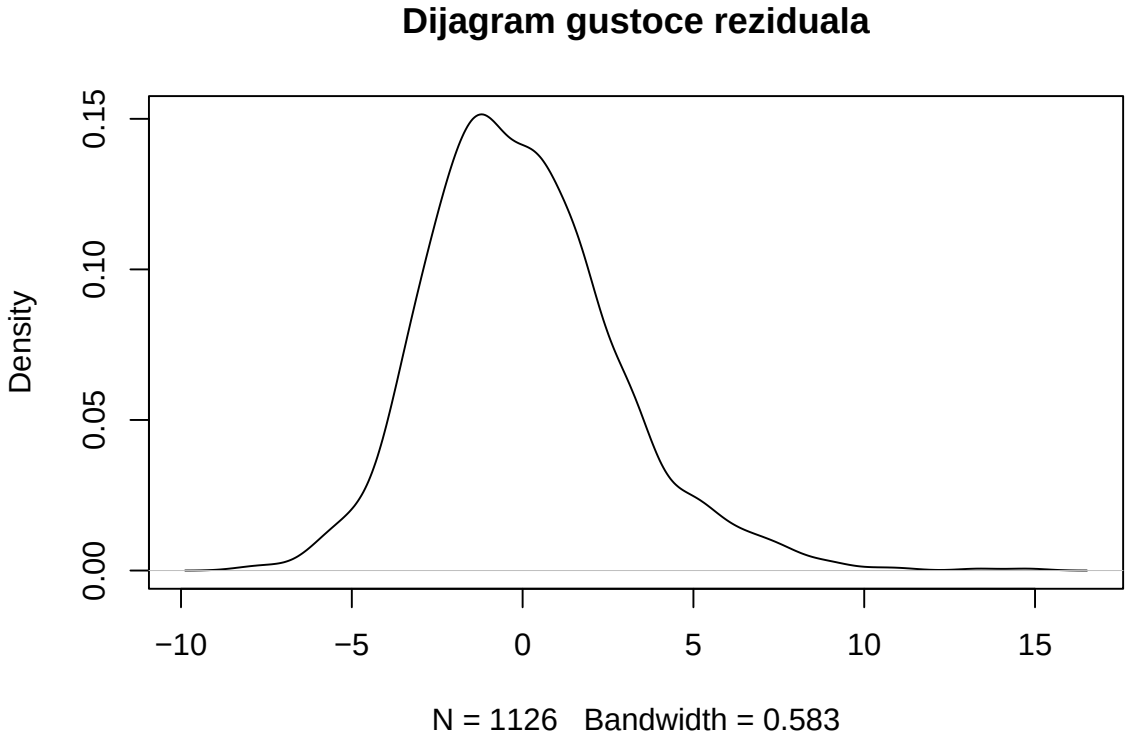
```
## Warning in test_data$ace - predicted_aces_test: longer object length is not a
## multiple of shorter object length
```

```
## Warning in test_data$ace - predicted_aces_test: longer object length is not a
## multiple of shorter object length
```

```
## [1] "Mean Squared Error (MSE): 18.42477788414"
```

```
## [1] "Squared Sum Error (SSE):  1.88114108205429"
## [2] "Squared Sum Error (SSE):  9.3159670769836"
## [3] "Squared Sum Error (SSE):  9.42057606365844"
## [4] "Squared Sum Error (SSE):  0.00491790823009026"
## [5] "Squared Sum Error (SSE):  2.73797345314097"
## [6] "Squared Sum Error (SSE):  19.9968211662635"
```





6.8 Zaključak

Gledajući dobivene rezultate, zaključuje se da je korištenjem linearne regresije moguće dobiti broj aseva koje će igrač odservirati u tekućoj sezoni, no s relativno malom sigurnošću. U modelima dobivenima u ovom zadatku, ta se sigurnost kreće između 30 i 36 posto. Evidendno je da je model na dobrom tragu, no

postoje “skriveni” prediktori koji nam nisu dostupni iz dobivenih podataka.

Pri kreiranju modela potrebno je analizirati svakog igrača zasebno. Pokazano je da prediktori koji su značajni za jednog igrača nemaju nikakvog utjecaja na broj aseva drugog, i samo narušavaju kvalitetu kreiranog modela.

7 6. istrazivačko pitanje: U kojoj je mjeri moguće predvidjeti ishod teniskog meča?

7.1 Uvod

Kako bi odgovorili na ovo pitanje, u analizi smo se fokusirali na teniske mečeve iz perioda 2010-2020 (mečevi “modernog tenisa” za koje imamo većinu podataka za veliki broj varijabli, Novak Đoković je 2011. osvojio 3/4 Grand Slam turnira i počela je formacija tzv. “Velike trojke”...) pri čemu nećemo uzimati u obzir mečeve Olimpijskih igara, Davis Cup-a i ATP Tour Finals-a zbog drugačijeg formata (ovo su jedini turniri za koje smo našli da su drugačijeg formata, vrlo je vjerojatno da ih ima još, ali izgleda da nam nisu stvarali previše problema kasnije):

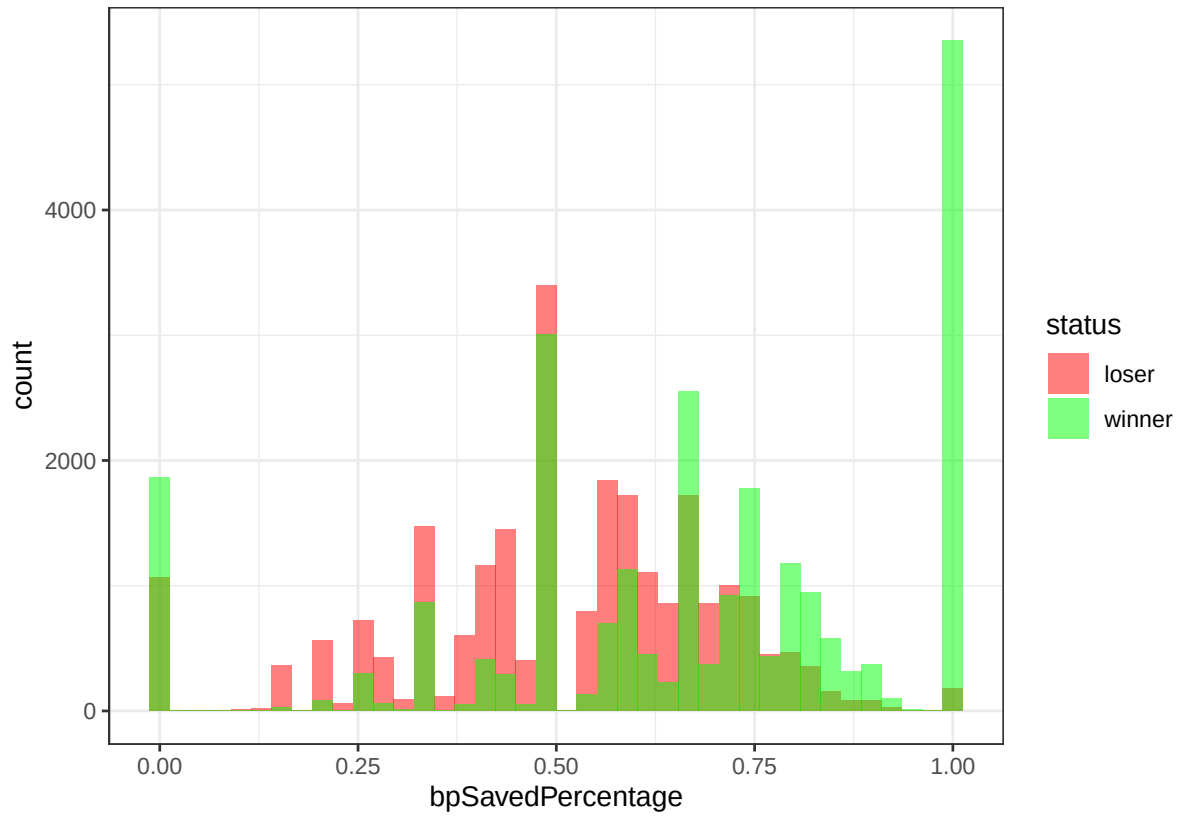
```
matches <- fetch_data(2010:2020)

matches$tourney_name %>%
  str_detect("(Olympics|Davis Cup|Finals)") %>%
    which -> differentFormatMatchesIndices
matches <- matches[-differentFormatMatchesIndices,]
```

Za predikciju ishoda meča koristit ćemo sljedeće varijable (vrijednosti varijabli sa oznakom * izračunavamo na temelju svih mečeva tenisača koji su odigrani prije meča za koji računamo odgovarajuću statistiku):

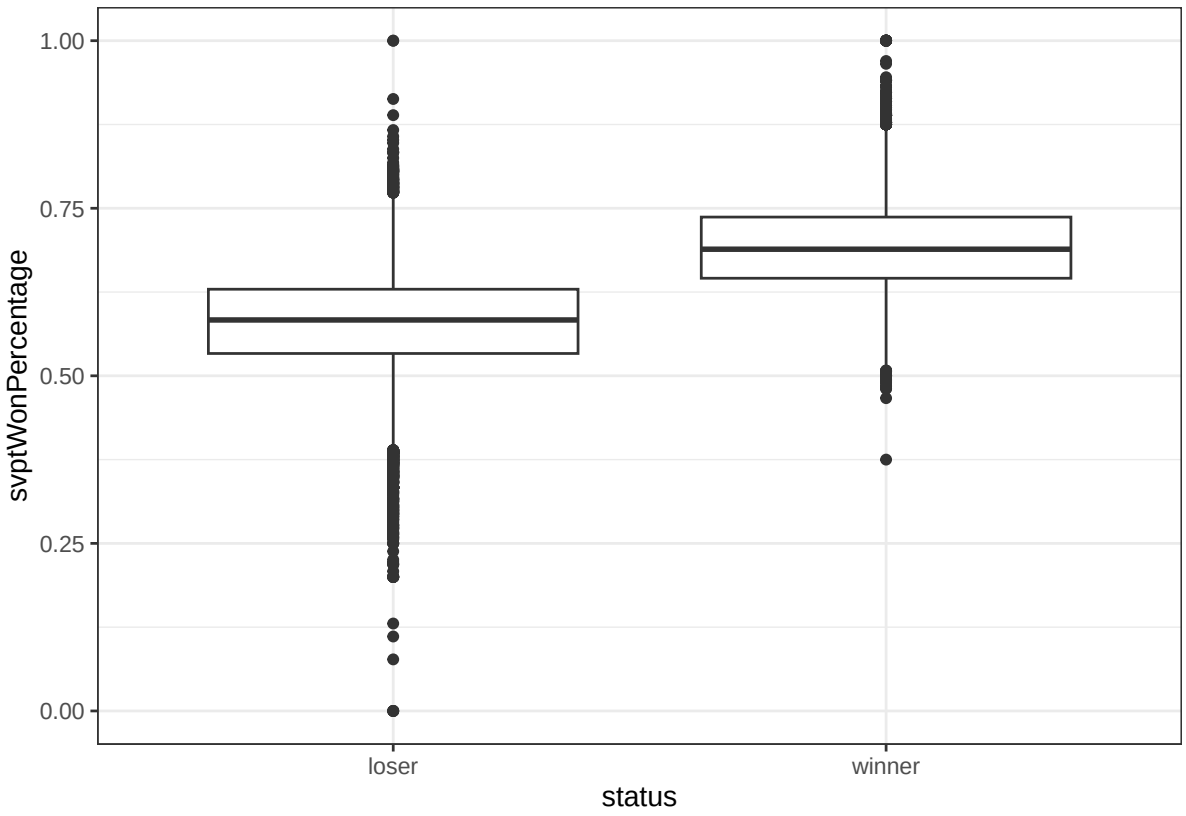
- height - visina igrača. Viši igrači imaju brži i efikasniji servis, lakše pokrivaju teren, bolja igra na mreži.
- age - broj godina. Veći broj godina donosi iskustvo, ali potencijalno i lošije fizičke sposobnosti.
- rank - rang na ATP ljestvici. Indikator sveukupne kvalitete igrača.
- avgAce* - prosječan broj asova po meču. Indikator kvalitete servisa tenisača.
- avgDf* - prosječan broj dvostrukih pogrešaka po meču. Indikator sklonosti pogreškama.
- avgSvptWon* - prosječan omjer osvojenih poena na servisu u usporedbi s igračevim brojem odigranih poena za vlastiti servis. Indikator snažnog i efikasnog servisa.
- avgBpSavedPercentage* - prosječan omjer “spašenih” breakpointova. Indikator mentalne snage i iskustva tenisača.
- winrate* - postotak pobjeda u dosadašnjoj karijeri. Indikator kvalitete tenisača.
- lastTenWinrate - postotak pobjeda u zadnjih 10 mečeva. Indikator forme tenisača.

U nastavku slijede grafovi za neke od prethodno odabranih statistika čiji nam je utjecaj na konačan ishod meča bio “upitan”:

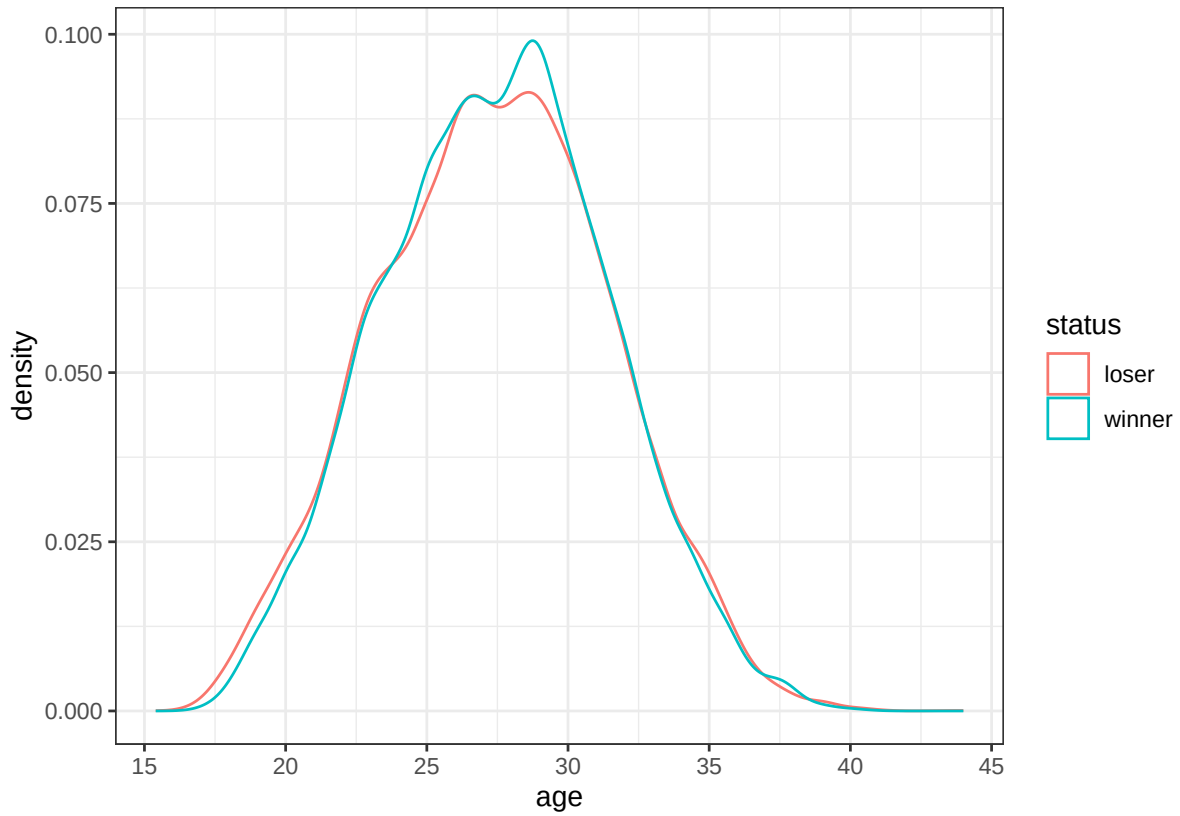
```
ifelse(matches$w_bpFaced == 0, NA,
       matches$w_bpSaved / matches$w_bpFaced) -> wBpSavedPercentage
ifelse(matches$l_bpFaced == 0, NA,
       matches$l_bpSaved / matches$l_bpFaced) -> lBpSavedPercentage

data <- data.frame(winner = wBpSavedPercentage, loser = lBpSavedPercentage)
data <- na.omit(data)
data <- gather(data, key = "status", value = "bpSavedPercentage")

plot <- ggplot(data, aes(x = bpSavedPercentage, fill = status)) +
  geom_histogram(alpha = 0.5, bins = 40, position = 'identity') +
  scale_fill_manual(values = c("winner" = "green", "loser" = "red")) + theme_bw()

plot
```




```
ifelse(matches$w_svpt == 0, NA,
       (matches$w_1stWon + matches$w_2ndWon) / matches$w_svpt) -> wSvptWonPercentage
ifelse(matches$l_svpt == 0, NA,
       (matches$l_1stWon + matches$l_2ndWon) / matches$l_svpt) -> lSvptWonPercentage

data <- data.frame(winner = wSvptWonPercentage, loser = lSvptWonPercentage)
data <- na.omit(data)
data <- gather(data, key = "status", value = "svptWonPercentage")

plot <- ggplot(data, aes(status, svptWonPercentage)) +
  geom_boxplot() + theme_bw()

plot
```




```
data <- data.frame(winner = matches$winner_age, loser = matches$loser_age)
data <- gather(data, key = "status", value = "age")

plot <- ggplot(data, aes(x = age, color = status)) +
  geom_density() + theme_bw()

plot
```



Iz priloženog možemo vidjeti da je naša intuicija bila na dobrom putu: svaka od prikazanih varijabli bi mogla utjecati na ishod meča. Sada ćemo izbaciti neregularne obzervacije:

```
faultyDataMatchesIndices <- which(
  apply(matches[, c("winner_rank", "loser_rank", "winner_ht", "loser_ht", "w_ace")],
        1, function(x) any(is.na(x))))
matches <- matches[-faultyDataMatchesIndices,]
```

te još malo pripremiti skup podataka za obradu:

```
matches$tourney_date <- ymd(matches$tourney_date)
matches <- matches %>% arrange(tourney_date, match_num)
matches$match_id <- seq(1, nrow(matches))
```

7.2 Obrada skupa podataka

Sada odabiremo one teniske mečeve na kojima ćemo kasnije bazirati naš logistički model. To su teniski mečevi u kojima je svaki od oba tenisača prethodno odigrao barem 10 mečeva (pojedine statistike kao što je npr. winrate su jako osjetljive na broj obzervacija). Ove mečeve ćemo u nastavku zvati "validni mečevi":

```
checkIfValidForProcessing = function(match) {
  winnerId <-  match["winner_id"] %>% as.numeric
  loserId <-  match["loser_id"] %>% as.numeric
  matchId <-  match["match_id"] %>% as.numeric

  lastMatchIndex <-  matchId - 1
  if (lastMatchIndex == 0) return(FALSE)

  winnersCandidates <-  pull(matches[1:lastMatchIndex, ], winner_id)
  losersCandidates <-  pull(matches[1:lastMatchIndex, ], loser_id)
  playersCandidates <-  c(winnersCandidates, losersCandidates)
```

```
  validPlayersIndices <-  table(playersCandidates) >= 10
  table(playersCandidates)[validPlayersIndices] %>%
    names %>% as.numeric -> validPlayers

  if (winnerId %in% validPlayers && loserId %in% validPlayers) return (TRUE)

  return (FALSE)
}

validMatchesIndices <- apply(matches, 1, checkIfValidForProcessing)
validMatches <- matches[validMatchesIndices, ]
```

Sada kreiramo pomoćni podatkovni okvir u kojemu će jedna obzervacija predstavljati statistike pojedinog igrača po pojedinom meču. NAPOMENA: ovdje gledamo performanse svakog igrača, i pobjednika i gubitnika, na svakom meču iz perioda 2010-2020 (osim na onim mečevima koje smo izbacili nakon prvog "čišćenja"):

```
allWinnersMatchStats <- matches[, c("match_id", "winner_id", "w_ace", "w_df", "w_svpt",
                                    "w_1stWon", "w_2ndWon", "w_bpSaved", "w_bpFaced")]
allWinnersMatchStats$won <- 1
names(allWinnersMatchStats) <- c("matchId", "playerId", "ace", "df", "svpt",
                                 "firstWon", "secondWon", "bpSaved", "bpFaced", "won")

allLosersMatchStats <- matches[, c("match_id", "loser_id", "l_ace", "l_df", "l_svpt",
                                   "l_1stWon", "l_2ndWon", "l_bpSaved", "l_bpFaced")]
allLosersMatchStats$won <- 0
names(allLosersMatchStats) <- c("matchId", "playerId", "ace", "df", "svpt",
                                "firstWon", "secondWon", "bpSaved", "bpFaced", "won")

allPlayersMatchStats <- rbind(allWinnersMatchStats, allLosersMatchStats)
allPlayersMatchStats <- allPlayersMatchStats %>% arrange(matchId)
```

Kreiramo dva podatkovna okvira (jedan za pobjednika, drugi za gubitnika) koji će sadržavati podatke o godinama, visini i rangu na ATP ljestvici u trenutku igranja validnog meča:

```
validMatchesWinnerInfo <- validMatches[, c("match_id", "winner_id", "winner_ht",
                                           "winner_age", "winner_rank")]
names(validMatchesWinnerInfo) <- c("matchId", "playerId", "height", "age", "rank")

validMatchesLoserInfo <- validMatches[, c("match_id", "loser_id", "loser_ht",
                                          "loser_age", "loser_rank")]
names(validMatchesLoserInfo) <- c("matchId", "playerId", "height", "age", "rank")
```

Za svaki validni meč računamo potrebne statistike za tenisača pobjednika i tenisača gubitnika na temelju svih njihovih prethodnih mečeva koji se nalaze u skupu "matches":

```
calculateValidMatchesStats <- function(x) {

  matchId = x["matchId"] %>% as.numeric
  playerId = x["playerId"] %>% as.numeric
  height = x["height"] %>% as.numeric
  age = x["age"] %>% as.numeric
  rank = x["rank"] %>% as.numeric

  pastMatches <- allPlayersMatchStats[allPlayersMatchStats$playerId == playerId &
                                        allPlayersMatchStats$matchId < matchId,]

  avgAce <- mean(pastMatches$ace, na.rm = T)

  avgDf <- mean(pastMatches$df, na.rm = T)
```

```
  svptWonPercentage <- ifelse(pastMatches$svpt == 0, NA,
                              (pastMatches$firstWon +
                                 pastMatches$secondWon) / pastMatches$svpt)
  avgSvptWonPercentage <- mean(svptWonPercentage, na.rm = T)

  bpSavedPercentage <- ifelse(pastMatches$bpFaced == 0, NA,
                              pastMatches$bpSaved / pastMatches$bpFaced)
  avgBpSavedPercentage <- mean(bpSavedPercentage, na.rm = T)

  winrate <- mean(pastMatches$won, na.rm = T)

  lastTenWinrate <- sum(tail(pastMatches$won, 10)) / 10

  result <- c(matchId, height, age, rank, avgAce, avgDf,
              avgBpSavedPercentage, avgSvptWonPercentage, winrate, lastTenWinrate)

  result
}

validMatchesWinnerData <- apply(validMatchesWinnerInfo, 1, calculateValidMatchesStats)
validMatchesWinnerData <- as.data.frame(t(validMatchesWinnerData))
colnames(validMatchesWinnerData) <- c("matchId", "height", "age", "rank",
                                      "avgAce", "avgDf", "avgBpSavedPercentage",
                                      "avgSvptWonPercentage", "winrate", "lastTenWinrate")

validMatchesLoserData <- apply(validMatchesLoserInfo, 1, calculateValidMatchesStats)
validMatchesLoserData <- as.data.frame(t(validMatchesLoserData))
colnames(validMatchesLoserData) <- c("matchId", "height", "age", "rank", "avgAce", "avgDf", "avgBpSa
```

Stvaramo 3 nova podatkovna okvira: playerOne, playerTwo i playerDifference. Prva 2 podatkovna okvira odgovaraju jedan drugom po retcima, npr. u oba okvira u retku 547 se nalaze podaci o istom meču, samo jedan od njih sadrži sve podatke o gubitniku, a drugi o pobjedniku, s tim da je okvir koji sadrži podatke za pobjednika, odnosno gubitnika, nasumično odabran. Okvir playerDifference predstavlja njihovu razliku:

```
winnerIndices <- sample(1:23101, size = length(1:23101) * 0.5, replace = FALSE)

playerOne <- rbind(validMatchesWinnerData[winnerIndices, ],
                   validMatchesLoserData[-winnerIndices, ]) %>% arrange(matchId)
playerTwo <- rbind(validMatchesWinnerData[-winnerIndices, ],
                   validMatchesLoserData[winnerIndices, ]) %>% arrange(matchId)
playerDifference <- playerOne - playerTwo

playerDifference$won <- 0
playerDifference[winnerIndices, "won"] <- 1
playerDifference$matchId <- NULL

head(playerDifference, 5)
```

```
##   height  age rank      avgAce      avgDf avgBpSavedPercentage
## 1      5 -4.0   30  2.54545455  0.7272727          -0.19861111
## 2      5  0.1   26  1.90151515  0.2045455          -0.01686344
## 3     30  2.3   -7 13.30000000 -0.1000000           0.25965608
## 4      5  1.9   21 -0.06666667  1.7333333          -0.08551828
## 5      8 -4.7  -26  4.98787879  2.4060606          -0.11030609
##   avgSvptWonPercentage      winrate lastTenWinrate won
## 1          -0.02974964 -0.090909091          -0.1   0
## 2          -0.01796746 -0.280303030          -0.2   0
## 3           0.02398847  0.200000000           0.2   1
```

```
## 4          -0.03218184 -0.033333333            0.0   0
## 5           0.03008055  0.006060606            0.0   0
```

Još je samo potrebno normalizirati podatke:

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
playerDifference <- as.data.frame(lapply(playerDifference, normalize))

head(playerDifference, 5)
```

```
##      height       age      rank    avgAce     avgDf avgBpSavedPercentage
## 1 0.5324675 0.3870192 0.4937606 0.5715404 0.5689291            0.2100828
## 2 0.5324675 0.4855769 0.4926262 0.5542194 0.5212143            0.5023474
## 3 0.8571429 0.5384615 0.4832672 0.8608209 0.4934153            0.9470128
## 4 0.5324675 0.5288462 0.4912082 0.5012784 0.6607628            0.3919450
## 5 0.5714286 0.3701923 0.4778786 0.6372378 0.7221696            0.3520842
##   avgSvptWonPercentage   winrate lastTenWinrate won
## 1            0.3922297 0.4197251      0.4444444   0
## 2            0.4252745 0.3025900      0.3888889   0
## 3            0.5429460 0.5996447      0.6111111   1
## 4            0.3854083 0.4553342      0.5000000   0
## 5            0.5600322 0.4796983      0.5000000   0
```

7.3 Treniranje modela, predikcija i zaključak

Dijelimo obzervacije u skupove podataka za treniranje i predikciju:

```
set.seed(123)

trainingIndices <- sample(1:nrow(playerDifference), nrow(playerDifference) * 0.7)
trainingSet <- playerDifference[trainingIndices, ]
testingSet <- playerDifference[-trainingIndices, ]
```

Provodimo logističku regresiju:

```
model <- glm(won ~ ., family = binomial(link = "logit"), data = trainingSet)
```

Sada pogledajmo ispis rezultata našeg modela:

```
summary(model)
```

```
##
## Call:
## glm(formula = won ~ ., family = binomial(link = "logit"), data = trainingSet)
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)              0.003552   0.511818   0.007 0.994462
## height                   0.076903   0.185073   0.416 0.677756
## age                     -1.381908   0.137975 -10.016  < 2e-16 ***
## rank                    -6.130800   0.914489  -6.704 2.03e-11 ***
## avgAce                  -1.129293   0.293756  -3.844 0.000121 ***
## avgDf                   -0.053404   0.180782  -0.295 0.767683
## avgBpSavedPercentage    -0.037413   0.251274  -0.149 0.881639
## avgSvptWonPercentage     1.985268   0.356640   5.567 2.60e-08 ***
## winrate                  5.570191   0.319137  17.454  < 2e-16 ***
## lastTenWinrate           1.291339   0.181492   7.115 1.12e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22416  on 16169  degrees of freedom
## Residual deviance: 19649  on 16160  degrees of freedom
## AIC: 19669
##
## Number of Fisher Scoring iterations: 3
```

Za početak bacimo pogled na stupac s imenom "Pr(>|z|)" koji nam ukazuje na to koje su se od varijabli u našem modelu pokazale statistički značajnim: za svaku varijablu nam daje P-vrijednost testa hipoteze da je pripadni koeficijent jednak 0. Možemo zaključiti da su nam za razinu značajnosti od 0.05 statistički značajne sve varijable osim height, avgDf i avgBpSavedPercentage koje ne doprinose efikasnosti modela. Sada pogledajmo koliko je naš model dobar:

```
predictions <- predict(model, newdata = testingSet, type = "response")
predictions <- ifelse(predictions > 0.5, 1, 0)

misClasificError <- mean(predictions != testingSet$won)
print(paste('Accuracy', 1 - misClasificError))
```

```
## [1] "Accuracy 0.664694849228106"
```

Preciznost od oko 66% na skupu podataka od 6931 obzervacija. Ajmo sada opet provesti logističku regresiju, ali bez prethodno spomenutih varijabli:

```
set.seed(123)

indices <- c(2, 3, 4, 7, 8, 9, 10)
playerDifference <- playerDifference[, indices]
trainingIndices <- sample(1:nrow(playerDifference), nrow(playerDifference) * 0.7)
trainingSet <- playerDifference[trainingIndices, ]
testingSet <- playerDifference[-trainingIndices, ]
model <- glm(won ~ ., family = binomial(link = "logit"), data = trainingSet)
summary(model)
```

```
##
## Call:
## glm(formula = won ~ ., family = binomial(link = "logit"), data = trainingSet)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.01101    0.50279  -0.022    0.983
## age                  -1.38687    0.13448 -10.313  < 2e-16 ***
## rank                 -6.14258    0.91365  -6.723 1.78e-11 ***
## avgAce               -1.11015    0.23998  -4.626 3.73e-06 ***
## avgSvptWonPercentage  2.00570    0.31580   6.351 2.14e-10 ***
## winrate               5.55461    0.31661  17.544  < 2e-16 ***
## lastTenWinrate        1.29140    0.18144   7.118 1.10e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22416  on 16169  degrees of freedom
## Residual deviance: 19649  on 16163  degrees of freedom
## AIC: 19663
##
## Number of Fisher Scoring iterations: 3
```

AIC vrijednost je sada malo manja, što je indikator da je ovaj model bolji. Da vidimo koliko je zapravo dobar:

```r
predictions <- predict(model, newdata = testingSet, type = "response")
predictions <- ifelse(predictions > 0.5, 1, 0)

misClasificError <- mean(predictions != testingSet$won)
print(paste('Accuracy', 1 - misClasificError))
```

```
## [1] "Accuracy 0.664839128552878"
```

Preciznost se nije znatno promijenila ("Dikod uđe, dikod ne uđe" - Mate Baturina). Međutim, unatoč tome možemo zaključiti da je itekako moguće predvidjeti ishod teniskog meča pomoću logističke regresije.

8 7. istraživačko pitanje: Postoji li razlika u broju odigranih aseva između igrača koji su osvojili Grand Slam naslov i onih koji nisu?

8.1 Učitavanje podataka

Podatke ćemo učitati iz CSV datoteka u listu podatkovnih okvira. Svaki podatkovni okvir sadržava podatke mečeva jedne godine. Nakon toga spajamo sve podatkove okvire u jedan te time imamo informacije o svim odigranim mečevima od 1968. do 2023. godine dostupne u jednom podatkovnom okviru. Iz tog okvira ćemo naknadno izvlačiti potrebne podatke za svrhe analize.

Proces učitavanja podataka je prikazan u sljedećem bloku koda:

```r
matches <- fetch_data(1968:2023)
```

8.2 Prilagodba tipova podataka

Nakon učitavanja podataka, potrebno je ispraviti tip podataka za stupac `tourney_level` koji je potreban kod istraživanja.

Stupac može sadržavati vrijednosti `F`, `A`, `D`, `M` i `G`. Te vrijednosti predstavljaju razine turnira na kojima su odigrani mečevi:

- `F` - ATP Finals
- `A` - ATP World Tour
- `D` - Davis Cup
- `M` - Masters 1000
- `G` - Grand Slam

Prilagodba je prikazana u sljedećem kodu:

```r
matches$tourney_level <- factor(
  matches$tourney_level,
  levels = c("F", "A", "D", "M", "G")
)
```

8.3 Statistička analiza

8.3.1 Priprema podataka

Potrebne su nam dvije glavne grupe u koje ćemo podijeliti igrače:

- igrači s barem jednom osvojenom Grand Slam titulom
- igrači bez osvojene Grand Slam titule (u daljnjem tekstu ćemo ih nazvati "ostali igrači")

Uz te dvije glavne grupe, stvorit ćemo dodatnu grupu za ostale igrače gdje ih dodatno filtriramo tako da uklonimo igrače koji su ukupno sudjelovali u manje od 7 mečeva. Budući da uspoređujemo prosjeke odigranih asova po meču, želimo izbjeći situacije gdje bi igrači koji su odigrali samo jedan ili dva meča imali velik utjecaj na prosjek cijele grupe. Budući da je za osvajanje titule potrebno odigrati i pobijediti u barem 7 mečeva (za grand slam turnire se inicijalno izvlači 128 sudionika), smatramo da je 7 mečeva

minimalna granica koju igrač mora ispuniti da bi bio uključen u usporedbu s igračima koji su osvojili Grand Slam titulu. U daljnjem tekstu ćemo se na tu grupu referirati kao "ostali igrači (n>=7)" i odvojeno ju uspoređivati s igračima koji su osvojili Grand Slam titulu.

Proces izvlačenja i pripreme podataka je prikazan i komentiran u sljedećem bloku koda:

```
# izrada novog okvira s podatcima o svim igračima (ID, ime, broj asova)
all_players <- matches %>%
  dplyr::select(winner_id, winner_name, w_ace) %>%
  rename(id = winner_id, name = winner_name, aces = w_ace) %>%
  bind_rows(matches %>%
    dplyr::select(loser_id, loser_name, l_ace) %>%
    rename(id = loser_id, name = loser_name, aces = l_ace))

# uklanjanje unosa za koje ne postoje podatci o broju odigranih asova
all_players <- all_players %>%
  filter(!is.na(aces))

# grupiranje igrača po ID-u i imenu te izračun prosjeka asova po meču (APM)
all_players <- all_players %>%
  group_by(id, name) %>%
  summarise(aces = sum(aces), matches = n()) %>%
  mutate(apm = aces / matches)
```

```
## `summarise()` has grouped output by 'id'. You can override using the `.groups`
## argument.
```

```
# filtriranje igrača koji su odigrali >=7 mečeva
filter_players <- all_players %>%
  filter(matches >= 7)

# izrada okvira za igrače s naslovom (pobjeda u finalu Grand Slam turnira)
gs_winners <- matches %>%
  filter(tourney_level == "G" & round == "F") %>%
  dplyr::select(winner_id, winner_name) %>%
  rename(id = winner_id, name = winner_name) %>%
  distinct(id, name)

# dodavanje stupca s prosjekom asova po meču (apm) za igrače s naslovom
gs_winners <- gs_winners %>%
  left_join(all_players, by = c("id", "name")) %>%
  filter(!is.na(apm))

# izrada okvira za ostale igrače (bez naslova)
gs_others <- all_players %>%
  filter(!id %in% gs_winners$id)

# izrada okvira za ostale igrače uz uvjet da su odigrali barem 7 mečeva
gs_filter1 <- filter_players %>%
  filter(!id %in% gs_winners$id)

# brisanje privremenih okvira koji su nepotrebni u daljnjem radu
rm(filter_players)
```

8.3.2 Deskriptivna statistika

Sada kada imamo podatke na kojima možemo provoditi analizu, izračunat ćemo osnovne deskriptivne statistike za svaku od grupa. Izračunat ćemo sljedeće vrijednosti:

- prosjek asova svake grupe

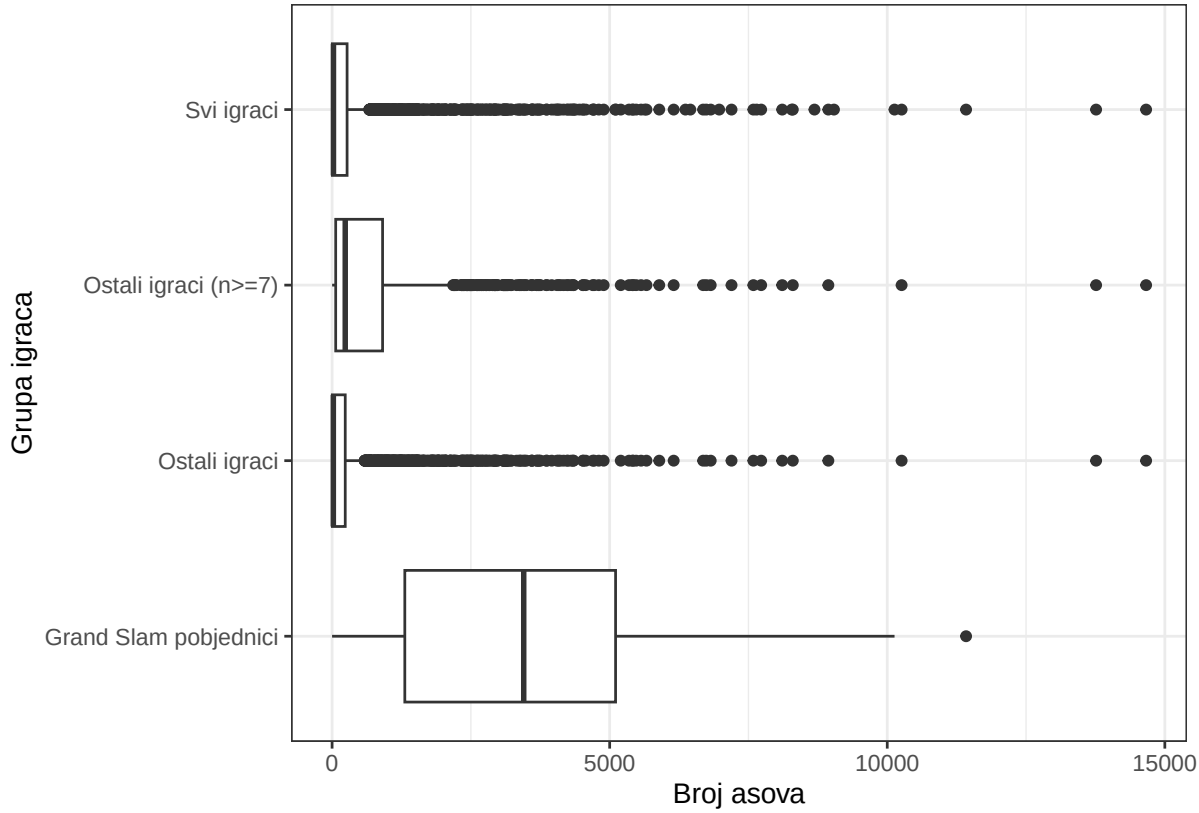
- medijan asova
- standardnu devijaciju asova
- minimalnu vrijednost asova svake grupe
- maksimalnu vrijednost asova svake grupe
- ukupan broj odigranih mečeva za svaku grupu
- prosjek asova po meču (APM) cijele grupe
- broj igrača u svakoj grupi

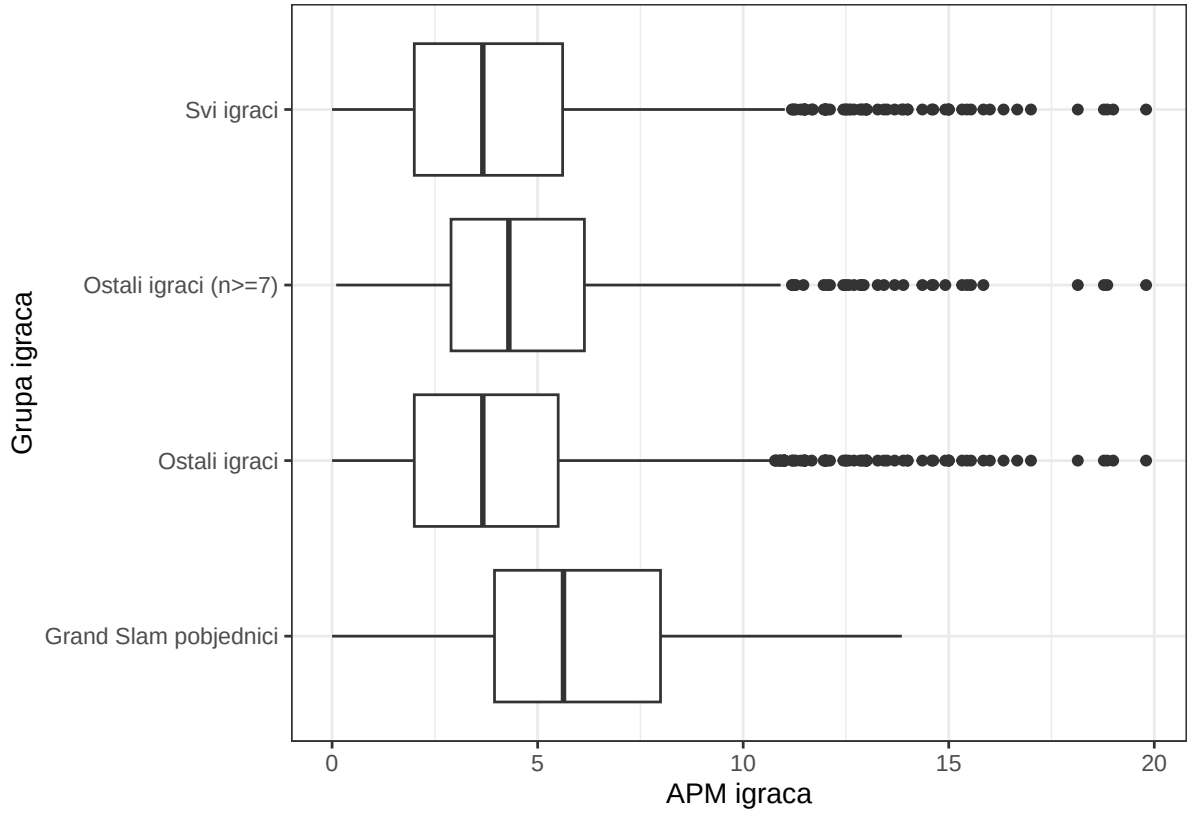
Table 3: Servirani asovi po grupama

id	mean_aces	median_aces	sd_aces	min_aces	max_aces	total_matches	group_apm	n_players
Grand Slam pobjednici	3703.77	3452	2982.47	0	11420	23677	6.73	43
Ostali igrači	386.25	23	1013.12	0	14663	165274	5.53	2368
Ostali igrači (n>=7)	764.82	237	1329.93	1	14663	162536	5.57	1183
Svi igrači	445.42	25	1164.47	0	14663	188951	5.68	2411

Uz tablicu s deskriptivnim statistikama, možemo i vizualizirati podatke kako bismo dobili bolji uvid u njih. Za to ćemo koristiti boxplotove za broj asova i srednje vrijednost asova po mečevima (APM) igrača te dijagrame gustoće za prosjek asova po meču kako bismo dobili uvid u distribuciju podataka. Za svaku od grupa ćemo izraditi zasebne vizualizacije kako bismo mogli usporediti podatke između grupa. Uz to, izradit ćemo i vizualizacije za sve igrače kako bismo dobili uvid u distribuciju podataka za cijelu populaciju igrača.

Boxplotovi broja asova i APM-ova za svaku grupu:





Iz boxplot dijagrama možemo vidjeti kako je distribucija asova igrača jako nakošena na lijevu stranu te kako ima jako puno ekstremnih vrijednosti, što i ima smisla jer broj asova ultimativno ovisi o ukupnom broju mečeva koje je igrač odigrao tijekom svoje karijere.

Zbog toga ćemo usporedbu broja asova između grupa raditi kroz srednju vrijednost asova po mečevima (APM) igrača. Kroz APM dobivamo jedinstvenu mjeru koja ne ovisi o broju mečeva koje je igrač odigrao te je stoga pogodnija za usporedbu između grupa.

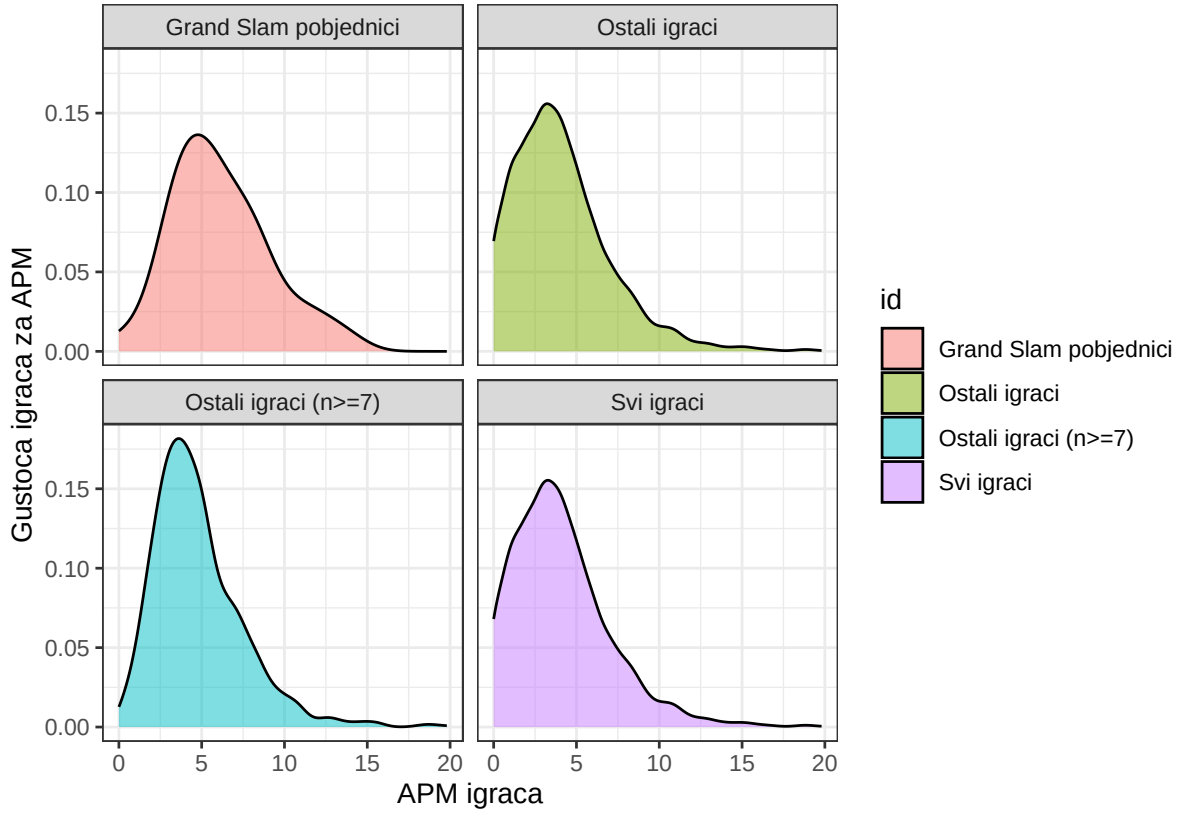
U boxplot dijagramu APM-ova možemo stoga primijetiti kako su distribucije APM-ova puno manje nakošene te imaju puno manje ekstremnih vrijednosti.

Deskriptivna statistika APM-ova za svaku grupu je sljedeća:

Table 4: APM igrača po grupama

id	mean_apm	median_apm	sd_apm	min_apm	max_apm	n_players
Grand Slam pobjednici	6.135	5.629	2.968	0.0	13.859	43
Ostali igrači	4.080	3.667	2.926	0.0	19.801	2368
Ostali igrači (n>=7)	4.791	4.300	2.709	0.1	19.801	1183
Svi igrači	4.117	3.667	2.938	0.0	19.801	2411

Kako bismo dobili bolji uvid u samu distribuciju APM-ova grupa, izradit ćemo dijagrame gustoće za svaku grupu:



Iz dijagrama gustoća svake od grupa možemo vidjeti kako su podatci uistinu još uvijek lijevo nakošeni te vjerojatno nisu normalno distribuirani. Zbog toga ćemo prvo trebati provjeriti normalnost podataka kako bismo mogli odlučiti koje testove ćemo koristiti za usporedbu grupa.

8.3.3 Statističko testiranje

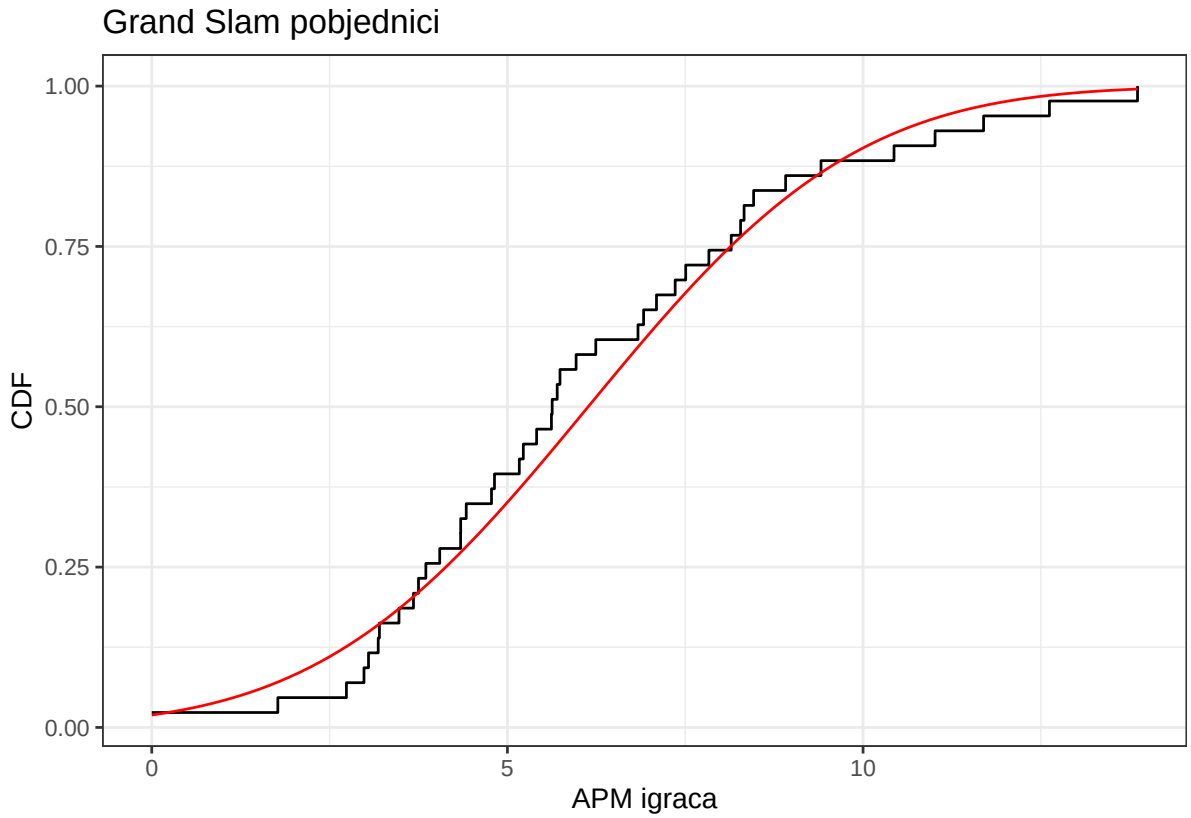
Nakon vizualnog uvida u podatke i izračunatih deskriptivnih statistika, možemo započeti statističko testiranje kako bismo dobili odgovor na temeljno pitanje ovog istraživanja: ***Postoji li razlika u broju odigranih aseva između igrača koji su osvojili Grand Slam naslov i onih koji nisu?***

8.3.3.1 Normalnost uzoraka Kako bismo odlučili hoćemo li koristiti parametarske ili neparametarske testove, potrebno je provjeriti normalnost podatako. Za to ćemo iskoristiti Kolmogorov-Smirnovljev test. Budući da smo već ranije zaključili da je APM pogodniji za usporedbu grupa, provjerit ćemo normalnost samo za APM-ove.

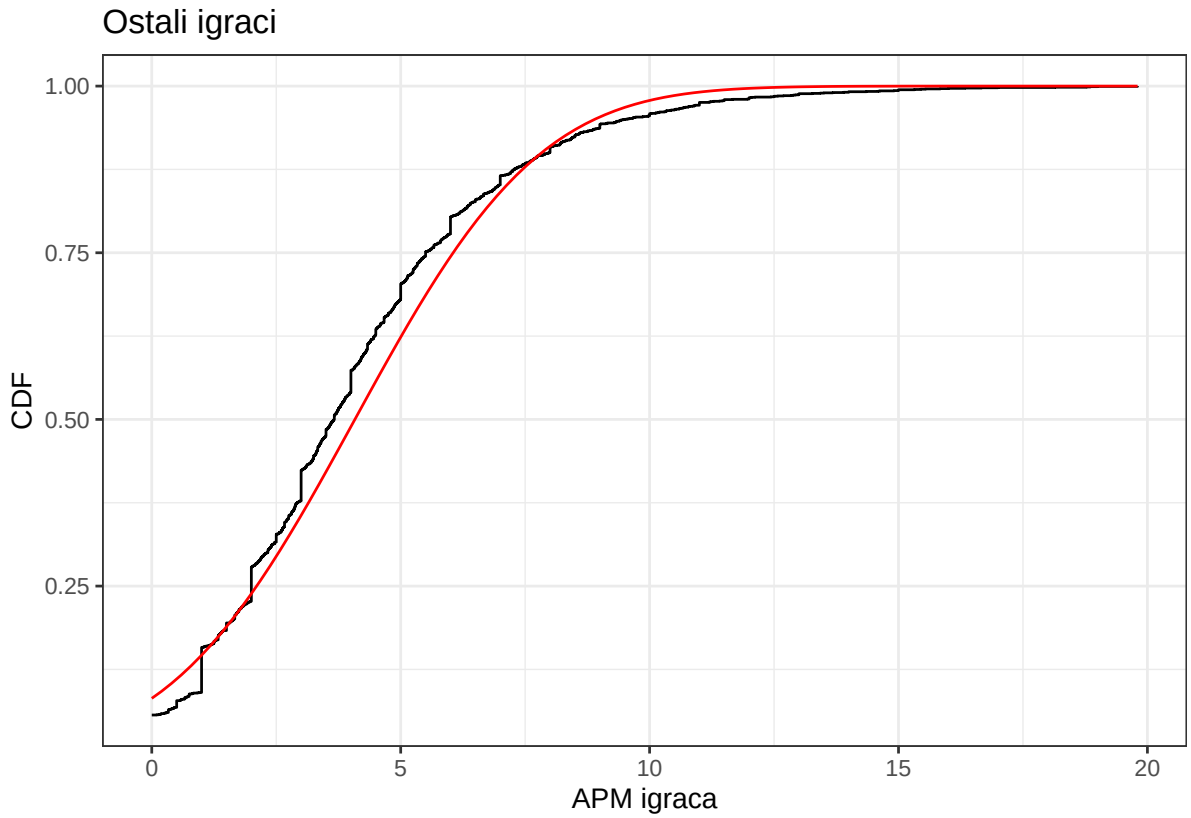
Prije provođenja Kolmogorov-Smirnovljevog testa postavljamo sljedeće hipoteze:

- H0: Podatci su normalno distribuirani
- H1: Podatci nisu normalno distribuirani

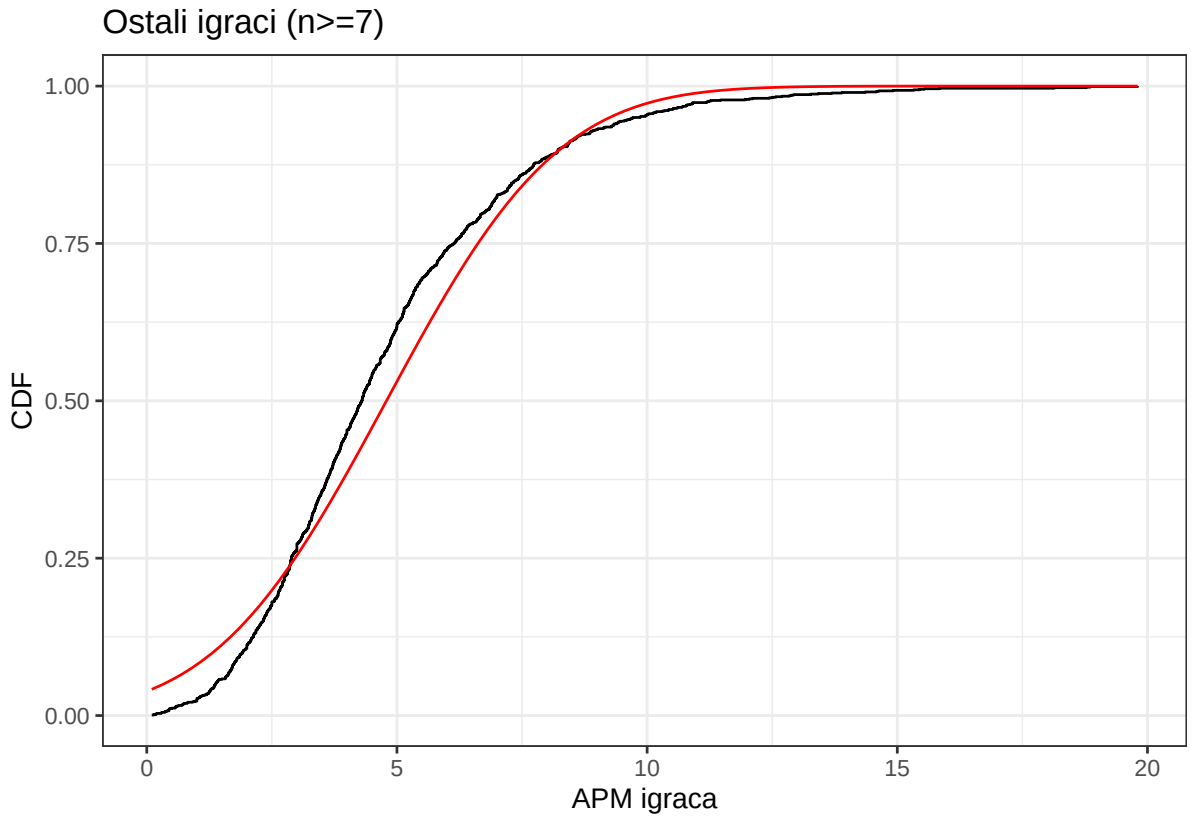
Za test uzimamo razinu značajnosti od 0.05.




```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  gs_winners$apm
## D = 0.11126, p-value = 0.6216
## alternative hypothesis: two-sided
```




```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  gs_others$apm
## D = 0.084809, p-value = 1.954e-14
## alternative hypothesis: two-sided
```




```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  gs_filter1$apm
## D = 0.09568, p-value = 7.062e-10
## alternative hypothesis: two-sided
```

Kao što možemo vidjeti iz dijagrama i rezultata Kolmogorov-Smirnovljevog testa, jedino uzorak igrača s Grand Slam naslovom prolazi test normalnosti, dok ostali, zbog činjenice da su uzorci puno veći, ne prolaze test te za njih odbacujemo H0 hipotezu uz razinu značajnosti od 0.05.

Zbog toga ćemo morati koristiti neparametarske testove za usporedbu grupa.

8.3.3.2 Neparametarsko testiranje Sada kada smo utvrdili da su uzorci nenoormalno distribuirani, možemo koristiti neparametarske testove za usporedbu grupa. Budući da uspoređujemo dvije grupe, koristit ćemo Mann-Whitney U test. Prije provođenja testa postavljamo sljedeće hipoteze:

- H0: Nema razlike u prosjeku APM-ova između grupa (uzorci dolaze iz iste populacije)
- H1: Postoji razlika u prosjeku APM-ova između grupa (jedan uzorak stohastički dominira)

Za ovaj test također uzimamo razinu značajnosti od 0.05.

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gs_winners$apm and gs_others$apm
## W = 72232, p-value = 2.427e-06
## alternative hypothesis: true location shift is not equal to 0
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  gs_winners$apm and gs_filter1$apm
## W = 32977, p-value = 0.0009439
```

```
## alternative hypothesis: true location shift is not equal to 0
```

Kao što možemo vidjeti iz rezultata testa, u oba slučaja odbacujemo H0 hipotezu uz razinu značajnosti od 0.05. To znači da postoji razlika u prosjeku APM-ova između grupa igrača s osvojenim Grand Slam naslovom i ostalih igrača. Uz to, možemo primijetiti kako je razlika u prosjeku APM-ova između uzorka igrača s naslovom i ostalih igrača puno veća nego između uzorka igrača s naslovom i filtriranih ostalih igrača.

8.4 Zaključak

U ovom istraživanju smo pokušali odgovoriti na pitanje postoji li razlika u broju odigranih aseva između igrača koji su osvojili Grand Slam naslov i onih koji nisu. Kako bismo odgovorili na to pitanje, analizirali smo podatke o svim odigranim mečevima od 1968. do 2023. godine. Nakon pripreme podataka, izračunavanja deskriptivnih statistika i vizualizacije podataka, zaključili smo da postoji razlika u prosjeku APM-ova između igrača koji su osvojili Grand Slam naslov i onih koji nisu. Uz to, možemo primijetiti kako je razlika u prosjeku APM-ova između uzorka igrača s naslovom i ostalih igrača puno veća nego između uzorka igrača s naslovom i filtriranih ostalih igrača. Iz toga možemo zaključiti da postoji razlika u broju odigranih aseva između igrača koji su osvojili Grand Slam naslov i onih koji nisu te da je ta razlika puno veća nego što se može objasniti činjenicom da igrači koji su osvojili Grand Slam naslov imaju više odigranih mečeva od ostalih igrača. Iz toga možemo zaključiti da bi broj odigranih aseva mogao biti jedan od faktora koji utječu na osvajanje Grand Slam naslova.