

Eigenvektori

MVedak

2022-12-06

Učitavanje potrebnih paketa:

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr 0.3.5
## v tibble 3.1.8       v dplyr 1.0.10
## v tidyr 1.2.1        v stringr 1.4.1
## v readr 2.1.3        v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## Loading required package: lattice
##
##
## Attaching package: 'caret'
##
##
## The following object is masked from 'package:purrr':
##
## lift
```

Inicijalno učitavanje i obrada podataka

Prije izvoda ikakve analize potrebno je učitati i očistiti podatke. Učitavanje podataka je vrlo jednostavno, no prije konkretnog korištenja potrebno je modificirati učitani DataFrame kako bi analiza bila moguća.

Prvi problem koji se pojavio je činjenica da R automatski prepoznaje tip podataka u kolumnama, te je kolumni "Min" prepoznao kao kolumnu koja sadrži stringove, dok je ona u realnosti numerička (do ovoga dolazi jer kolumna sadrži ', ' character). Nažalost, ovaj problem se ne može instantno riješiti funkcijom `as.numeric` već moramo prije toga eliminirati ', ' koji odvaja tisućice od stotica.

```
players <- read.csv("Statistika nogometaša engleske Premier lige.csv", stringsAsFactors = FALSE)
players$Min <- as.numeric(gsub(",", "", players$Min))
```

Pogledajmo sada naš dataset:

```
head(players)
```

```
##           Player   Team   Nation   Pos Age MP Starts   Min X90s Gls Ast
## 1      Bukayo Saka Arsenal eng\xa0ENG FW,MF  19 38      36 2978 33.1  11   7
## 2 Gabriel Dos Santos Arsenal br\xa0BRA   DF  23 35      35 3063 34.0   5   0
## 3   Aaron Ramsdale Arsenal eng\xa0ENG   GK  23 34      34 3060 34.0   0   0
## 4       Ben White Arsenal eng\xa0ENG   DF  23 32      32 2880 32.0   0   0
## 5 Martin \xd8degaard Arsenal no\xa0NOR   MF  22 36      32 2785 30.9   7   4
## 6      Granit Khaka Arsenal ch\xa0SUI MF,DF  28 27      27 2327 25.9   1   2
##   G.PK PK PKatt CrdY CrdR Gls.1 Ast.1 G.A G.PK.1 G.A.PK  xG npKG  xA npKG.xA
```

```
## 1  9  2    2  6  0  0.33  0.21  0.54  0.27  0.48  9.7  8.2  6.9  15.2
## 2  5  0    0  8  1  0.15  0.00  0.15  0.15  0.15  2.7  2.7  0.8   3.5
## 3  0  0    0  1  0  0.00  0.00  0.00  0.00  0.00  0.0  0.0  0.0   0.0
## 4  0  0    0  3  0  0.00  0.00  0.00  0.00  0.00  1.0  1.0  0.6   1.6
## 5  7  0    0  4  0  0.23  0.13  0.36  0.23  0.36  4.8  4.8  6.8  11.6
## 6  1  0    0 10  1  0.04  0.08  0.12  0.04  0.12  1.2  1.2  2.3   3.5
##   xG.1 xA.1 xG.xA npxG.1 npxG.xA.1
## 1 0.29 0.21 0.50 0.25 0.46
## 2 0.08 0.02 0.10 0.08 0.10
## 3 0.00 0.00 0.00 0.00 0.00
## 4 0.03 0.02 0.05 0.03 0.05
## 5 0.16 0.22 0.38 0.16 0.38
## 6 0.05 0.09 0.14 0.05 0.14
```

```
str(players)
```

```
## 'data.frame': 691 obs. of 30 variables:
## $ Player : chr "Bukayo Saka" "Gabriel Dos Santos" "Aaron Ramsdale" "Ben White" ...
## $ Team : chr "Arsenal" "Arsenal" "Arsenal" "Arsenal" ...
## $ Nation : chr "eng\xa0ENG" "br\xa0BRA" "eng\xa0ENG" "eng\xa0ENG" ...
## $ Pos : chr "FW,MF" "DF" "GK" "DF" ...
## $ Age : int 19 23 23 22 28 28 24 21 20 ...
## $ MP : int 38 35 34 32 36 27 24 22 33 29 ...
## $ Starts : int 36 35 34 32 32 27 23 22 21 21 ...
## $ Min : num 2978 3063 3060 2880 2785 ...
## $ X90s : num 33.1 34 34 32 30.9 25.9 22.5 21.3 21.3 20.7 ...
## $ GlS : int 11 5 0 0 7 1 2 1 10 6 ...
## $ Ast : int 7 0 0 0 4 2 1 3 2 6 ...
## $ G.PK : int 9 5 0 0 7 1 2 1 10 5 ...
## $ PK : int 2 0 0 0 0 0 0 0 0 1 ...
## $ PKatt : int 2 0 0 0 0 0 0 0 0 1 ...
## $ CrdY : int 6 8 1 3 4 10 6 0 1 3 ...
## $ CrdR : int 0 1 0 0 0 1 0 0 0 1 ...
## $ GlS.1 : num 0.33 0.15 0 0 0.23 0.04 0.09 0.05 0.47 0.29 ...
## $ Ast.1 : num 0.21 0 0 0 0.13 0.08 0.04 0.14 0.09 0.29 ...
## $ G.A : num 0.54 0.15 0 0 0.36 0.12 0.13 0.19 0.56 0.58 ...
## $ G.PK.1 : num 0.27 0.15 0 0 0.23 0.04 0.09 0.05 0.47 0.24 ...
## $ G.A.PK : num 0.48 0.15 0 0 0.36 0.12 0.13 0.19 0.56 0.53 ...
## $ xG : num 9.7 2.7 0 1 4.8 1.2 2.5 0.7 5.8 7.2 ...
## $ npxG : num 8.2 2.7 0 1 4.8 1.2 2.5 0.7 5.8 6.5 ...
## $ xA : num 6.9 0.8 0 0.6 6.8 2.3 1.3 1.9 2.2 3.3 ...
## $ npxG.xA : num 15.2 3.5 0 1.6 11.6 3.5 3.8 2.6 8 9.8 ...
## $ xG.1 : num 0.29 0.08 0 0.03 0.16 0.05 0.11 0.03 0.27 0.35 ...
## $ xA.1 : num 0.21 0.02 0 0.02 0.22 0.09 0.06 0.09 0.1 0.16 ...
## $ xG.xA : num 0.5 0.1 0 0.05 0.38 0.14 0.17 0.12 0.37 0.51 ...
## $ npxG.1 : num 0.25 0.08 0 0.03 0.16 0.05 0.11 0.03 0.27 0.31 ...
## $ npxG.xA.1 : num 0.46 0.1 0 0.05 0.38 0.14 0.17 0.12 0.37 0.47 ...
```

```
summary(players)
```

```
##      Player      Team      Nation      Pos
## Length:691 Length:691 Length:691 Length:691
## Class :character Class :character Class :character Class :character
## Mode :character Mode :character Mode :character Mode :character
##
```

```

##
##
##
##      Age           MP           Starts           Min           X90s
## Min.    :15.00    Min.    : 0.00    Min.    : 0.0    Min.    : 1    Min.    : 0.00
## 1st Qu.:20.00    1st Qu.: 1.00    1st Qu.: 0.0    1st Qu.: 398    1st Qu.: 4.35
## Median :24.00    Median :14.00    Median : 9.0    Median :1328    Median :14.70
## Mean    :24.49    Mean    :15.17    Mean    :12.1    Mean    :1376    Mean    :15.26
## 3rd Qu.:28.00    3rd Qu.:28.00    3rd Qu.:22.0    3rd Qu.:2154    3rd Qu.:23.90
## Max.    :39.00    Max.    :38.00    Max.    :38.0    Max.    :3420    Max.    :38.00
## NA's    :4                                NA's    :145    NA's    :144
##      GlS           Ast           G.PK           PK
## Min.    : 0.000    Min.    : 0.000    Min.    : 0.000    Min.    :0.0000
## 1st Qu.: 0.000    1st Qu.: 0.000    1st Qu.: 0.000    1st Qu.:0.0000
## Median : 1.000    Median : 1.000    Median : 1.000    Median :0.0000
## Mean    : 1.896    Mean    : 1.362    Mean    : 1.742    Mean    :0.1536
## 3rd Qu.: 2.000    3rd Qu.: 2.000    3rd Qu.: 2.000    3rd Qu.:0.0000
## Max.    :23.000    Max.    :13.000    Max.    :23.000    Max.    :6.0000
## NA's    :144      NA's    :144      NA's    :144      NA's    :144
##      PKatt         CrdY         CrdR         GlS.1
## Min.    :0.0000    Min.    : 0.000    Min.    :0.00000    Min.    :0.0000
## 1st Qu.:0.0000    1st Qu.: 0.000    1st Qu.:0.00000    1st Qu.:0.0000
## Median :0.0000    Median : 2.000    Median :0.00000    Median :0.0300
## Mean    :0.1883    Mean    : 2.452    Mean    :0.07861    Mean    :0.1104
## 3rd Qu.:0.0000    3rd Qu.: 4.000    3rd Qu.:0.00000    3rd Qu.:0.1500
## Max.    :7.0000    Max.    :11.000    Max.    :2.00000    Max.    :2.0300
## NA's    :144      NA's    :144      NA's    :144      NA's    :145
##      Ast.1         G.A         G.PK.1         G.A.PK
## Min.    : 0.0000    Min.    : 0.0000    Min.    :0.0000    Min.    : 0.0000
## 1st Qu.: 0.0000    1st Qu.: 0.0000    1st Qu.:0.0000    1st Qu.: 0.0000
## Median : 0.0300    Median : 0.1000    Median :0.0300    Median : 0.1000
## Mean    : 0.1003    Mean    : 0.2107    Mean    :0.1032    Mean    : 0.2034
## 3rd Qu.: 0.1200    3rd Qu.: 0.2900    3rd Qu.:0.1400    3rd Qu.: 0.2800
## Max.    :11.2500    Max.    :11.2500    Max.    :2.0300    Max.    :11.2500
## NA's    :145      NA's    :145      NA's    :145      NA's    :145
##      xG           npG         xA           npG.xA
## Min.    : 0.000    Min.    : 0.000    Min.    : 0.000    Min.    : 0.000
## 1st Qu.: 0.100    1st Qu.: 0.100    1st Qu.: 0.100    1st Qu.: 0.300
## Median : 0.800    Median : 0.750    Median : 0.650    Median : 1.600
## Mean    : 1.929    Mean    : 1.785    Mean    : 1.301    Mean    : 3.089
## 3rd Qu.: 2.500    3rd Qu.: 2.400    3rd Qu.: 1.900    3rd Qu.: 4.300
## Max.    :21.800    Max.    :17.100    Max.    :11.200    Max.    :27.400
## NA's    :145      NA's    :145      NA's    :145      NA's    :145
##      xG.1         xA.1         xG.xA         npG.1
## Min.    :0.0000    Min.    :0.00000    Min.    :0.00    Min.    :0.0000
## 1st Qu.:0.0200    1st Qu.:0.01000    1st Qu.:0.05    1st Qu.:0.0125
## Median :0.0600    Median :0.06000    Median :0.13    Median :0.0600
## Mean    :0.1372    Mean    :0.09262    Mean    :0.23    Mean    :0.1301
## 3rd Qu.:0.1700    3rd Qu.:0.12000    3rd Qu.:0.33    3rd Qu.:0.1600
## Max.    :4.4800    Max.    :6.50000    Max.    :6.50    Max.    :4.4800
## NA's    :145      NA's    :145      NA's    :145      NA's    :145
##      npG.xA.1
## Min.    :0.000
## 1st Qu.:0.050

```

```
## Median :0.130
## Mean   :0.223
## 3rd Qu.:0.310
## Max.   :6.500
## NA's   :145
```

Podatci izgledaju dobro i možemo primjetiti par stvari:

1. Igrači koji su odigrali 0 utakmica (MP=0) nemaju podatke o golovima, asistencijama i sličnim kolumnama (X90s, GLs, ... = NA) te za neke fali informacija o minutama igre. Ovo nije problem jer to znači da ove vrijednosti koje fale trebaju biti 0.
2. Za određene igrače fali informacija o njihovim godinama. Ovo predstavlja problem koji treba riješiti. Najjednostavnije rješenje ovog problema bi bilo pronalazak tih informacija i ručna nadopuna. Postoje i razni drugi načini nadopune podataka koji fale, načini koji su utemeljeni na statističkim svojstvima svih podataka. Mi smo se odlučili za pristup izbacivanja takvih podataka - dataset je dovoljno velik (691 podatak), i broj igrača za koje ne postoji informacija o godinama (njih 4) je dovoljno malen da bi ovakav pristup funkcionirao.

```
players <- players[!is.na(players$Age),]
```

1. zadatak: postoji li razlika u broju odigranih minuta mladih igrača (do 25 godina) među premierligaškim ekipama?

Da bi odgovorili na ovaj zadatak, potrebne su nam samo dvije kolumne iz našeg dataseta: kolumna o godinama (Age) te kolumna o odigranim minutama (Min). Kolumnu Age smo već očistili od nepostojećih podataka, dok kolumnu Min trebamo popraviti - popuniti nepostojeće podatke nulama. Učinimo to sada.

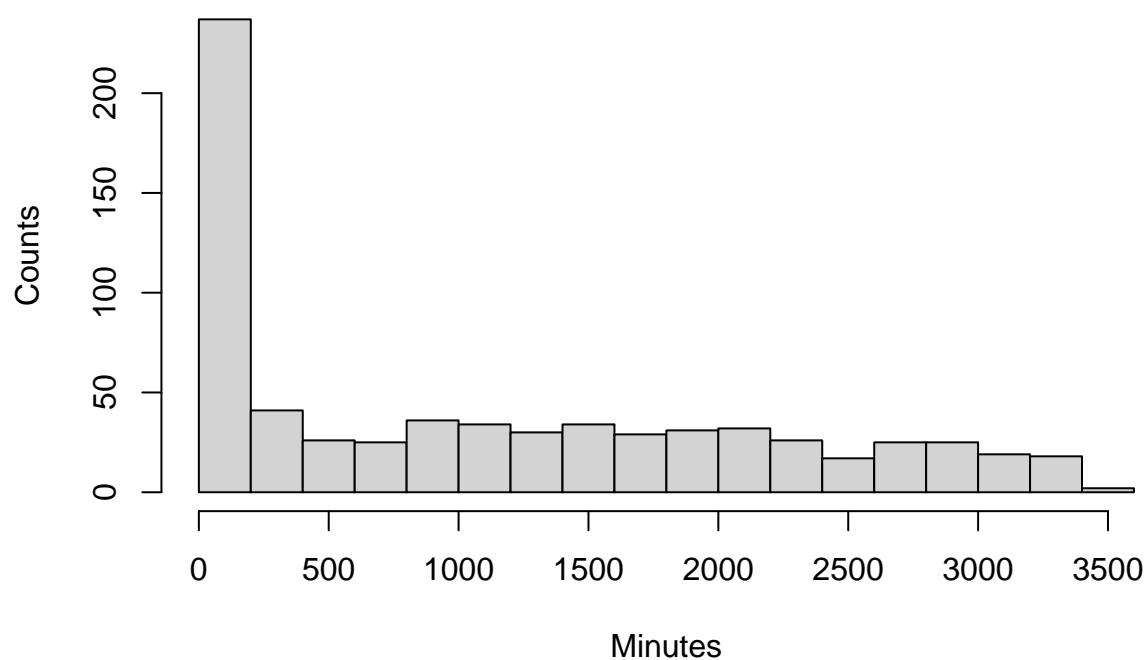
```
players <- players %>% mutate(Min = coalesce(Min, 0))
```

Prije konkretne analize, pogledajmo distribuciju odigranih minuta mladih igrača.

```
# Izdvajanje mladih igrača (do 25 godina)
young_players <- players[players$Age <= 25,]
```

```
hist(players$Min, breaks=20, main="Distribution of played minutes for young players", xlab="Minutes", ylab="Frequency")
```

Distribution of played minutes for young players

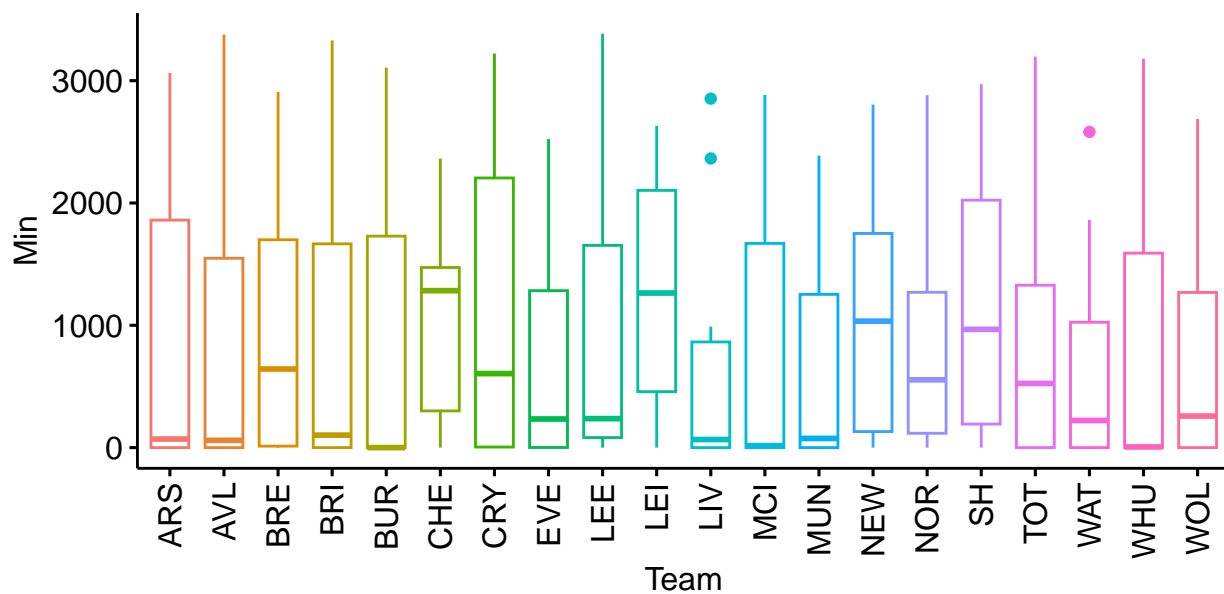


Iz grafa možemo zaključiti da velik broj igrača ne igra utakmice ili igraju jako malo, dok uspješniji igrači imaju podjednaku distribuciju odigranih minuta sve do 3500.

Pregled distribucija minuta po timovima je malo složenija vizualizacija - iskoristit ćemo box-plot po timovima.

```
ggboxplot(young_players, x = "Team", y = "Min", color = "Team", font.label = list(size=20)) + scale_x_d
```

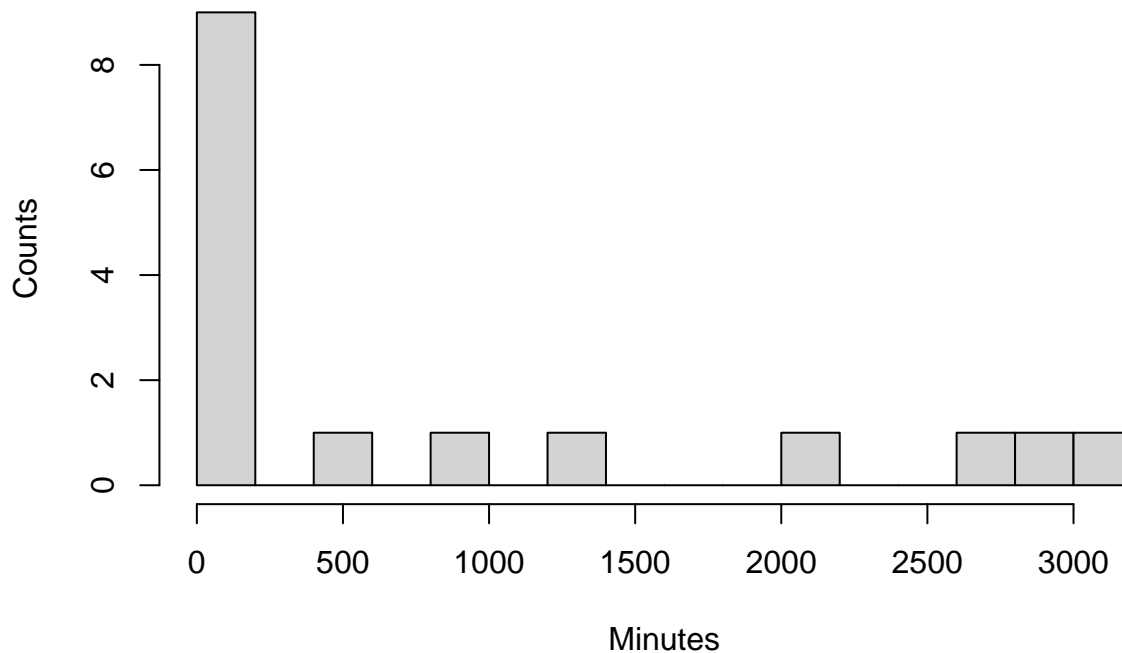
Arsenal	Burnley	Leeds United	Manchester United	Totterham
Aston Villa	Chelsea	Leicester City	Newcastle United	Watford
Brentford	Crystal Palace	Liverpool	Norwich City	West Ham United
Brighton & Hove Albion	Everton	Manchester City	Southampton	Wolverhampton



Pogledajmo još distribuciju odigranih minuta za određenu ekipu, recimo West Ham United.

```
westham_young_players <- young_players[young_players$Team == "West Ham United",]
hist(westham_young_players$Min, breaks=20, main="Distribution of played minutes by young players for West Ham United")
```

Distribution of played minutes by young players for West Ham Unite



Iz danog grafa mislimo da je jasno vidljivo da distribucija uzorka ne prati normalnu distribuciju. Ne-normalnost distribucija može se testirati Lilliefors testom (podaci moraju biti neovisni, i veličina uzorka mora biti dovoljno velika). Nulta hipoteza test je da podaci dolaze iz normalne distribucije, dok je alternativna hipoteza da ne dolaze.

```
teams <- unique(young_players$Team)

for (team in teams) {

  team_data <- subset(young_players, Team == team)

  lillie_test <- lillie.test(team_data$Min)

  print(paste("Team:", team))
  print(lillie_test)
}

## [1] "Team: Arsenal"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.28489, p-value = 1.928e-06
##
## [1] "Team: Aston Villa"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
```

```

##
## data: team_data$Min
## D = 0.29213, p-value = 1.176e-05
##
## [1] "Team: Brentford"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.21197, p-value = 0.00235
##
## [1] "Team: Brighton & Hove Albion"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.27577, p-value = 0.0001309
##
## [1] "Team: Burnley"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.33648, p-value = 0.001034
##
## [1] "Team: Chelsea"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.17467, p-value = 0.2509
##
## [1] "Team: Crystal Palace"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.24696, p-value = 0.01437
##
## [1] "Team: Everton"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.24631, p-value = 0.0004142
##
## [1] "Team: Leeds United"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.26152, p-value = 0.000266
##
## [1] "Team: Leicester City"

```



```

##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.12818, p-value = 0.5655
##
## [1] "Team: Liverpool"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.26641, p-value = 0.0009823
##
## [1] "Team: Manchester City"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.32571, p-value = 2.019e-05
##
## [1] "Team: Manchester United"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.32509, p-value = 1.112e-05
##
## [1] "Team: Newcastle United"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.21013, p-value = 0.1909
##
## [1] "Team: Norwich City"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.17087, p-value = 0.06807
##
## [1] "Team: Southampton"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.14578, p-value = 0.3964
##
## [1] "Team: Tottenham Hotspur"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.19895, p-value = 0.00949

```

```
##
## [1] "Team: Watford"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.24017, p-value = 0.005183
##
## [1] "Team: West Ham United"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.32487, p-value = 7.83e-05
##
## [1] "Team: Wolverhampton Wanderers"
##
## Lilliefors (Kolmogorov-Smirnov) normality test
##
## data: team_data$Min
## D = 0.23191, p-value = 0.003263
```

Krenimo sada s analizom pitanja. Da bi odgovorili na ovo pitanje moramo usporediti distribuciju odigranih minuta svih timova premier lige (njih 20). Činjenica da je broj skupina koje uspoređujemo veći od 2 odbacuje mogućnost korištenja “jednostavnih” statističkih testova poput t-testa. Metoda koja nam omogućuje statistički odgovor na zadano pitanje je ANOVA (Analysis of Variance).

ANOVA je statistički test koji nam govori jesu li sredine dviju ili više populacija jednake, te je generalizacija t-testa na više od dvije distribucije. Drugim riječima, nulta hipoteza ANOVA je da su srednje vrijednosti svih testiranih populacija jednake, a sukladna p-vrijednost nam govori kolika je vjerojatnost da dobijemo videnu populacijom pod pretpostavkom nasumičnog uzorkovanja iz distribucija jednakih srednjih vrijednosti.

ANOVA koristi sljedeće pretpostavke:

1. Normalnost: podaci moraju biti normalno distribuirani u svakoj skupini.
2. Homogenost varijance: varijanca svake skupine mora biti jednaka.
3. Nezavisnost: podaci u svakoj skupini moraju biti neovisni jedni od drugih. Iz prijašnjih grafova, možemo vidjeti da pretpostavka normalnosti ne vrijedi za sve ekipe, štoviše za većinu ekipa ne vrijedi. To znači da ANOVA vjerojatno neće dati dobre rezultate.

Srećom, postoji alternativa: neparametarski test, Kruskal-Wallisov test. Kruskal-Wallis test će izračunati p-vrijednost koja odgovara na pitanje: postoji li značajna razlika u broju odigranih minuta među timovima za mlađe igrače. Ako je p-vrijednost manja od zadane razine značajnosti (obično 0,05), onda možemo zaključiti da postoji značajna razlika u broju odigranih minuta među timovima za mlađe igrače. Uvjet provođenja Kruskal-Wallisovog testa je da veličina svakog uzorka mora biti barem 5, što u našem slučaju vrijedi.

1. H_0 : medijani distribucija svih uzoraka su jednaki. H_1 : barem dva medijana nisu jednaka
2. Uzmimo razinu značajnosti $\alpha = 0.05$.

```
# Kruskal-Wallis test
result <- kruskal.test(Min ~ Team, data = young_players)

print(result)

##
## Kruskal-Wallis rank sum test
##
## data: Min by Team
```

```
## Kruskal-Wallis chi-squared = 15.753, df = 19, p-value = 0.6737
```

Dobivena p-vrijednost je 0.6737, mnogo veća od 0.05. Ne možemo odbaciti nultu hipotezu H_0 te zaključujemo da ne postoji razlika među odigranim minutama mladih igrača između timova premier lige.

2. zadatak: dobivaju li u prosjeku više žutih kartona napadači ili igrači veznog reda?

Da bi odgovorili na ovo pitanje, koristit ćemo tri kolumne u zadanom datasetu: broj dobivenih žutih karton (CrdY), poziciju igrača (Pos), te broj odigranih utakmica (MP).

Kao i prije, imamo problem s jednom od kolumni: neke vrijednosti CrdY fale. Za igrače koji nisu odigrali niti jednu minutu utakmica logično da vrijednost broja dobivenih žutih kartona fali. Taj broj dobivenih žutih kartona je tehnički 0, no smatramo da ovdje treba razlikovati igrače koji su odigrali neke utakmice i nisu dobili niti jedan žuti karton (valjana pretpostavka je da igraju “čisto”, ne krše protivnike), za razliku od igrača koji uopće nisu igrali - ne možemo zaključiti da igraju “čisto” ili “prljavo”. Igrače koji nisu uopće igrali ćemo izbaciti.

```
players_who_played <- players[players$MP > 0, c("Player", "Team", "Pos", "MP", "CrdY")]
```

Dalje, vjerojatnost da igrač dobije žuti karton sigurno raste s količinom odigranih minuta. No, kako igrač može dobiti maksimalno 2 žuta kartona po utakmici, više smisla ima gledati broj žutih kartona po utakmici. U tu svrhu, umjesto direktne usporedbe broja žutih kartona, uspoređivati ćemo broj žutih kartona po broju odigranih utakmica (nazovimo to CrdY.MP)

```
players_who_played$CrdY.MP <- players_who_played$CrdY / players_who_played$MP
```

Pogledajmo kako izgledaju ti podaci.

```
summary(players_who_played)
```

```
##      Player      Team      Pos      MP
## Length:546   Length:546   Length:546   Min.    : 1.00
## Class :character Class :character Class :character 1st Qu.: 9.00
## Mode  :character Mode  :character Mode  :character Median :20.00
##                                     Mean  :19.20
##                                     3rd Qu.:29.75
##                                     Max.   :38.00
##      CrdY      CrdY.MP
## Min.    : 0.000   Min.    :0.00000
## 1st Qu.: 0.000   1st Qu.:0.00000
## Median : 2.000   Median :0.09091
## Mean    : 2.454   Mean    :0.11597
## 3rd Qu.: 4.000   3rd Qu.:0.18182
## Max.    :11.000   Max.    :1.00000
```

```
position_counts <- table(players_who_played$Pos)
kable(position_counts, caption = "Table 1: Number of players by their positions", align="c")
```

Table 1: Table 1: Number of players by their positions

Var1	Freq
DF	185
DF,FW	4
DF,MF	5
FW	83
FW,DF	2

Var1	Freq
FW,MF	59
GK	42
MF	116
MF,DF	11
MF,FW	39

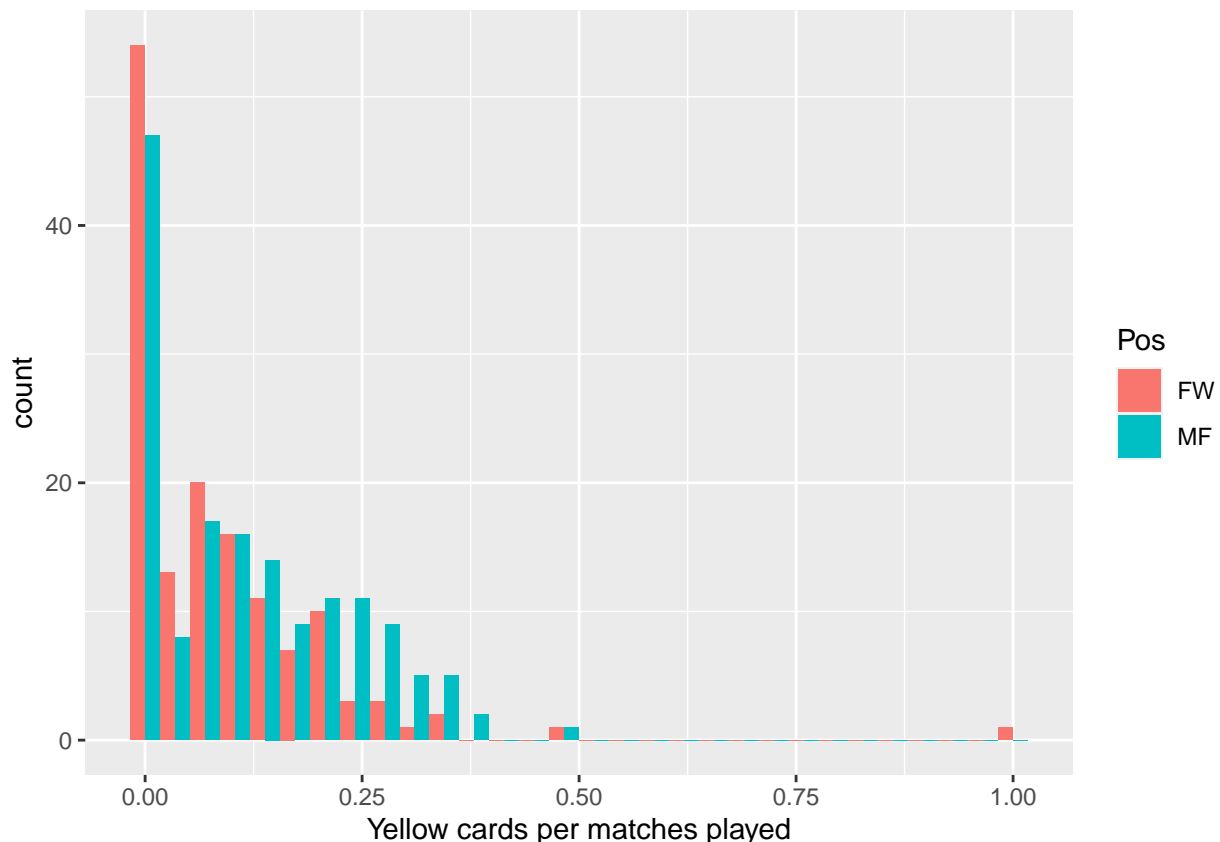
Napokon, preostaje diskusija o kolumni pozicije (Pos). Ona je vrlo jednostavna: opisuje koju poziciju igra koji igrač. Pojedini igrači su svrstani u više kategorija, poput napadača i veznog igrača. U ovom zadatku, nas interesiraju samo napadači (FW) te vezni igrači (MF). Iz tablice 1 možemo vidjeti da imamo podatke o 83 napadača, 116 veznih igrača, 59 napadača/veznih, 39 veznih/napadača. Kod kombiniranih pozicija, pretpostavljamo da je prvo napisana ona pozicija koju napadač preferira/većinom igra, te ćemo tako napadača/veznog (FW,MF) igrača svrstati kao čistog napadača (FW), a veznog/napadača (MF,FW) kao veznog igrača (MF).

```
players_who_played$Pos[players_who_played$Pos == "FW,MF"] <- "FW"
players_who_played$Pos[players_who_played$Pos == "MF,FW"] <- "MF"
midfielders_and_forwards = players_who_played[players_who_played$Pos == "FW" | players_who_played$Pos == "MF"]
```

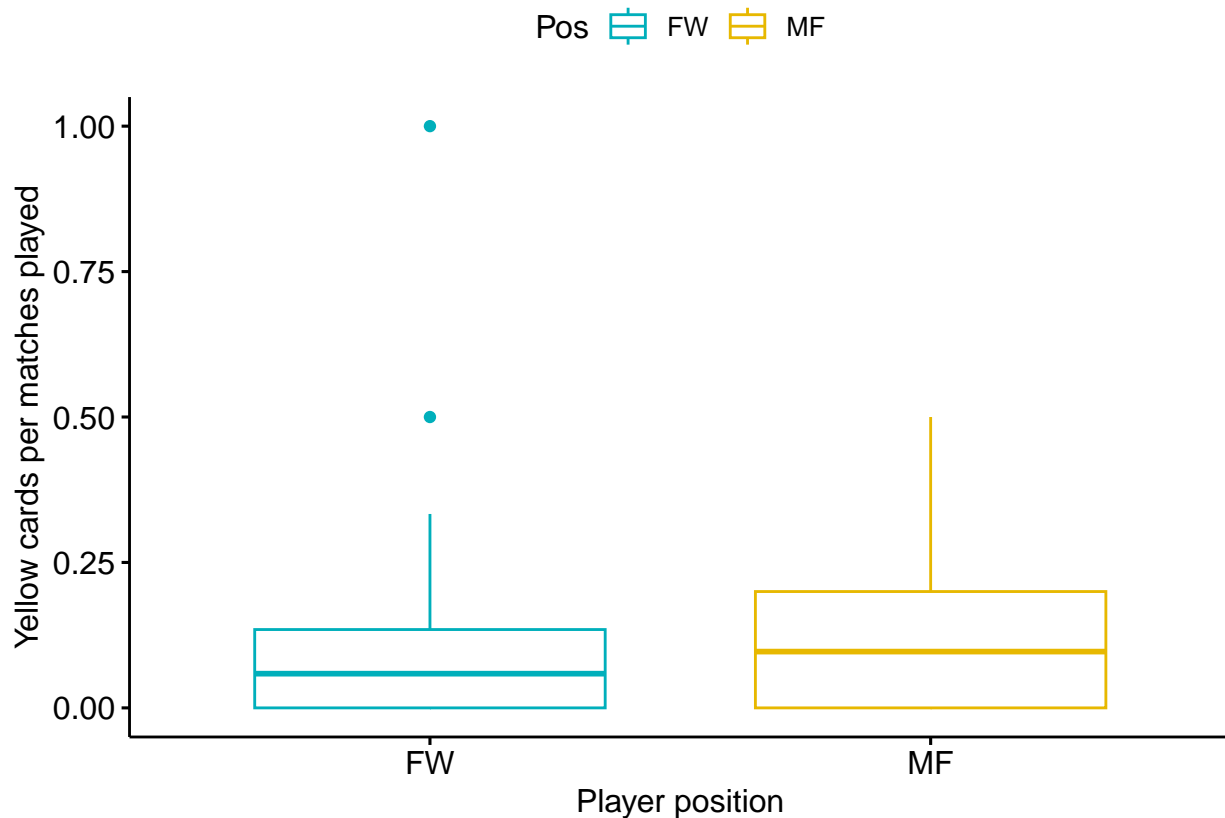
Kao i u prijašnjem zadatku, prije ikakve analize vizualizirat ćemo dane podatke.

```
# Create a histogram of the column 'CrdY.MP' split by the column 'MP'
ggplot(midfielders_and_forwards, aes(x = CrdY.MP, fill = Pos)) +
  geom_histogram(position = "dodge") + xlab("Yellow cards per matches played")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggboxplot(midfielders_and_forwards, x = "Pos", y = "CrdY.MP",
          color = "Pos", palette = c("#00AFBB", "#E7B800"),
          ylab = "Yellow cards per matches played", xlab = "Player position")
```



Dvije populacije koje uspoređujemo nisu normalne. To znači da ne možemo koristiti standardan t-test. U spas ponovno dolaze neparametarski testovi - specifično Wilcoxonov rank-sum test. On uspoređuje medijan dva uzorka. Test je baziran na ukupnim rangovima observacija, a ne na konkretnim vrijednostima.

Postava eksperimenta je slijedeća:

- Nulta hipoteza H_0 : medijani su jednaki
- Alternativna hipoteza H_1 : medijan veznih igrača je veći od medijana napadača
- Uzimamo razinu signifikantnosti od $\alpha = 0.05$

```
wilcox.test(midfielders_and_forwards$CrdY.MP[midfielders_and_forwards$Pos == "FW"], midfielders_and_forwards$CrdY.MP[midfielders_and_forwards$Pos == "MF"],
            data.name = "Yellow cards per matches played",
            alternative = "less",
            continuity = TRUE)

##
## Wilcoxon rank sum test with continuity correction
##
## data:  midfielders_and_forwards$CrdY.MP[midfielders_and_forwards$Pos == "FW"] and midfielders_and_forwards$CrdY.MP[midfielders_and_forwards$Pos == "MF"]
## W = 8999.5, p-value = 0.002828
## alternative hypothesis: true location shift is less than 0
```

Dobivena p-vrijednost iznosi 0.0028, stoga možemo s razinom signifikantnosti od 1% odbaciti nultu hipotezu.

3. zadatak: možete li na temelju zadanih parametara odrediti uspješnost pojedinog igrača?

Predvidljivost uspješnosti pojedinog igrača zahtjeva definiciju uspješnosti. Budući da je zadani dataset relativno slabo informativan (nedostaju informacije poput broju dodavanja, broju driblinga i slično, što definira uspješnost braniča) odlučili smo se za jednostavnu metriku uspješnosti: broj zabijenih golova + broj asistencija. Također, predviđanje će biti odrađeno samo na napadačima jer pozicija napadača je definirana brojem zabijenih golova i asistencijama, dok braniči i vezni igrači generalno imaju nešto drugačije definicije uspješnosti.

U linearnu regresiju su ugrađene neke pretpostavke koje valja spomenuti: 1. Linearnost zavisnih varijabli o varijablama koje objašnjavaju 2. Normalnost grešaka 3. Neovisnost grešaka 4. Homoskedastičnost (jednakost varijanci??) 5. Ne smije biti multikolinearnosti

Kao i u prijašnjem zadatku, uzimamo samo igrače koji su odigrali barem jednu utakmicu. Također, igrače kojima piše da igraju poziciju "FW,MF" svrstavamo u napadače.

```
forward_players = players[players$MP > 0 & (players$Pos == "FW" | players$Pos == "FW,MF"),]
```

```
# Create new column 'Gls+Ast'
forward_players$GlsAst <- forward_players$Gls + forward_players$Ast

# Split data into train and test sets
set.seed(110) # for reproducibility
train_index <- sample(1:nrow(forward_players), 0.9*nrow(forward_players))
train_data <- forward_players[train_index, ]
test_data <- forward_players[-train_index, ]

# Perform linear regression on train data
model <- lm(GlsAst ~ MP + Min + PK + CrdY + CrdR + xG + xA, data = train_data)

# Print summary of model
summary(model)
```

```
##
## Call:
## lm(formula = GlsAst ~ MP + Min + PK + CrdY + CrdR + xG + xA,
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.9477 -0.9753  0.0532  0.6860  5.6336
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.4180695  0.3490420  -1.198   0.233
## MP           0.0384632  0.0403128   0.954   0.342
## Min          -0.0004090  0.0006586  -0.621   0.536
## PK           -0.0453743  0.2038567  -0.223   0.824
## CrdY          -0.0068625  0.1154469  -0.059   0.953
## CrdR           0.8037322  0.7172929   1.121   0.265
## xG            1.0552688  0.0776411  13.592 < 2e-16 ***
## xA            0.9688707  0.1532142   6.324 4.64e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.885 on 119 degrees of freedom
## Multiple R-squared:  0.9297, Adjusted R-squared:  0.9255
## F-statistic: 224.7 on 7 and 119 DF,  p-value: < 2.2e-16
```

```
tidy(model)
```

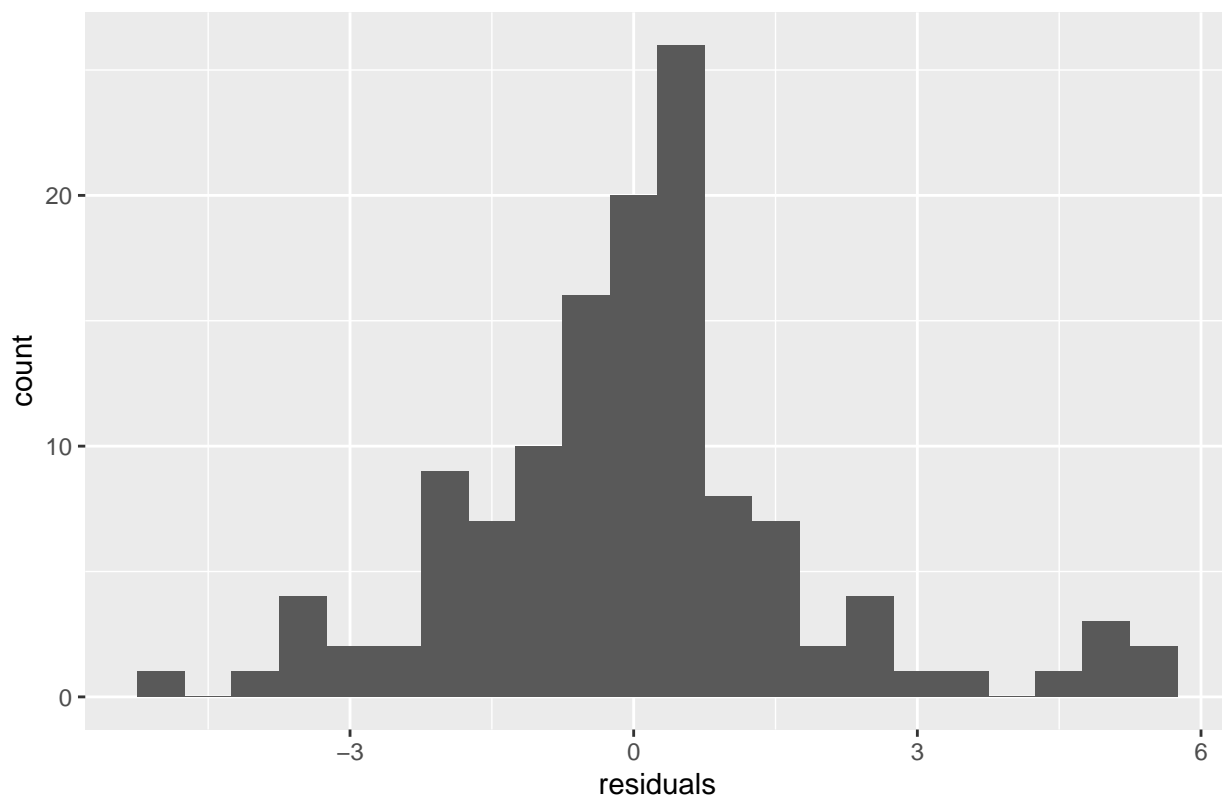
```
## # A tibble: 8 x 5
##   term      estimate std.error statistic  p.value
##   <chr>      <dbl>    <dbl>    <dbl>   <dbl>
## 1 (Intercept) -0.418    0.349    -1.20  2.33e- 1
## 2 MP          0.0385   0.0403     0.954  3.42e- 1
## 3 Min        -0.000409 0.000659   -0.621  5.36e- 1
## 4 PK         -0.0454   0.204    -0.223  8.24e- 1
## 5 CrdY       -0.00686   0.115    -0.0594 9.53e- 1
## 6 CrdR        0.804    0.717     1.12   2.65e- 1
## 7 xG          1.06    0.0776    13.6   5.70e-26
## 8 xA          0.969    0.153     6.32   4.64e- 9
```

Provjerimo prvo kakva je distribucija reziduala.

```
residuals <- residuals(model)
```

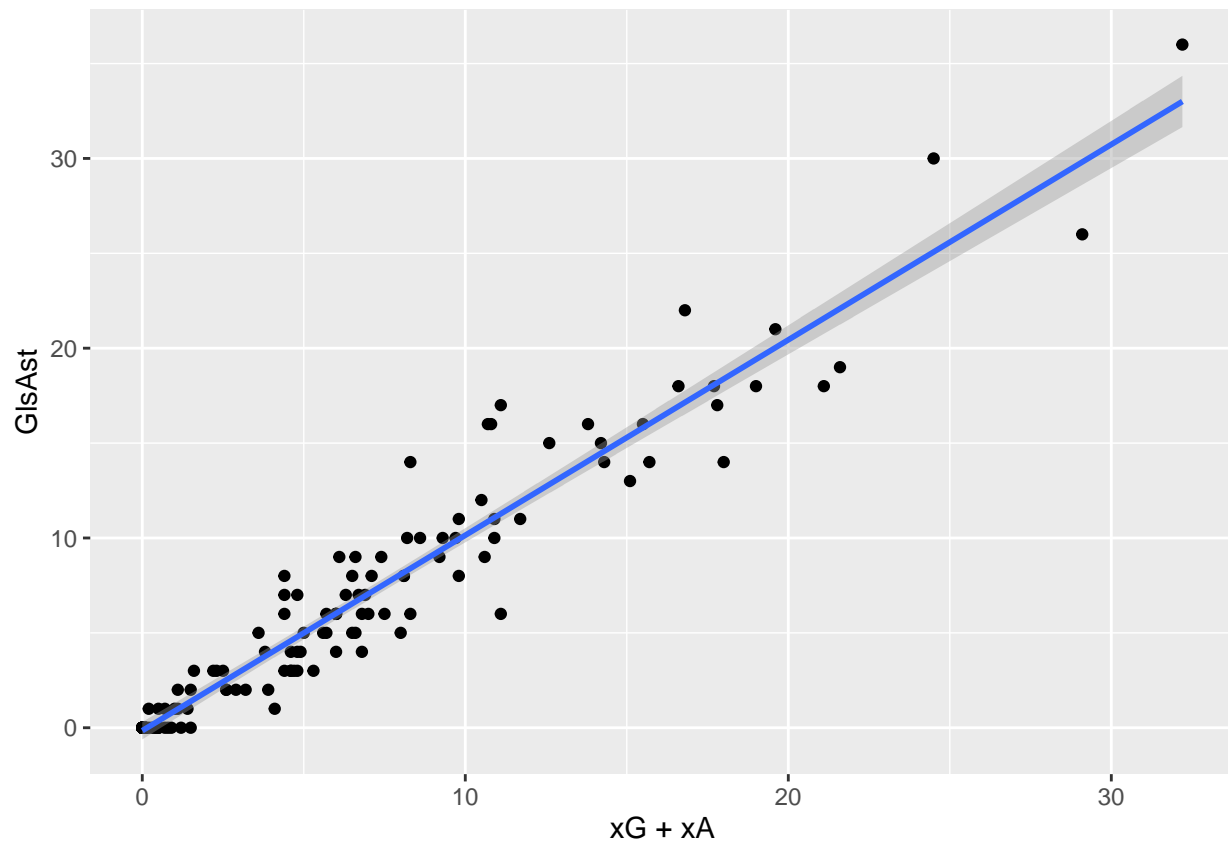
```
ggplot(data = data.frame(residuals), aes(x = residuals)) + geom_histogram(binwidth=0.5) + ggtitle("Dist
```

Distribution of Residuals



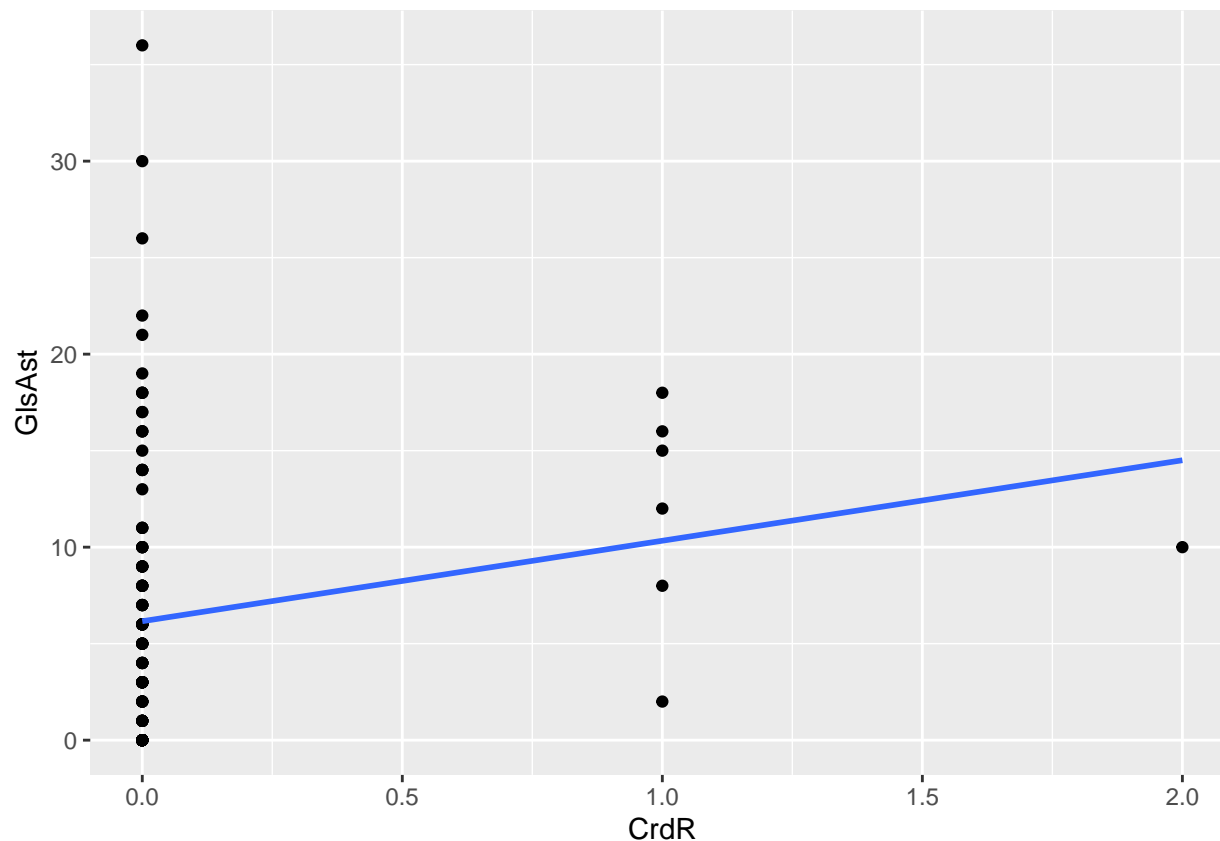
Rezultati linearni regresije su očekivani: očekivani golovi i očekivane asistencije najviše doprinose našoj definiciji uspješnosti te možemo zaključiti da je broj golova liarno ovisan o te dvije varijable na razini značajnosti boljoj od 1%. Pogledajmo tu linearnu ovisnost na grafu.

```
ggplot(train_data, aes(x = xG + xA, y = GlsAst)) + geom_point() + geom_smooth(aes(y=GlsAst), method = "lm")
## `geom_smooth()` using formula = 'y ~ x'
```



Jedan zanimljiv rezultat linearne regresije je taj što broj dobivenih crvenih kartona ima relativno velik utjecaj na konačan broj zabijenih golova i asistencija. Doduše, p -vrijednost CrdR kolumne je ipak 0.26 što je daleko od statistički značajnog. Svejedno, pogledajmo kako broj crvenih kartona utječe na broj zabijenih golova i asistencija.

```
ggplot(train_data, aes(x = CrdR, y = GlsAst)) + geom_point() + geom_smooth(aes(y=GlsAst), method = "lm")
## `geom_smooth()` using formula = 'y ~ x'
```

Odokativno, čini se da broj crvenih kartona ipak nije linearno koreliran s brojem zabijenih golova i asistencija. To možemo i potvrditi eliminacijom varijabli koje ne objašnjavaju varijablu nad kojom regresiramo.

```
model2 <- step(model)
```

```
## Start:  AIC=168.78
## GlsAst ~ MP + Min + PK + CrdY + CrdR + xG + xA
##
##           Df Sum of Sq    RSS    AIC
## - CrdY    1      0.01  422.92 166.78
## - PK      1      0.18  423.09 166.83
## - Min     1      1.37  424.28 167.19
## - MP      1      3.24  426.14 167.75
## - CrdR    1      4.46  427.37 168.11
## <none>                    422.91 168.78
## - xA      1     142.11  565.02 203.57
## - xG      1     656.51 1079.42 285.78
##
## Step:  AIC=166.78
## GlsAst ~ MP + Min + PK + CrdR + xG + xA
##
##           Df Sum of Sq    RSS    AIC
## - PK      1      0.19  423.11 164.84
## - Min     1      1.65  424.57 165.27
## - MP      1      3.29  426.21 165.77
## - CrdR    1      5.20  428.12 166.33
```

```

## <none>          422.92 166.78
## - xA    1    145.06 567.98 202.23
## - xG    1    656.61 1079.53 283.79
##
## Step: AIC=164.84
## GlsAst ~ MP + Min + CrdR + xG + xA
##
##      Df Sum of Sq    RSS    AIC
## - Min   1      1.73  424.85 163.36
## - MP    1      3.73  426.85 163.96
## - CrdR   1      5.08  428.19 164.35
## <none>          423.11 164.84
## - xA    1    145.43  568.55 200.36
## - xG    1    776.08 1199.20 295.14
##
## Step: AIC=163.36
## GlsAst ~ MP + CrdR + xG + xA
##
##      Df Sum of Sq    RSS    AIC
## - MP    1      2.12  426.97 161.99
## - CrdR   1      4.02  428.86 162.55
## <none>          424.85 163.36
## - xA    1    150.12  574.96 199.78
## - xG    1    923.72 1348.57 308.05
##
## Step: AIC=161.99
## GlsAst ~ CrdR + xG + xA
##
##      Df Sum of Sq    RSS    AIC
## - CrdR   1      5.44  432.41 161.60
## <none>          426.97 161.99
## - xA    1    213.93  640.90 211.57
## - xG    1   1069.28 1496.25 319.25
##
## Step: AIC=161.6
## GlsAst ~ xG + xA
##
##      Df Sum of Sq    RSS    AIC
## <none>          432.41 161.60
## - xA    1    218.05  650.47 211.46
## - xG    1   1077.37 1509.79 318.39

```

```

summary(model2)

```

```

##
## Call:
## lm(formula = GlsAst ~ xG + xA, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.1288 -0.9325  0.0462  0.6183  5.6391
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.15077    0.23381  -0.645    0.52

```

```
## xG          1.04525    0.05947   17.577 < 2e-16 ***
## xA          0.99402    0.12570    7.908 1.23e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.867 on 124 degrees of freedom
## Multiple R-squared:  0.9281, Adjusted R-squared:  0.9269
## F-statistic: 800.2 on 2 and 124 DF,  p-value: < 2.2e-16
```

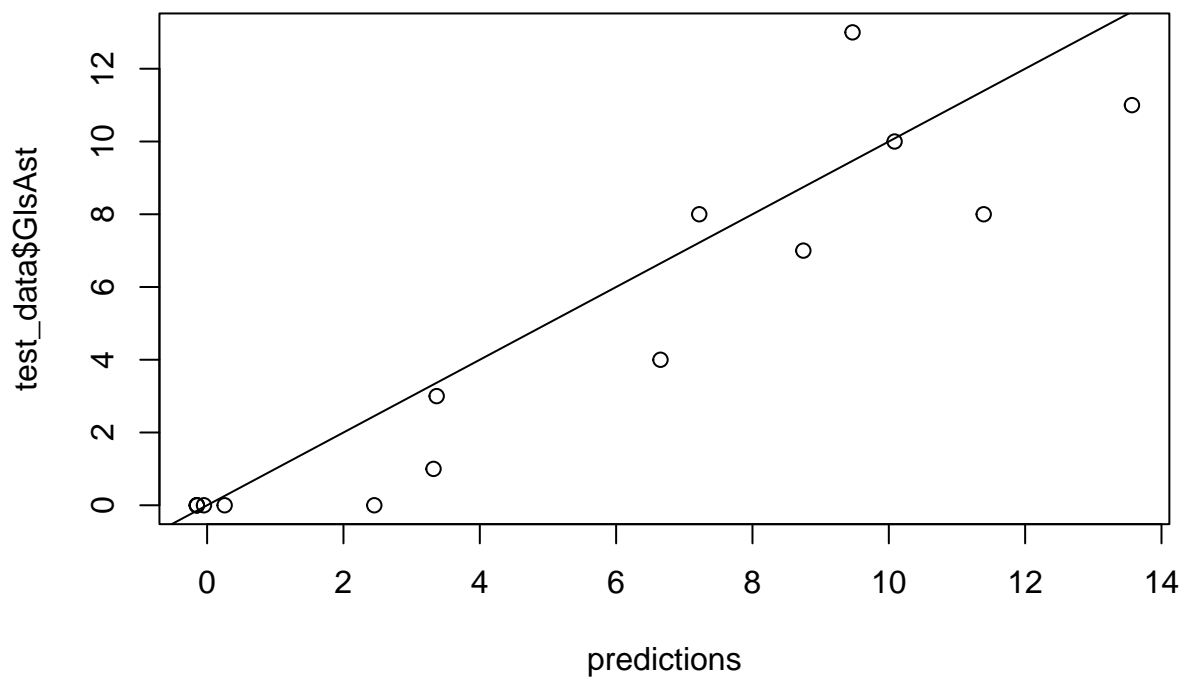
Pogledajmo sada kako generalizira naša istrenirana linearna regresija.

```
predictions <- predict(model2, newdata = test_data)
MSE <- mean((predictions - test_data$GlsAst)^2)
print(MSE)
```

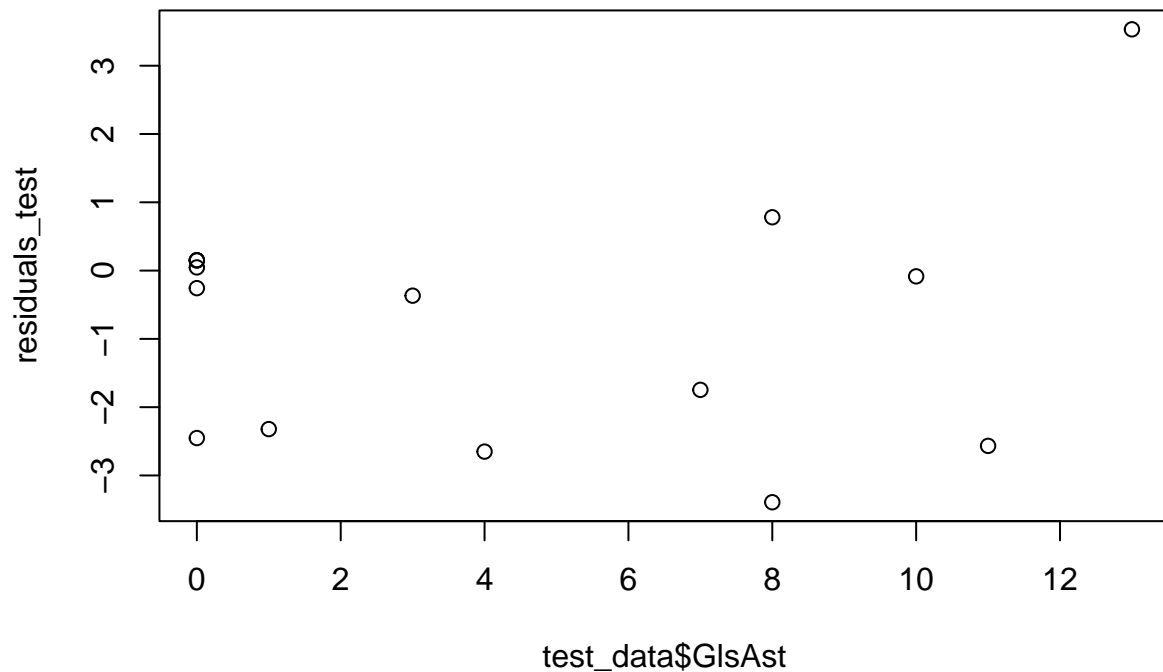
```
## [1] 3.52899
```

Prosječno kvadratno odstupanje je 3.52899. Pogledajmo na grafu stvarne vrijednosti i naše predikcije.

```
plot(predictions, test_data$GlsAst) + abline(0, 1)
```



```
## integer(0)
residuals_test <- test_data$GlsAst - predictions
plot(residuals_test ~ test_data$GlsAst)
```



4. zadatak: Doprinos li sveukupnom uspjehu svoga tima više "domaći" igrači (tj. igrači engleske nacionalnosti) ili strani igrači?

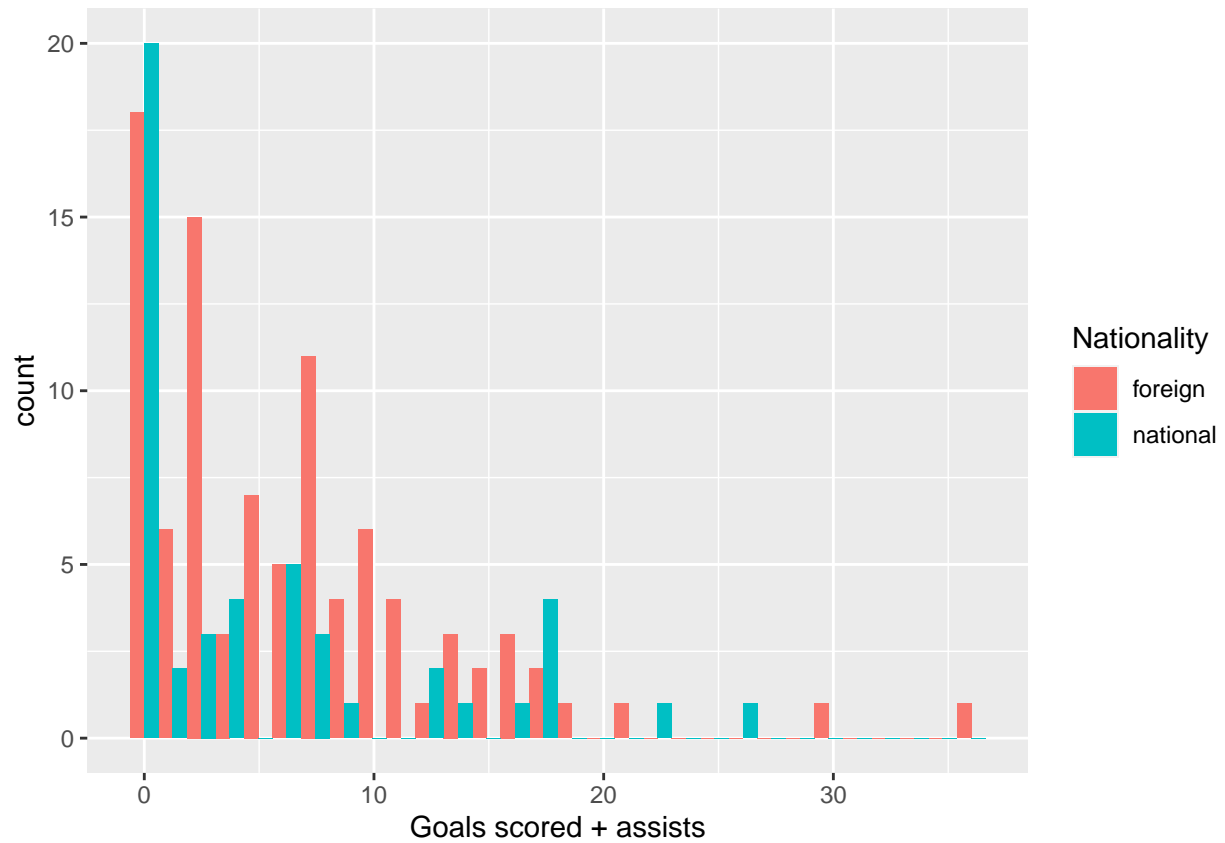
Kao i u prošlom zadatku, analizu provodimo samo za napadače.

```
forward_players$Nation <- str_sub(forward_players$Nation, -3)

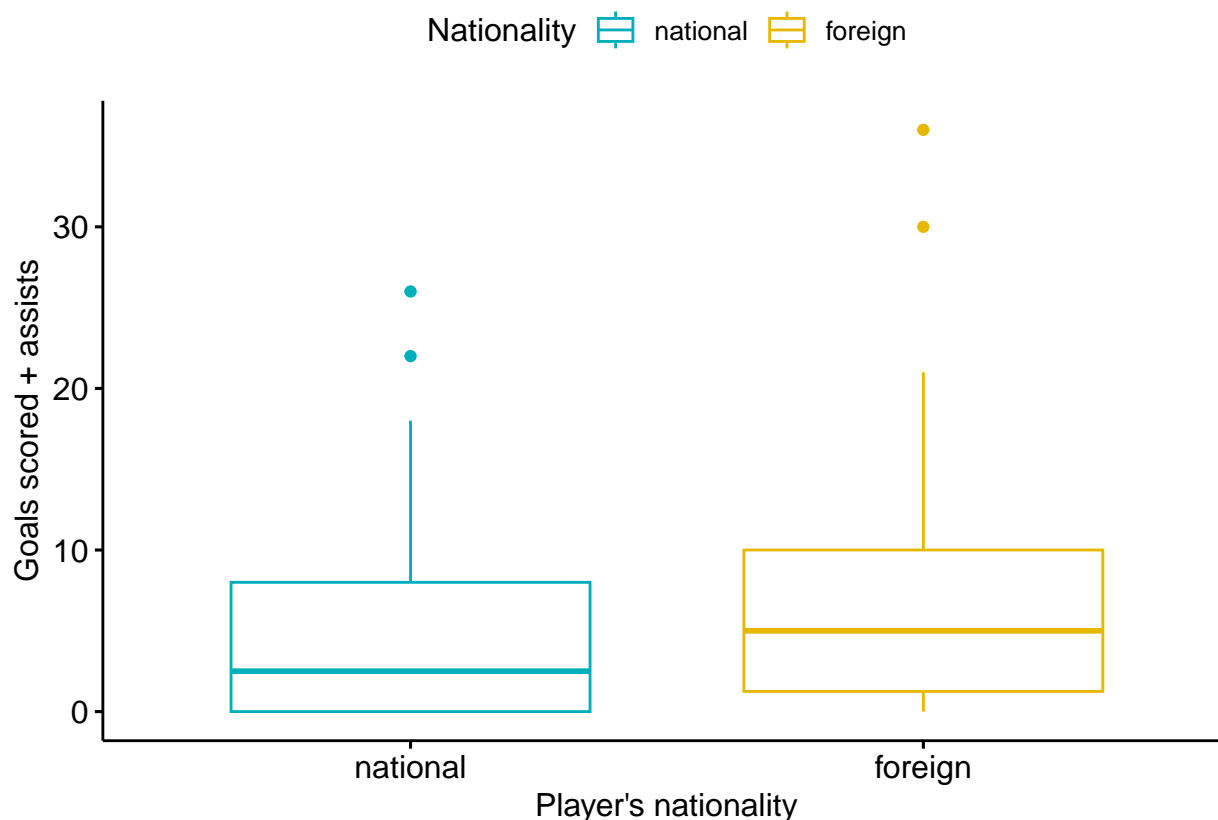
forward_players$Nationality <- ifelse(forward_players$Nation == "ENG", "national", "foreign")

# Create a histogram of the column 'GlsAst' split by the column 'Nationality'
ggplot(forward_players, aes(x = GlsAst, fill = Nationality)) +
  geom_histogram(position = "dodge") + xlab("Goals scored + assists")

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggboxplot(forward_players, x = "Nationality", y = "GlsAst",  
  color = "Nationality", palette = c("#00AFBB", "#E7B800"),  
  ylab = "Goals scored + assists", xlab = "Player's nationality")
```



Distribucije uzoraka očito nisu normalno distribuirane. To znači da ne možemo koristiti standardan t-test, već moramo primijeniti neparametarsku metodu - Wilcoxon rank-sum test. Nulta hipoteza je da su dvije distribucije jednake, tj da nema signifikantne razlike između uspjeha nacionalnih i stranih igrača. Alternativna hipoteza je da postoji neka signifikantna razlika i to specifično da strani igrači doprinose više nego nacionalni (jednostrani test). Uzmimo razinu značajnosti $\alpha = 0.05$.

```
result <- wilcox.test(GlsAst ~ Nationality, data = forward_players, alternative = "greater")
```

```
result
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data: GlsAst by Nationality
## W = 2670, p-value = 0.03575
## alternative hypothesis: true location shift is greater than 0
```

Dobivena p -vrijednost iznosi 0.03575. To znači da možemo zaključiti da strani igrači doprinose više no nacionalni igrači s razinom značajnosti od 5%.