

Eksploratorna analiza statistike o partijama društvene igre Catan

Projekt iz kolegija Osnove statističkog programiranja

Josip Hajpek, Borna Odobašić

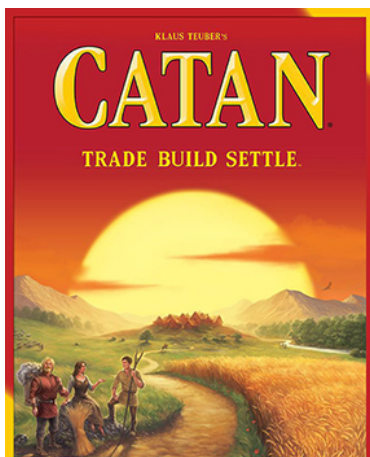
2025-01-31

Sadržaj

Uvod	1
Učitavanje paketa i podataka	2
Prilagodba podataka	3
Provjera nedostajućih vrijednosti	3
Razdioba bacanja	4
Najbolja startna pozicija	6
Analiza o najboljim resursima	8
Utjecaj razmjene resursa na pobjedu	14
Strategija naseljavanja luke na početku partije	21
Zaključak	25

Uvod

U ovoj analizi istražuju se podatci o partijama društvene igre **Catan** odigranih na web stranici playcatan.com kako bi se odgovorilo na razna pitanja o najboljim strategijama za pobjedu. Ispitat će se utjecaj startne pozicije, razmjene resursa, lovca i korištenja luka.



Učitavanje paketa i podataka

U narednim poglavljima koristit će se funkcije iz raznih paketa od kojih je većina dostupna kroz paket `tidyverse`. Nakon učitavanja spomenutog te paketa `gridExtra`, podatci csv datoteke bit će prebačeni u podatkovni okvir pomoću R-ove funkcije `read_csv`.

```
library(tidyverse)
library(gridExtra)
library(dplyr)
library(nortest)
library(RColorBrewer)
library(corrplot)
library(pROC)
library(caret)
"data/SettlersOfCatanStats.csv" %>% read_csv -> my_data
my_data %>% glimpse
```

```
## Rows: 200
## Columns: 36
## $ gameNum      <dbl> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, ~
## $ player       <dbl> 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, ~
## $ points       <dbl> 5, 9, 10, 5, 10, 6, 4, 9, 5, 10, 7, 7, 7, 10, 9, 9, 8, ~
## $ me          <dbl> NA, 1, NA, NA, NA, NA, 1, NA, NA, NA, 1, NA, NA, 1, NA~
## $ '2'         <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, ~
## $ '3'         <dbl> 3, 3, 3, 3, 6, 6, 6, 6, 3, 3, 3, 3, 6, 6, 6, 6, 3, 3, ~
## $ '4'         <dbl> 5, 5, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 7, 7, ~
## $ '5'         <dbl> 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 10, 12, 12, 12, ~
## $ '6'         <dbl> 7, 7, 7, 7, 10, 10, 10, 10, 10, 10, 10, 10, 10, 14, 14, ~
## $ '7'         <dbl> 10, 10, 10, 10, 8, 8, 8, 8, 4, 4, 4, 4, 20, 20, 20, 20~
## $ '8'         <dbl> 6, 6, 6, 6, 14, 14, 14, 14, 5, 5, 5, 5, 12, 12, 12, 12~
## $ '9'         <dbl> 7, 7, 7, 7, 9, 9, 9, 9, 5, 5, 5, 5, 11, 11, 11, 11, 9, ~
## $ '10'        <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 6, 6, 6, 6, 4, 4, 4, 4, 3, 3, ~
## $ '11'        <dbl> 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, ~
## $ '12'        <dbl> 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, ~
## $ settlement1 <dbl> 6, 5, 5, 6, 6, 5, 5, 4, 5, 5, 6, 5, 5, 6, 9, 5, 6, 5, ~
## $ ...17        <chr> "L", "W", "S", "O", "W", "C", "C", "C", "L", "W", "S", ~
## $ ...18        <dbl> 3, 8, 6, 9, 9, 8, 6, 9, 4, 8, 9, 6, 6, 9, 10, 9, 9, 6, ~
## $ ...19        <chr> "C", "O", "S", "L", "O", "S", "W", "W", "L", "O", "W", ~
## $ ...20        <dbl> 11, 10, 12, 3, 10, 10, 11, 11, 10, 10, 3, 11, 11, 3, 1~
## $ ...21        <chr> "C", "W", "W", "L", "O", "O", "O", "O", "C", "C", "L", ~
## $ settlement2 <dbl> 9, 4, 8, 4, 8, 8, 5, 6, 6, 4, 8, 8, 8, 5, 4, 8, 8, 5, ~
## $ ...23        <chr> "L", "L", "O", "L", "W", "W", "C", "L", "W", "L", "O", ~
## $ ...24        <dbl> 10, 5, 4, 8, 10, 4, 9, 4, 3, 9, 10, 4, 3, 8, 8, 4, 10, ~
## $ ...25        <chr> "W", "S", "S", "L", "L", "L", "W", "C", "L", "W", "C", ~
## $ ...26        <dbl> 11, 11, 3, 10, 0, 3, 10, 3, 11, 11, 0, 3, 0, 10, 10, 3~
## $ ...27        <chr> "O", "O", "C", "S", "2L", "W", "L", "S", "S", "S", "3G~
## $ production  <dbl> 38, 48, 44, 42, 60, 57, 44, 61, 44, 41, 47, 53, 64, 97~
## $ tradeGain    <dbl> 5, 8, 14, 12, 15, 12, 10, 16, 5, 4, 6, 2, 15, 21, 18, ~
## $ robberCardsGain <dbl> 2, 6, 9, 0, 16, 1, 8, 11, 5, 9, 5, 2, 12, 15, 7, 8, 5, ~
## $ totalGain    <dbl> 45, 62, 67, 54, 91, 70, 62, 88, 54, 54, 58, 57, 91, 13~
## $ tradeLoss    <dbl> 10, 11, 24, 24, 28, 26, 18, 25, 11, 8, 10, 4, 23, 47, ~
## $ robberCardsLoss <dbl> 2, 1, 4, 6, 10, 6, 6, 6, 1, 3, 7, 4, 5, 15, 5, 19, 11, ~
## $ tribute      <dbl> 4, 8, 0, 0, 0, 8, 8, 4, 9, 0, 0, 8, 12, 10, 0, 0, 0, 4~
## $ totalLoss    <dbl> 16, 20, 28, 30, 38, 40, 32, 35, 21, 11, 17, 16, 40, 72~
```

```
## $ totalAvailable <dbl> 29, 42, 39, 24, 53, 30, 30, 53, 33, 43, 41, 41, 51, 61~
```

Podatkovni okvir sastoji se od 200 zapisa koji pomoću 36 varijabli opisuju ključne podatke od 50 odigranih partija društvene igre Catan. Varijable opisuju startnu poziciju igrača, bodove igrača s kraja partije, koliko se bacanja svake vrijednosti dogodilo tijekom igre, početne odabire naselja, broj dobivenih resursa putem bacanja kockica, broj osvojenih te izgubljenih resursa uzrokovanih lovcem te broj odbačenih karata uzrokovanih vrtnjom broja 7.

U igri Catan postoji 5 vrsta resursa: drvo, glina, ovca, žito i kamen. Redom su označeni s: *L*, *C*, *S*, *W* i *O*. Postoje još neke kratice poput *D* za pustinju, ali za samo razumijevanje najbitniji su gornji resursi.

Prilagodba podataka

Kratki predložak iz prethodnog poglavlja daje adekvatan pogled na podatke partija, no promjena naziva varijabli, tipova varijabli ili kategorizacija varijabli doprinosi boljoj organizaciji podatkovnog okvira.

```
names(my_data)[names(my_data) == "settlement1"] <- "settlement1_01_number"
names(my_data)[names(my_data) == "...17"] <- "settlement1_01_resource"

names(my_data)[names(my_data) == "...18"] <- "settlement1_02_number"
names(my_data)[names(my_data) == "...19"] <- "settlement1_02_resource"

names(my_data)[names(my_data) == "...20"] <- "settlement1_03_number"
names(my_data)[names(my_data) == "...21"] <- "settlement1_03_resource"

names(my_data)[names(my_data) == "settlement2"] <- "settlement2_01_number"
names(my_data)[names(my_data) == "...23"] <- "settlement2_01_resource"

names(my_data)[names(my_data) == "...24"] <- "settlement2_02_number"
names(my_data)[names(my_data) == "...25"] <- "settlement2_02_resource"

names(my_data)[names(my_data) == "...26"] <- "settlement2_03_number"
names(my_data)[names(my_data) == "...27"] <- "settlement2_03_resource"

my_data$me <- !is.na(my_data$me)

ind <- my_data$settlement2_03_resource == "B"
my_data$settlement2_03_resource[ind] <-
  my_data$settlement2_02_resource[ind]
my_data$settlement2_02_resource[ind] <- "SE"

categoric_variables <- c("gameNum", "player", names(my_data)[16:27])
my_data[categoric_variables] <- lapply(my_data[categoric_variables], as.factor)
```

Provjera nedostajućih vrijednosti

Nakon prilagodbe podataka, provjerava se prisutnost nedostajućih vrijednosti u podatkovnom okviru.

```
my_data %>% lapply(is.na) %>% unlist %>% sum
```

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

Budući da nedostajućih vrijednosti nema, nije potrebno ni izbacivati obzervacije s nedostajućim podatcima niti nadopunjavati nedostajuće podatke što je pozitivna činjenica s obzirom na relativno malen broj redaka podatkovnog okvira. Sada slijede vizualizacije i analize koje će pokušati dati odgovor na određena istraživačka pitanja.

Razdioba bacanja

Potez svakog igrača, osim eventualnim aktiviranjem razvojne kartice **Vitez**, započinje vrtnjom dvije kockice nakon čega svaki igrač uzima kartice resursa koji se nalaze uz njihova naselja i gradove ako je na kartici resursa prikazana suma bačenih brojeva.

Bacanje dvije kockice modelira se pomoću diskretne slučajne varijable X čija je razdioba

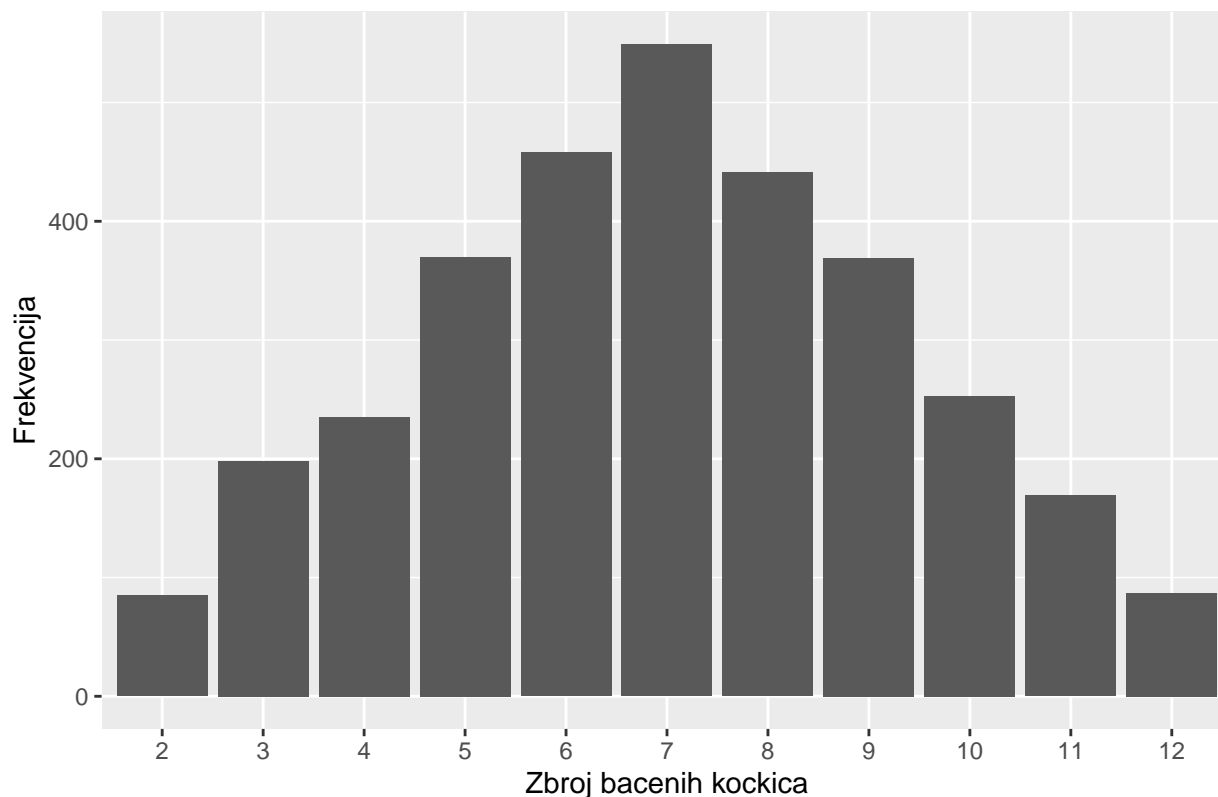
$$X \sim \left(\begin{array}{cccccccccccc} 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \frac{1}{36} & \frac{2}{36} & \frac{3}{36} & \frac{4}{36} & \frac{5}{36} & \frac{6}{36} & \frac{5}{36} & \frac{4}{36} & \frac{3}{36} & \frac{2}{36} & \frac{1}{36} \end{array} \right).$$

Očekivanje slučajne varijable X iznosi $E(X) = \sum_k x_k p_k = 7$. Dakle najvjerojatnija suma bacanja dvije kockice je 7, zatim 6 i 8 sve do 2 i 12 što su najneizgledniji ishodi.

Stupčasti dijagram ishoda svih bacanja

```
all_throws <- data.frame(x = 2:12, y = unlist(lapply(my_data[seq(1, 200, 4), 5:15], sum)))
ggplot(all_throws, aes(x = factor(x, levels = 2:12), y)) +
  geom_bar(stat = "identity") +
  labs(title = "Stupčasti dijagram ishoda svih bacanja",
       x = "Zbroj bačenih kockica", y = "Frekvencija")
```

Stupčasti dijagram ishoda svih bacanja

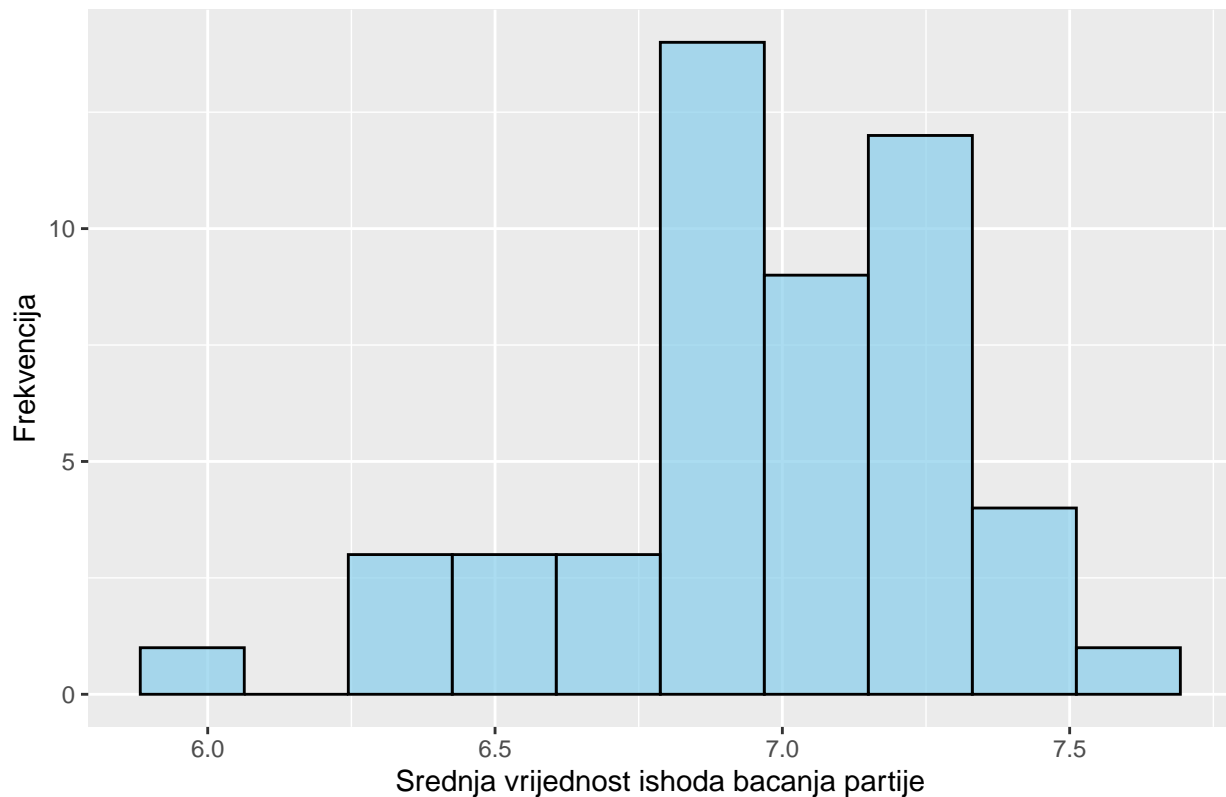


Iz priloženog stupčastog dijagrama primjetno je da su se bacanja kockica odvijala po očekivanoj distribuciji definiranoj na početku poglavlja. Osim stupčastog dijagrama pokazatelj toga može biti i histogram koji prati prosječne vrijednosti ishoda bacanja kockica u cijeloj partiji.

Histogram srednjih vrijednosti svih bacanja partije

```
throws_means <- data.frame(x = apply(as.matrix(my_data[seq(1, 200, 4), 5:15]), 1,
                                     function(x) { sum(x * (seq_along(x) + 1)) / sum(x) })),
                           aes(x)) +
ggplot(throws_means, aes(x)) +
  geom_histogram(bins = 10, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Histogram srednjih vrijednosti ishoda svih bacanja partije",
        x = "Srednja vrijednost ishoda bacanja partije", y = "Frekvencija")
```

Histogram srednjih vrijednosti ishoda svih bacanja partije



Na primjeru histograma, može se primjetiti da se srednje vrijednosti gomilaju oko vrijednosti očekivanja slučajne varijable X što ide u korist kredibiliteta web stranice playcatan.com.

Najbolja startna pozicija

Igra počinje tako da prvi igrač odabire mjesto na ploči gdje će postaviti svoje prvo naselje nakon čega ostali igrači rade upravo istu stvar. Neupućen čitatelj bi u ovom trenutku pomislio da ovakva situacija uopće nije pravedna za drugog, trećeg i četvrtog igrača, no nakon postavljanja prvog naselja, postavlja se i drugo, ali obrnutim redoslijedom tako da pravo odabira onda ima četvrti igrač, zatim treći te na kraju drugi pa prvi. Postavlja se pitanje: Koja je startna pozicija najbolja?

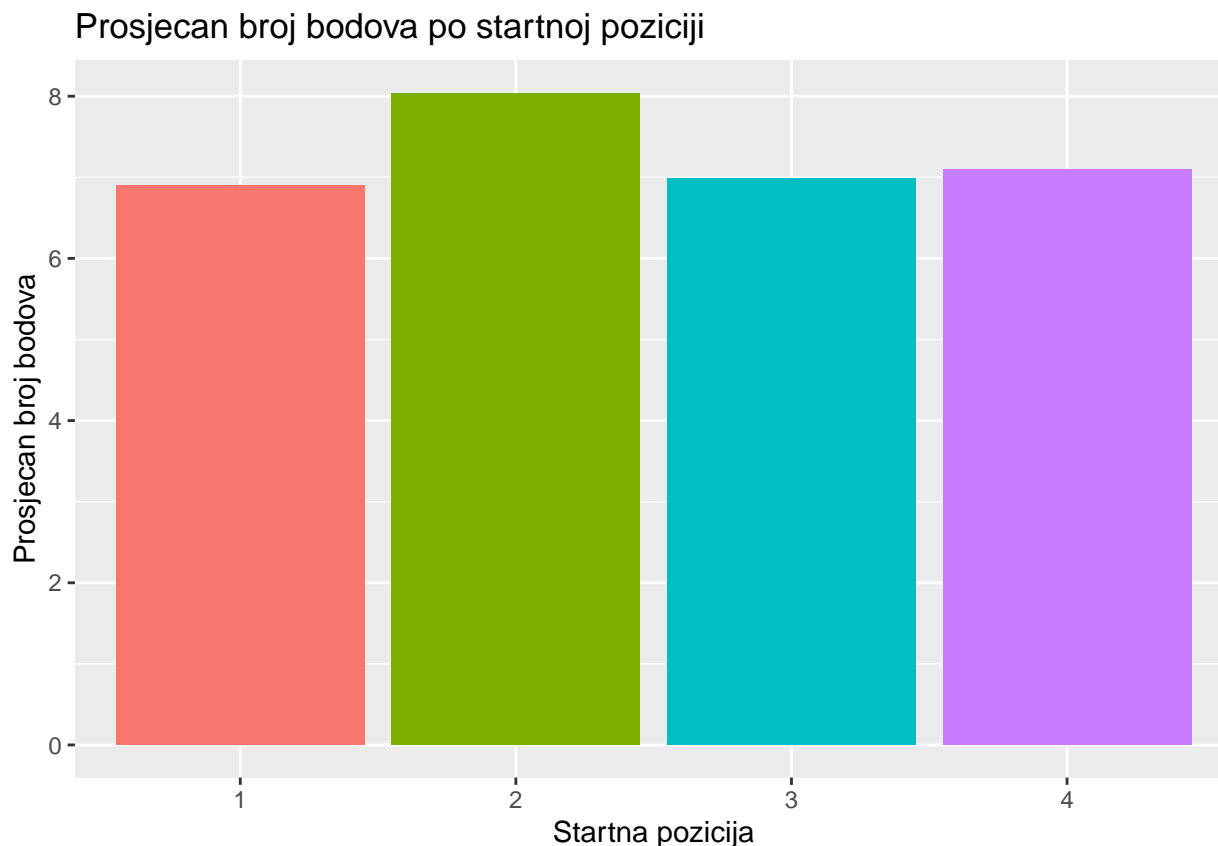
Vizualizacija prosječnog broja bodova

U nastavku prikazan je prosječan broj bodova sa završetka partije u odnosu na startnu poziciju igrača u obliku stupčastog dijagrama.

```
avg_points <- my_data %>%
  group_by(player) %>%
  summarise(prosjecni_bodovi = mean(points, na.rm = TRUE))

avg_points %>%
  ggplot(aes(x = player, y = prosjecni_bodovi, fill = player)) +
  geom_bar(stat = "identity") +
  labs(title = "Prosječan broj bodova po startnoj poziciji",
       x = "Startna pozicija",
```

```
y = "Prosječan broj bodova" +  
theme(legend.position = "none")
```



Prosječan broj bodova drugog igrača je otprilike 8 dok prvi, treći i četvrti igrač u prosjeku osvoje oko 7 bodova. Na prvi pogled, čini se da je najbolje započeti kao drugi, no detaljnijom analizom doći će se do čvršćih zaključaka.

Usporedba prosječnog broja bodova igrača različite startne pozicije

Da bi se provjerila statistički značajna razlika u prosjeku bodova pojedinog igrača koristit će se ANOVA (eng. **A**nalysis of **V**ariance). U ovom slučaju nulta hipoteza glasi Ne postoji značajna razlika u očekivanom broju bodova između startnih pozicija dok alternativna hipoteza glasi Barem jedna startna pozicija odskaače u očekivanom broju bodova.

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$$

$$H_1 : \neg H_0$$

Prilikom provođenja ANOVA testa koristit će se razina značajnosti $\alpha = 0.05$. Ova razina označava prihvatljivu granicu pogreške I. vrste. Ako je p-vrijednost manja od 0.05, odbacuje se H_0 u korist H_1 sa zaključkom da postoji značajna razlika u prosječnom broju bodova između igrača različitih startnih pozicija. Međutim prije donošenja ikakvih zaključaka potrebno je provjeriti jesu li reziduali modela iz normalne distribucije.

```
model <- aov(points ~ player, data = my_data)
shapiro.test(residuals(model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model)
## W = 0.95598, p-value = 7.447e-06
```

Zbog iznimno male p-vrijednosti Shapiro-Wilk testa normalnosti, ne smije se koristiti ANOVA, već Kruskal-Wallisov test koji je neparametarska alternativa analize varijance.

```
kruskal.test(points ~ player, data = my_data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  points by player
## Kruskal-Wallis chi-squared = 8.9346, df = 3, p-value = 0.03017
```

Na razini značajnosti $\alpha = 0.05$, odbacuje se H_0 u korist H_1 koja glasi Barem jedna startna pozicija odskaače u očekivanom broju bodova. Za kraj, preostaje jedino analizirati koje startne pozicije se značajno razlikuju u odnosu na očekivani broj bodova.

```
pairwise.wilcox.test(my_data$points, my_data$player, p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  my_data$points and my_data$player
##
##    1      2      3
## 2 0.034 -      -
## 3 0.858 0.055 -
## 4 0.826 0.055 0.858
##
## P value adjustment method: BH
```

Ako opet koristimo razinu značajnosti $\alpha = 0.05$, zaključujemo da u prosjeku, drugi igrač statistički značajno osvoji više bodova od prvog igrača što zaključujemo po p-vrijednosti od 0.034, ali nema značajne razlike između prosječnog broja bodova drugog i trećeg, odnosno drugog i četvrtog igrača gdje je u oba slučaja P-vrijednost 0.055. Iako na razini značajnosti $\alpha = 0.05$, statistički nije pronađena značajna razlika između drugog igrača i trećeg, odnosno četvrtog, postoje indicije koje ukazuju da bi drugi igrač mogao biti najpo-voljnija pozicija na početku partije.

Analiza o najboljim resursima

S obzirom na to da se u Catanu bodovi mogu ostvariti na više načina, razvijene su i neke taktike oko optimizacije načina za dolazak do pobjede. No, ipak se najviše bodova ostvaruje kroz gradove, koji se moraju nadograditi kroz naselja. Zato je u početku iznimno važno brzo proširiti mrežu cesta i sagraditi nova

naselja što je prije moguće kako bi se pokrio veći opseg brojeva i naravno pobijediti u utrci s drugim igračima za određeno mjesto.

Stoga, za očekivati je da će u početku same igre iznimno važno biti doći do drveta i gline, a kasnije tijekom igre sve važniji postaju kamen i žito.

Naravno, ulogu bi igrali i brojevi resursa, ali to ćemo zasad ostaviti po strani.~ S tim svim saznanjima, nameće se pitanje kod pobjednika partija, jesu li pri samom postavljanju pazili na koje resurse se više trebaju orijentirati. Dodatno, kartica ovce, barem u osnovnoj verziji **Catana** čini se najmanje značajnom.

Hipoteze ovog pitanja su:

H_0 : Ne postoji preferirani resurs u postavljanju početnih naselja.

H_1 : Postoji preferirani resurs u postavljanju početnih naselja.

Testove ćemo provoditi na razini značajnosti od 0.05.

```
winners <- my_data |> filter(points >= 10) |> select(contains("resource"))
```

```
winners <- winners |> mutate(across(everything(), as.character))
```

```
df <- winners |>
  mutate(
    L = rowSums(across(everything(), ~ . == "L")),
    C = rowSums(across(everything(), ~ . == "C")),
    S = rowSums(across(everything(), ~ . == "S")),
    W = rowSums(across(everything(), ~ . == "W")),
    O = rowSums(across(everything(), ~ . == "O"))
  )
```

```
df <- df[, 7:11]
```

```
mean(df$L) #drvo
```

```
## [1] 1.14
```

```
mean(df$C) #glina
```

```
## [1] 1.04
```

```
mean(df$S) #ovca
```

```
## [1] 1.3
```

```
mean(df$W) #žito
```

```
## [1] 1.28
```

```
mean(df$O) #kamen
```

```
## [1] 1.06
```

Suprotno nekim našim pretpostavkama, možemo vidjeti da u prosjeku pobjednici upravo najviše polja u svoja prva dva naselja imaju na ovcima. Glina i kamen značajno su manji, no naravno za neke točnije zaključke, provest ćemo statistički test. Moramo prvo provjeriti normalnost podataka, ovaj put *Lillieforsovom inačicom Kolmogorov-Smirnovljevog testa*.

```
lillie.test(df$L)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  df$L  
## D = 0.27618, p-value = 2.353e-10
```

```
lillie.test(df$C)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  df$C  
## D = 0.30283, p-value = 1.39e-12
```

```
lillie.test(df$S)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  df$S  
## D = 0.26812, p-value = 1.001e-09
```

```
lillie.test(df$W)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  df$W  
## D = 0.39758, p-value < 2.2e-16
```

```
lillie.test(df$O)
```

```
##  
##  Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data:  df$O  
## D = 0.27503, p-value = 2.899e-10
```

Očekivano, kako je više manje realizacija jednaka 0, 1 ili 2 nemoguće je postići normalnost u bilo kojem slučaju. Zato ćemo se odlučiti još jednom na *Kruskal-Wallis* test.

```
kruskal.test(list(df$L, df$C, df$S, df$W, df$O))
```

```
##  
##  Kruskal-Wallis rank sum test  
##  
## data:  list(df$L, df$C, df$S, df$W, df$O)  
## Kruskal-Wallis chi-squared = 5.1097, df = 4, p-value = 0.2762
```

Nakon provedbe Kruskal-Wallis testa vidimo da smo dobili p-vrijednost od 0.276. To dakako znači da ne možemo odbaciti H_0 .

Ipak, na umu treba imati i brojke koje se nalaze uz određeni resurs. Postoji mogućnost da na ovaj rezultat utječe npr. to da su pobjednici pukim slučajem imali početna naselja s poljem ovce na *lošim* brojevima. Zato ćemo zapravo ponoviti cijeli postupak (s izostavljenim testom normalnosti jer je više nego izvjesno da ga ovi podaci ne mogu proći). Jedina promjena će biti što ćemo ovaj put zanemariti mjesta s 4 najslabija broja *gledano po vjerojatnosti*. Oni bi trebali činiti $\frac{1}{6}$ bacanja i dok svakako mogu biti od koristi, nije za očekivati da bi igrači ciljano postavljali naselja na 2, 3, 11 ili 12 već je vjerojatnije da je riječ u susjednom polju nekog ciljanog *jačeg*.

```
winnersTrim <- my_data |> filter(points >= 10) |> select(settlement1_01_number:settlement2_03_resource)

winnersTrim$settlement1_01_resource[winnersTrim$settlement1_01_number %in% c(2,3, 11, 12)] <- NA
winnersTrim$settlement1_02_resource[winnersTrim$settlement1_02_number %in% c(2,3, 11, 12)] <- NA
winnersTrim$settlement1_03_resource[winnersTrim$settlement1_03_number %in% c(2,3, 11, 12)] <- NA
winnersTrim$settlement2_01_resource[winnersTrim$settlement2_01_number %in% c(2,3, 11, 12)] <- NA
winnersTrim$settlement2_02_resource[winnersTrim$settlement2_02_number %in% c(2,3, 11, 12)] <- NA
winnersTrim$settlement2_03_resource[winnersTrim$settlement2_03_number %in% c(2,3, 11, 12)] <- NA

winnersTrim <- winnersTrim |> select(contains("resource"))

winnersTrim <- winnersTrim |> mutate(across(everything(), as.character))

dfTrim <- winnersTrim |>
  mutate(
    L = rowSums(across(everything(), ~ . == "L"), na.rm = TRUE),
    C = rowSums(across(everything(), ~ . == "C"), na.rm = TRUE),
    S = rowSums(across(everything(), ~ . == "S"), na.rm = TRUE),
    W = rowSums(across(everything(), ~ . == "W"), na.rm = TRUE),
    O = rowSums(across(everything(), ~ . == "O"), na.rm = TRUE)
  )

dfTrim <- dfTrim[, 7:11]

mean(dfTrim$L) #drvo

## [1] 0.98

mean(dfTrim$C) #glina

## [1] 0.76

mean(dfTrim$S) #ovca

## [1] 0.92
```

```
mean(dfTrim$W) #žito
```

```
## [1] 1.06
```

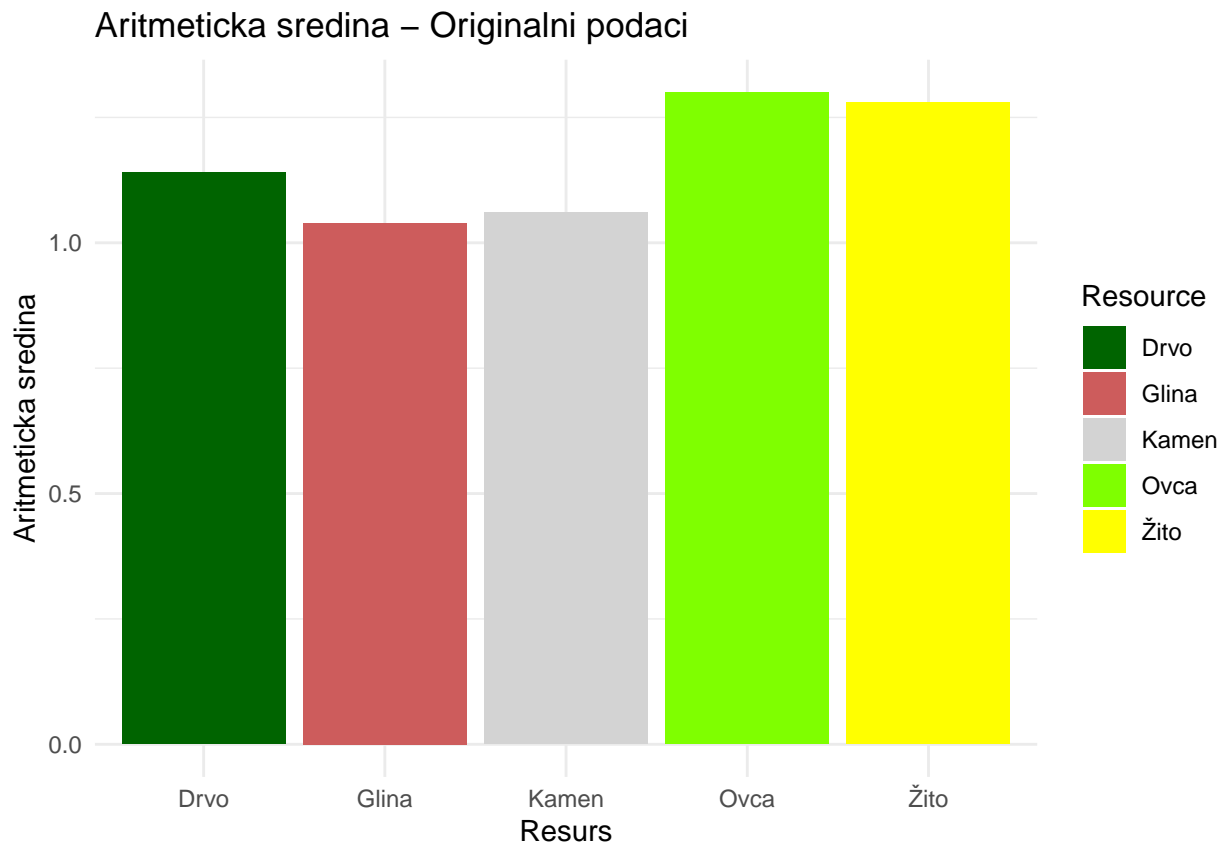
```
mean(dfTrim$O) #kamen
```

```
## [1] 0.88
```

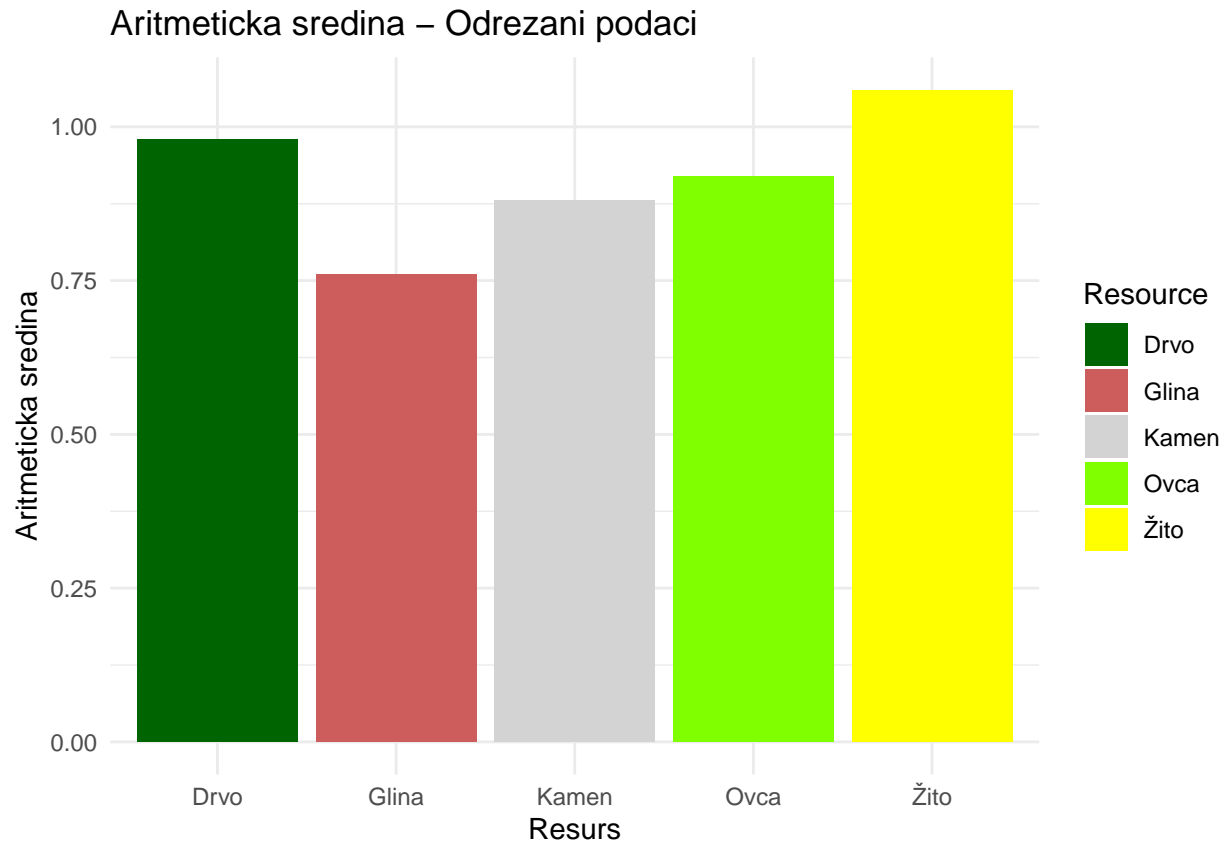
```
means_original <- colMeans(df)
df_original <- data.frame(
  Resource = c("Drvo", "Glina", "Ovca", "Žito", "Kamen"),
  Mean = means_original
)
```

```
means_trimmed <- colMeans(dfTrim)
df_trimmed <- data.frame(
  Resource = c("Drvo", "Glina", "Ovca", "Žito", "Kamen"),
  Mean = means_trimmed
)
```

```
ggplot(df_original, aes(x=Resource, y=Mean, fill=Resource)) +
  geom_bar(stat="identity") +
  labs(title="Aritmetička sredina - Originalni podaci", x="Resurs", y="Aritmetička sredina") +
  scale_fill_manual(values=c("darkgreen", "indianred", "lightgray", "chartreuse", "yellow")) +
  theme_minimal()
```



```
ggplot(df_trimmed, aes(x=Resource, y=Mean, fill=Resource)) +
  geom_bar(stat="identity") +
  labs(title="Aritmetička sredina - Odrezani podaci", x="Resurs", y="Aritmetička sredina") +
  scale_fill_manual(values=c("darkgreen", "indianred", "lightgray", "chartreuse", "yellow")) +
  theme_minimal()
```



Sredine ovih resursa predstavljaju značajno drukčiju sliku. Ovce više nisu najzastupljeniji resurs, već su puno negdje u sredini, a za razliku od prvog dijela ispitivanja sada možemo reći da su se drvo i pšenica profilirali kao najvažniji resursi. Kamen se svakako mora nalaziti iznad gline, a to u ovom slučaju i značajno jest. Konačno, provedimo još jednom *Kruskal-Wallisov test*.

```
kruskal.test(list(dfTrim$L, dfTrim$C, dfTrim$S, dfTrim$W, dfTrim$O))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  list(dfTrim$L, dfTrim$C, dfTrim$S, dfTrim$W, dfTrim$O)
## Kruskal-Wallis chi-squared = 4.7603, df = 4, p-value = 0.3128
```

Iako se poredak malo promijenio, i dalje su srednje vrijednosti prilično blizu jedne drugima i ova p-vrijednost od 0.31 ne može previše čuditi. I u ovom slučaju dakle ne odbijamo H_0 i možemo zaključiti da ne postoji nekakva razlika između resursa.

Utjecaj razmjene resursa na pobjedu

Još jedan važan aspekt *Catana* je razmjena kartica. Bilo s bankom, povlaštenim lukama ili drugim igračima. Osim toga, loše upravljanje samom količinom kartica zbog *sedmice* na kockici može biti fatalno i odlučiti partiju.

U podatkovnom skupu postoji nekoliko stupaca koji se bave dobivanjem/gubljenjem kartica. Zbog toga bi bilo odlično iskoristiti logističku regresiju kako bismo vidjeli utjecaj tih varijabli na pobjedu.

```
my_data <- my_data |> mutate(Win = ifelse(points >= 10, 1, 0)) |>
  mutate(Win = factor(Win, levels = c(0, 1)))
```

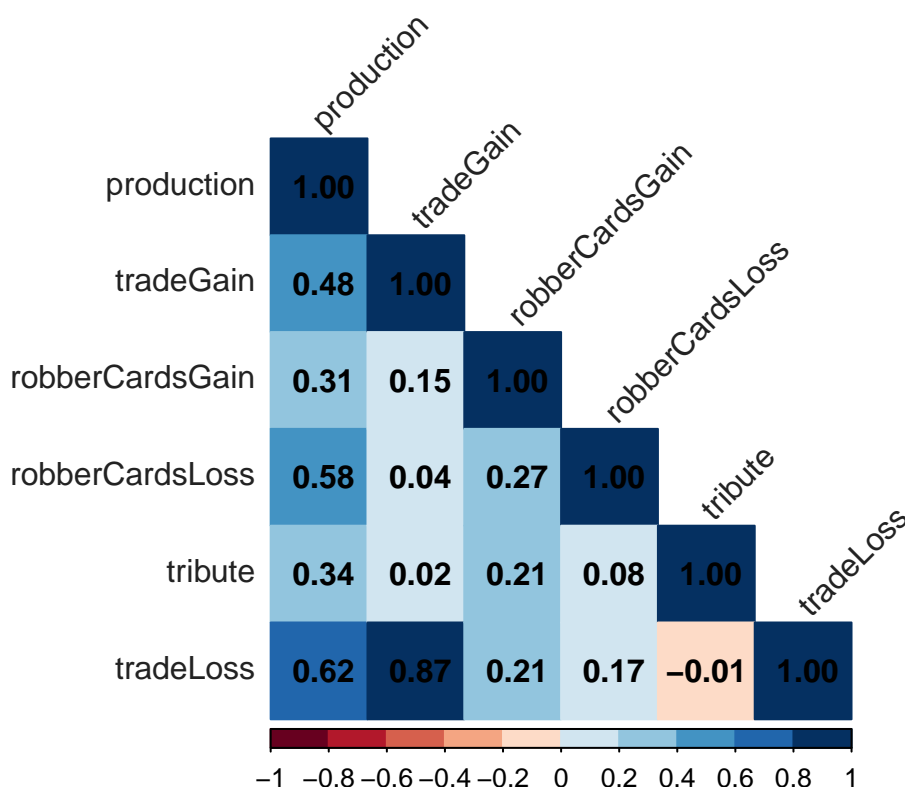
U sami podatkovni skup morali smo dodati faktor pobjede kako bismo znali rezultat partije.

Pravo pitanje prije same provedbe logističke regresije je o kojim varijablama razmjene može ovisiti pobjeda. Odmah ćemo isključiti *totale* jer su oni vezani za ostale varijable koje su puno zanimljivije za individualnu procjenu.

Varijable koje više manje sigurno dolaze u obzir su *production*, *tradeGain*, *robberCardsGain*, *robberCardsLoss* i *tribute*. Potencijalno je tu i varijabla *tradeLoss*, ali za nju bismo trebali provjeriti kolinearnost prije svega.

```
cardVars <- my_data |> select(production, tradeGain, robberCardsGain, robberCardsLoss, tribute, tradeLoss)

corrplot(cor(cardVars), type = "lower",
  col = brewer.pal(n = 10, name = "RdBu"),
  method = "color", addCoef.col = "black",
  tl.col = "#222", tl.srt = 45)
```



Sukladno nekoj pretpostavci, *tradeGain* i *tradeLoss* varijable su iznimno korelirane što je potpuno logično, tako da ćemo izbaciti *tradeLoss* i nastaviti bez te varijable.

```
cardVars <- cardVars |> select(-tradeLoss)
cardVars <- c(cardVars, my_data |> select(Win))
cardVars <- as.data.frame(cardVars)
```

Pogledajmo još neke *box-plotove* kako bismo mogli pretpostaviti na prvu kako bi se varijable mogle ponašati.

```
g1 <- ggplot(cardVars, aes(x = Win, y = production, fill = Win)) +
  geom_boxplot(color = "black") +
  labs(title = "Boxplot dobivenih kartica od naselja i gradova")

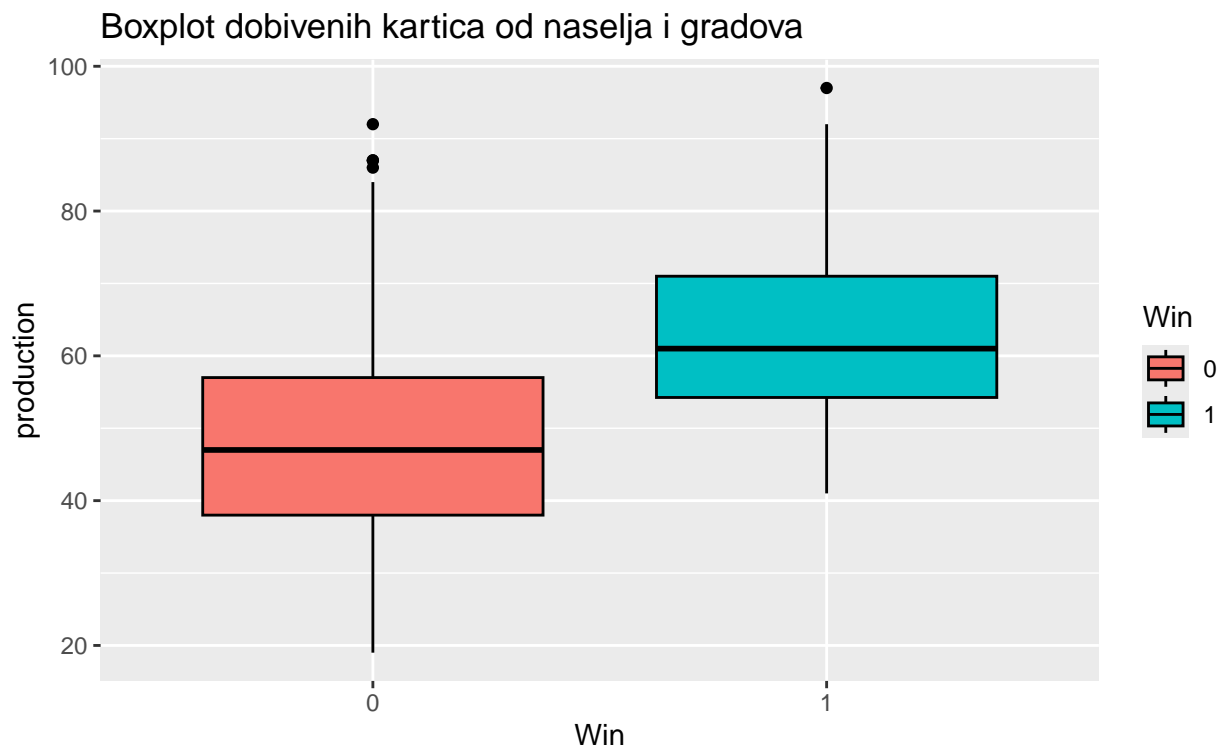
g2 <- ggplot(cardVars, aes(x = Win, y = tradeGain, fill = Win)) +
  geom_boxplot(color = "black") +
  labs(title = "Boxplot dobivenih kartica od razmjene")

g3 <- ggplot(cardVars, aes(x = Win, y = robberCardsGain, fill = Win)) +
  geom_boxplot(color = "black") +
  labs(title = "Boxplot dobivenih kartica od lopova i specijalnih kartica")

g4 <- ggplot(cardVars, aes(x = Win, y = robberCardsLoss, fill = Win)) +
  geom_boxplot(color = "black") +
  labs(title = "Boxplot izgubljenih kartica od lopova i specijalnih kartica")

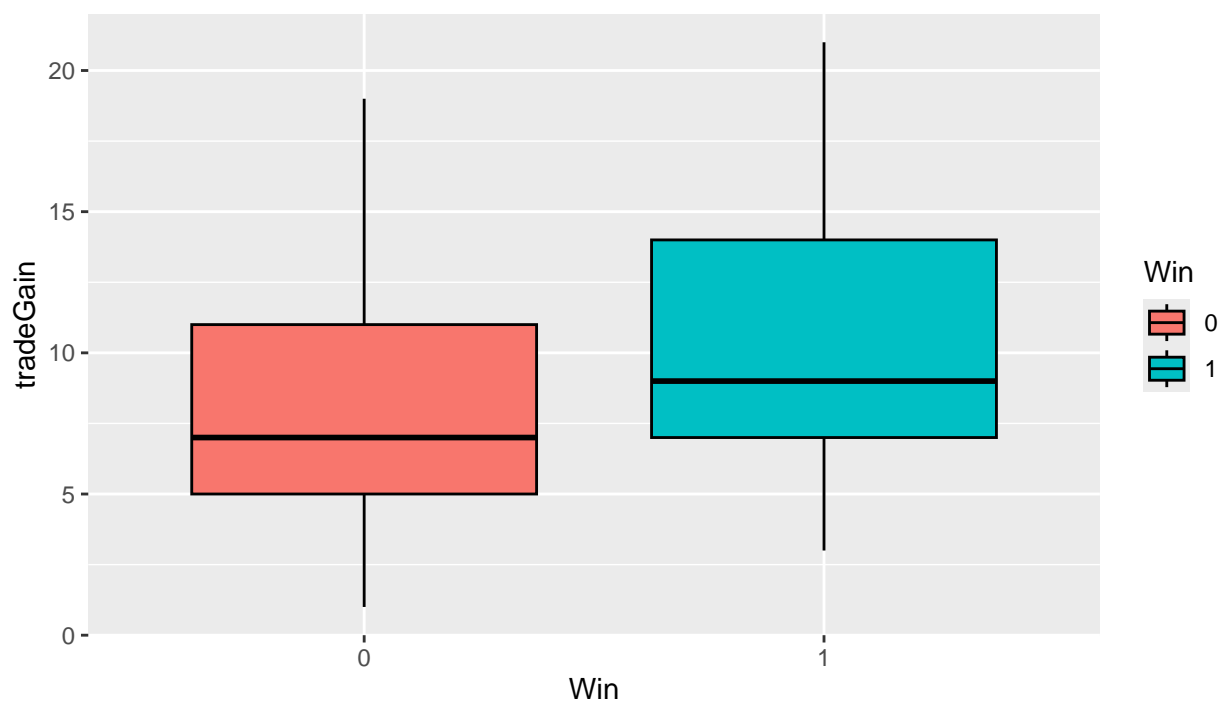
g5 <- ggplot(cardVars, aes(x = Win, y = tribute, fill = Win)) +
  geom_boxplot(color = "black") +
  labs(title = "Boxplot izbačenih kartica padom sedmice")
```

g1



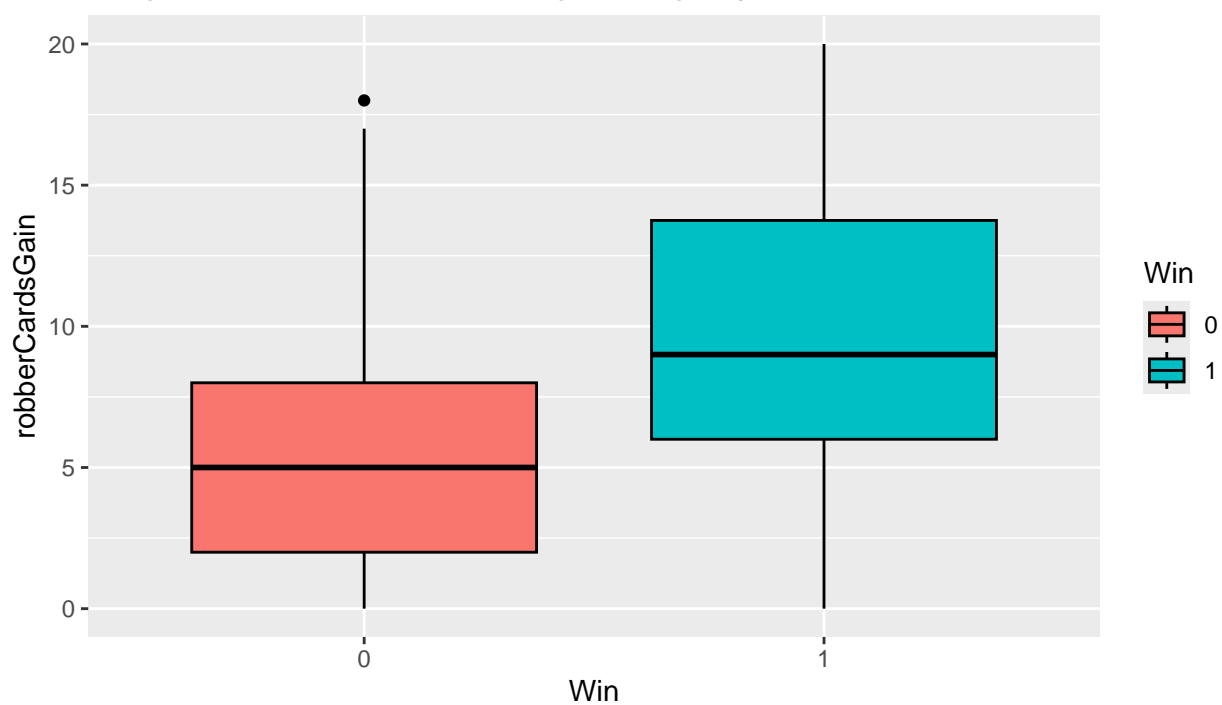
g2

Boxplot dobivenih kartica od razmjene

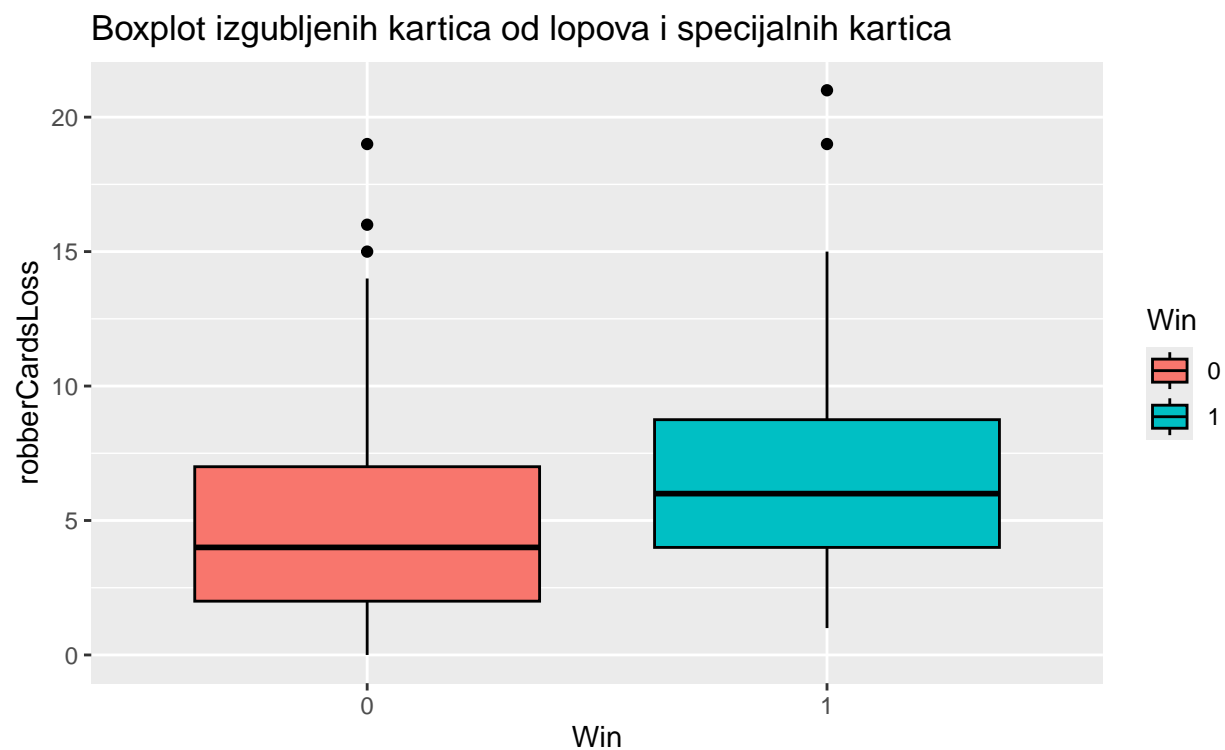


g3

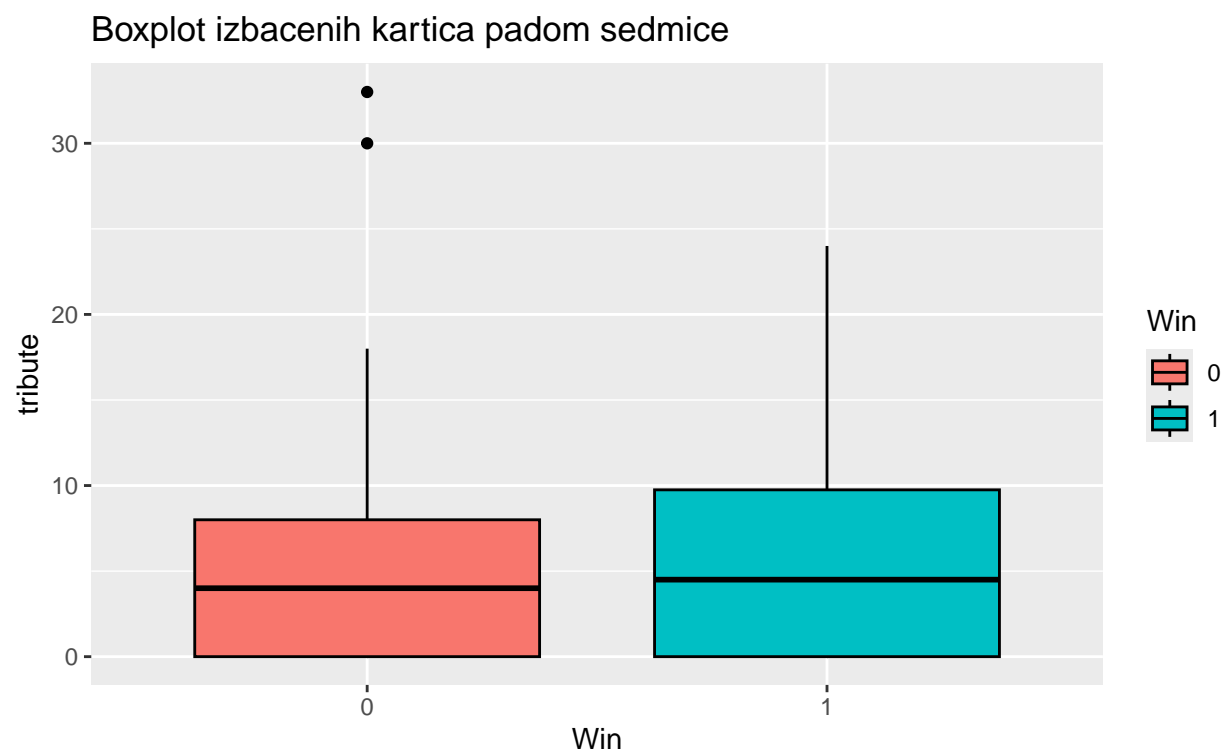
Boxplot dobivenih kartica od lopova i specijalnih kartica



g4



g5



Box-plotovi su pokazali nekoliko očekivanih stvari, koje želimo potvrditi modelom. Pobjednici u pravilu izvuku iz svojih naselja i gradova više resursa. Nešto više kartica pobjednici dobiju u razmjenama i očekivano imaju veći prinos od specijalnih kartica što im može svakako pomoći u pobjedi. Kako su igrači koji su pobijedili često i najnapredniji igrači tijekom partije, uglavnom su oni meta ostalim igrača tako da ne čudi da im se uzme više kartica nego svima ostalima.

```
model <- glm(Win ~ ., data = cardVars, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = Win ~ ., family = binomial, data = cardVars)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.91571    0.93559  -6.323 2.57e-10 ***
## production     0.08850    0.02339   3.783 0.000155 ***
## tradeGain    -0.05469    0.05555  -0.985 0.324836
## robberCardsGain 0.18912    0.04364   4.333 1.47e-05 ***
## robberCardsLoss -0.09646    0.06587  -1.464 0.143091
## tribute      -0.07670    0.03990  -1.923 0.054533 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 224.93  on 199  degrees of freedom
## Residual deviance: 170.50  on 194  degrees of freedom
## AIC: 182.5
##
## Number of Fisher Scoring iterations: 5
```

Na temelju logističkog modela, možemo zaključiti da je produkcija, odnosno broj kartica dobivenim iz naselja i gradova jako značajan i povećava vjerojatnosti za pobjedu. Isto tako vrijedi i za broj dobivenih kartica kada igrač *krade* od ostalih ili ima neku specijalnu karticu poput izgradnje cesta. Vidimo da *tradeGain* čak po ovom modelu ispada negativan, -0.05. Zato se postavlja pitanje jesu li ove varijable u nekoj interferenciji, odnosno, možemo li poboljšati ovaj model.

To ćemo pokušati iterativnom izgradnjom prediktivnog modela.

```
model2 <- MASS::stepAIC(model, direction = "backward")
```

```
## Start:  AIC=182.5
## Win ~ production + tradeGain + robberCardsGain + robberCardsLoss +
##      tribute
##
##              Df Deviance    AIC
## - tradeGain     1   171.48 181.48
## <none>              170.50 182.50
## - robberCardsLoss 1   172.70 182.70
## - tribute         1   174.50 184.50
## - production     1   187.46 197.46
## - robberCardsGain 1   191.76 201.76
##
```

```
## Step: AIC=181.48
## Win ~ production + robberCardsGain + robberCardsLoss + tribute
##
##           Df Deviance   AIC
## - robberCardsLoss 1  172.88 180.88
## <none>              171.48 181.48
## - tribute          1  174.65 182.65
## - production       1  191.64 199.64
## - robberCardsGain 1  191.88 199.88
##
## Step: AIC=180.88
## Win ~ production + robberCardsGain + tribute
##
##           Df Deviance   AIC
## <none>              172.88 180.88
## - tribute          1  175.31 181.31
## - robberCardsGain 1  192.21 198.21
## - production       1  194.57 200.57
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = Win ~ production + robberCardsGain + tribute, family = binomial,
##      data = cardVars)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.53257    0.86822  -6.372 1.86e-10 ***
## production     0.06272    0.01465   4.281 1.86e-05 ***
## robberCardsGain 0.17335    0.04131   4.196 2.71e-05 ***
## tribute       -0.05459    0.03600  -1.517  0.129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 224.93  on 199  degrees of freedom
## Residual deviance: 172.88  on 196  degrees of freedom
## AIC: 180.88
##
## Number of Fisher Scoring iterations: 5
```

```
exp(coef(model2))
```

```
##      (Intercept)      production robberCardsGain      tribute
##      0.003955809      1.064732239      1.189287573      0.946874198
```

Vidimo nešto drukčiju situaciju sada, izgleda da ovaj model sasvim dobro mogu opisati i tri varijable. Osim već gore navedene i statistički značajne dvije, imamo i *tribute*, koji govori koliko je kartica vraćeno banci zbog pada *sedmice*. Naravno da ta varijabla ima negativan utjecaj, ali po rezultatu modela i po box-plotu kojeg smo vidjeli ipak možemo reći da nije statistički značajan.

```
cardVars <- cardVars |> mutate(predicted_prob = predict(model2, type = "response"))

cardVars <- cardVars |> mutate(predicted_class = ifelse(predicted_prob > 0.5, 1, 0))

confMatrix <- confusionMatrix(factor(cardVars$predicted_class), factor(cardVars$Win))
confMatrix
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##           0 138  32
##           1  12  18
##
##           Accuracy : 0.78
##           95% CI : (0.7161, 0.8354)
##    No Information Rate : 0.75
##    P-Value [Acc > NIR] : 0.185238
##
##           Kappa : 0.3231
##
## Mcnemar's Test P-Value : 0.004179
##
##           Sensitivity : 0.9200
##           Specificity : 0.3600
##           Pos Pred Value : 0.8118
##           Neg Pred Value : 0.6000
##           Prevalence : 0.7500
##           Detection Rate : 0.6900
##    Detection Prevalence : 0.8500
##           Balanced Accuracy : 0.6400
##
##           'Positive' Class : 0
##
```

Confusion matrix prikazuje podatke solidno, ali ipak neke stvari nam malo smetaju, npr. *specificity*. Imamo puno problema s *false negative* rezultatima, teško prepoznavamo ispravno pobjednike. Kako bismo doskočili tom problemu, pokušajmo pronaći bolji *threshold*.

```
roc_obj <- roc(cardVars$Win, cardVars$predicted_prob)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

best_threshold <- coords(roc_obj, "best", ret = "threshold")
print(best_threshold)

## threshold
## 1 0.2287972
```

```
cardVars <- cardVars |> mutate(predicted_class = ifelse(predicted_prob > 0.228, 1, 0))

confMatrix2 <- confusionMatrix(factor(cardVars$predicted_class), factor(cardVars$Win))
confMatrix2
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    0    1
##              0 115    8
##              1   35   42
##
##              Accuracy : 0.785
##              95% CI : (0.7215, 0.8398)
##      No Information Rate : 0.75
##      P-Value [Acc > NIR] : 0.1438
##
##              Kappa : 0.5141
##
##  McNemar's Test P-Value : 7.341e-05
##
##              Sensitivity : 0.7667
##              Specificity : 0.8400
##              Pos Pred Value : 0.9350
##              Neg Pred Value : 0.5455
##              Prevalence : 0.7500
##              Detection Rate : 0.5750
##      Detection Prevalence : 0.6150
##              Balanced Accuracy : 0.8033
##
##              'Positive' Class : 0
##
```

Konačno, analizom najboljeg *thresholda* pronašli smo novi broj i ponovno napravili confusion matrix. Sada imamo nešto više *false positive* vrijednosti, ali općenito ovaj model puno je bolji od prvog, isto tako vidimo da je puno značajnih vrijednosti oko 0.8 što nam govori da imamo dosta kvalitetan model.

Strategija naseljavanja luke na početku partije

Slika Catan ploče iz [uvoda](#) prikazuje razmještaj kopnenih i morskih heksagona. Na nekim granicama kopnenih i morskih heksagona, nalaze se luke. Moguće je s bankom razmjenjivati kartice resursa ako na luci postoji sagrađeno naselje ili grad. Postoji po jedna luka za svaki od pet resursa, a na njima se za dvije kartice tog resursa iz banke može dobiti kartica po volji. Osim navedenih pet luka, postoje četiri univerzalne luke gdje se za bilo koja tri ista resursa može iz banke dobiti kartica po volji.

Odabir prvog naselja na lukama

Je li izbor univerzalne luke kao prvog naselja opravdan? Koliko se puta u 50 zabilježenih partija univerzalna luka odabrala kao destinacija gradnje naselja? Koji su susjedni heksagoni pri takvim odabirima? Odgovori na ova pitanja slijede u nastavku.

```

first_turn_universal_harbor <- my_data[, 16:27] %>% filter(
  str_detect(settlement1_01_resource, "^3") |
  str_detect(settlement1_02_resource, "^3") |
  str_detect(settlement1_03_resource, "^3"))

numbers <- first_turn_universal_harbor %>% select(settlement1_01_number,
  settlement1_02_number, settlement1_03_number) %>%
  unlist %>% as.vector %>% '['(. != "0")

resources <- first_turn_universal_harbor %>% select(settlement1_01_resource,
  settlement1_02_resource, settlement1_03_resource) %>%
  unlist %>% as.vector %>% '['(. != "3G")

resources <- dictionary$names[match(resources, dictionary$code)]

cat("Broj izgrađenih prvih naselja na univerzalnim lukama:",
  nrow(first_turn_universal_harbor),
  "\nBrojevi na susjednim heksagonima univerzalne luke:",
  numbers,
  "\nResursi na susjednim heksagonima univerzalne luke:",
  resources, sep = " ")

```

```

## Broj izgrađenih prvih naselja na univerzalnim lukama: 2
## Brojevi na susjednim heksagonima univerzalne luke: 8 8 4 10
## Resursi na susjednim heksagonima univerzalne luke: Glina Glina Drvo Ovca

```

Dakle, izbor univerzalne luke kao prvog naselja nije popularan. Samo u dva slučaja od 200, igrač je odabrao univerzalnu luku kao svoje prvo naselje. Tada su susjedni resursi bili glina dva puta te drvo i ovca jedanput. Ovdje valja napomenuti da su glina i drvo ključni za razvoj cesta što je korisno u početku igre. Od susjednih brojeva, ističe se broj 8 koji je uz broj 6 vjerojatnosno najbolja opcija za dohvaćanje resursa.

```

first_turn_special_harbor <- my_data[, 16:27] %>% filter(
  str_detect(settlement1_01_resource, "^2") |
  str_detect(settlement1_02_resource, "^2") |
  str_detect(settlement1_03_resource, "^2"))

numbers <- first_turn_special_harbor %>% select(settlement1_01_number,
  settlement1_02_number, settlement1_03_number) %>%
  unlist %>% as.vector %>% '['(. != "0")

resources <- first_turn_special_harbor %>% select(settlement1_01_resource,
  settlement1_02_resource, settlement1_03_resource) %>%
  unlist %>% as.vector %>% '['(!str_detect(., "^2"))
resources <- dictionary$names[match(resources, dictionary$code)]

cat("Broj izgrađenih prvih naselja na specijalnim lukama:",
  nrow(first_turn_special_harbor),
  "\nBrojevi na susjednim heksagonima specijalne luke:",
  numbers,
  "\nResursi na susjednim heksagonima specijalne luke:",
  resources[1:7],
  "\n",
  resources[8:length(resources)], sep = " ")

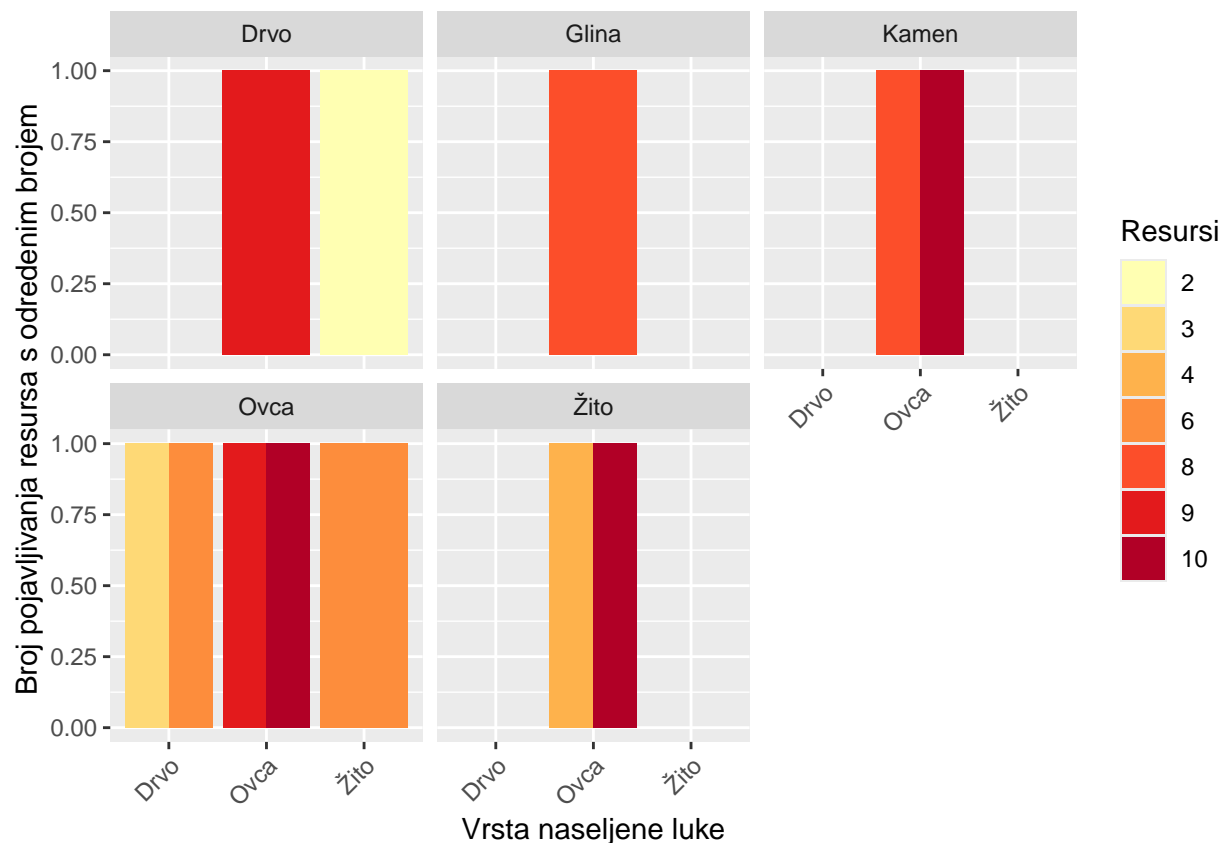
```

```
## Broj izgrađenih prvih naselja na specijalnim lukama: 6
## Brojevi na susjednim heksagonima specijalne luke: 8 9 6 8 6 9 4 10 2 10 3 10
## Resursi na susjednim heksagonima specijalne luke: Glina Drvo Ovca Kamen Ovca Ovca Žito
## Ovca Drvo Žito Ovca Kamen
```

Specijalnih luka kao prvih naselja ima nešto više nego univerzalnih luka, a vjerojatno su primarno odabirane zbog povoljnog susjednog broja poput 6, 8 ili 9. Budući da je najčešći susjedni resurs u ovom slučaju ovca, koja je jedna od najrjeđe korištenih u igri, treba se zapitati je li naseljena luka upravo specijalizirana za razmjenu ovaca.

```
first_neighbour <- first_turn_special_harbor[, c(1, 2, 6)]
names(first_neighbour) <- c("settlement_number", "settlement_resource", "settlement_harbor")
second_neighbour <- first_turn_special_harbor[, c(3, 4, 6)]
names(second_neighbour) <- c("settlement_number", "settlement_resource", "settlement_harbor")
data_to_plot <- rbind(first_neighbour, second_neighbour)
data_to_plot$settlement_number <- factor(data_to_plot$settlement_number, levels = 2:12)
data_to_plot$settlement_resource <- as.factor(dictionary$names[
  match(data_to_plot$settlement_resource, dictionary$code)])
data_to_plot$settlement_harbor <- as.factor(dictionary$names[
  match(data_to_plot$settlement_harbor, paste0("2", dictionary$code))])

ggplot(data_to_plot, aes(x = settlement_harbor, fill = settlement_number)) +
  geom_bar(position = "dodge") +
  facet_wrap(~ settlement_resource) +
  scale_fill_brewer(palette = "YlOrRd") +
  labs(x = "Vrsta naseljene luke", y = "Broj pojavljivanja resursa s određenim brojem",
       fill = "Resursi") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Specijalizirana luka za neki resurs samo u 2 navrata ima susjedni heksagon koji je predodređen za taj resurs i to s brojevima 9 i 10. Dakle, zaista je riječ o rijetkoj strategiji, no preostaje još provjeriti je li odabir naselja na luci u ovih 50 partija ikada doveo do pobjede.

```
cat("Broj pobjeda pri odabiru prvog naselja na luci:", my_data %>% filter((
  str_detect(settlement1_01_resource, "^3") | str_detect(settlement1_02_resource, "^3") |
  str_detect(settlement1_03_resource, "^3") | str_detect(settlement1_01_resource, "^2") |
  str_detect(settlement1_02_resource, "^2") | str_detect(settlement1_03_resource, "^2")) &
  points >= 10) %>% nrow, sep = " ")
```

```
## Broj pobjeda pri odabiru prvog naselja na luci: 2
```

Odabir drugog naselja na lukama

Odabir drugog naselja na luci je učestalija pojava od odabira prvog naselja na luci. Tada igrač ima već odabrano naselje koje graniči s određenim heksagonima s resursima koje može dobiti kroz bacanje kockica. Upravo ti resursi mogu biti misao vodilja za odabir neke specijalizirane luke kao drugog naselja. Također je moguće da prvi igrač, koji odabire svoje drugo naselje zadnji, ne može staviti naselje kod resursa kojemu još nema pristup što ga potiče na gradnju naselja kod luke gdje samostalno može dolaziti do takvih resursa pomoću razmjene.

```
second_turn_harbors <- my_data %>% filter(
  str_detect(settlement2_01_resource, "^3") |
  str_detect(settlement2_02_resource, "^3") |
  str_detect(settlement2_03_resource, "^3") |
```



```

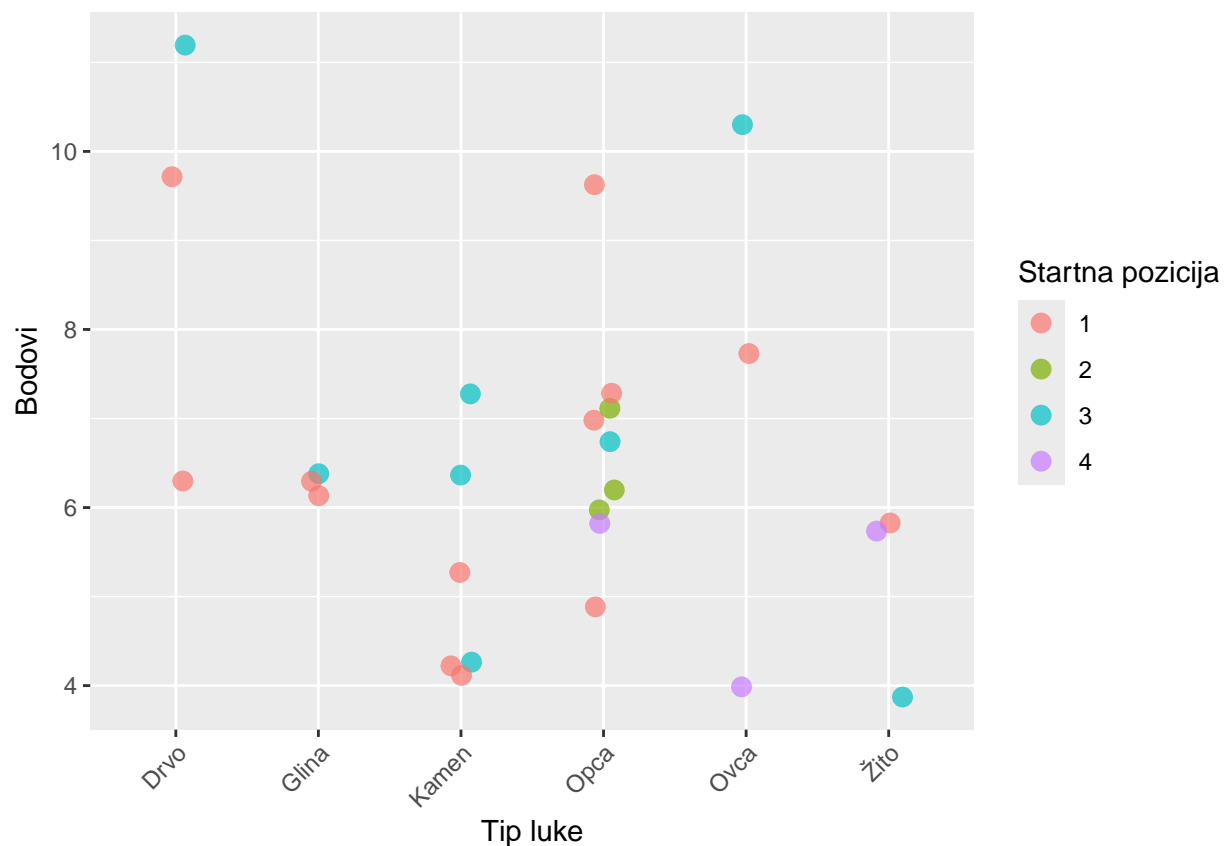
str_detect(settlement2_01_resource, "^2") |
str_detect(settlement2_02_resource, "^2") |
str_detect(settlement2_03_resource, "^2"))

cat("Broj izgrađenih drugih naselja na lukama:", nrow(second_turn_harbors), sep = " ")

## Broj izgrađenih drugih naselja na lukama: 27

ggplot(second_turn_harbors, aes(x = settlement2_03_resource, y = points, color = player)) +
  geom_jitter(width = 0.1, size = 3, alpha = 0.7) +
  labs(x = "Tip luke", y = "Bodovi", color = "Startna pozicija") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



Za gornji graf korišten je `geom_jitter` zbog preglednosti jer se točke okupljaju oko identičnih iznosa broja bodova. Prvi igrač ističe se po najčešćem odabiru luke kao svog drugog naselja što ide u korist pretpostavki iz uvodnog dijela potpoglavlja. Ukupan broj odabira je 27, što je veće od broja odabira luka kao prvog naselja, no ne pridonosi nužno pobjedi jer je igrač pobijedio samo 4 puta kada je stavio luku kao svoje drugo naselje.

Zaključak

Društvena igra Catan godinama slovi za jednu od najboljih i najpopularnijih društvenih igara na svijetu. Jednostavnost igre zbog faktora sreće u kockicama, ali i dodatnog taktičkog aspekta dovela je igru na sami vrh, a isto tako i zaintrigirala mnoge za dublju analizu.

Na temelju podataka koje smo imali mogli smo doći do solidnog broja zaključaka, ali isto tako potrebno je reći da je riječ o 50 igara, koje je igrala određena skupina ljudi i pitanje je same kvalitete njih, jesu li korištene optimalne strategije i slično. Nastavno, ne postoji dovoljno podataka o strategijama s korištenjem kartica napretka.

Očekivano, bacanja kockice potpuno se prilagođavaju normalnoj razdiobi zbog zakona velikih brojeva. Kod najbolje startne pozicije, zaključak je da je vjerojatno najbolje postavljati drugi, a sigurno prije drugi nego prvi. To nije skroz iznenađujuće, jer jednostavno postoji toliko brojeva i kombinacija, da je gotovo nemoguće složiti mapu gdje bi samo jedno mjesto bilo značajno bolje od drugih. A kako drugi igrač postavlja u drugom krugu zapravo sedmo po redu naselje, gdje su kalkulacije nešto veće, za očekivati je da prvom igraču, koji postavlja osmo naselje ne ostane toliko dobrih mogućnosti.

U radu je već objašnjena strategija i potencijalna ideja o tome koji resursi bi bili najznačajniji. No, detaljnim testiranjima nismo uspjeli doći do zaključka da su igračima određeni resursi privlačniji u početku.

Provjerom o tome kako razmjena resursa te dobivanje/gubljenje istih utječe na pobjedu. Uspjeli smo pronaći dobar model, koji nam govori da količina kartica osvojena od naselja i gradova igra jako bitnu ulogu, ali da možda i presudno može biti korištenje specijalnih kartica ili uzimanje kartica drugim igračima premještanjem lopova, što je isto i više nego razumljivo jer direktno značajno preokreće situaciju u korist igrača nauštrb drugih.

Postavljanje naselja uz luku automatski znači da je jedan heksagon more, odnosno ne može pridonijeti resursima, već učinkovitijoj razmjeni s bankom. Ipak, dvije stvari su iz naše analize potpuno očite. Prvo je vidljivo da se događa rijetko. Jednostavno, u prva dva postavljanja, igrači žele pokriti sve resurse (ima ih 5), a maksimalno mogu izvući 6 polja s kojima graniče ako ne odabiru luke. Uz faktor brojeva i vjerojatnosti da padnu, teško je očekivati da bi s lukom sve lijepo pokrili. Zato se u početnom postavljanju želi težiti prema sredini, a onda kasnije tijekom partije prema obali i boljim uvjetima za trgovinu. Nadalje, vidljivo je da je najčešće igrač koji i postavi naselje na luci prvi igrač jer on u drugom krugu postavlja osmi pa mu je i takvo mjesto vjerojatno u tom slučaju bilo najbolje.

Za kraj, valja napomenuti da bi ovakve analize sigurno mogle koristiti pogotovo novijim igračima, ali isto tako i nama bi za dodatnu analizu puno bolje došli veći skupovi te nekakva informacija o kvaliteti samih igrača jer zaključivanje određenih strategija, nastalih na suboptimalnim podacima ne bi bilo previše korisno. Dakako, *Catan* sa svojim dodatnim proširenjima igre unosi puno veću kompleksnost i još više umanjuje faktor sreće i tek bi za takve podatke bilo iznimno korisno doći do određenih zaključaka.