Analiza balansu stron w grze Starcraft 2 na podstawie zarobków najlepszych 800 graczy do roku 2022.

Krótkie wprowadzenie:

Gra Starcraft 2 jest strategią czasu rzeczywistego (RTS) gdzie gracze mogą wcielić się w 3 niesymetryczne gatunki/rasy nazywane dalej stronami. Gracze mają do wyboru: Zergi, Protosi, Terranie. Poniższa analiza skupi się na sprawdzeniu czy gra jest należycie zbalansowana, tj. czy któraś ze stron nie ma przewagi nad innymi. Jako miarę sukcesu przyjmie się zarobki 800 najlepszych graczy, zdobywane na turniejach e-sprtowych na przestrzeni 12 lat istnienia gry (lata 2010-2022).

Przyjmę 3 hipotezy zerowe:

1) Zarobki graczy wybierających Zergi różniły się od zarobków graczy grających Protosami
2) Zarobki graczy wybierających Zergi różniły się od zarobków graczy grających Terranami
3) Zarobki graczy wybierających Protosów różniły się od zarobków graczy grających Terranami

Każdą hipotezę będę testował oddzielnie.

```
#Ładowanie potrzebnych bibliotek
library("RSQLite")
if (!require(RColorBrewer)){
  install.packages("RColorBrewer")
  library(RColorBrewer)
}
```

```
## Ładowanie wymaganego pakietu: RColorBrewer
```

Na początku ładuję do R przygotowaną wcześniej bazę danych z zarobkami graczy:

```
db<-dbConnect(SQLite(), dbname="starcraft.db")
dbListTables(db)
```

```
## [1] "Players"
```

```
dbListFields(db, "Players")
```

```
## [1] "nick"          "race"          "country"       "name"
## [5] "birthday"      "team"          "totalEarnings"
```

Najistotniejsze będą kolumny “race” i “totalEarnings”. Mając wczytane dane możemy przyjrzeć się jak rozkładają się preferencje graczy i zarobki poszczególnych grup:

```
zerg_count<-dbGetQuery(db, "select count(nick) from
  Players where race = 'Zerg' ")[1,1]
protoss_count<-dbGetQuery(db, "select count(nick) from
  Players where race = 'Protoss' ")[1,1]
terran_count<-dbGetQuery(db, "select count(nick) from
  Players where race = 'Terran' ")[1,1]
```

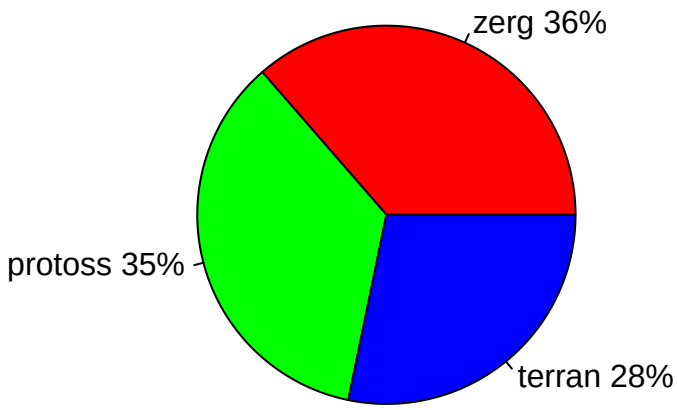
```
zerg_earnings<-dbGetQuery(db, "select sum(totalEarnings) from
  Players where race = 'Zerg' ")[1,1]
protoss_earnings<-dbGetQuery(db, "select sum(totalEarnings) from
  Players where race = 'Protoss' ")[1,1]
terran_earnings<-dbGetQuery(db, "select sum(totalEarnings) from
  Players where race = 'Terran' ")[1,1]
```

```
earnings<-c(zerg_earnings, protoss_earnings, terran_earnings)
counts<-c(zerg_count, protoss_count, terran_count)
# create matrix with 3 columns and 3 rows
dane= matrix(c(counts, earnings), ncol=2, byrow=FALSE)
 # specify the column names and row names of matrix
colnames(dane) = c('count','totalEarnings')
rownames(dane) <- c('zerg','protoss','terran')
 # assign to table
wyniki=as.table(dane)
 # display
wyniki
```

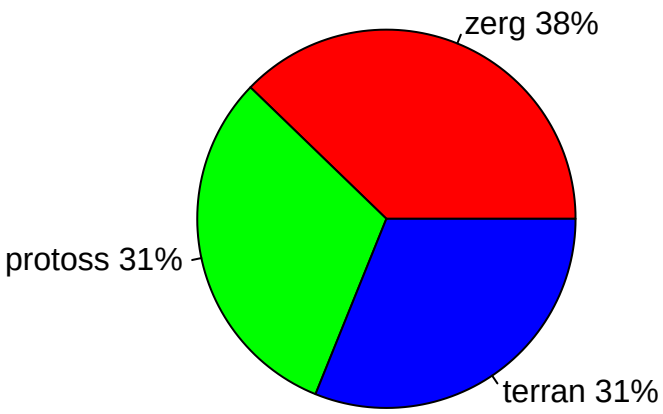
```
##            count totalEarnings
## zerg         290      14513277
## protoss      282      11957412
## terran       225      11955930
```

Co można przedstawić na wykresach kołowych:

```
#wykres kołowy - liczebność
pct <- paste(round(wyniki[1:3,1]/sum(wyniki[1:3,1])*100), "%", sep="")
lbls <- paste(rownames(dane), pct, sep=" ")
pie(wyniki[1:3, 1],labels = lbls, col=rainbow(length(lbls)),
   main="Rasy wybierane przez graczy")
```




```
#wykres kołowy - zarobki
pct <- paste(round(wyniki[1:3,2]/sum(wyniki[1:3,2])*100), "%", sep="")
lbls <- paste(rownames(dane), pct, sep=" ")
pie(wyniki[1:3, 2],labels = lbls, col=rainbow(length(lbls)),
   main="Zarobki graczy z podziałem na rasy")
```




```
#zdefiniujmy funkcję zwracającą modę z wektora
get_mode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Możemy tekże przyjrzeć się wskaźnikom położenia (odpowiednio: średnia, mediana, moda i dodatkowo suma) dla całej populacji:

```
#wskaźniki położenia w populacji
for (earnings_var in dbGetQuery(db, "select totalEarnings from
  Players")){}
print(paste('średnia: ', mean(earnings_var)))
```

```
## [1] "średnia:  48044.58875"
```

```
print(paste('mediana: ', median(earnings_var)))
```

```
## [1] "mediana:  6114"
```

```
print(paste('moda: ', get_mode(earnings_var)))
```

```
## [1] "moda:  1199"
```

```
print(paste('suma: ', sum(earnings_var)))
```

```
## [1] "suma:  38435671"
```

```
#średni wynik z SQL Query
#avg_earnings<-dbGetQuery(db, "select avg(totalEarnings) from Players")
```

oraz dla poszczególnch grup graczy:

```
#wskaźniki położenia dla grupy:
  #zerg:
for (zerg_earnings_var in dbGetQuery(db, "select totalEarnings from
  Players where race = 'Zerg'")){}
print('Zergi:')
```

```
## [1] "Zergi:"
```

```
print(paste('średnia: ', mean(zerg_earnings_var)))
```

```
## [1] "średnia:  50045.7827586207"
```

```
print(paste('mediana: ', median(zerg_earnings_var)))
```

```
## [1] "mediana:  5794.5"
```

```
print(paste('moda: ', get_mode(zerg_earnings_var)))
```

```
## [1] "moda:  2000"
```

```
print(paste('suma: ', sum(zerg_earnings_var)))
```

```
## [1] "suma:  14513277"
```

```
#wskaźniki położenia dla grupy:
  #protoss:
for (protoss_earnings_var in dbGetQuery(db, "select totalEarnings from
  Players where race = 'Protoss'")){}
print('Protosi:')
```

```
## [1] "Protosi:"
```

```
print(paste('średnia: ', mean(protoss_earnings_var)))
```

```
## [1] "średnia:  42402.170212766"
```

```
print(paste('mediana: ', median(protoss_earnings_var)))
```

```
## [1] "mediana:  6114"
```

```
print(paste('moda: ', get_mode(protoss_earnings_var)))
```

```
## [1] "moda:  1199"
```

```
print(paste('suma: ', sum(protoss_earnings_var)))
```

```
## [1] "suma:  11957412"
```

```
#wskaźniki położenia dla grupy:
  #terran:
for (terran_earnings_var in dbGetQuery(db, "select totalEarnings from
  Players where race = 'Terran'")){}
print('Terranie:')
```

```
## [1] "Terranie:"
```

```
print(paste('średnia: ', mean(terran_earnings_var)))
```

```
## [1] "średnia:  53137.4666666667"
```

```
print(paste('mediana: ', median(terran_earnings_var)))
```

```
## [1] "mediana:  6993"
```

```
print(paste('moda: ', get_mode(terran_earnings_var)))
```

```
## [1] "moda:  1864"
```

```
print(paste('suma: ', sum(terran_earnings_var)))
```

```
## [1] "suma:  11955930"
```

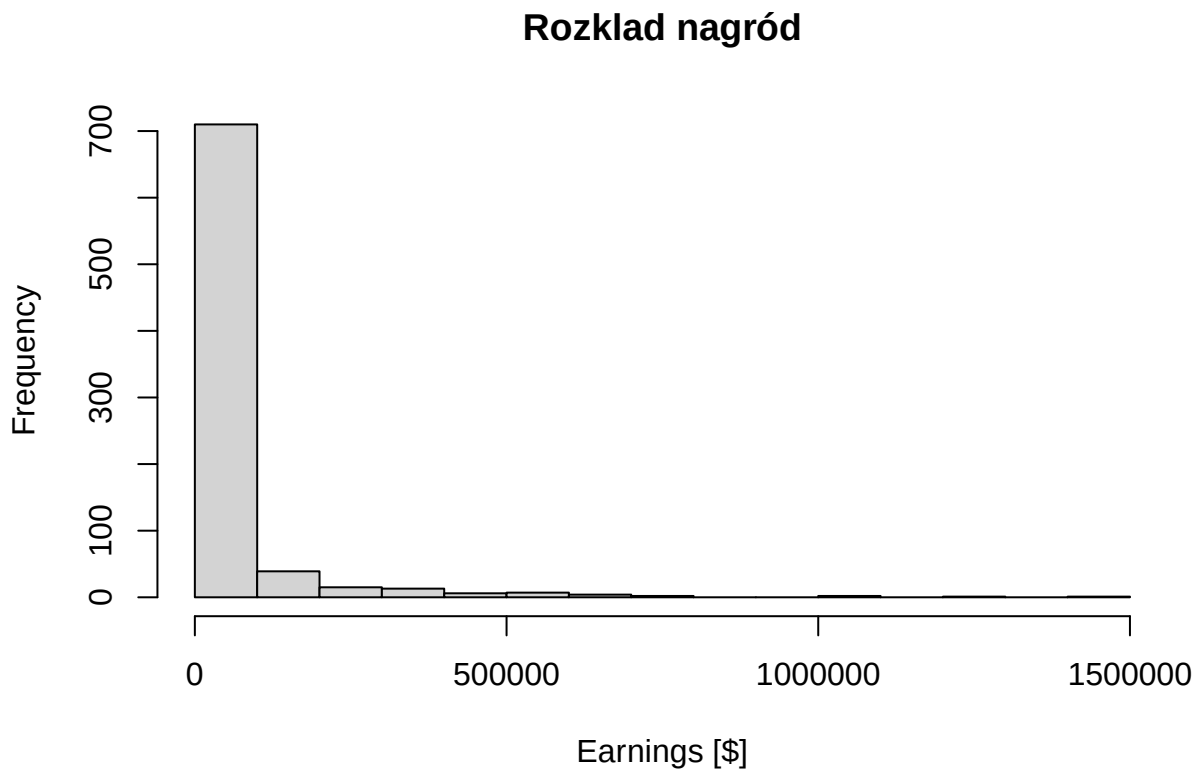
Widzimy, że średnie zarobki najbardziej różnią się dla grup graczy wybierających Protosówi Terran, dlatego dla uproszczenia przeprowadzę dalszą analizę tylko tych dwóch grup.

Po zgrupowaniu poszczególnych zawodników ze względu na zarobki możemy zauważyć (co widaćna poniższym histogramie), że zdecydowana większość graczy (ok. 700 z 800) nie zarobiła więcej niż 100'000 dolarów w całej swojej karierze. Pokazuje to dwie rzeczy:

Po pierwsze, istnieje garstka topowych graczy o umiejętnościach poza zasięgiem przeciętnego użytkownika (zgodnie z zasadą "Easy to lern, hard to master", jak często mówi się w kontekście tej gry).

Po drugie, być może nie był to najszczęśliwszy dobór danych źródłowych do analizy. Ale przez brak czasu, zdecydowałem się kontynuować pracę.

```r
hist(earnings_var, xlab = "Earnings [$]", main = "Rozkład nagród")
```



Porównajmy zatem zarobki graczy wybierających Protosów i Terran: Do porównania rozkładów użyłem testu t-studenta:

```r
#analiza protoss vs terran

for (races in dbGetQuery(db, "select race from
  Players where race = 'Protoss' or race ='Terran' ")){}
for (PvT_earnings_var in dbGetQuery(db, "select totalEarnings from
  Players where race = 'Protoss' or race ='Terran' ")){}

data_test <- data.frame(race=races,data=PvT_earnings_var)
#data_test
print('Porównanie średnich zarobków: ')
```

```
## [1] "Porównanie średnich zarobków: "
```

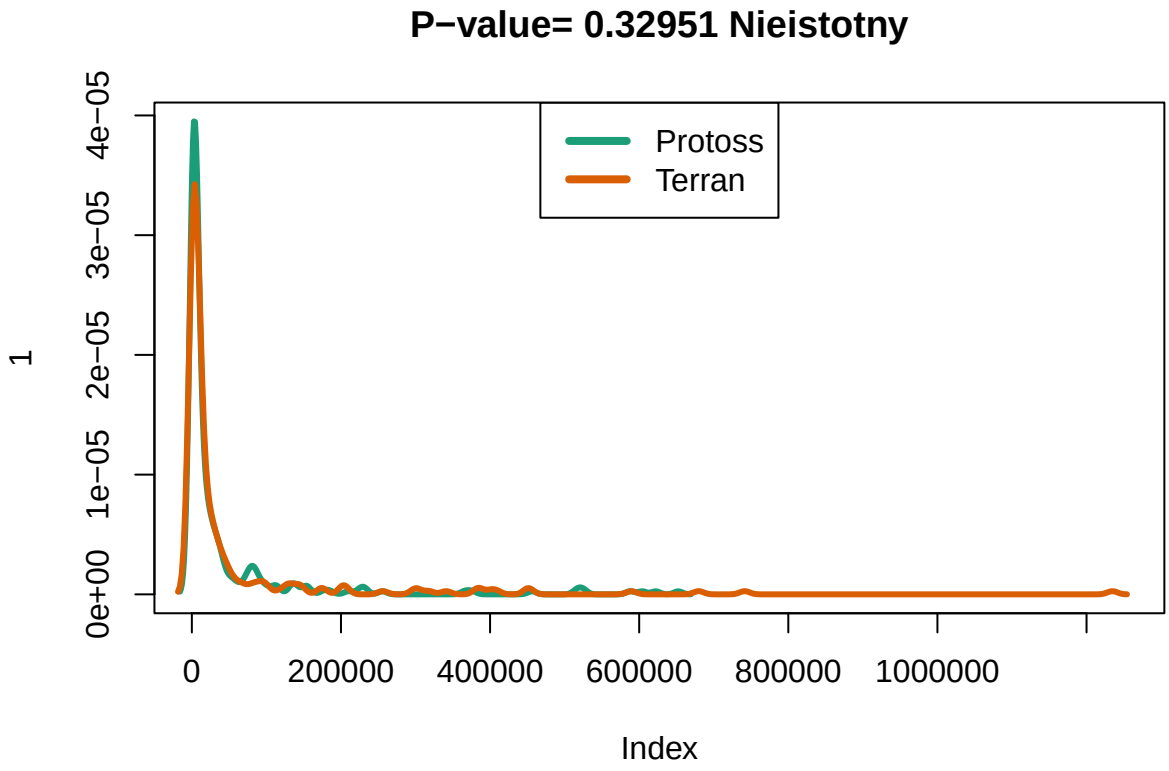
```r
sapply(split(data_test$data,list(data_test$race)),mean)
```

```
##  Protoss   Terran
## 42402.17 53137.47
```

Poniższy kod definiuje funkcję rysującą graf przedstawiający rozkłady obu grup oraz dokonujący jednoczesnego porównania obu grup testem studenta:

```
Analiza <- function(data_test,cols){
pvalue <- t.test(data_test$data ~ data_test$race)$p.value
istota <- c("Nieistotny","Istotny")[as.numeric(pvalue<=0.05)+1]
density_val <- lapply(split(data_test$data,data_test$race),density)
plot(1,xlim=c(0,max(sapply(density_val,function(x)
  max(x$x)))),ylim=c(0,max(sapply(density_val,function(x) max(x$y)))))
lapply(1:length(density_val),function(x)
  lines(density_val[[x]],col=cols[x],lwd=3))
legend("top",legend=names(density_val),col=cols,lwd=4)
title(paste("P-value=",format.pval(pvalue),istota))
}
```

```
par(mfrow=c(1,1))
Analiza(data_test,cols=brewer.pal(8,"Dark2"))
```



Jak widzimy różnice między obiema grupami są nieistotne statystycznie, co prowadzi do wniosku, że przynajmniej w przypadku meczy typu Protoss vs Terran gra jest nalezycie zbalansowana.

```
#zamknij bazę danych
dbDisconnect(db)
```