vgsales

Case study analiza prodaje videoigara

Mato Gudelj, Ivan Gadža, Lovro Glogar, Renato Jurišić

Video games sales data

```
# ucitavanje podataka
vgsales = read.csv("vgsales.csv")
vgsales$Year = as.integer(vgsales$Year)
```

Grupiramo igre po imenu, izdavaču i žanru jer su to iste igre i ne želimo ih razlikovati. Inicijalno se razlikuju po platformi.

```
# sales sumiramo
vgsales.grouped = aggregate( vgsales[c(7, 8, 9, 10, 11)],
                             vgsales[c("Name", "Publisher", "Genre")],
                             FUN = sum)

# za godinu uzimamo srednju vrijednost
vgsales.grouped = merge( vgsales.grouped,
                         aggregate( vgsales["Year"],
                         vgsales[c("Name", "Publisher", "Genre")],
                         FUN = mean),
                         by = c( "Name", "Publisher", "Genre"))
```

```
summary( vgsales.grouped)
```

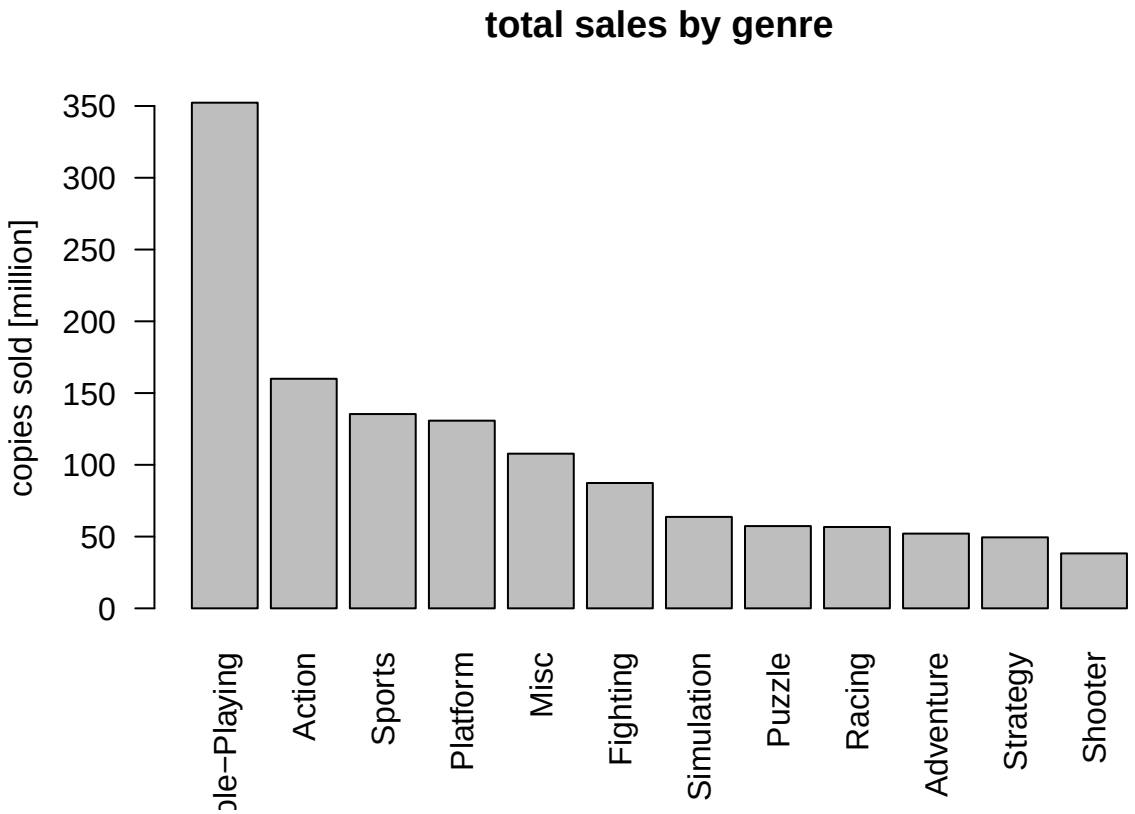
```
##      Name            Publisher            Genre             NA_Sales
##  Length:11920       Length:11920       Length:11920       Min.   : 0.0000
##  Class :character   Class :character   Class :character   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Mode  :character   Median : 0.0700
##                                                           Mean   : 0.3685
##                                                           3rd Qu.: 0.2900
##                                                           Max.   :41.4900
##
##     EU_Sales          JP_Sales        Other_Sales        Global_Sales
##  Min.   : 0.0000   Min.   : 0.0000   Min.   : 0.00000   Min.   : 0.0100
##  1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.00000   1st Qu.: 0.0600
##  Median : 0.0200   Median : 0.0000   Median : 0.01000   Median : 0.1900
##  Mean   : 0.2042   Mean   : 0.1083   Mean   : 0.06692   Mean   : 0.7484
##  3rd Qu.: 0.1200   3rd Qu.: 0.0700   3rd Qu.: 0.04000   3rd Qu.: 0.5900
##  Max.   :29.0200   Max.   :10.2200   Max.   :10.72000   Max.   :82.7400
##
##       Year
##  Min.   :1980
##  1st Qu.:2002
##  Median :2007
```

```
##  Mean   :2006
##  3rd Qu.:2010
##  Max.   :2020
##  NA's   :235
```

Pitanje 1: Jesu li u Japanu RPG igre značajno prodavanije od FPS igara?

Grupirajmo igre prema žanru, izračunavamo ukupnu prodaju te ih silazno sortiramo. RPG igre ostvarile su najviše, a FPS igre najmanje prodaja, što ide u korist istraživačkoj hipotezi.

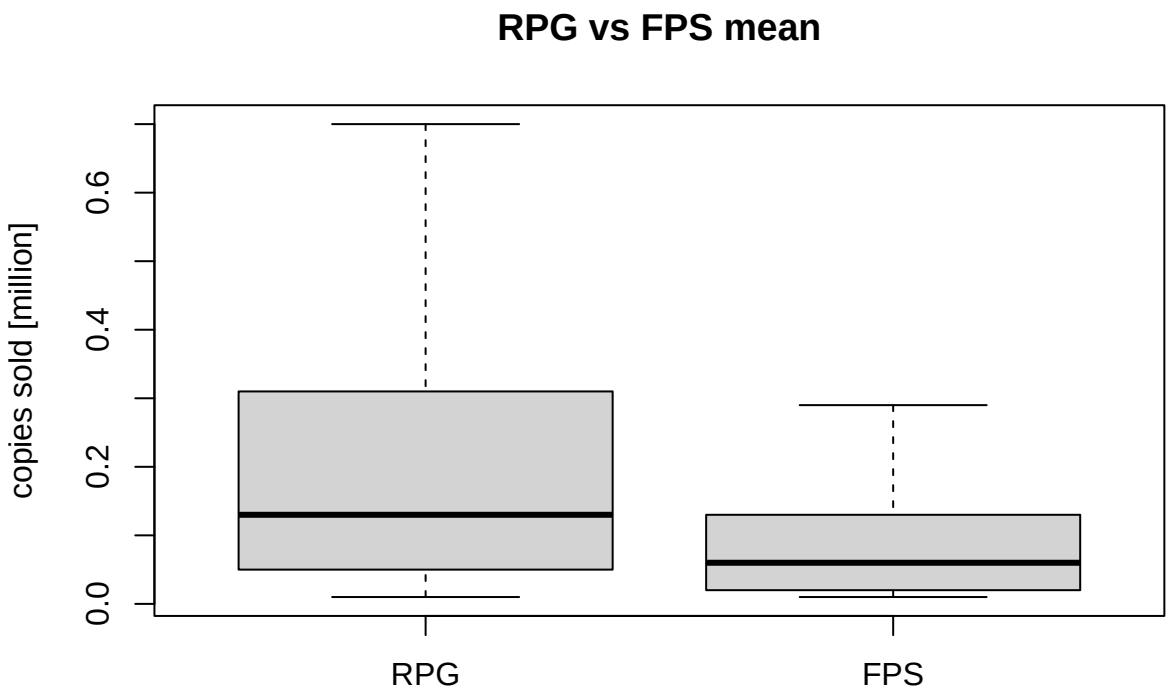
```
total_sales_genre = aggregate( vgsales.grouped["JP_Sales"], vgsales.grouped["Genre"], sum)
total_sales_genre_in_order = order( total_sales_genre["JP_Sales"], decreasing=TRUE)

barplot( total_sales_genre$JP_Sales[ total_sales_genre_in_order],
         main="total sales by genre",
         ylab="copies sold [million]",
         names.arg = total_sales_genre$Genre[ total_sales_genre_in_order],
         las=2)
```



Srednja vrijednost prodaja također je veća kod RPG igara.

```
jp_rpg_sales = vgsales.grouped[ vgsales.grouped["Genre"]=="Role-Playing"
                                & vgsales.grouped["JP_Sales"] != 0,]$JP_Sales
jp_fps_sales = vgsales.grouped[ vgsales.grouped["Genre"]=="Shooter"
                                & vgsales.grouped["JP_Sales"] != 0,]$JP_Sales
```

```
boxplot( jp_rpg_sales, jp_fps_sales,
         main="RPG vs FPS mean",
         ylab="copies sold [million]",
         names=c("RPG", "FPS"),
         outline=FALSE)
```



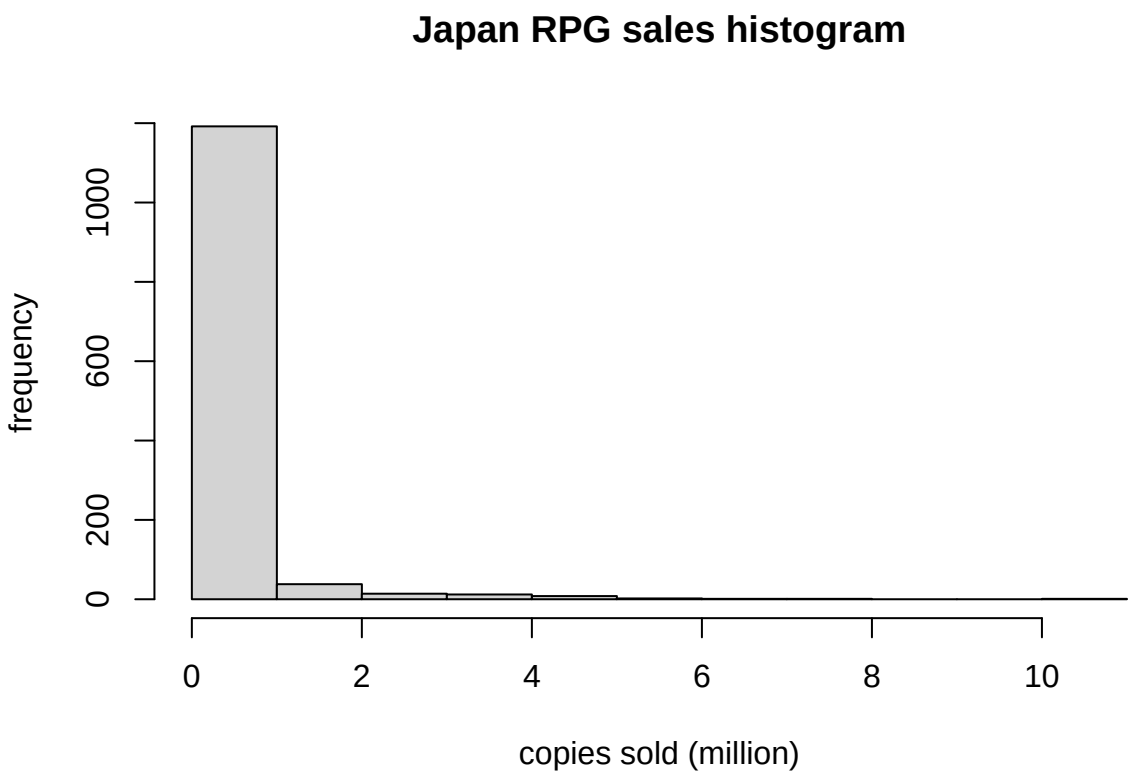
Motivirani prethodnim grafom formuliramo sljedeću statističku hipotezu.

$$\textbf{H0: mean(RPG sales)} = \textbf{mean(FPS sales)}$$

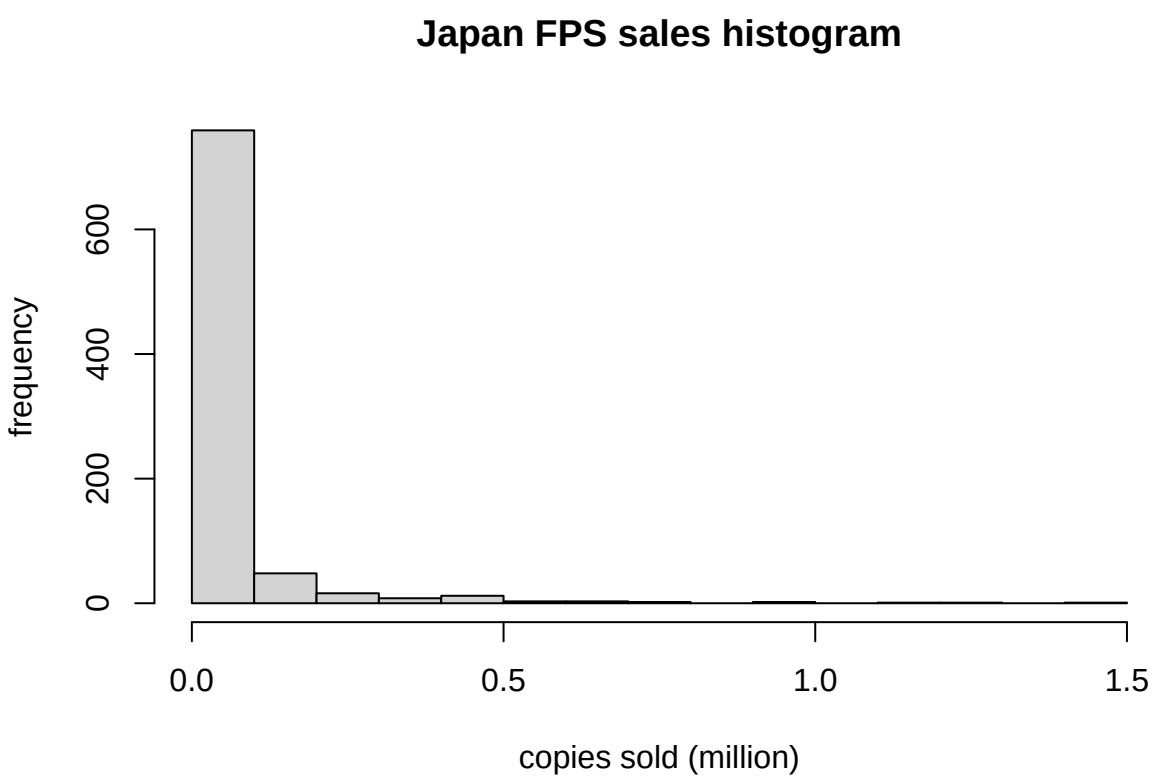
$$\textbf{H1: mean(RPG sales)} > \textbf{mean(RPG sales)}$$

Ako želimo provesti T-test moramo provjeriti normalnost podataka. Ovi podaci nisu normalni, više djeluju da dolaze iz eksponencijalne distribucije.

```
jp_rpg = vgsales.grouped[ vgsales.grouped["Genre"]=="Role-Playing", ]$JP_Sales
jp_fps = vgsales.grouped[ vgsales.grouped["Genre"]=="Shooter",]$JP_Sales

hist( jp_rpg,
      main="Japan RPG sales histogram",
      xlab="copies sold (million)",
      ylab="frequency")
```




```
hist( jp_fps,
      main="Japan FPS sales histogram",
      xlab="copies sold (million)",
      ylab="frequency")
```



Mogli bismo transformirati podatke da dobijemo normalnu distribuciju i nakon toga provesti T-test. Također, možemo provesti i bootstrap koji ne zahtjeva normalnost. Boostrap je nešto slabiji test, ali imamo veliki uzorak koji će to kompenzirati.

```
rpg_mean = mean( jp_rpg)
fps_mean = mean( jp_fps)

n = length( jp_rpg)
m = length( jp_fps)

t = (rpg_mean - fps_mean)/sqrt( var( jp_rpg)/n + var( jp_fps)/m)

z = mean( c(jp_rpg,jp_fps))

x = jp_rpg - rpg_mean + z
y = jp_fps - fps_mean + z

t_boot = vector(mode="numeric", length=10)
B = 1000

for( b in 1:B){
  x_temp = sample( x, size=n)
  y_temp = sample( y, size=m)

  t_b = (mean( x_temp) - mean( y_temp))/sqrt( var(x_temp)/n + var( y_temp)/m)
  t_boot[b] = t_b
}
```

```
p = sum( t_boot > t)/B
print( p)
```

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

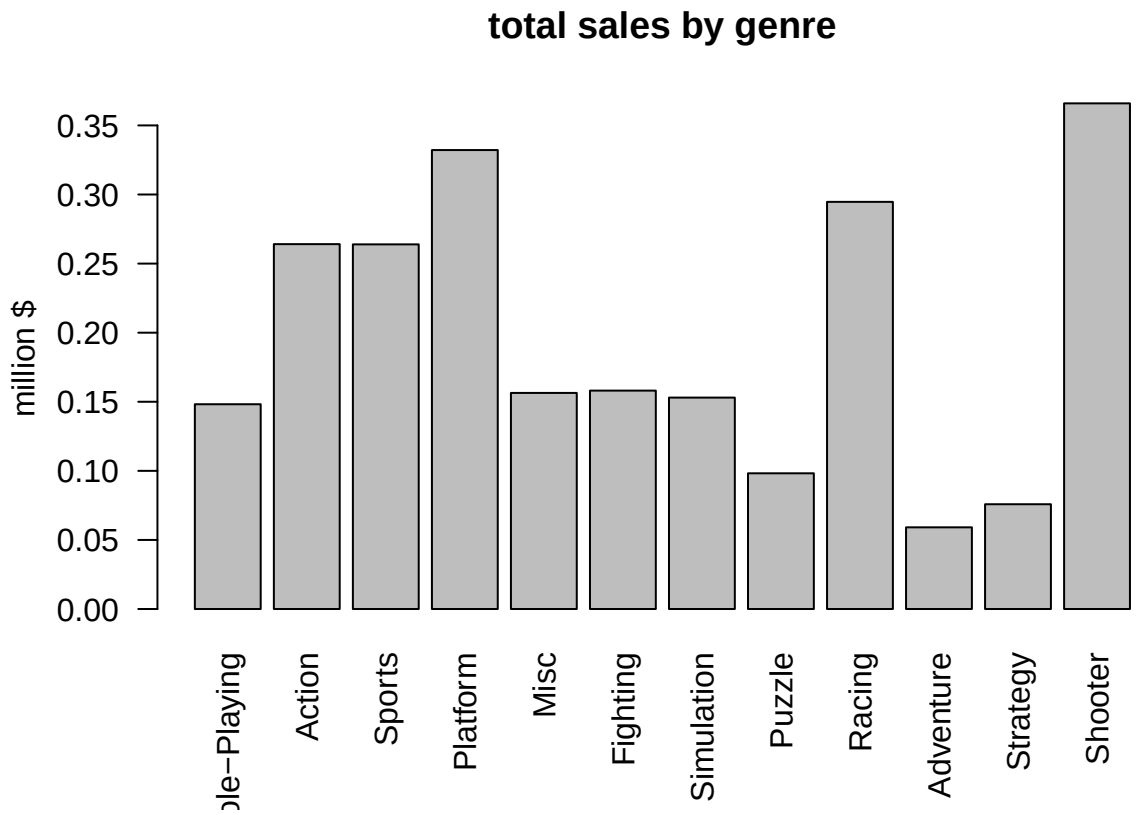
Zaključak

P-vrijednost ispada 0 što nam daje veliku sigurnost da možemo odbaciti H0 i prihvatiti alternativu. Dakle, zaključujemo da su, u Japanu, RPG igre popularnije od FPS igri.

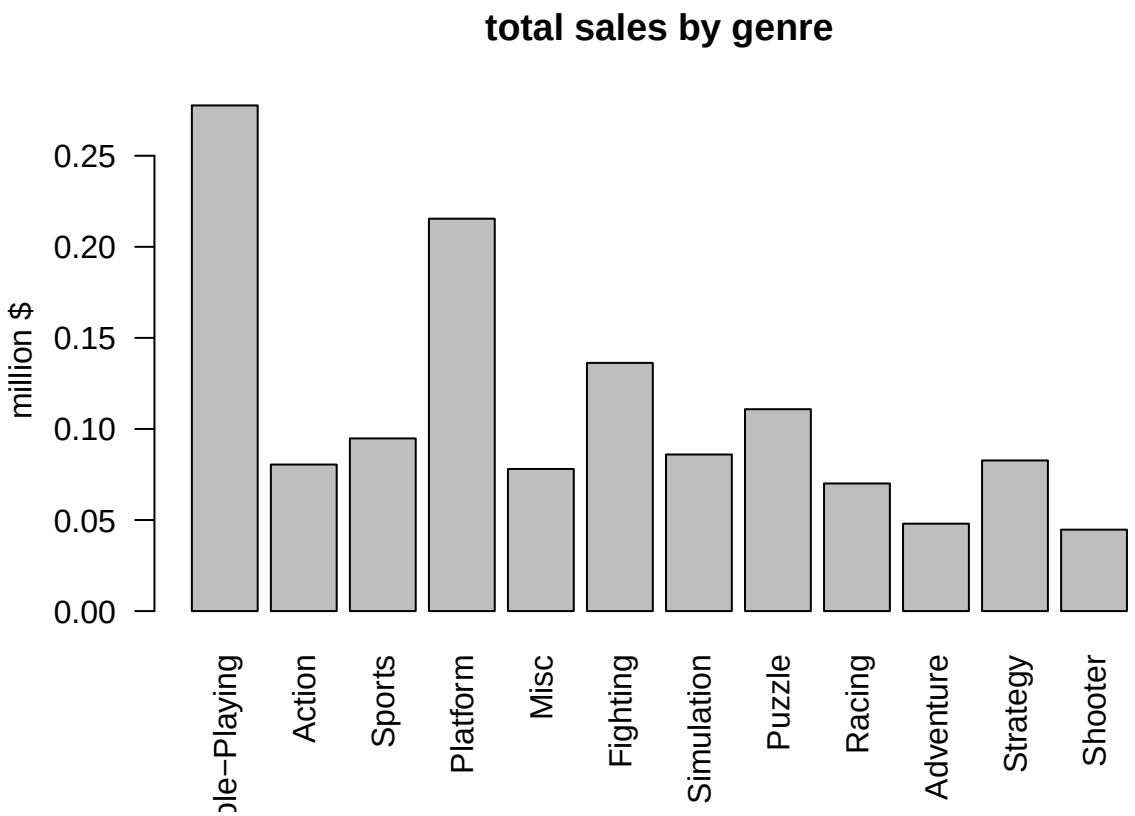
Pitanje 2: Možete li pronaći neki žanr koji je značajno popularniji u Europi nego u Japanu?

Usporedimo prvo prodaje po žanrovima posebno u Europi i Japanu, te razlike prodaja između Europe i Japana.

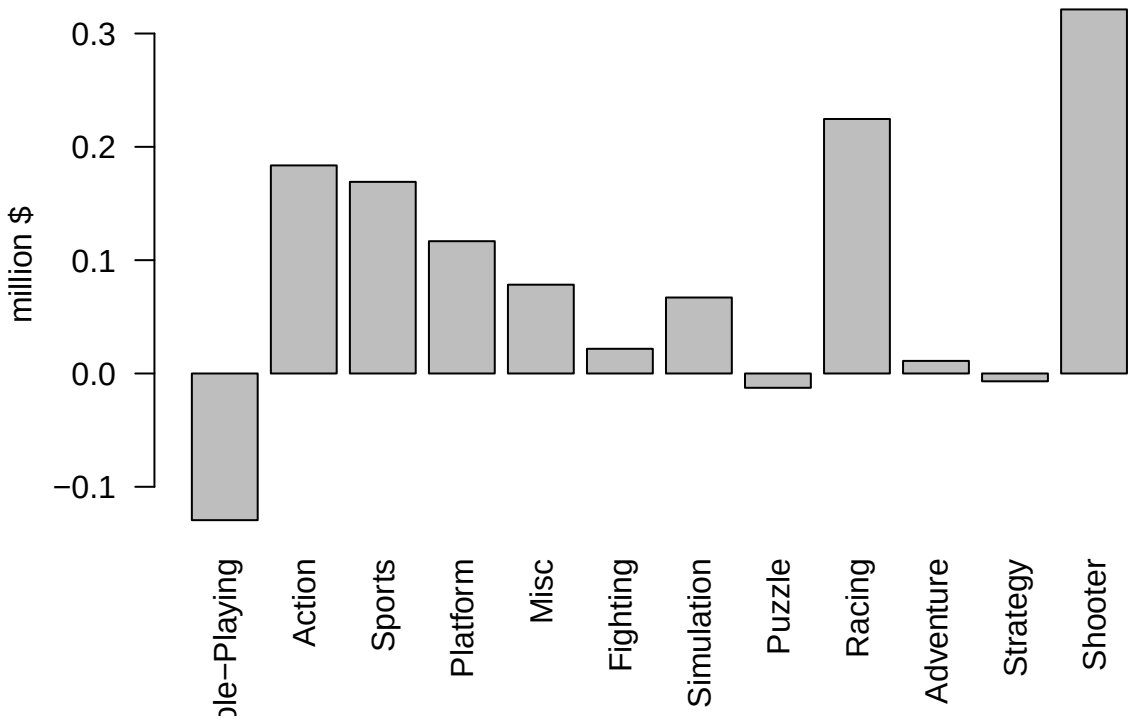
```
# Žanrovi u europi
europe.genre_sales = aggregate( vgsales.grouped[c("EU_Sales")],
                                vgsales.grouped["Genre"], mean)

barplot( europe.genre_sales$EU_Sales[ total_sales_genre_in_order],
         main="total sales by genre",
         ylab="million $",
         names.arg = total_sales_genre$Genre[ total_sales_genre_in_order],
         las=2)
```




```
# Žanrovi u japanu
japan.genre_sales = aggregate( vgsales.grouped[c("JP_Sales")],
                               vgsales.grouped["Genre"], mean)

barplot( japan.genre_sales$JP_Sales[ total_sales_genre_in_order],
         main="total sales by genre",
         ylab="million $",
         names.arg = total_sales_genre$Genre[ total_sales_genre_in_order],
         las=2)
```




```
# razlika između popularnosti žanrova u europi i japanu
difference.genre_sales = aggregate( vgsales.grouped["EU_Sales"] - vgsales.grouped["JP_Sales"],
                                    vgsales.grouped["Genre"], mean)

barplot( difference.genre_sales$EU_Sales[ total_sales_genre_in_order],
         main="total sales by genre",
         ylab="million $",
         names.arg = total_sales_genre$Genre[ total_sales_genre_in_order],
         las=2)
```



Prema dobivenim grafovima dobivamo osjećaj o tome koji bi žanrovi mogli biti popularniji u Europi u odnosu na Japan.

Provjerimo je li žanr 'Shooter' značajno pupularniji u Europi nego u Japanu. Mičemo outliere iz prodaje igara koje pripadaju žanru 'Shooter' u Europi i Japanu:

```
shooter = vgsales[vgsales$Genre == "Shooter", ]

outliers.shooter_eu_sales = boxplot(shooter$EU_Sales, plot = FALSE)$out
shooter_eu_sales_wihtout_outliers = shooter[! shooter$EU_Sales %in% outliers.shooter_eu_sales, ]$EU_Sale

outliers.shooter_jp_sales = boxplot(shooter$JP_Sales, plot = FALSE)$out
shooter_jp_sales_wihtout_outliers = shooter[! shooter$JP_Sales %in% outliers.shooter_jp_sales, ]$JP_Sale

#Prije početka testiranja, skaliramo shooter_eu_sales_without_outliers zbog razlika tržišta Europe i Ja
shooter_eu_sales_wihtout_outliers = shooter_eu_sales_wihtout_outliers * sum(shooter$JP_Sales) / sum(shoo
```

Radimo bootstrap test sredina žanra 'Shooter'.

H0: Sredine su jednake; $\mu_{shooter\_eu} = \mu_{shooter\_jp}$

H1 Sredina Europe je veća; $\mu_{shooter\_eu} > \mu_{shooter\_jp}$

```
eu_shooter_mean = mean(shooter_eu_sales_wihtout_outliers)
jp_shooter_mean = mean(shooter_jp_sales_wihtout_outliers)

n = length( shooter_eu_sales_wihtout_outliers)
m = length( shooter_jp_sales_wihtout_outliers)
```

```
t = (eu_shooter_mean - jp_shooter_mean)/sqrt( var( shooter_eu_sales_wihtout_outliers)/n + var( shooter_

z = mean( c(shooter_eu_sales_wihtout_outliers,shooter_jp_sales_wihtout_outliers))

x = shooter_eu_sales_wihtout_outliers - eu_shooter_mean + z
y = shooter_jp_sales_wihtout_outliers - jp_shooter_mean + z

t_boot = vector(mode="numeric", length=10)
B = 1000

for( b in 1:B){
  x_temp = sample( x, size=n)
  y_temp = sample( y, size=m)

  t_b = (mean( x_temp) - mean( y_temp))/sqrt( var(x_temp)/n + var( y_temp)/m)
  t_boot[b] = t_b
}
p = sum( t_boot > t)/B
print( p)
```

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

Zaključak

Prema rezultatu Bootstrap-a, odbacujemo hipotezu da su sredine iste i zaključujemo da je žanr 'Shooter' značajno popularniji u Europi nego u Japanu.

Pitanje 3: Jesu li izdavači jednako popularni u svim regijama?

Zanima nas imaju li izdavači prednost na nekom tržištu. Npr. da li izdavači prodaju više kopija igri na domaćem tržištu. Postavljamo statističku hipotezu i provodimo test homogenosti.

H0: Za svakog izdavača, proporcije prodaja po regijama su jednake

H1: Za svakog izdavača, proporcije prodaja po regijama nisu jednake

Kontingencijska tablica izgleda ovako:

```
publisher_region = vgsales.grouped[ vgsales.grouped$Publisher == "Nintendo" |
                            vgsales.grouped$Publisher == "Microsoft Game Studios" |
                            vgsales.grouped$Publisher == "Sony Computer Entertainment" |
                            vgsales.grouped$Publisher == "Electronic Arts",]

publisher_region = publisher_region[ c("Publisher", "NA_Sales", "EU_Sales",
                                       "JP_Sales", "Other_Sales")]
publisher_region = aggregate( publisher_region[c("NA_Sales", "EU_Sales",
                                       "JP_Sales", "Other_Sales")],
                              list(publisher_region$Publisher), sum)
table = as.table( as.matrix(publisher_region[-1]))
rownames( table) = publisher_region$Group.1
table = addmargins( table)
print( table)
```

```
##                             NA_Sales EU_Sales JP_Sales Other_Sales     Sum
## Electronic Arts               595.07   371.27    14.04      129.77 1110.15
## Microsoft Game Studios        155.35    68.61     3.26       18.56  245.78
## Nintendo                      816.87   418.74   455.42       95.33 1786.36
## Sony Computer Entertainment   265.22   187.72    74.10       80.45  607.49
## Sum                          1832.51  1046.34   546.82      324.11 3749.78
```

Prije nego što provedemo chi-kvadrat test moramo provjeriti jesu li zadovoljene pretpostavke testa. Očekivane frekvencije svih razreda moraju biti > 5. U ovom slučaju je to zadovoljeno.

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

```
## Očekivane frekvencije za razred  NA_Sales - Electronic Arts :  542.5281
## Očekivane frekvencije za razred  NA_Sales - Microsoft Game Studios :  120.1122
## Očekivane frekvencije za razred  NA_Sales - Nintendo :  872.9906
## Očekivane frekvencije za razred  NA_Sales - Sony Computer Entertainment :  296.8792
## Očekivane frekvencije za razred  EU_Sales - Electronic Arts :  309.7767
## Očekivane frekvencije za razred  EU_Sales - Microsoft Game Studios :  68.58254
## Očekivane frekvencije za razred  EU_Sales - Nintendo :  498.4666
## Očekivane frekvencije za razred  EU_Sales - Sony Computer Entertainment :  169.5142
## Očekivane frekvencije za razred  JP_Sales - Electronic Arts :  161.8901
## Očekivane frekvencije za razred  JP_Sales - Microsoft Game Studios :  35.84141
## Očekivane frekvencije za razred  JP_Sales - Nintendo :  260.4999
```

```
## Očekivane frekvencije za razred  JP_Sales - Sony Computer Entertainment :  88.58858
## Očekivane frekvencije za razred  Other_Sales - Electronic Arts :  95.95515
## Očekivane frekvencije za razred  Other_Sales - Microsoft Game Studios :  21.24385
## Očekivane frekvencije za razred  Other_Sales - Nintendo :  154.403
## Očekivane frekvencije za razred  Other_Sales - Sony Computer Entertainment :  52.50804
```

Provodimo test:

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

```
##
##  Pearson's Chi-squared test
##
## data:  table
## X-squared = 411.91, df = 16, p-value < 2.2e-16
```

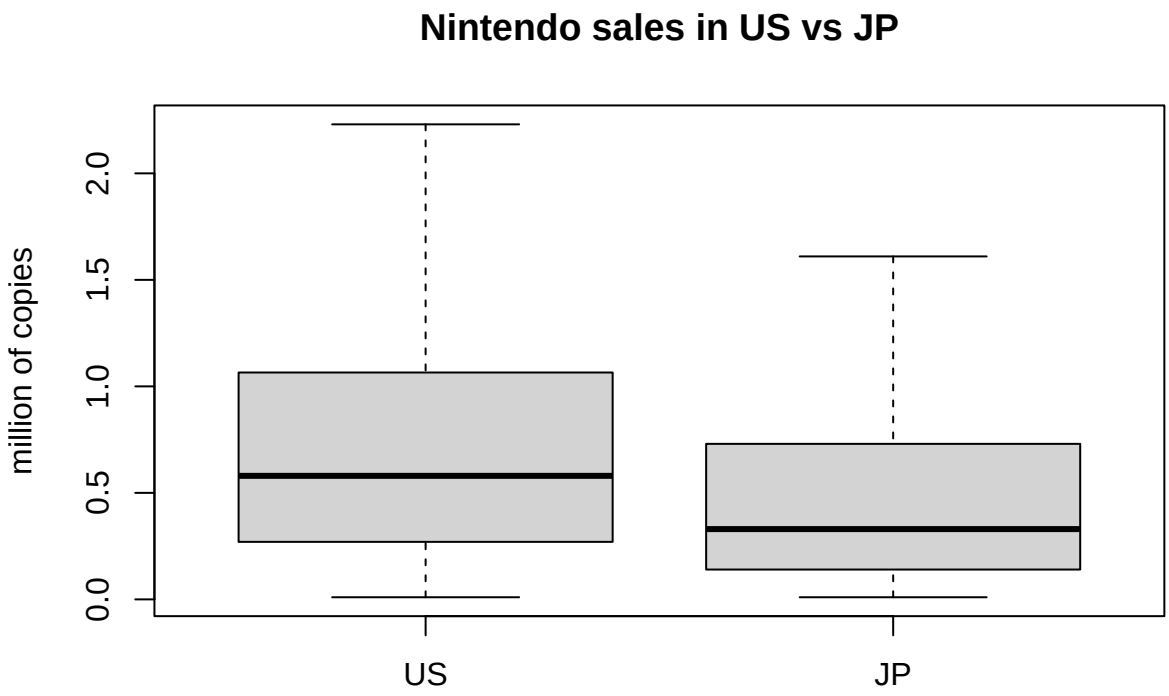
Zaključak

Nisu svi izdavači jednako popularni u svakoj regiji.

Pitanje 4: Prodaje li se, u prosjeku, više primjeraka igrica u Sjedinjenim državama ili Japanu za proizvođača Nintendo?

Zbog toga što jedna videoigra može biti veliki uspjeh u nekoj državi (npr. Super Mario) i ne reprezentira prodaju ostalih igara, potrebno je maknuti stršeće vrijednosti. Prvo ćemo maknuti one igrice koje odskaču u Sjedinjenim državama pa one koje odskaču u Japanu. Gledati ćemo iste igrice (koje nisu pretežno popularne) koje prodajemo u Sjedinjenim državama i Japanu.

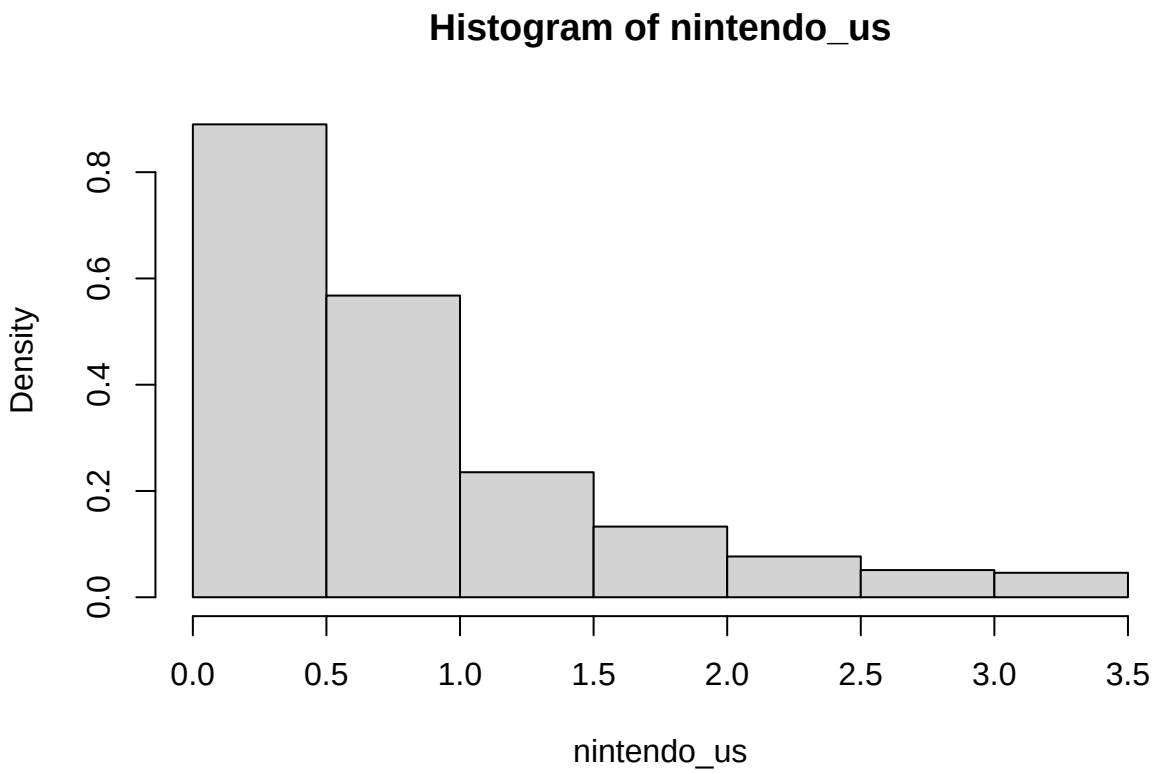
```
nintendo = vgsales[vgsales["Publisher"] == "Nintendo" &
                     vgsales["NA_Sales"] != 0.00 &
                     vgsales["JP_Sales"] != 0.00,]

Q <- quantile(nintendo$NA_Sales, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(nintendo$NA_Sales)

nintendo <- subset(nintendo, nintendo$NA_Sales > (Q[1] - 1.5*iqr)
                   & nintendo$NA_Sales < (Q[2]+1.5*iqr))

Q <- quantile(nintendo$JP_Sales, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(nintendo$JP_Sales)

nintendo <- subset(nintendo, nintendo$JP_Sales > (Q[1] - 1.5*iqr)
                   & nintendo$JP_Sales < (Q[2]+1.5*iqr))

nintendo_us <- nintendo$NA_Sales
nintendo_jp <- nintendo$JP_Sales

boxplot( nintendo_us, nintendo_jp,
         main="Nintendo sales in US vs JP",
         ylab="million of copies",
         names=c("US", "JP"),
         outline=FALSE)
```



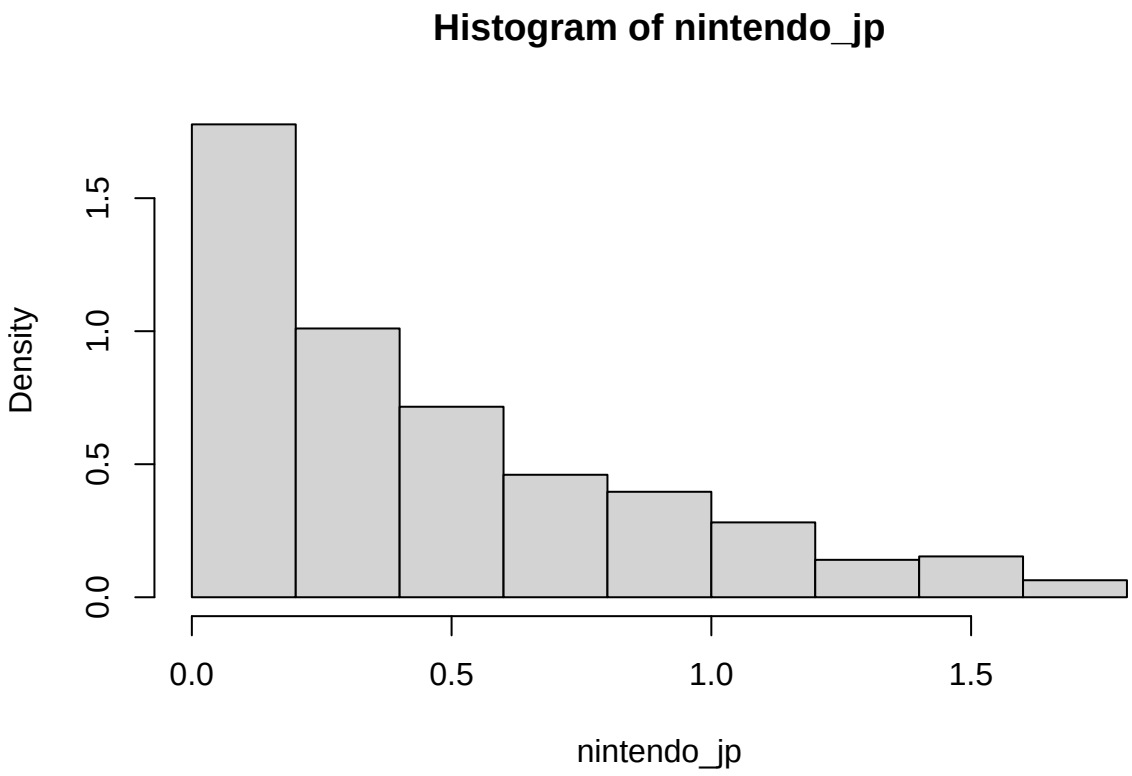
Na grafu se može vidjeti da je medijan veći za Sjedinjene države negoli za Japan što je dobra naznaka za ono što želimo testirati.

Pogledajmo sada kakva je distribucija podataka.

```
hist(nintendo_us, freq=FALSE)
```

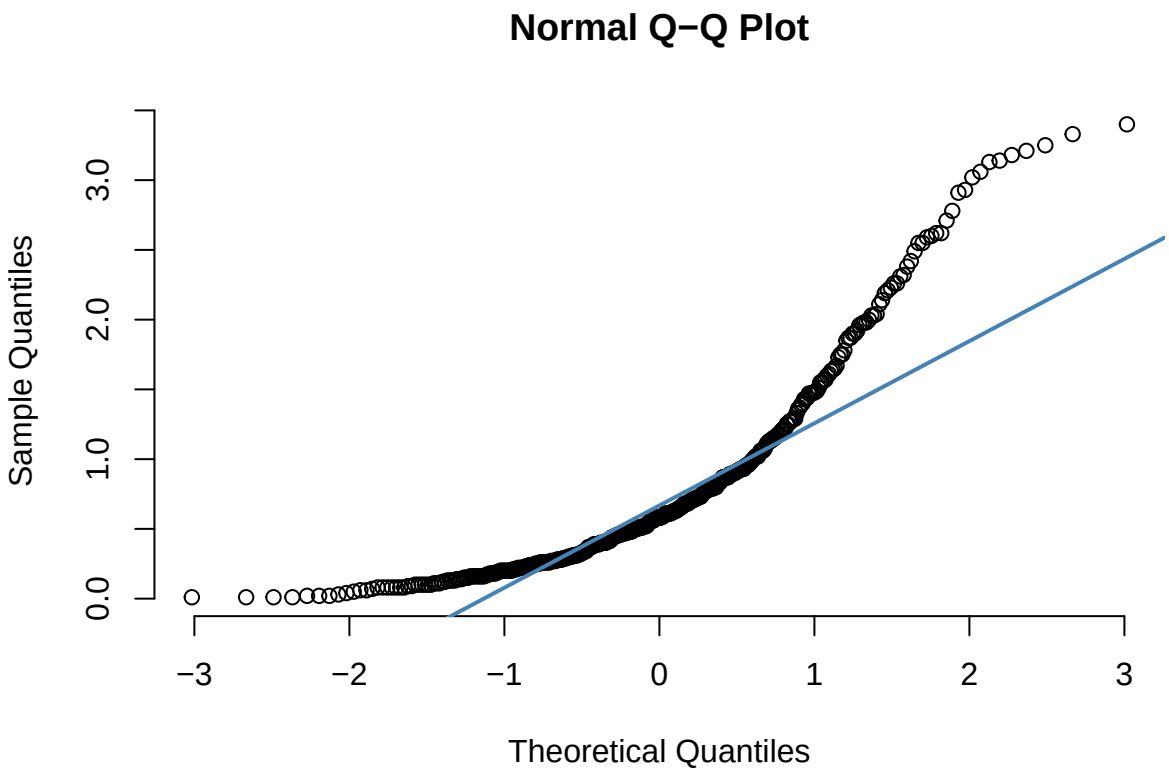



```
hist(nintendo_jp, freq=FALSE)
```

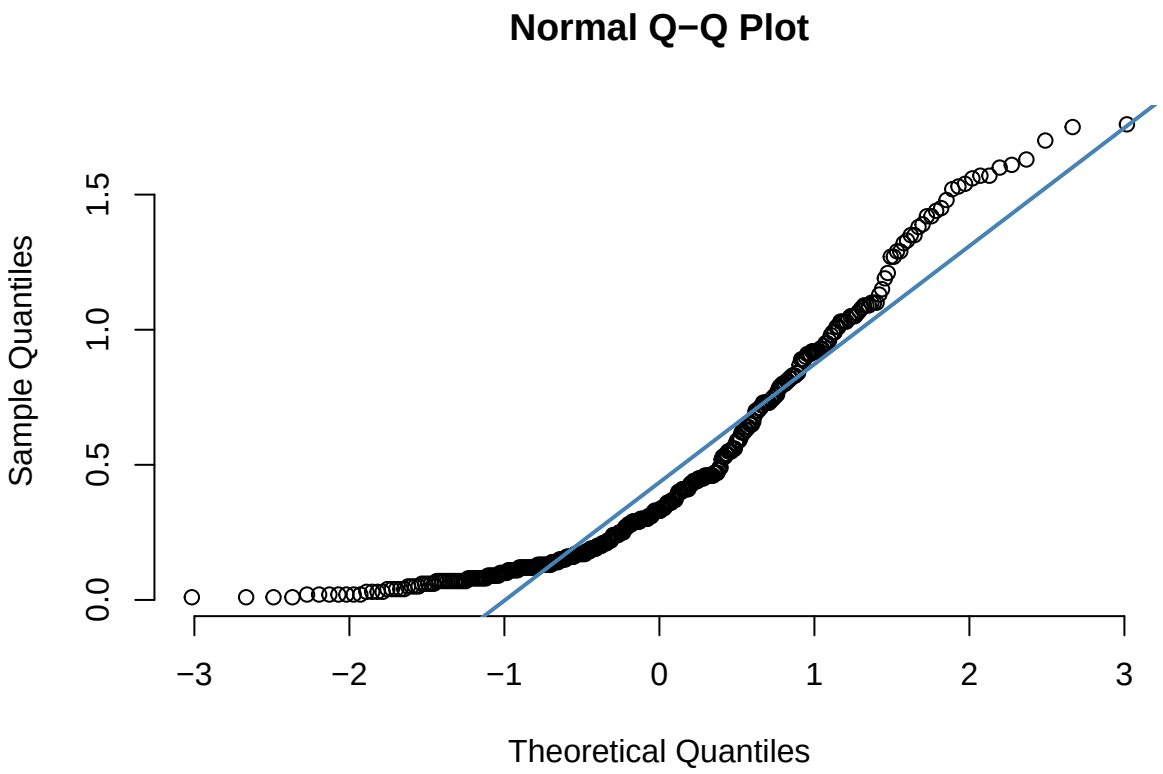


Ovo nam je naznaka da podaci nisu normalno distribuirani. Pogledajmo još qq-plot.

```
qqnorm(nintendo_us, pch = 1, frame = FALSE)
qqline(nintendo_us, col = "steelblue", lwd = 2)
```




```
qqnorm(nintendo_jp, pch = 1, frame = FALSE)
qqline(nintendo_jp, col = "steelblue", lwd = 2)
```



Iako se iz histograma i qq-plota može dobiti dobra slika o normalnosti podatka, ipak je bolje provesti statistički test kako bismo to i dokazali.

```
ks.test(nintendo_us, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_us
## D^- = 0.50399, p-value < 2.2e-16
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

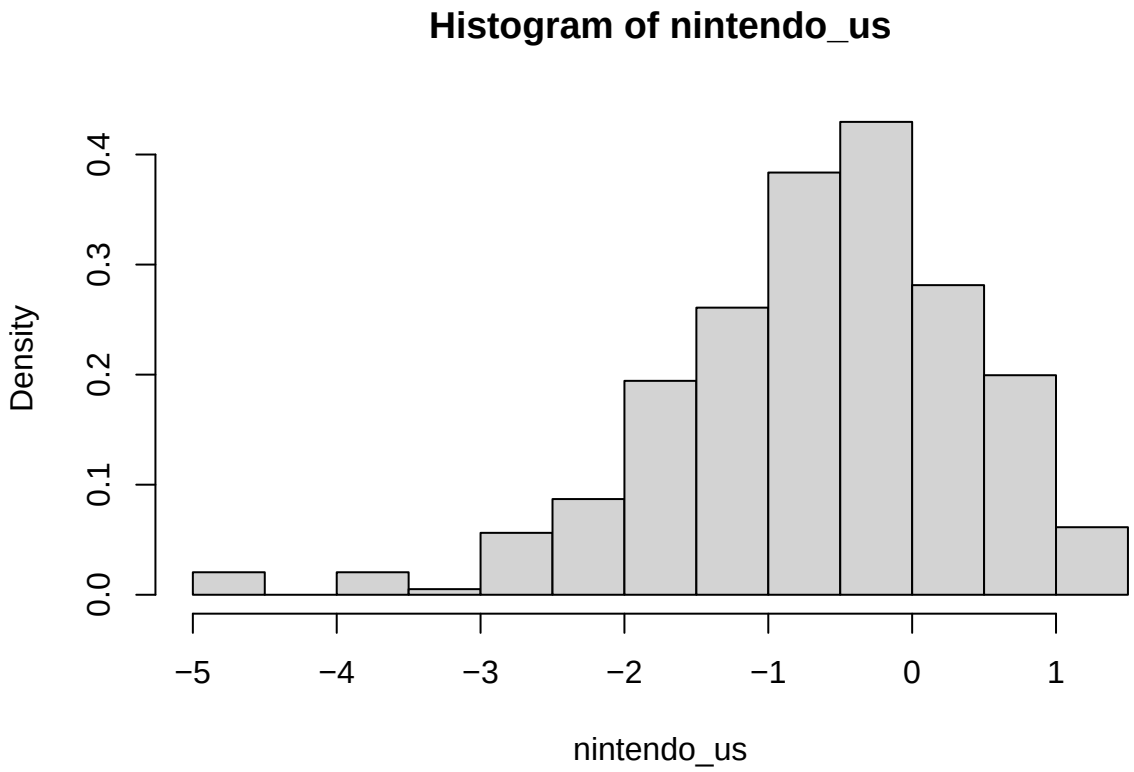
```
ks.test(nintendo_jp, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_jp
## D^- = 0.50399, p-value < 2.2e-16
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

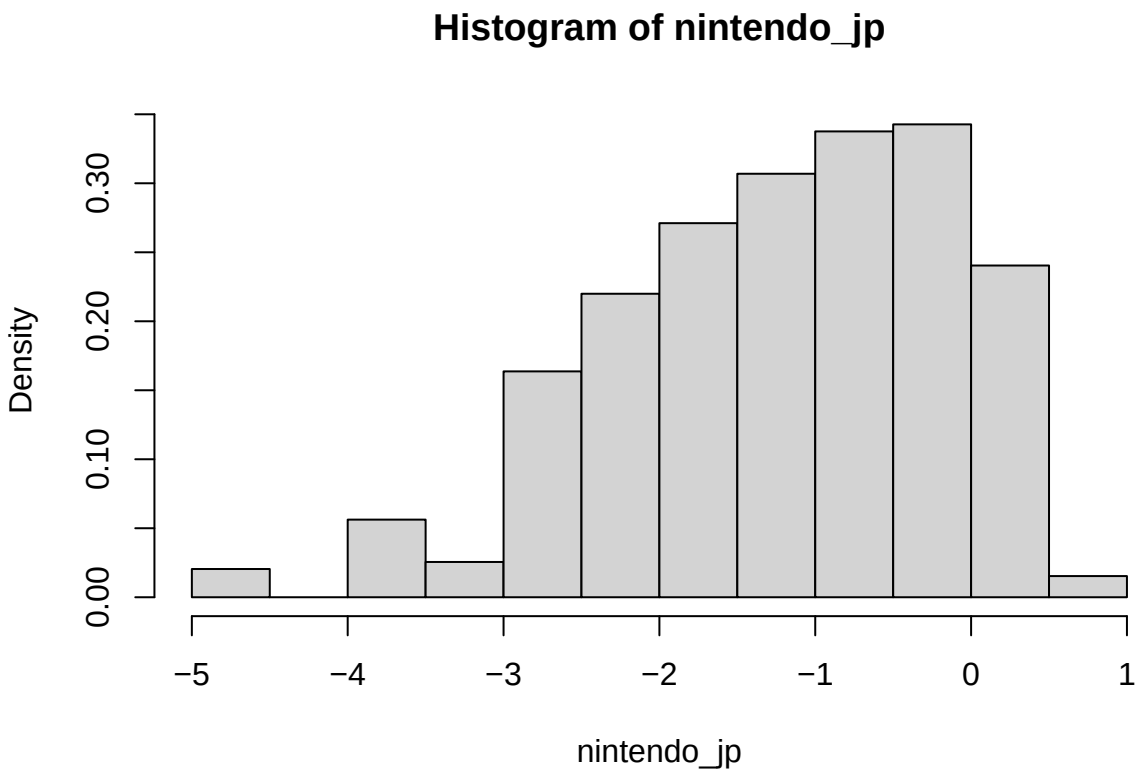
U oba slučaja odbacujemo nul-hipotezu (podaci su normalno distribuirani) u korist alternativne hipoteze (podaci nisu normalno distribuirani) zato što je p-vrijednost manja od 0.05. Zato ćemo pokušati transformirati podatke kako bismo dobili "normalnije" podatke.

```
nintendo_us <- log(nintendo_us)
nintendo_jp <- log(nintendo_jp)
```

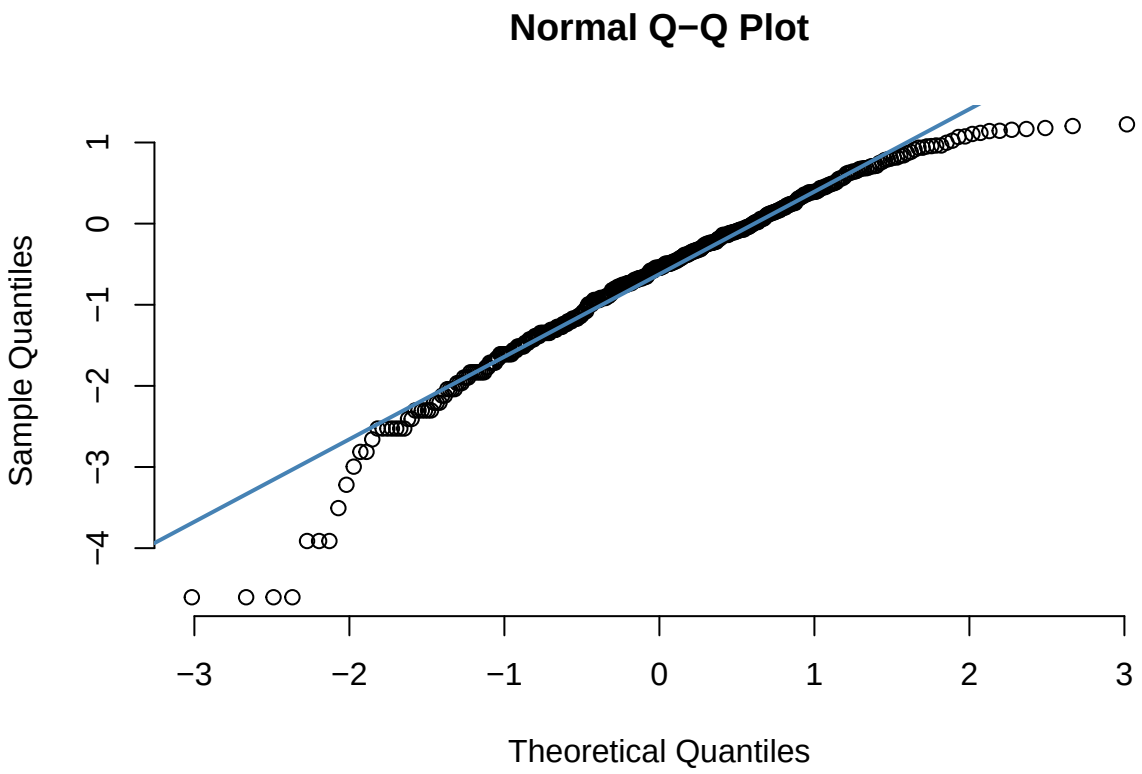
```
hist(nintendo_us, freq=FALSE)
```



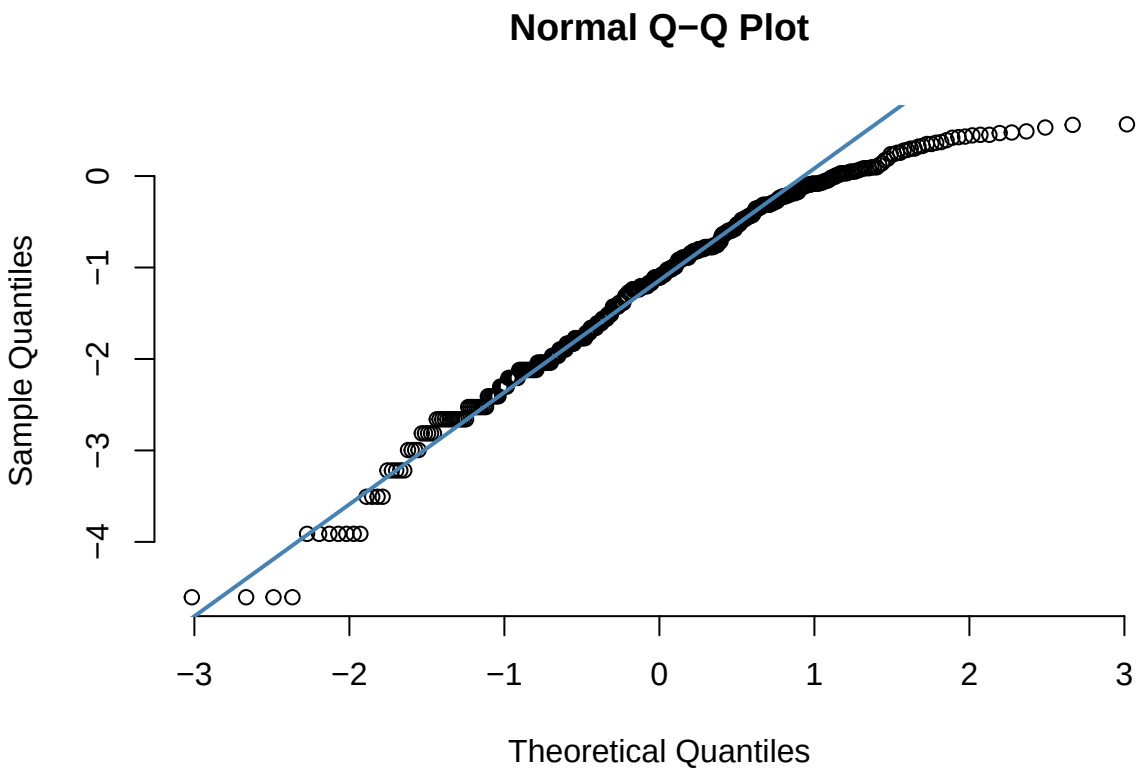

```
hist(nintendo_jp, freq=FALSE)
```




```
qqnorm(nintendo_us, pch = 1, frame = FALSE)
qqline(nintendo_us, col = "steelblue", lwd = 2)
```




```
qqnorm(nintendo_jp, pch = 1, frame = FALSE)
qqline(nintendo_jp, col = "steelblue", lwd = 2)
```



Vidimo da su sada podaci bliži normalnoj distribuciji nego što su bili, pa će i t-test biti točniji. Također, opet provodimo test za normalnost podataka.

```
ks.test(nintendo_us, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_us
## D^- = 2.0606e-06, p-value = 1
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

```
ks.test(nintendo_jp, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_jp
## D^- = 2.0606e-06, p-value = 1
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

Sada kada znamo da su naši podaci približno normalno distribuirani, potrebno je i provjeriti jednakost varijanci statističkim testom (graf je dobar uvid, ali nije dovoljan).

```
var.test(nintendo_us, nintendo_jp, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
```

```
## data:  nintendo_us and nintendo_jp
## F = 0.9639, num df = 390, denom df = 390, p-value = 0.7167
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7901338 1.1758773
## sample estimates:
## ratio of variances
##          0.9638985
```

Budući da je p-vrijednost veća od 0.05, prihvaćamo nul hipotezu i zaključujemo da nema značajne razlike u varijancama ovih dvaju uzoraka.

```
t.test(nintendo_us, nintendo_jp, paired = TRUE, alternative = "greater", var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  nintendo_us and nintendo_jp
## t = 11.574, df = 390, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.4839649       Inf
## sample estimates:
## mean of the differences
##               0.5643643
```

Zaključak

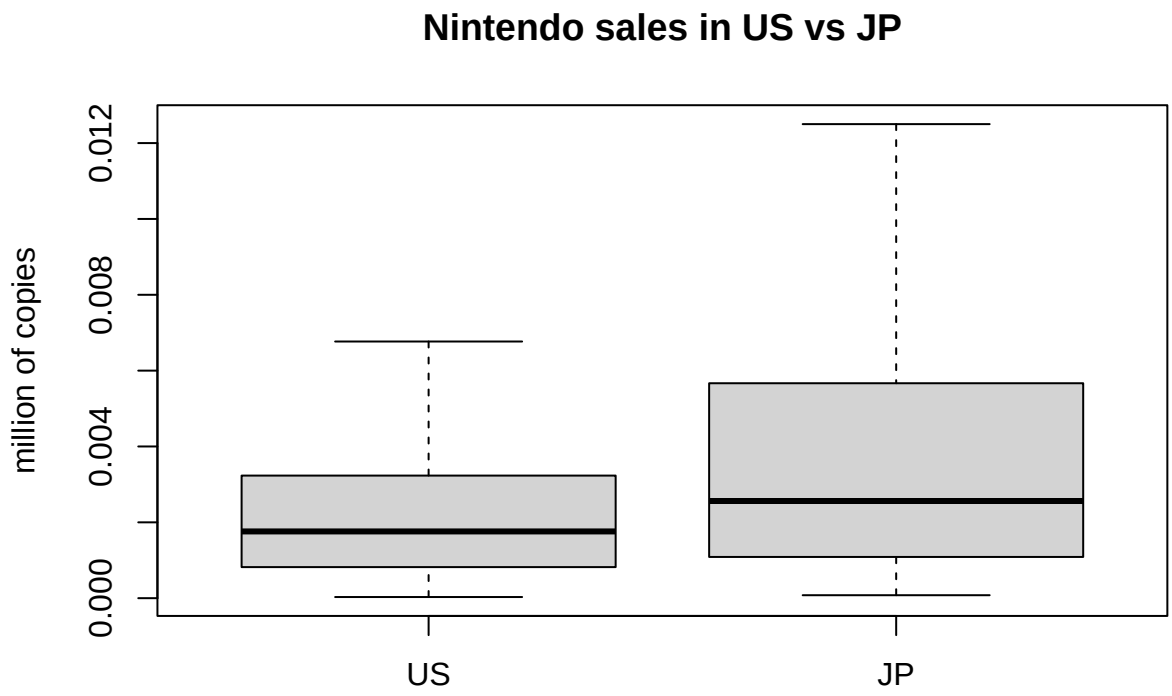
P vrijednost je manja od 0.05 i te odbacujemo H0 i zaključujemo da je u prosjeku prodano više igara proizvođača Nintendo u Sjedinjenim državama što je i očekivano budući da u Sjedinjenim državama živi daleko više ljudi.

Usporedba per capita

Pogledajmo sada što se događa "per capita", odnosno po glavi stanovnika. Broj ljudi koji živi u Sjedinjenim državama jest 329,5 milijuna, dok u Japanu živi 125,8 milijuna ljudi.

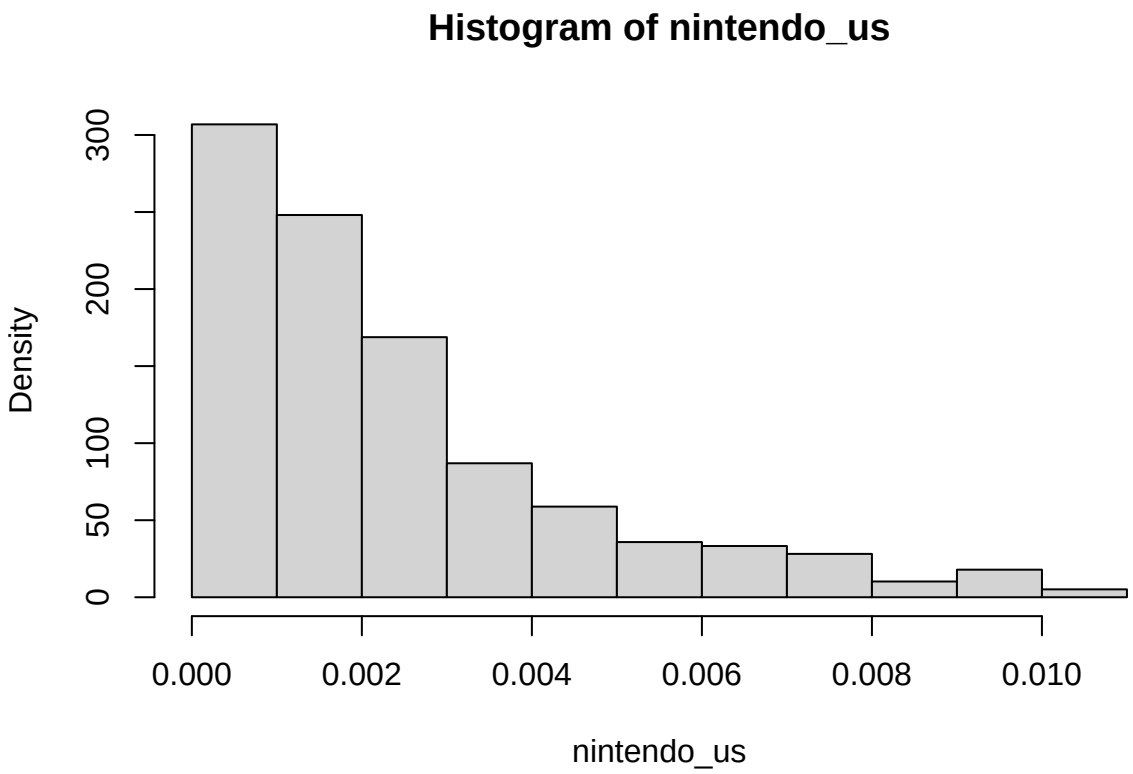
```
nintendo = vgsales[vgsales["Publisher"] == "Nintendo"
                   & vgsales["NA_Sales"] != 0.00
                   & vgsales["JP_Sales"] != 0.00,]

Q <- quantile(nintendo$NA_Sales, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(nintendo$NA_Sales)

nintendo <- subset(nintendo, nintendo$NA_Sales > (Q[1] - 1.5*iqr)
                   & nintendo$NA_Sales < (Q[2]+1.5*iqr))

Q <- quantile(nintendo$JP_Sales, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(nintendo$JP_Sales)

nintendo <- subset(nintendo, nintendo$JP_Sales > (Q[1] - 1.5*iqr)
                   & nintendo$JP_Sales < (Q[2]+1.5*iqr))

nintendo_us = nintendo$NA_Sales / 329.5
nintendo_jp = nintendo$JP_Sales / 128.8
```

```
boxplot( nintendo_us, nintendo_jp,
         main="Nintendo sales in US vs JP",
         ylab="million of copies",
         names=c("US", "JP"),
         outline=FALSE)
```

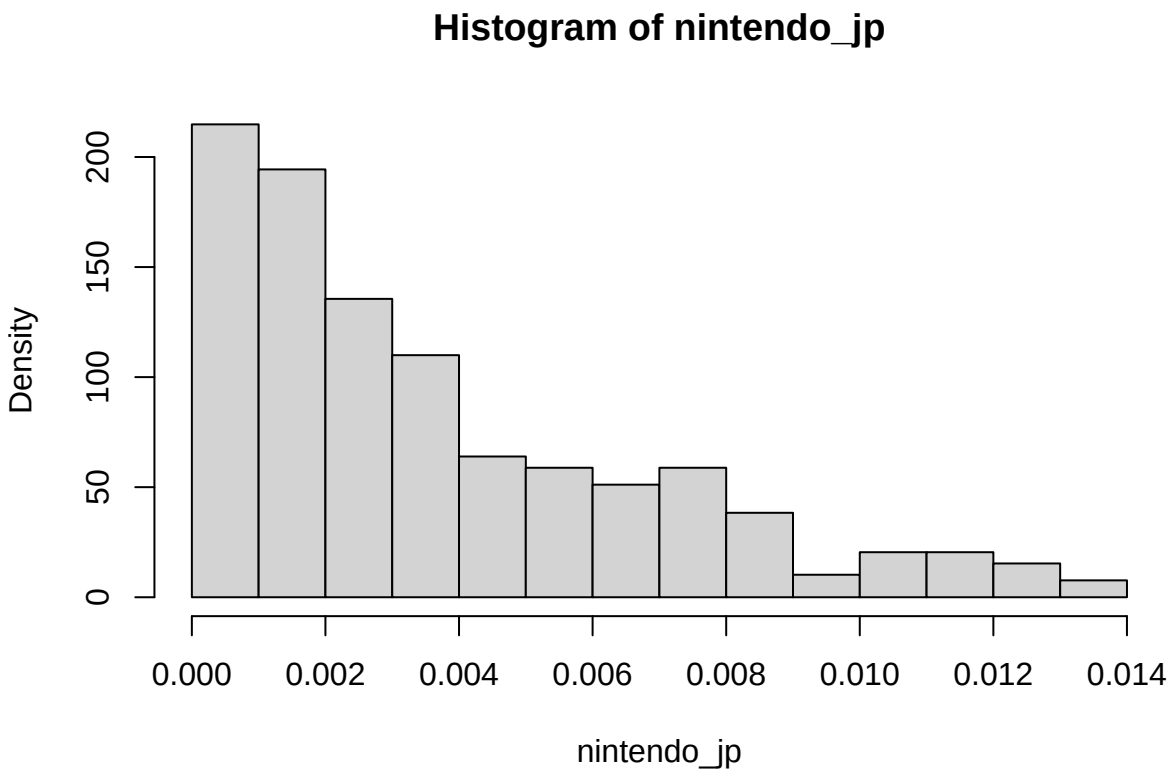


Vidimo da je situacija sada obrnuta. Pogledajmo distribuciju podataka.

```
hist(nintendo_us, freq=FALSE)
```




```
hist(nintendo_jp, freq=FALSE)
```



Vidimo da podaci vjerojatno nisu normalno distribuirani, pa provodimo test.

```
ks.test(nintendo_us, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_us
## D^- = 0.50001, p-value < 2.2e-16
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

```
ks.test(nintendo_jp, "pnorm", alternative ="less")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_jp
## D^- = 0.50003, p-value < 2.2e-16
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

Zaključujemo da podaci nisu normalno distribuirani pa ih pokušamo transformirati.

```
nintendo_us <- log(nintendo_us)
nintendo_jp <- log(nintendo_jp)
```

Sada opet provodimo test kako bi se uvjerili u distribuciju podataka.

```
ks.test(nintendo_us, "pnorm", alternative ="less")
```

```
##
```

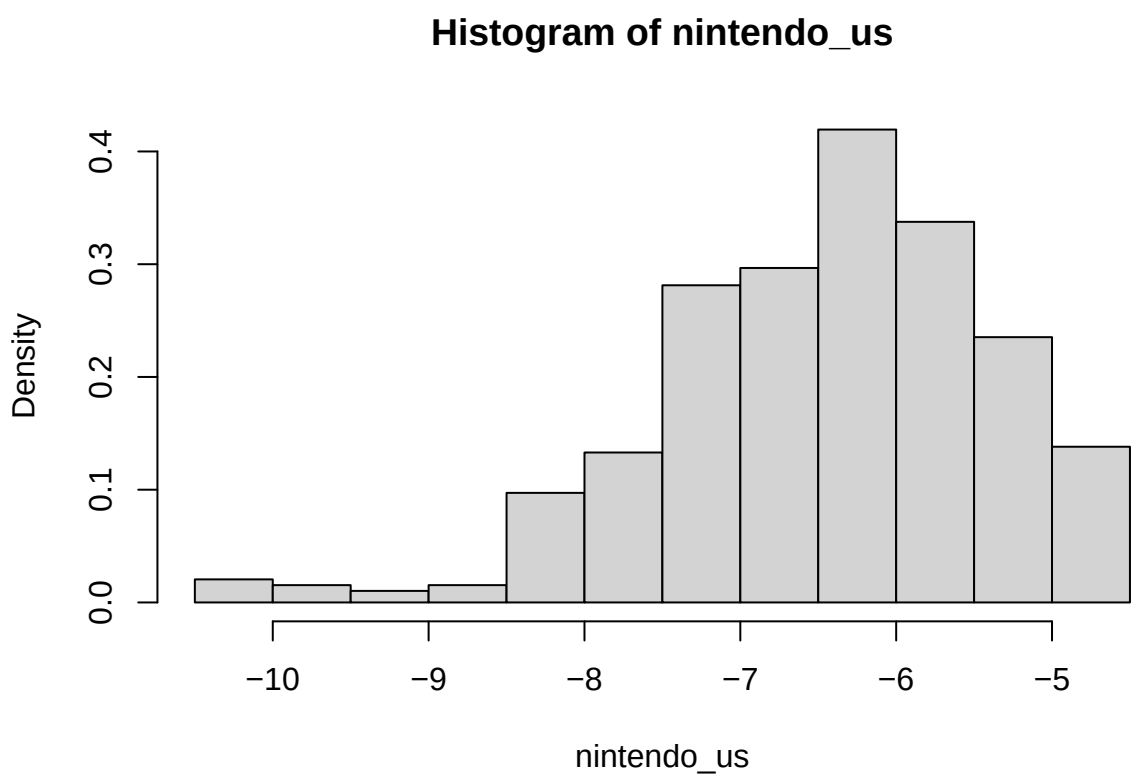
```
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_us
## D^- = 1.2044e-25, p-value = 1
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

```
ks.test(nintendo_jp, "pnorm", alternative ="less")
```

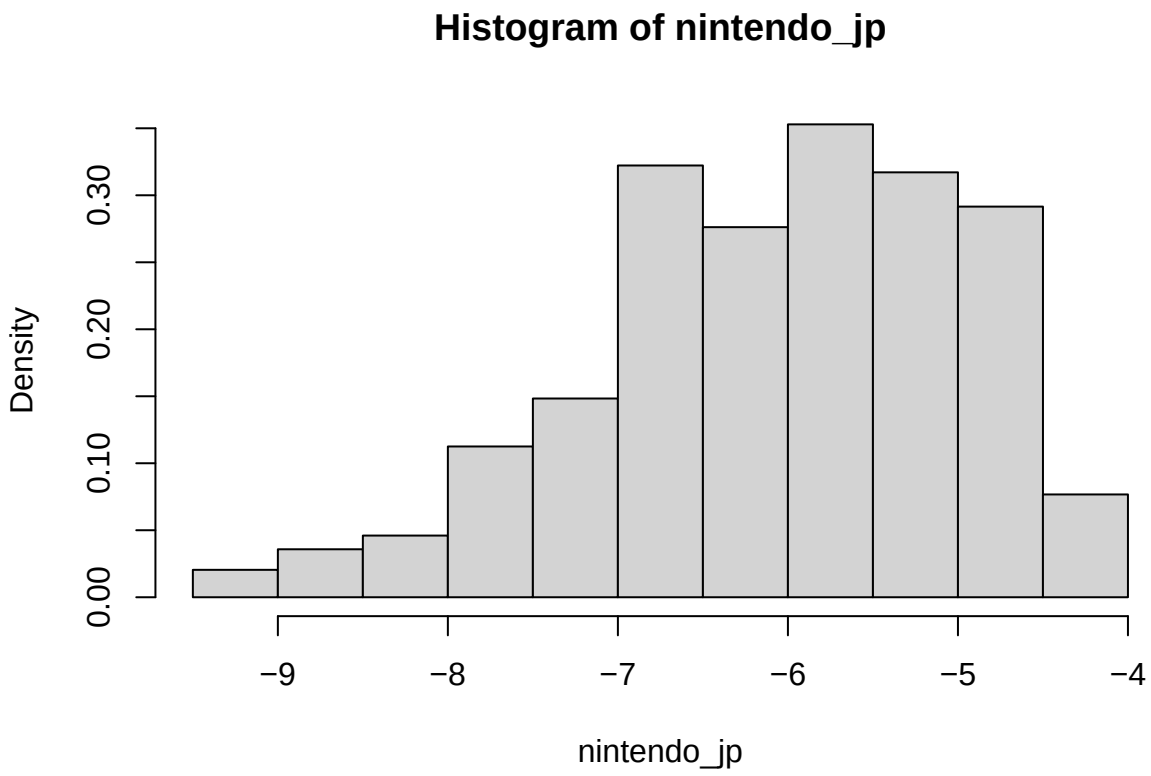
```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nintendo_jp
## D^- = 1.49e-21, p-value = 1
## alternative hypothesis: the CDF of x lies below the null hypothesis
```

Zaključujemo da su podaci približno normalno distribuirani. Pogledajmo sada i grafički prikaz podataka.

```
hist(nintendo_us, freq=FALSE)
```




```
hist(nintendo_jp, freq=FALSE)
```



Vidimo da su sada podaci bliži normalnoj te opet testiramo jednakost varijanci kako bi mogli što točnije provesti t-test.

```
var.test(nintendo_us, nintendo_jp, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  nintendo_us and nintendo_jp
## F = 0.9639, num df = 390, denom df = 390, p-value = 0.7167
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7901338 1.1758773
## sample estimates:
## ratio of variances
##          0.9638985
```

Zaključujemo da nema značajne razlike u varijancama.

```
t.test(nintendo_us, nintendo_jp, paired = TRUE, alternative = "less", var.equal = TRUE)
```

```
##
##  Paired t-test
##
## data:  nintendo_us and nintendo_jp
## t = -7.6892, df = 390, p-value = 6.089e-14
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
```

```
##        -Inf -0.2945518
## sample estimates:
## mean of the differences
##              -0.3749512
```

Zaključak - per capita

Dakle, ukoliko uklonimo stršeće vrijednosti, odnosno ne gledamo igrice koje su postale izrazito popularne u Sjedinjenim državama i Japanu možemo zaključiti da se u prosjeku po glavi stanovnika više igrica prozvođača Nintendo proda u Japanu nego u Sjedinjenim državama.

Pitanje 5: Promatrajući prodaju u Sjevernoj Americi, jesu li neki žanrovi značano popularniji?

Za odgovor na ovo pitanje moramo definirati 'mjeru popularnosti'. Uzimamo da je srednja vrijednost broja prodaja dobar analog popularnosti. Zbog toga se problem svodi na uspoređivanje srednjih vrijednosti broja prodaja po žanrovima.

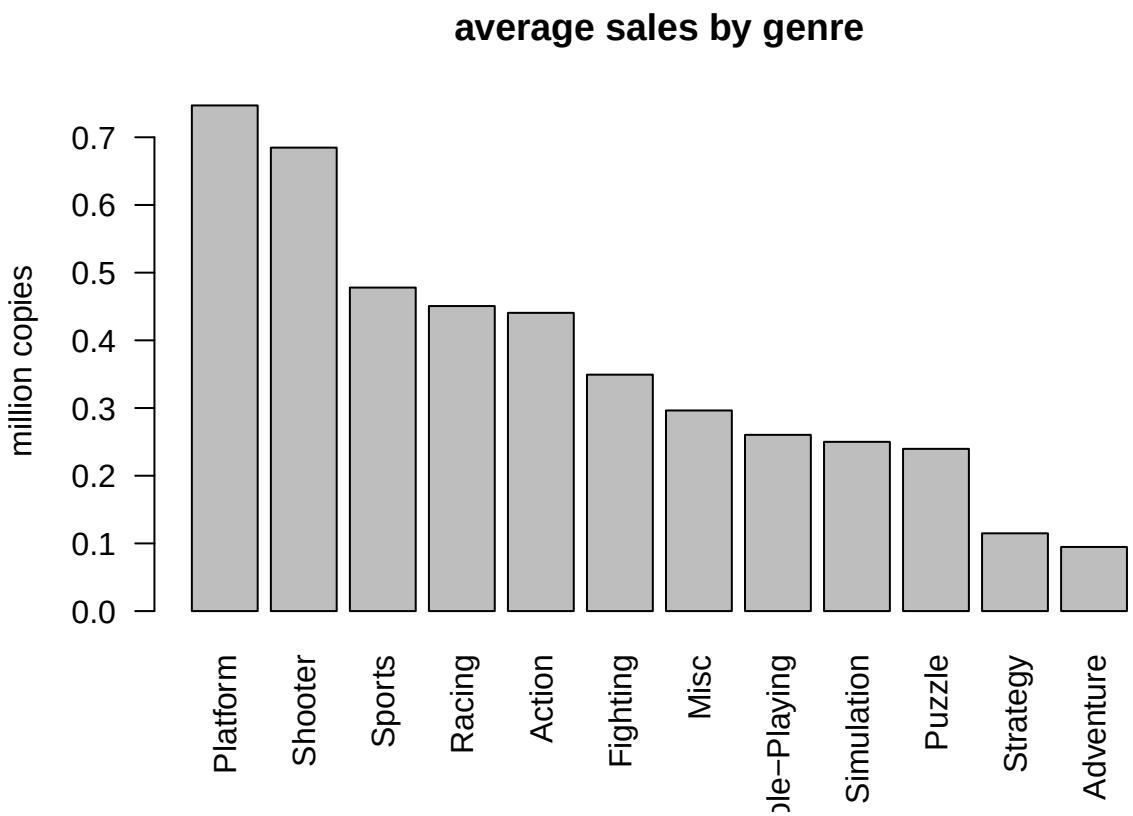
Jednakost srednjih vrijednosti možemo provjeriti ANOVA-om. Za primjenu ANOVA-e trebamo provjeriti normalnost podataka i homogenost varijance.

```
# ucitavanje podataka
vgsales_p3 = read.csv("vgsales.csv")
vgsales_p3$Year = suppressWarnings(as.integer(vgsales_p3$Year))
```

```
# stvaramo korisne varijable, micemo retke s NA (not available) vrijednostima
svi_zanrovi <- unique(vgsales_p3$Genre)
vgsales_p3 <- na.omit(vgsales_p3)

# grupiramo podatke kao i prije
agr = aggregate(vgsales_p3[c(7, 8, 9, 10, 11)], vgsales_p3[c("Name", "Publisher", "Genre")], FUN = sum)

# za godinu uzimamo srednju vrijednost
vgsales_p3 = merge(agr, aggregate(vgsales_p3["Year"],
                                  vgsales_p3[c("Name", "Publisher", "Genre")],
                                  FUN = mean),
                                  by = c("Name", "Publisher", "Genre"))
```

Prvo želimo grafički prikazati ljestvicu prosječnih prodaja po žanru u NA. Rezultat nam može dati moguće kandidate za daljnju analizu.

```
# ljestvica najpopularinijih žanrova u NA
mean_po_zanrovima <- aggregate(vgsales_p3["NA_Sales"], vgsales_p3["Genre"], mean)
poredani_meanovi <- mean_po_zanrovima[order(-mean_po_zanrovima$NA_Sales),]

barplot(poredani_meanovi$NA_Sales,
        main = "average sales by genre",
        ylab = "million copies",
        names.arg = poredani_meanovi$Genre,
        las = 2)
```



Vidimo da su mogući kandidati Platform i Shooter.

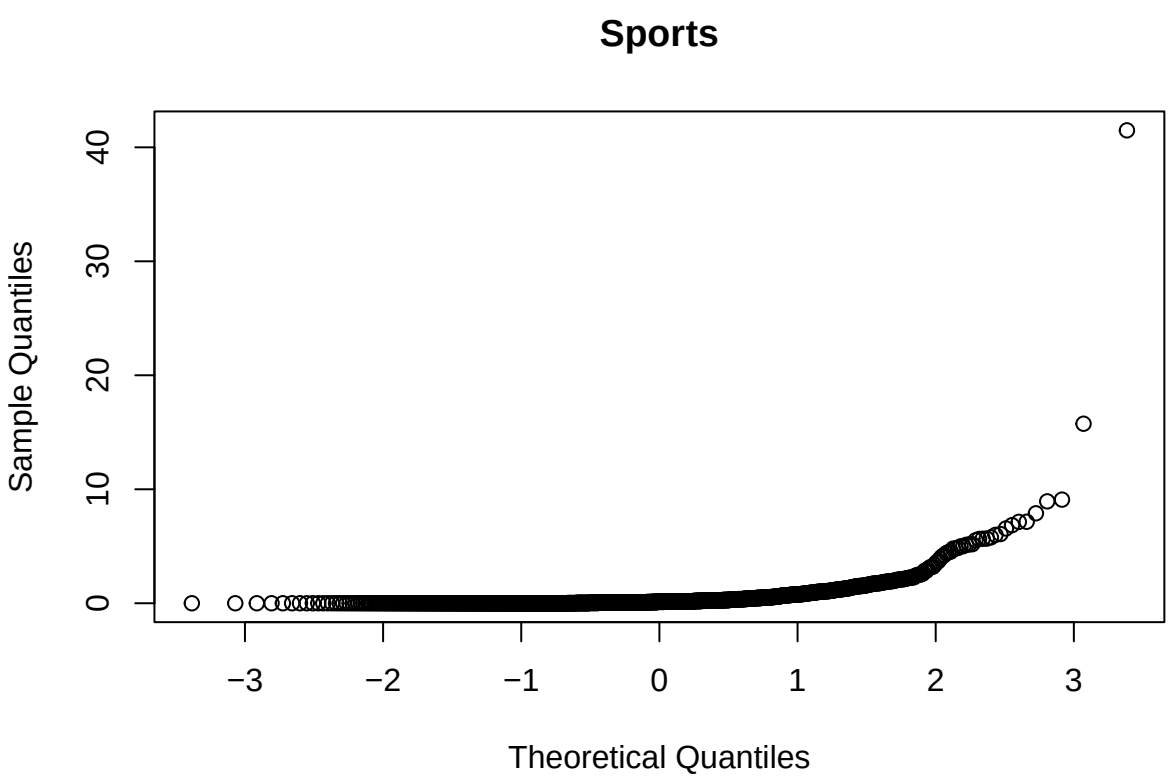
Zbog pretpostavke ANOVA-e trebamo provjeriti normalnost populacija i homogenost varijanci. Odnosno želimo provjeriti:

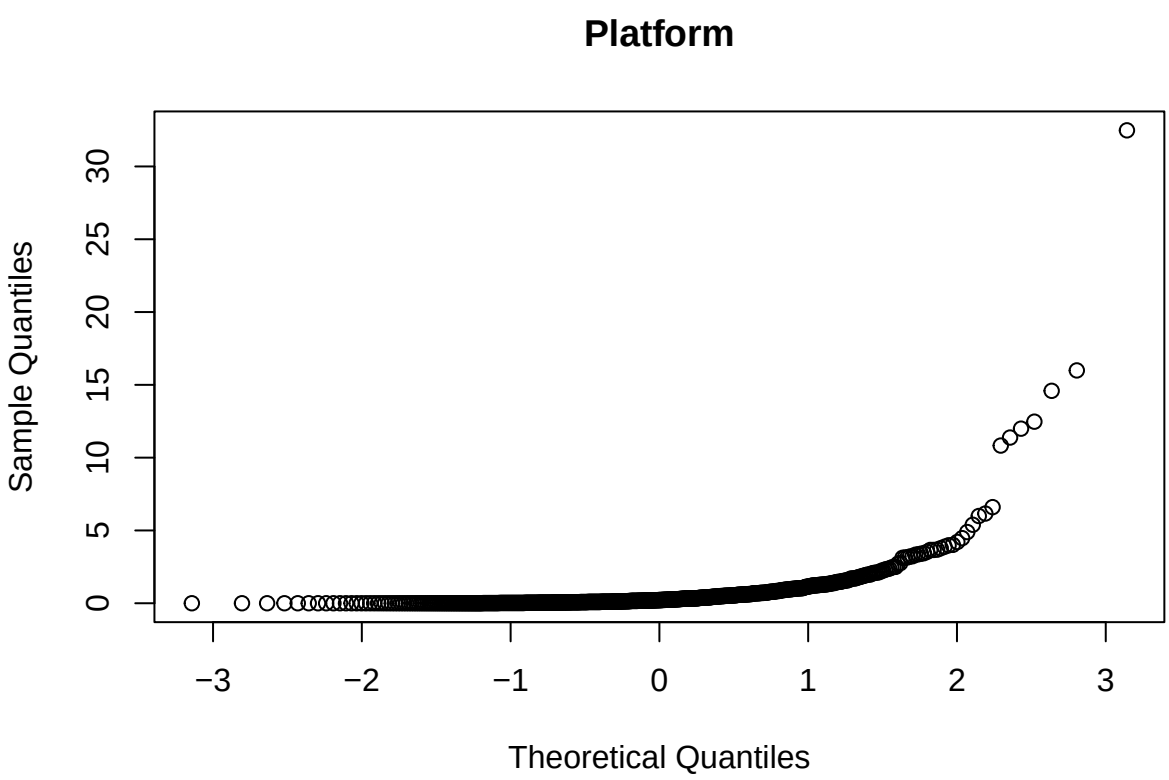
$$\mathbb{X}_i \sim \mathcal{N}, \quad i \in [1, n]$$
$$\sigma_1^2 = \sigma_2^2 = ... = \sigma_n^2$$

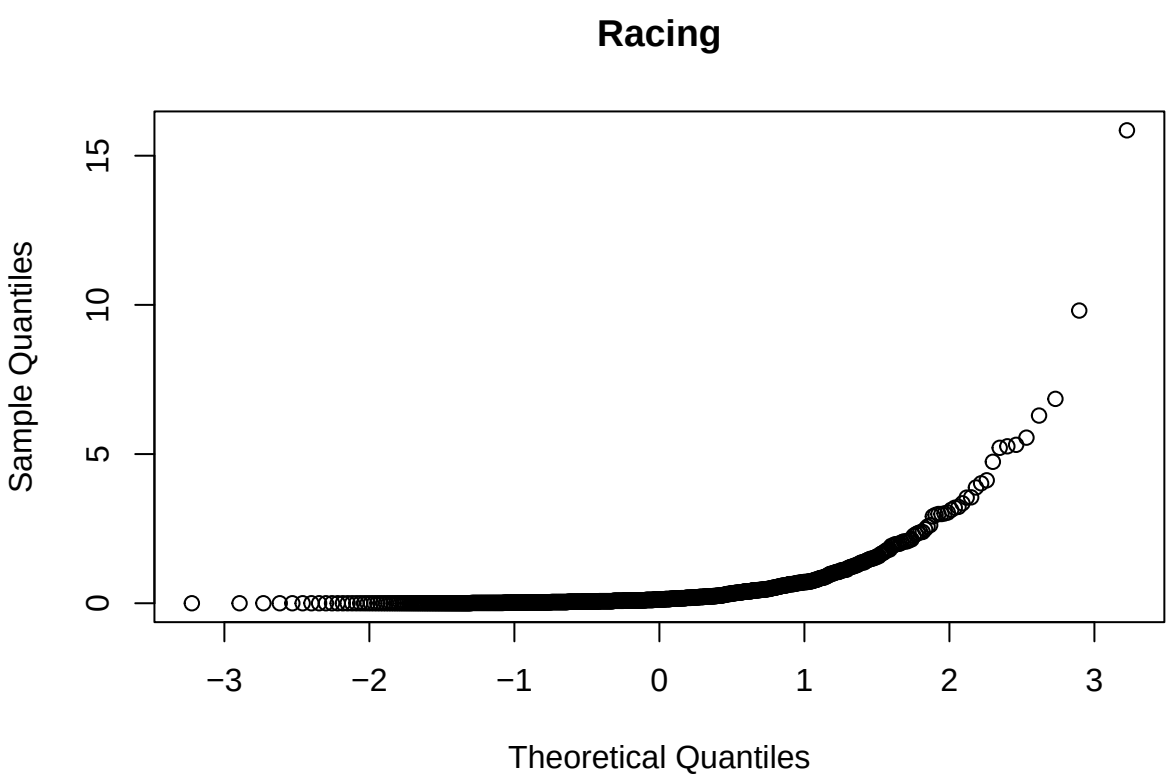
Gdje je 'X_i' populacija broja prodanih kopija žanra i, a 'n' ukupan broj žanrova.

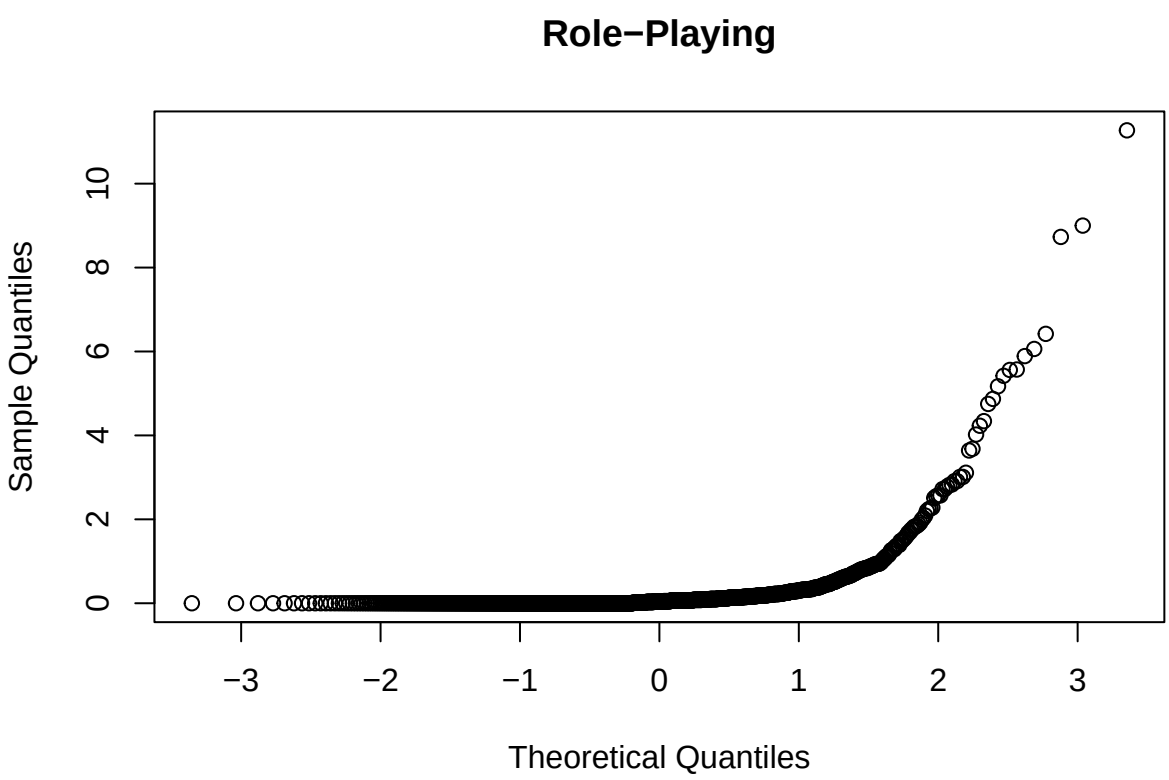
Za provjeru normalnosti vizualiziramo populacije QQ-plotom:

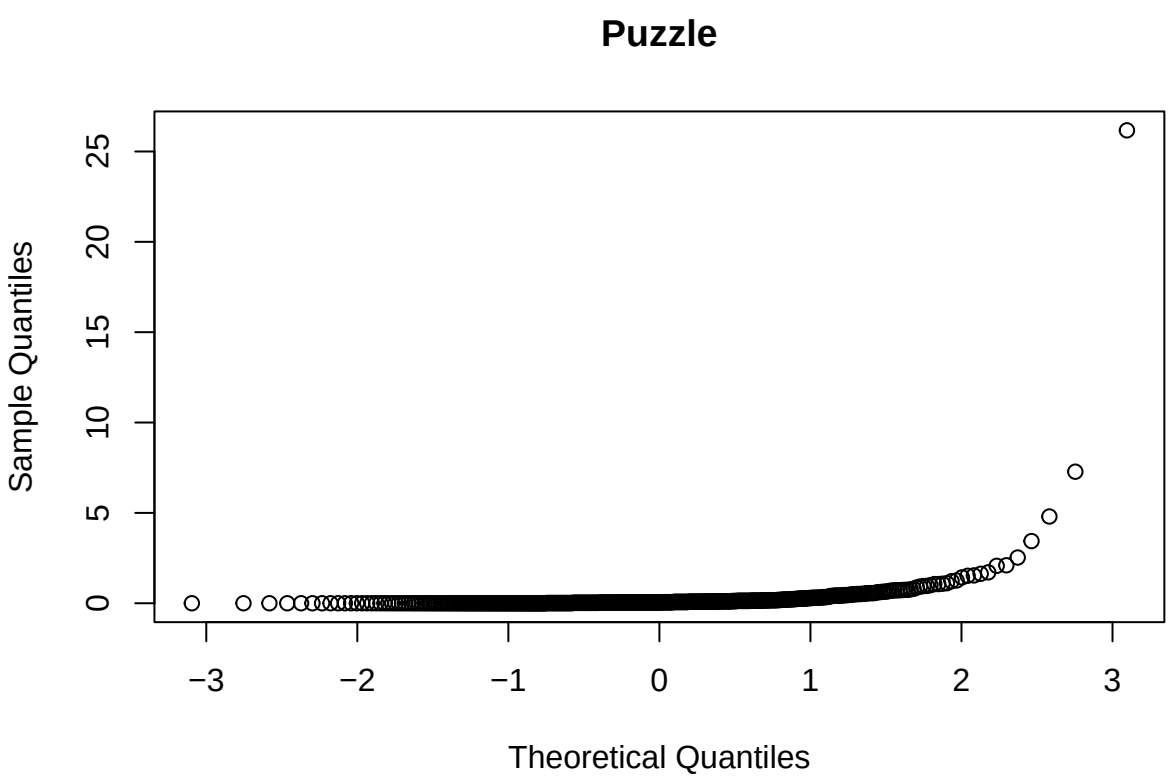
```
for (zanr in svi_zanrovi) {
  qqnorm(vgsales_p3[vgsales_p3$Genre == zanr,]$NA_Sales, main=zanr)
}
```

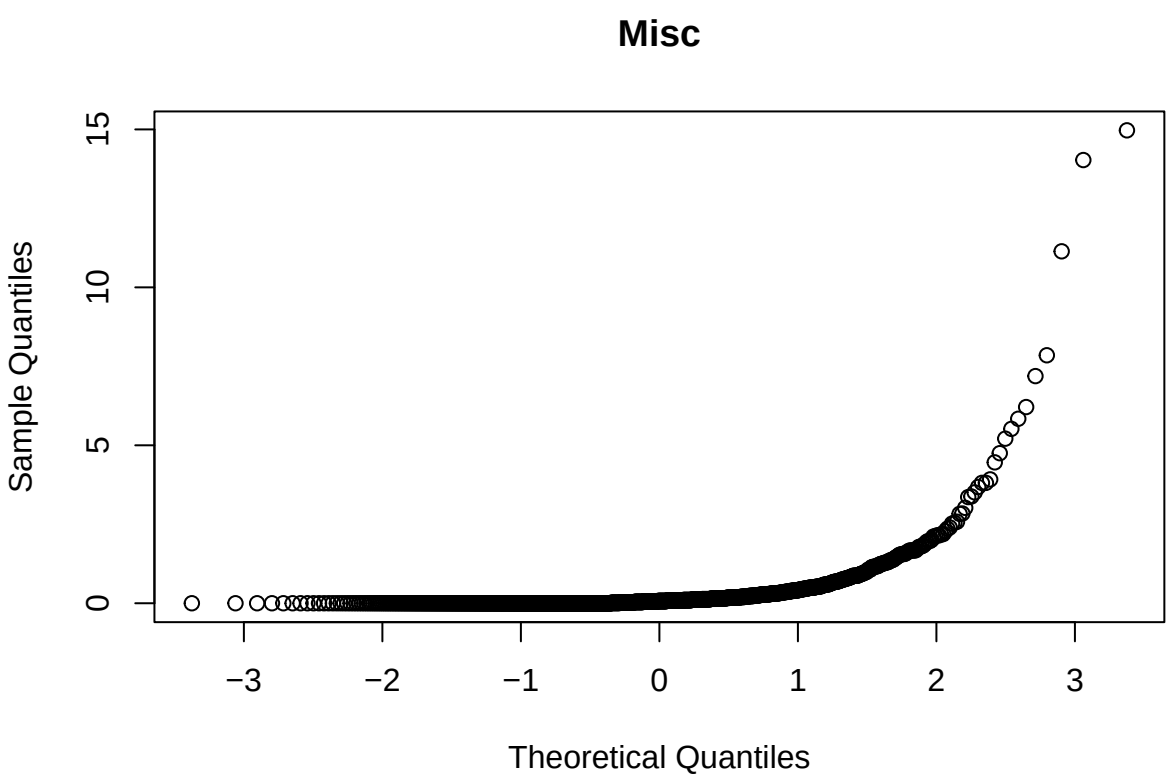

Sports
Sample Quantiles
0
10
20
30
40
−3
−2
−1
0
1
2
3
Theoretical Quantiles

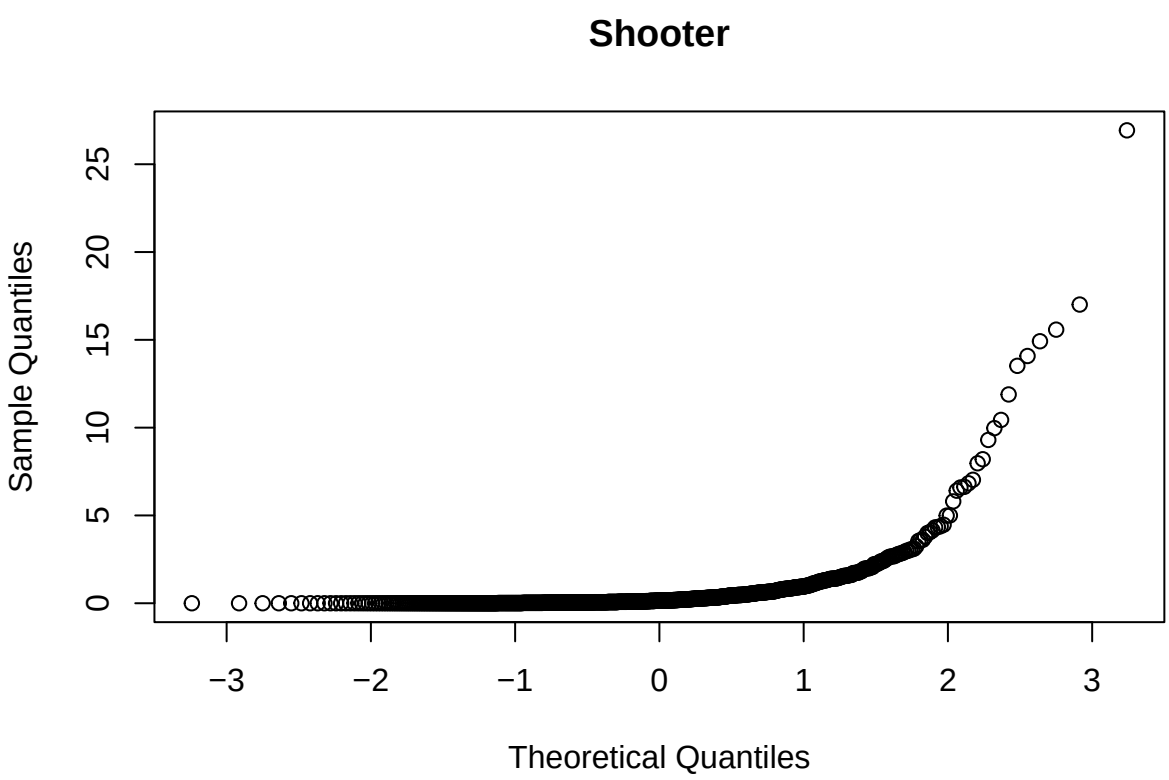

Platform
Sample Quantiles
0
5
10
15
20
25
30
−3
−2
−1
0
1
2
3
Theoretical Quantiles

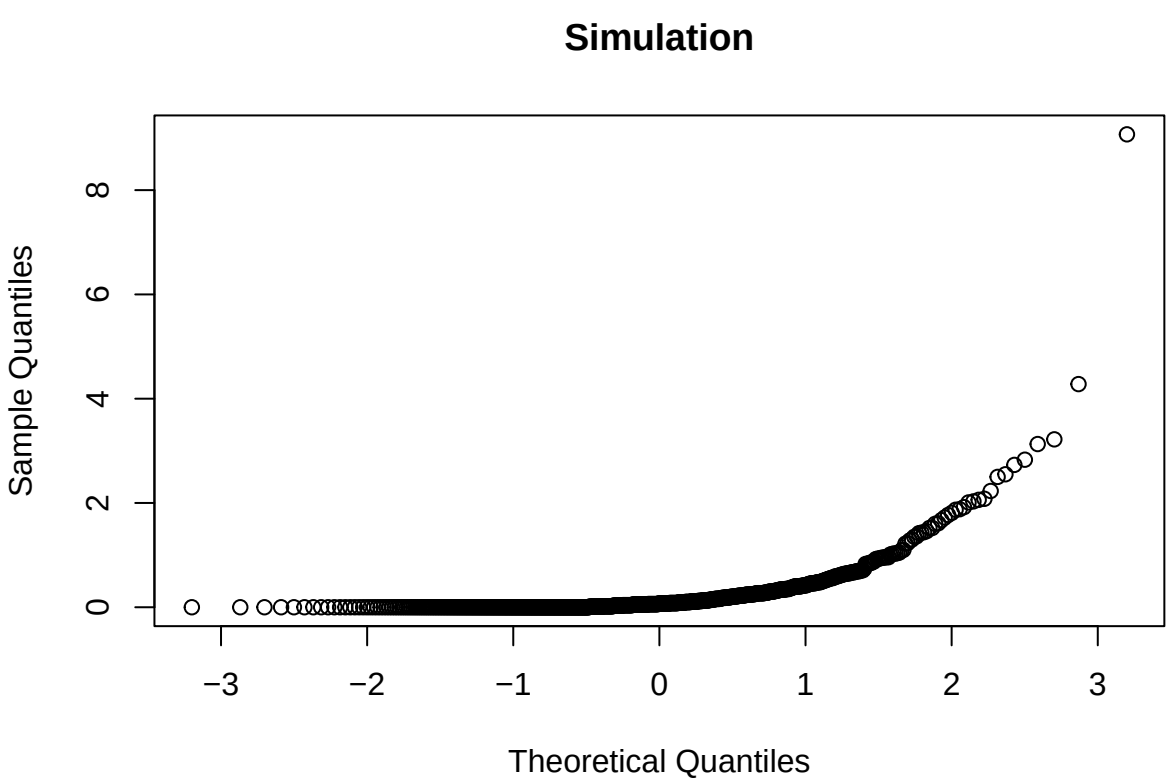

Racing
Sample Quantiles
0
5
10
15
−3
−2
−1
0
1
2
3
Theoretical Quantiles

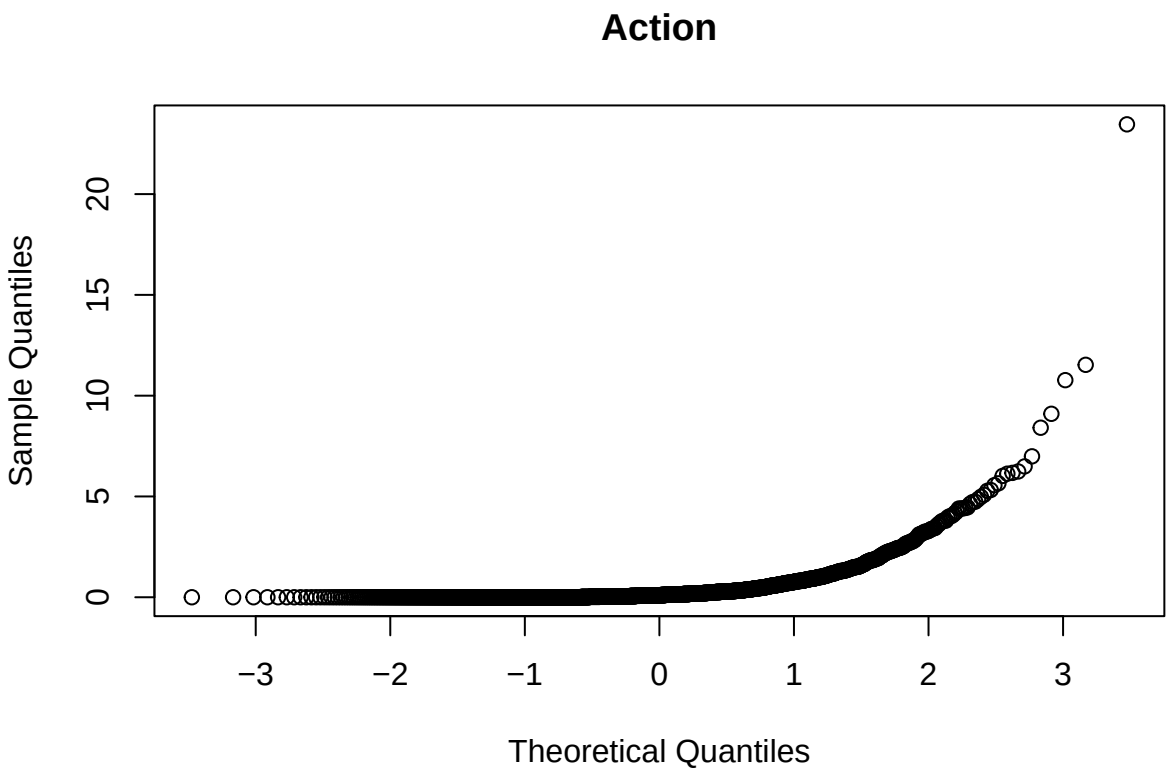

Role−Playing
Sample Quantiles
0
2
4
6
8
10
−3
−2
−1
0
1
2
3
Theoretical Quantiles

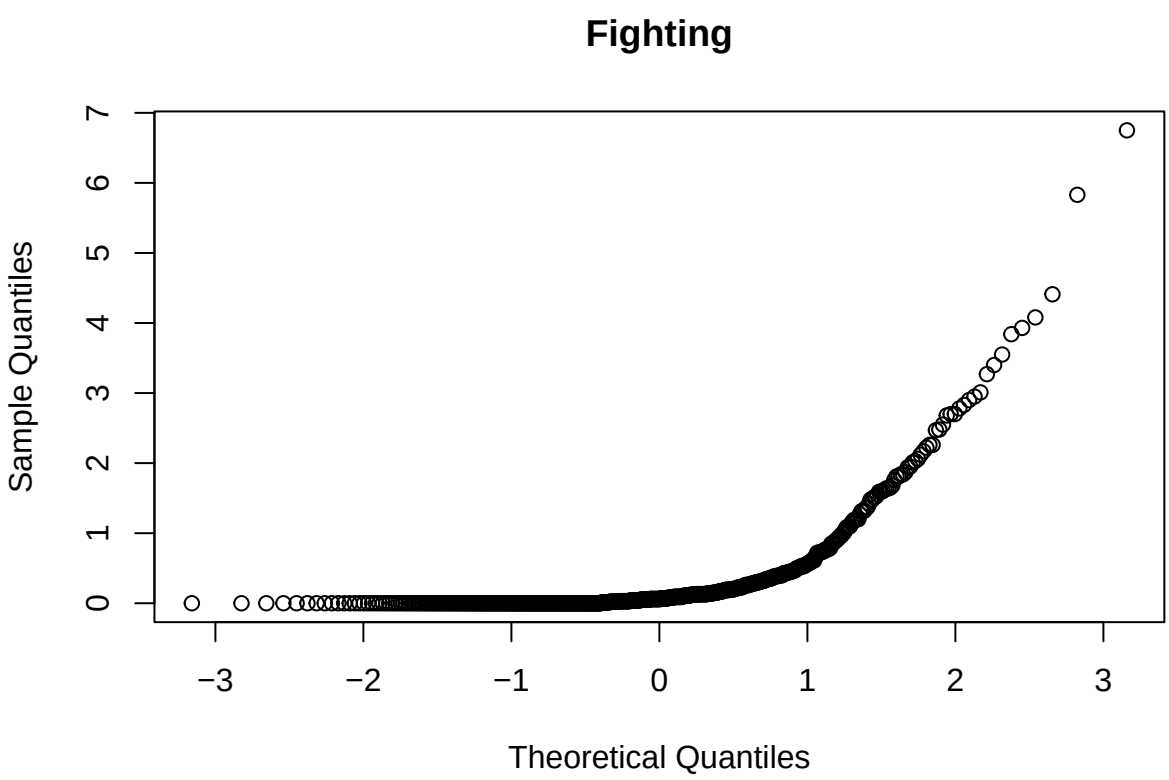

Puzzle
Sample Quantiles
0
5
10
15
20
25
−3
−2
−1
0
1
2
3
Theoretical Quantiles

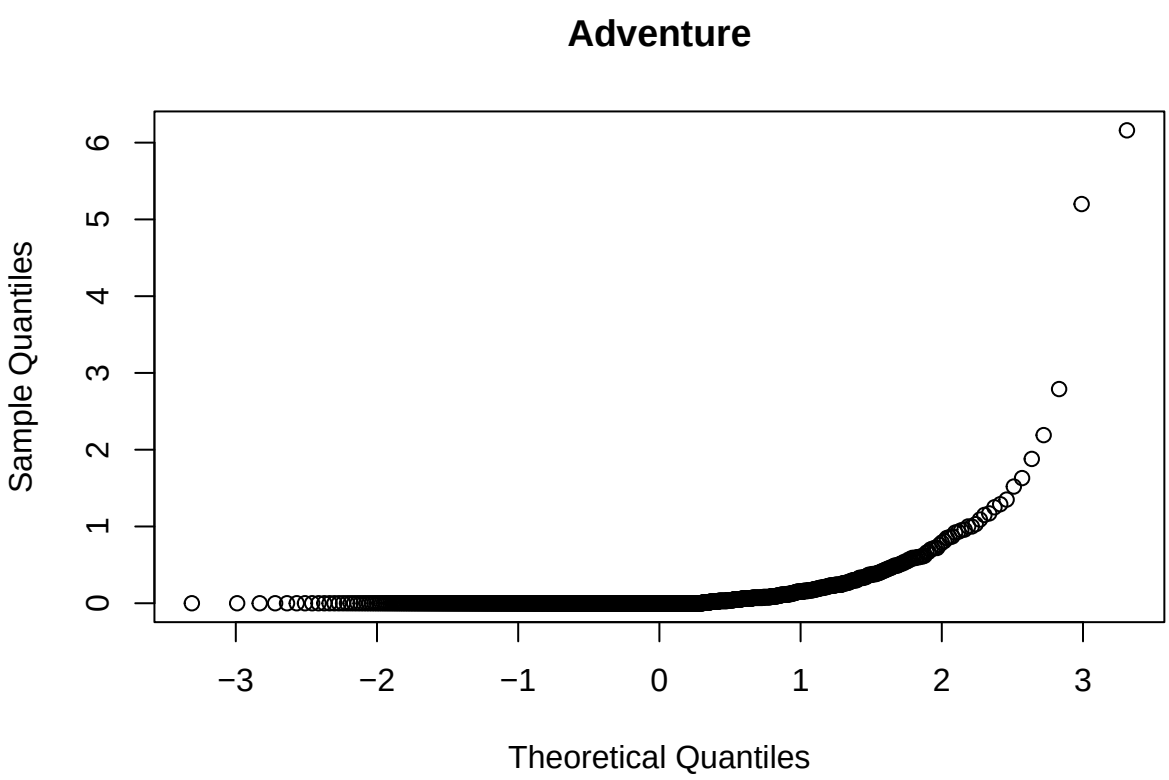

Misc
Sample Quantiles
Theoretical Quantiles
0
5
10
15
−3
−2
−1
0
1
2
3

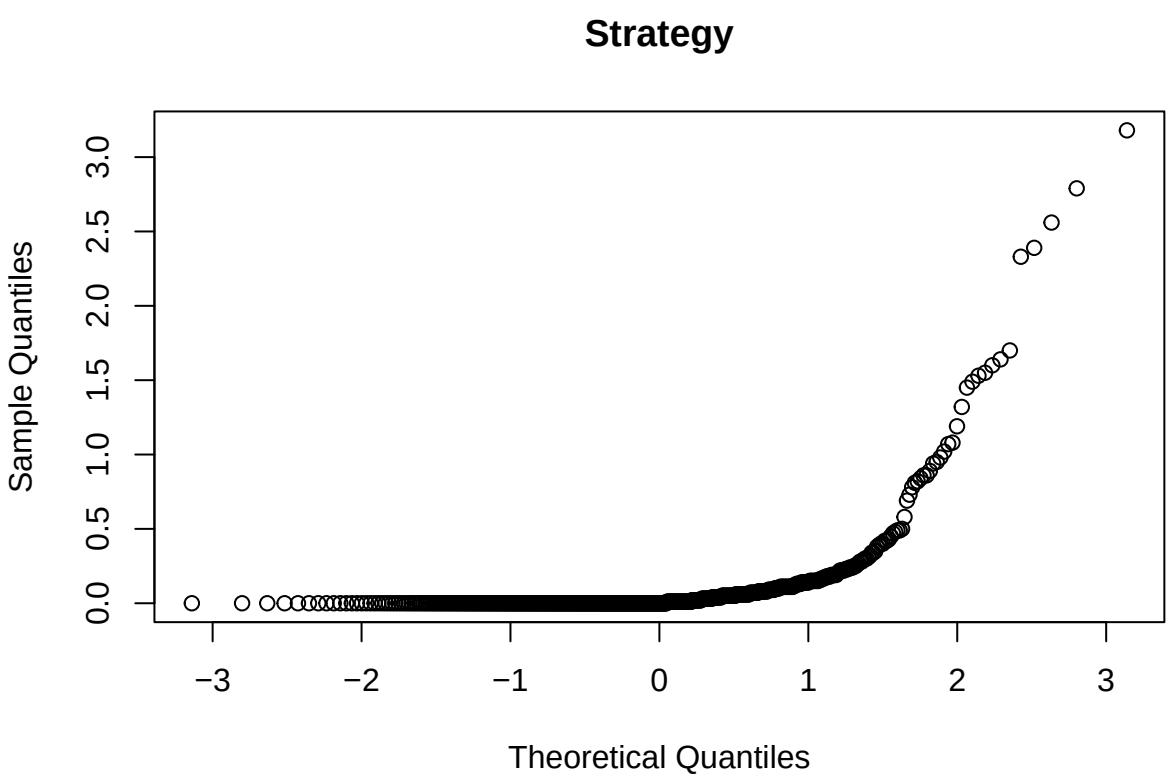

Shooter
Sample Quantiles
0
5
10
15
20
25
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Simulation
Sample Quantiles
0
2
4
6
8
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Action
Sample Quantiles
0
5
10
15
20
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Fighting
Sample Quantiles
0
1
2
3
4
5
6
7
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Adventure
Sample Quantiles
0
1
2
3
4
5
6
−3
−2
−1
0
1
2
3
Theoretical Quantiles

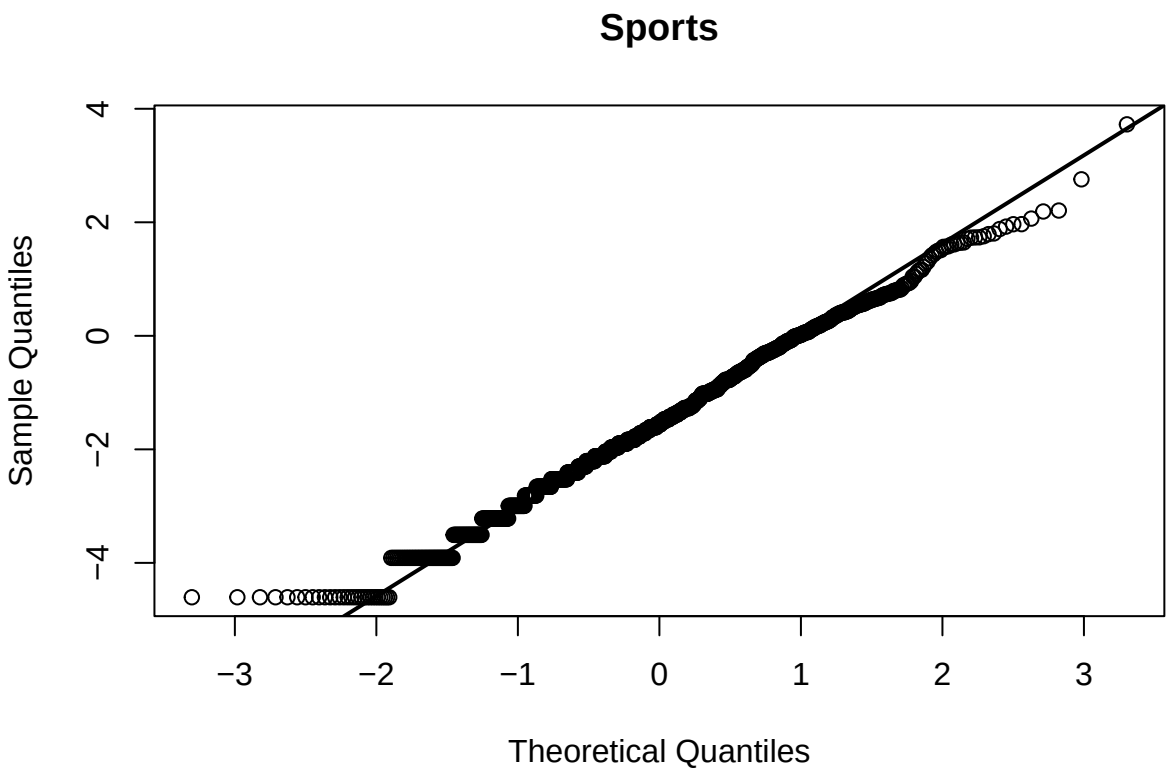


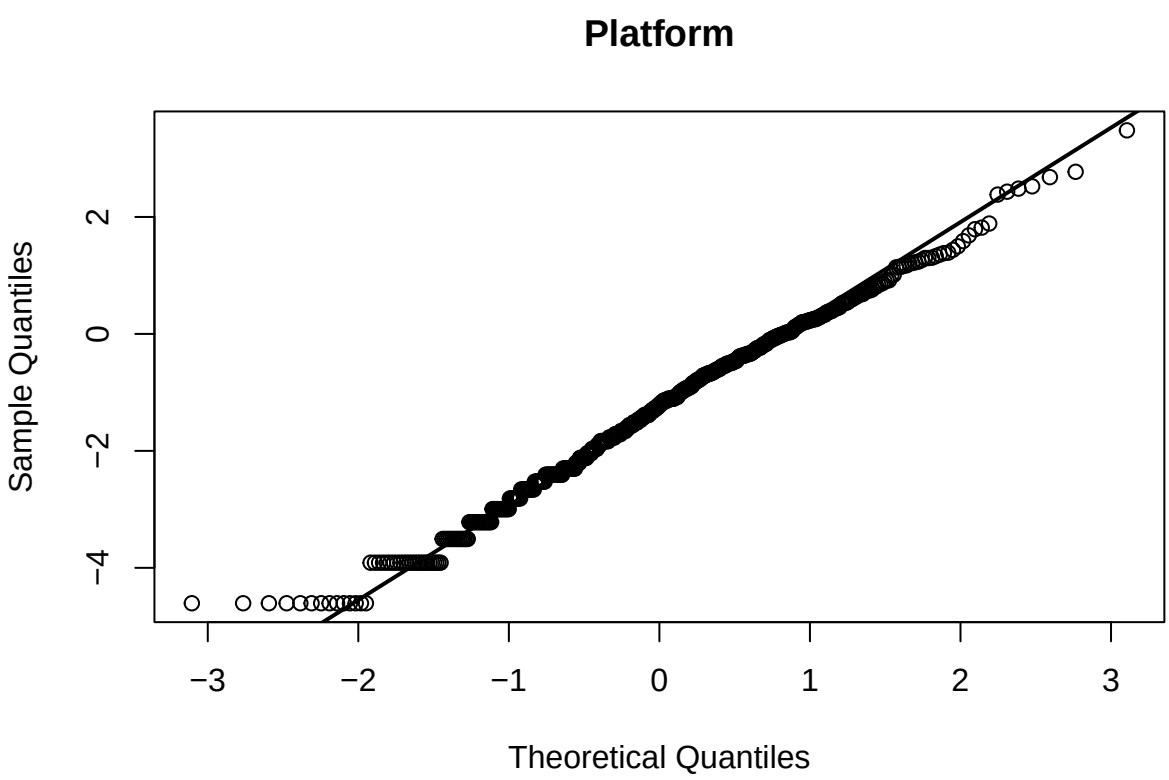
Iz QQ-plota možemo vidjeti da populacije ne izgledaju normalno.

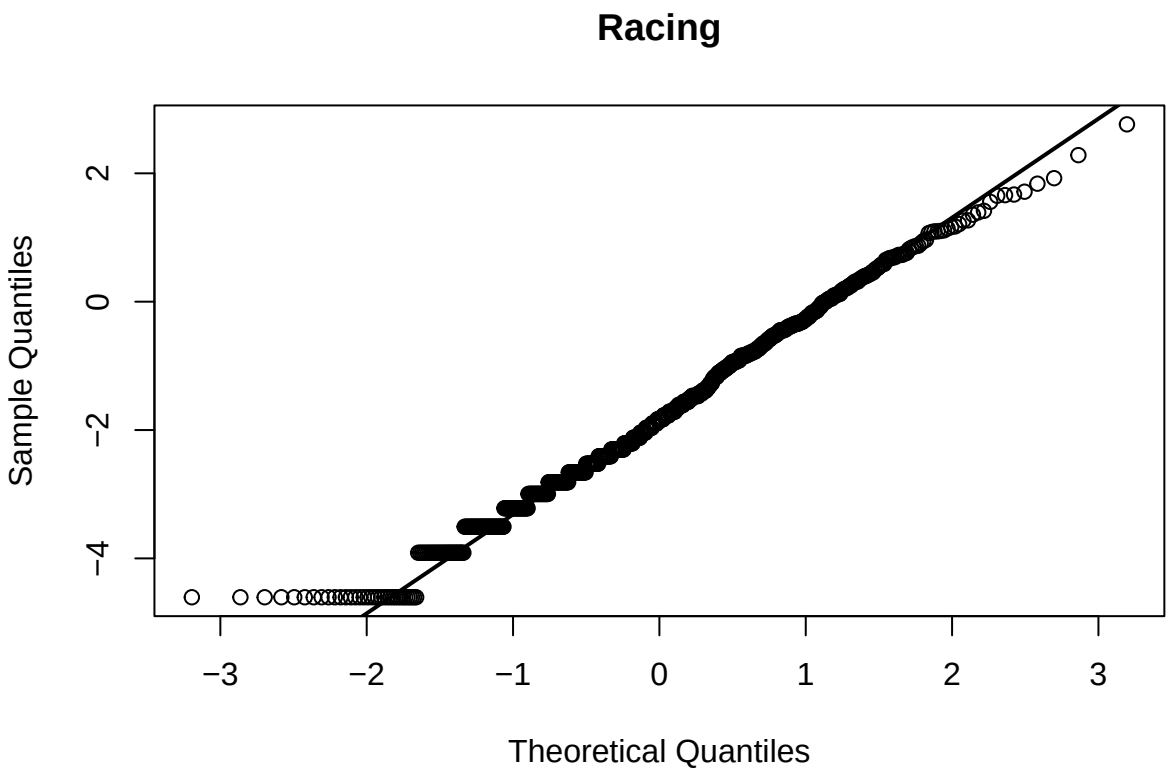
Za svrhe daljnje vizualizacije mičemo retke s NA_Sales == 0. Razlog tomu je što je preciznost podatka 2 decimale, pa su *sve* prodaje ispod 0.005 milijuna kopija obilježene s 0. Zbog toga je odnos x i y u podacima u intervalu [0, 0.005] gotovo sigurno kriv, i vrlo lako može dati krivu sliku.

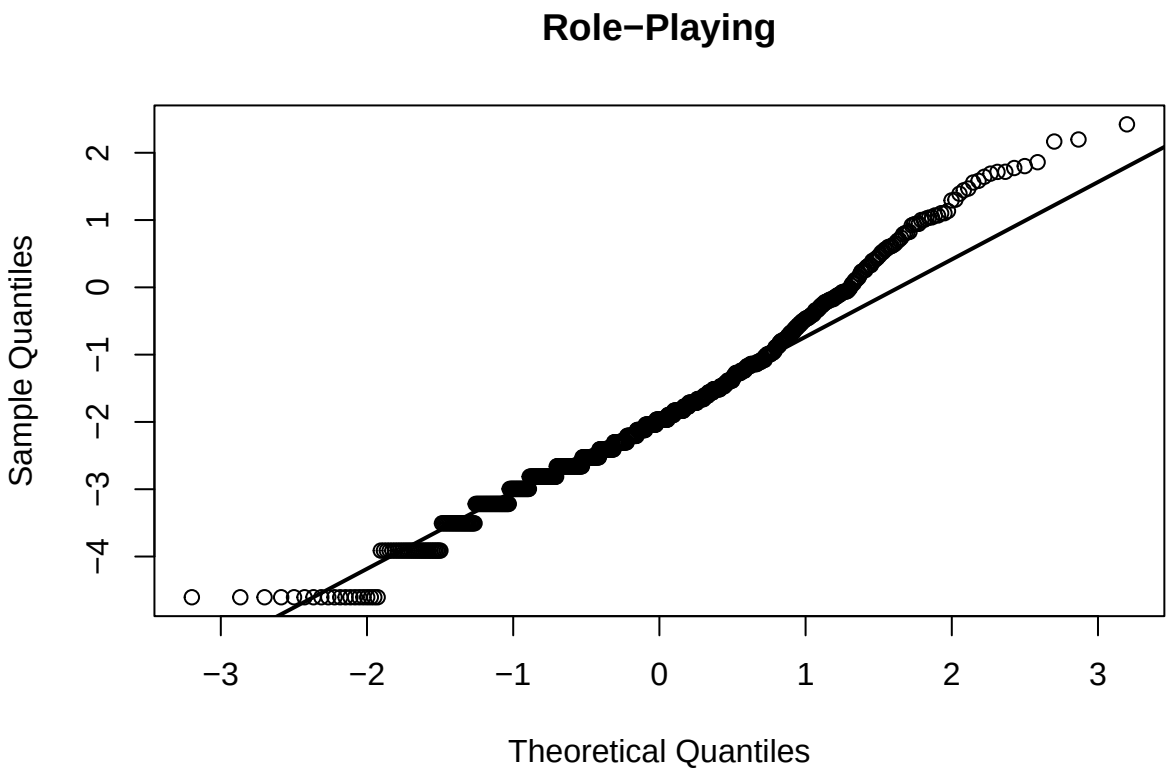
Ako transformacijom uspostavimo eksponencijalni odnos x i y i ponovimo QQ-plot dobivamo:

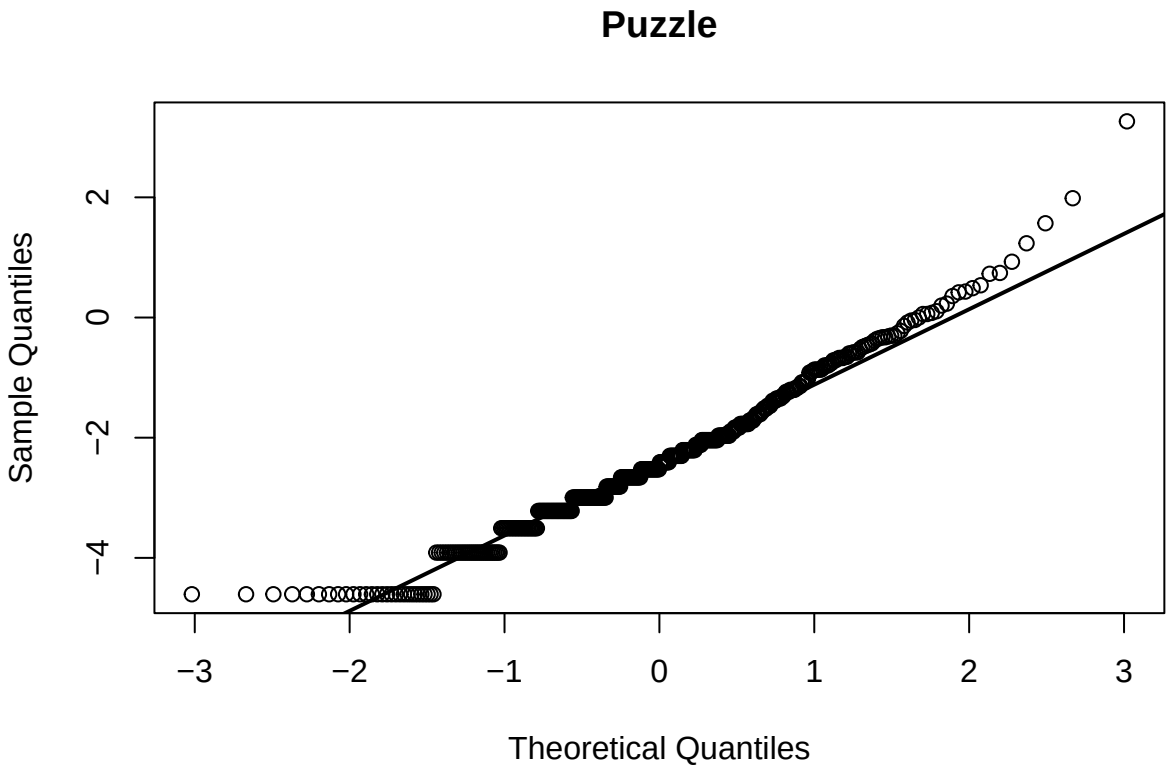
```
# Za svrhe vizualizacije mičemo retke s NA_Sales == 0.
bez_nula <- vgsales_p3[vgsales_p3$NA_Sales != 0,]

for (zanr in svi_zanrovi) {
  qqnorm(log(bez_nula[bez_nula$Genre == zanr,]$NA_Sales), main=zanr)
  qqline(log(bez_nula[bez_nula$Genre == zanr,]$NA_Sales), lwd = 2)
}
```

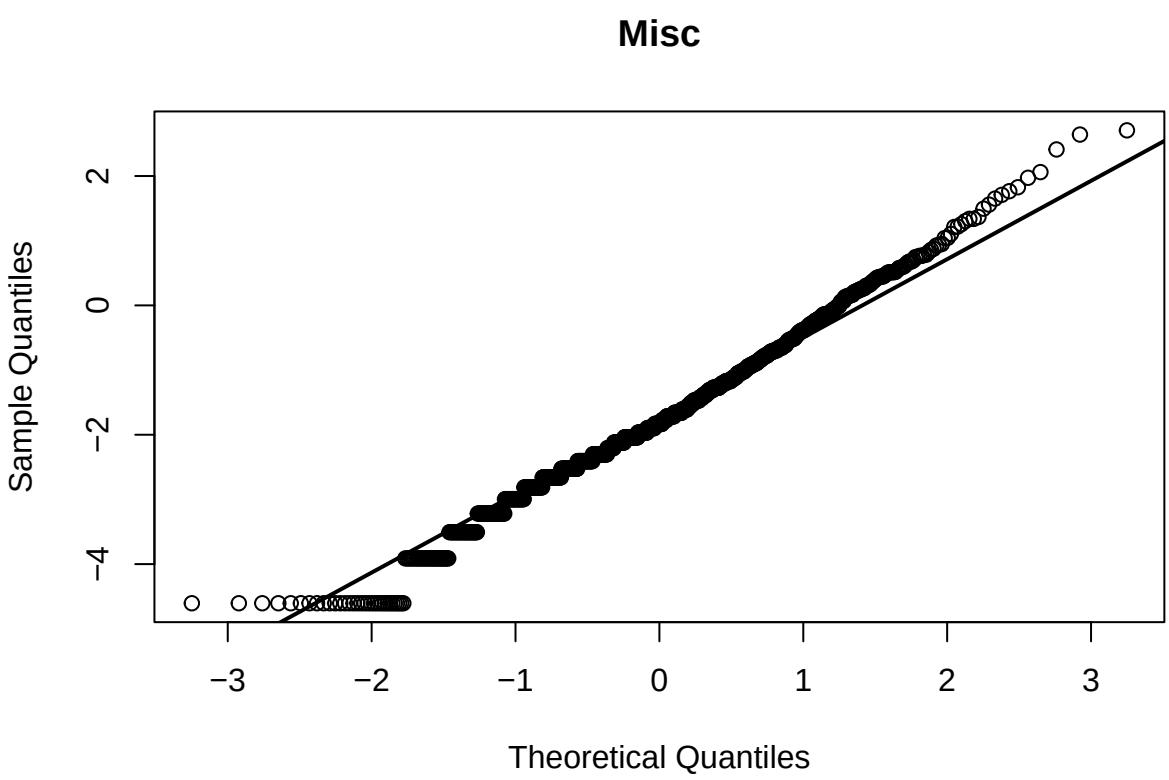

Sports
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4
−3
−2
−1
0
1
2
3

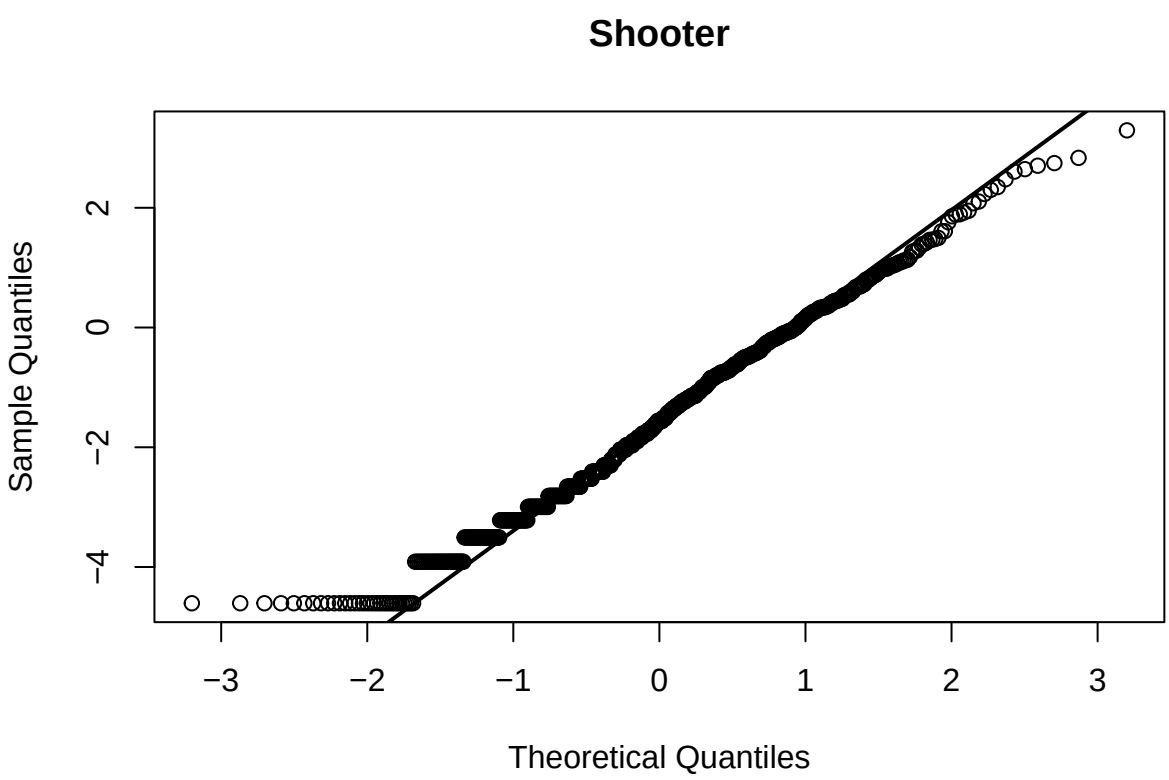

Platform
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
−3
−2
−1
0
1
2
3

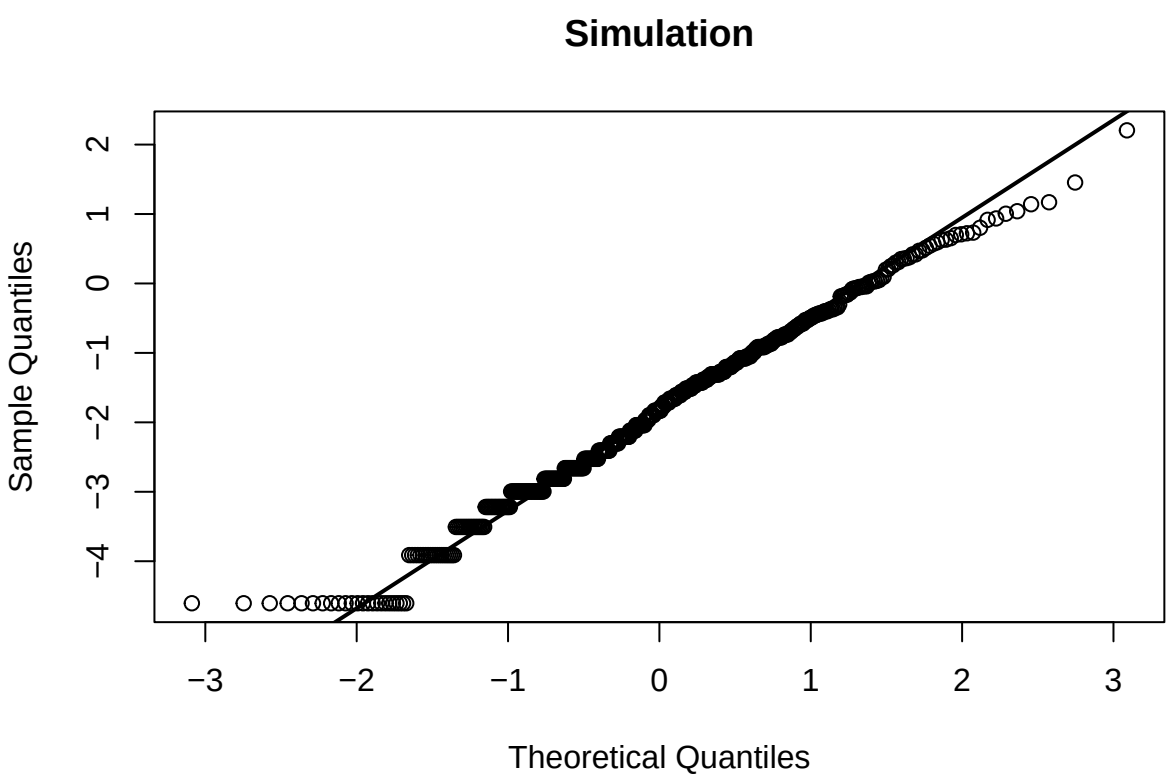

Racing
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−3
−2
−1
0
1
2
3

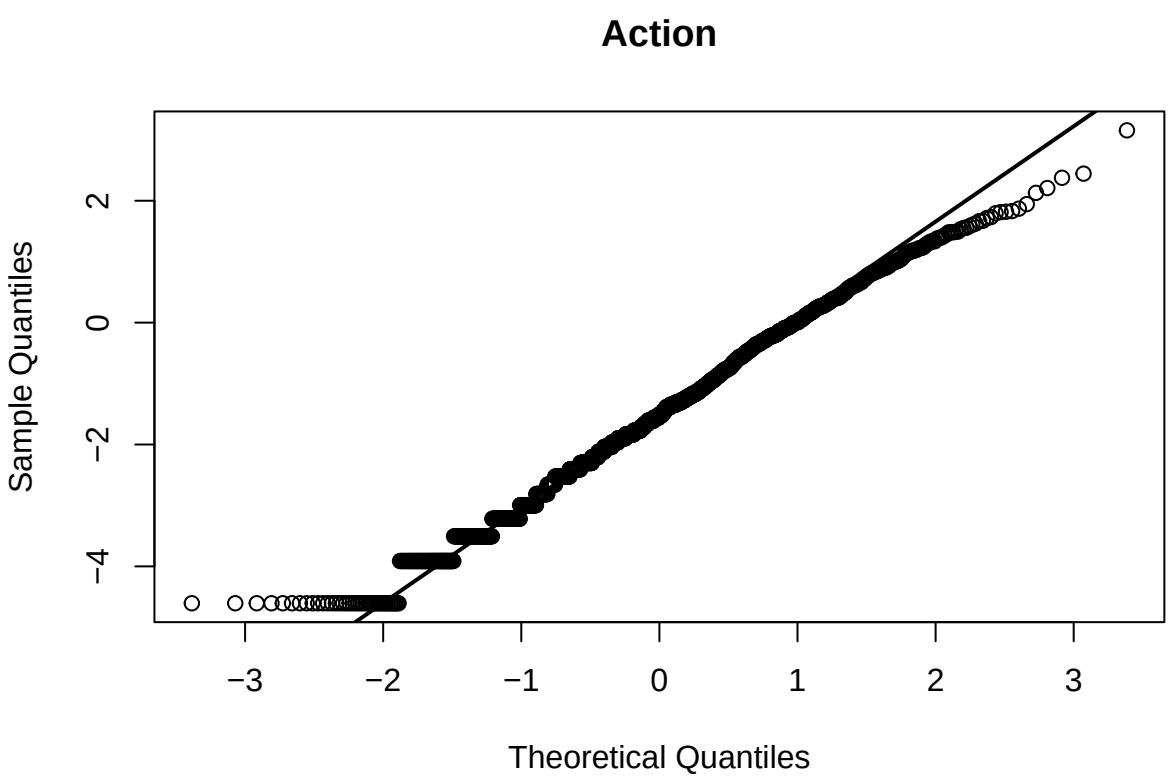

Role−Playing
Sample Quantiles
Theoretical Quantiles
−4
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3

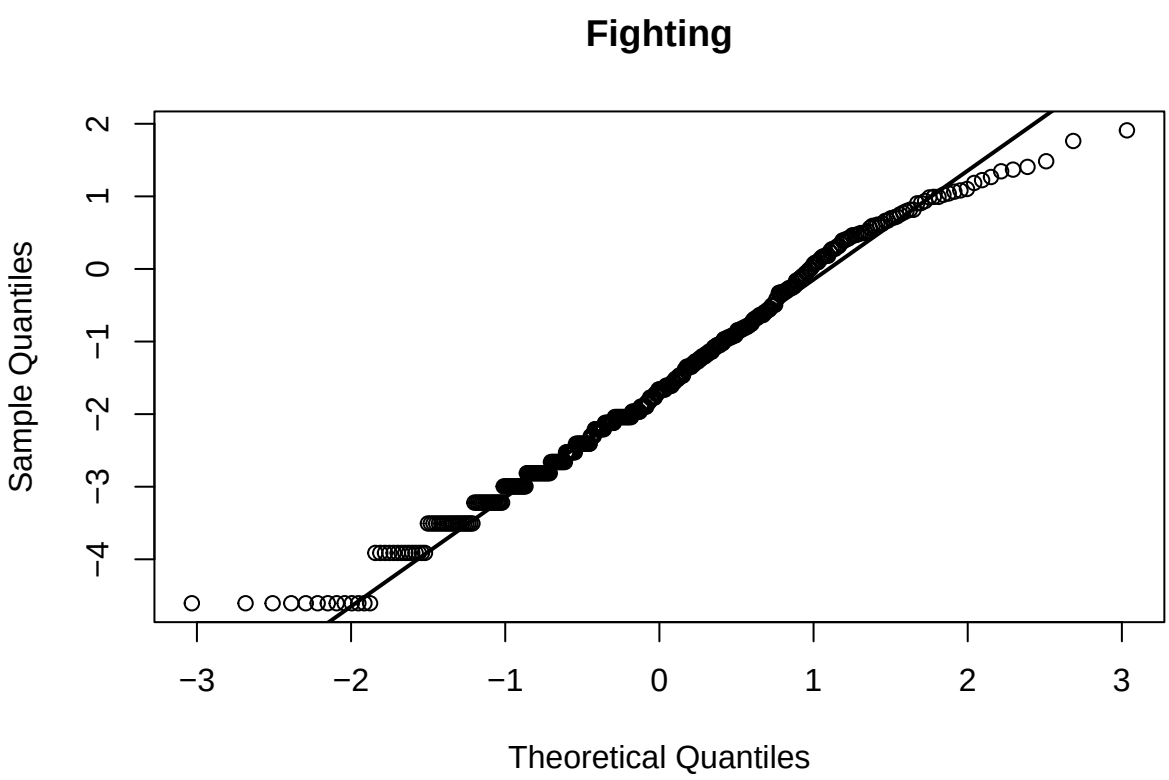

Puzzle
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−3
−2
−1
0
1
2
3

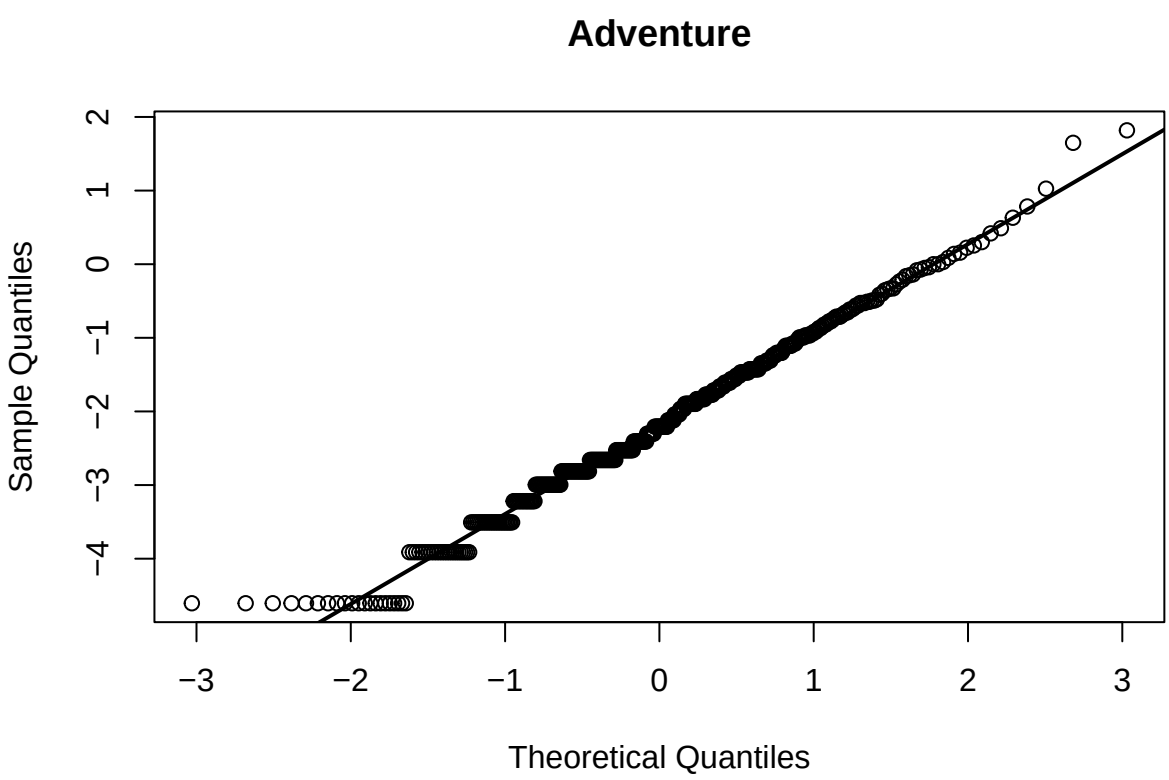

Misc
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−3
−2
−1
0
1
2
3

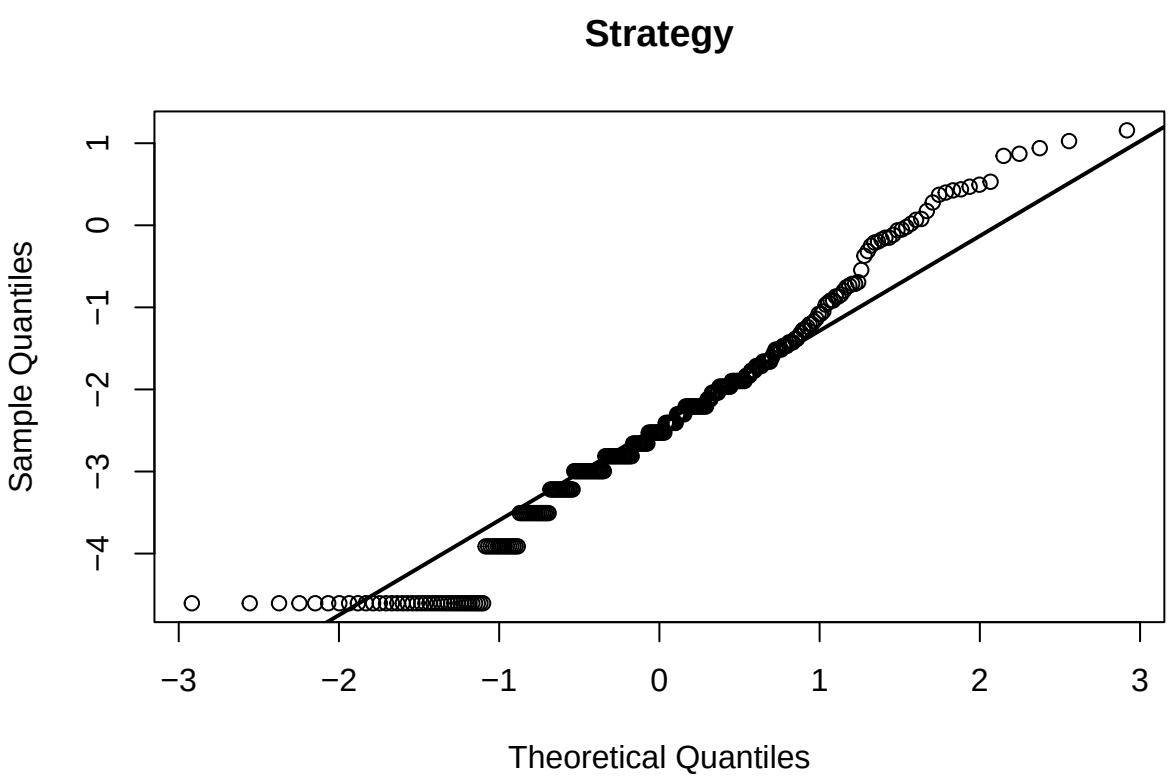

Shooter
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
−3
−2
−1
0
1
2
3


Simulation
Sample Quantiles
Theoretical Quantiles
−4
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3


Action
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−3
−2
−1
0
1
2
3


Fighting
Sample Quantiles
Theoretical Quantiles
−4
−3
−2
−1
0
1
2
−3
−2
−1
0
1
2
3


Adventure
Sample Quantiles
Theoretical Quantiles
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3



Ovi QQ-plotovi izgledaju puno normalnije.

Na lijevom repu imamo 'stepeničast' rast zbog (kao i prethodno spomenute) nedostatne preciznosti podatka NA_Sales. To donekle daje lažnu sliku, no veći dio problema smo riješili odbacivanjem vrijednosti s NA_Sales == 0 iz vizualizacije.

Prije korištenja ANOVA-e trebamo još provjeriti homogenost varijance. Kako imamo više od 2 populacije, ne možemo koristiti f-test. Zato koristimo Bartlettov test.

Testiramo hipotezu:

$$H_0 : \sigma_1^2 = \sigma_2^2 = ... = \sigma_n^2$$
$$H_1 : \neg H_0$$

```
bartlett.test(log(NA_Sales) ~ Genre, bez_nula)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  log(NA_Sales) by Genre
## Bartlett's K-squared = 73.525, df = 11, p-value = 2.599e-11
```

```
for (zanr in svi_zanrovi) {
  print(zanr)
  subset <- bez_nula[bez_nula$Genre == zanr,]$NA_Sales
  print(var(log(subset)))
}
```

```
## [1] "Sports"
## [1] 2.093295
## [1] "Platform"
## [1] 2.339982
## [1] "Racing"
## [1] 2.157539
## [1] "Role-Playing"
## [1] 1.784694
## [1] "Puzzle"
## [1] 1.759668
## [1] "Misc"
## [1] 1.797986
## [1] "Shooter"
## [1] 2.630241
## [1] "Simulation"
## [1] 1.767986
## [1] "Action"
## [1] 2.128215
## [1] "Fighting"
## [1] 2.08947
## [1] "Adventure"
## [1] 1.496922
## [1] "Strategy"
## [1] 1.929365
```

Odbacujemo H_0. Vidimo da *nemamo* homogenost varijanci. Zbog toga ne možemo primijeniti klasičnu ANOVA-u.

Ipak, kao što je i napravljeno na auditornim vježbama, nastavljamo s analizom iako nismo zadovoljili pretpostavku ANOVA-e.

U ovom slučaju bi inače koristili drugi pristup koji ne zahtijeva homogenost varijanci. Jedan takav pristup je prikazan za svrhe usporedbe pri kraju poglavlja - Welch-ova ANOVA. Usporedbom tog rezultata s analizom koja slijedi moći ćemo vidjeti da je kršenje pretpostavke napravilo malu razliku u donesenom zaključku.

Želimo provesti ANOVA-u na punom skupu podataka, no problematične su nam vrijednosti y = 0 (zbog transformacije log(y)). Odbacivanje tih vrijednosti ne bi bilo dobro rješenje problema, jer nerazmjerno utječe na neke žanrove kao što je prikazano ovdje:

```
samo_nule <- vgsales_p3[vgsales_p3$NA_Sales == 0, ]
n_zanr <- setNames(aggregate(Genre ~ samo_nule$Genre, data = samo_nule, FUN = length), c("Genre", "N"))
n_zanr_sve <- setNames(aggregate(Genre ~ vgsales_p3$Genre, data = vgsales_p3, FUN = length), c("Genre",

n_zanr
```

```
##           Genre   N
## 1        Action 548
## 2     Adventure 668
## 3      Fighting 219
## 4          Misc 493
## 5      Platform  70
## 6        Puzzle 115
## 7        Racing  77
## 8  Role-Playing 531
## 9       Shooter 110
## 10   Simulation 229
## 11       Sports 353
```

```
## 12     Strategy 307
```

```
udio_nula <- data.frame(Zanr=character(), Udio=double(), stringsAsFactors=FALSE)

for (zanr in svi_zanrovi) {
  p <- n_zanr[n_zanr$Genre == zanr, ]$N/n_zanr_sve[n_zanr_sve$Genre == zanr, ]$N
  udio_nula <- rbind(udio_nula, data.frame(Zanr=zanr, Udio=p))
}

udio_nula[order(-udio_nula$Udio),]
```

```
##            Zanr       Udio
## 11    Adventure 0.62024141
## 12     Strategy 0.51945854
## 4  Role-Playing 0.42344498
## 6          Misc 0.36303387
## 10     Fighting 0.34651899
## 8    Simulation 0.31499312
## 9        Action 0.28016360
## 1        Sports 0.25178317
## 5        Puzzle 0.22593320
## 7       Shooter 0.13095238
## 2      Platform 0.11725293
## 3        Racing 0.09722222
```

Vidimo da postoji velika razlika u udjelu igrica s 0 prodaja među žanrovima. Od čak 62.02% za 'Adventure' do ispod 10% (točnije 9.72%) za 'Racing'.

Zbog toga koristimo drugi pristup. Dodajemo malu epsilon vrijednost svim prodajama. Vrijednost je izabrana takva da bude upola manja od preciznosti podatka, odnosno 0.01/2 (0.005).

ANOVA testira hipotezu:

$$H_0 : \mu_1 = \mu_2 = ... = \mu_k$$
$$H_1 : \neg H_0$$

```
eps <- 0.005
rezultat_aov <- aov(log(NA_Sales + eps) ~ Genre, data = vgsales_p3)
summary(rezultat_aov)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Genre          11   4845   440.4     116 <2e-16 ***
## Residuals   11723  44498     3.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Odbacujemo H_0. Vidimo da zaista postoji značajna razlika u popularnosti žanrova u NA, no ANOVA nam ne kaže točno *koji* su žanrovi popularniji od drugih.

Za to trebamo provesti parne usporedbe srednjih vrijednosti žanrova (svih 11*10/2 = 55) i pronaći one žanrove koji imaju značajno *veću* srednju vrijednost od ostalih žanrova.

Za to možemo koristiti Tukey-ov test [Walpole, 13.6 - p.546]:

```
tukey <- TukeyHSD(rezultat_aov)
tukey
```

```
##   Tukey multiple comparisons of means
```

```
##     95% family-wise confidence level
##
## Fit: aov(formula = log(NA_Sales + eps) ~ Genre, data = vgsales_p3)
##
## $Genre
##                                diff         lwr           upr     p adj
## Adventure-Action        -1.55466467 -1.79630322 -1.3130261177 0.0000000
## Fighting-Action         -0.32219149 -0.61357265 -0.0308103290 0.0158867
## Misc-Action             -0.46513331 -0.69007204 -0.2401945809 0.0000000
## Platform-Action          0.80440040  0.50663371  1.1021670882 0.0000000
## Puzzle-Action           -0.41028476 -0.72715937 -0.0934101551 0.0014001
## Racing-Action            0.47473326  0.20651791  0.7429486232 0.0000005
## Role-Playing-Action     -0.74427429 -0.97465262 -0.5138959675 0.0000000
## Shooter-Action           0.54383542  0.28113134  0.8065394928 0.0000000
## Simulation-Action       -0.36592795 -0.64254631 -0.0893095896 0.0009393
## Sports-Action            0.10920541 -0.11364027  0.3320510809 0.9088839
## Strategy-Action         -1.34804685 -1.64696935 -1.0491243525 0.0000000
## Fighting-Adventure       1.23247318  0.91337308  1.5515732772 0.0000000
## Misc-Adventure           1.08953136  0.82968655  1.3493761655 0.0000000
## Platform-Adventure       2.35906507  2.03412370  2.6840064363 0.0000000
## Puzzle-Adventure         1.14437991  0.80184318  1.4869166321 0.0000000
## Racing-Adventure         2.02939793  1.73130169  2.3274941768 0.0000000
## Role-Playing-Adventure   0.81039038  0.54582268  1.0749580696 0.0000000
## Shooter-Adventure        2.09850009  1.80535281  2.3916473580 0.0000000
## Simulation-Adventure     1.18873672  0.88305779  1.4944156503 0.0000000
## Sports-Adventure         1.66387008  1.40583503  1.9219051239 0.0000000
## Strategy-Adventure       0.20661782 -0.11938303  0.5326186643 0.6432468
## Misc-Fighting           -0.14294182 -0.44959001  0.1637063705 0.9343697
## Platform-Fighting        1.12659189  0.76313531  1.4900484639 0.0000000
## Puzzle-Fighting         -0.08809328 -0.46736257  0.2911760213 0.9998313
## Racing-Fighting          0.79692475  0.45725556  1.1365939456 0.0000000
## Role-Playing-Fighting   -0.42208281 -0.73274316 -0.1114224520 0.0005569
## Shooter-Fighting         0.86602691  0.53069257  1.2013612361 0.0000000
## Simulation-Fighting     -0.04373646 -0.39007935  0.3026064311 0.9999997
## Sports-Fighting          0.43139689  0.12628073  0.7365130636 0.0002415
## Strategy-Fighting       -1.02585536 -1.39025945 -0.6614512709 0.0000000
## Platform-Misc            1.26953371  0.95681157  1.5822558462 0.0000000
## Puzzle-Misc              0.05484855 -0.27611920  0.3858162962 0.9999945
## Racing-Misc              0.93986657  0.65513931  1.2245938344 0.0000000
## Role-Playing-Misc       -0.27914098 -0.52854890 -0.0297330706 0.0135711
## Shooter-Misc             1.00896873  0.72942702  1.2885104329 0.0000000
## Simulation-Misc          0.09920536 -0.19345118  0.3918619015 0.9944069
## Sports-Misc              0.57433872  0.33187155  0.8168058865 0.0000000
## Strategy-Misc           -0.88291354 -1.19673641 -0.5690906705 0.0000000
## Puzzle-Platform         -1.21468516 -1.59888201 -0.8304883116 0.0000000
## Racing-Platform         -0.32966713 -0.67482968  0.0154954091 0.0773562
## Role-Playing-Platform   -1.54867469 -1.86533204 -1.2320173445 0.0000000
## Shooter-Platform        -0.26056498 -0.60146252  0.0803325603 0.3408066
## Simulation-Platform     -1.17032835 -1.52206038 -0.8185963198 0.0000000
## Sports-Platform         -0.69519499 -1.00641501 -0.3839749780 0.0000000
## Strategy-Platform       -2.15244725 -2.52197717 -1.7829173282 0.0000000
## Racing-Puzzle            0.88501803  0.52324227  1.2467937853 0.0000000
## Role-Playing-Puzzle     -0.33398953 -0.66867803  0.0006989716 0.0510605
## Shooter-Puzzle           0.95412018  0.59641129  1.3118290726 0.0000000
```

```
## Simulation-Puzzle         0.04435682 -0.32369201  0.4124056384 0.9999998
## Sports-Puzzle             0.51949017  0.18994137  0.8490389744 0.0000169
## Strategy-Puzzle          -0.93776209 -1.32285542 -0.5526687478 0.0000000
## Role-Playing-Racing      -1.21900756 -1.50805142 -0.9299636963 0.0000000
## Shooter-Racing            0.06910215 -0.24631125  0.3845155516 0.9999053
## Simulation-Racing        -0.84066121 -1.16775432 -0.5135681110 0.0000000
## Sports-Racing            -0.36552786 -0.64860448 -0.0824512333 0.0014710
## Strategy-Racing          -1.82278011 -2.16894025 -1.4766199776 0.0000000
## Shooter-Role-Playing      1.28810971  1.00417256  1.5720468627 0.0000000
## Simulation-Role-Playing   0.37834634  0.08148848  0.6752042091 0.0018570
## Sports-Role-Playing       0.85347970  0.60595784  1.1010015568 0.0000000
## Strategy-Role-Playing    -0.60377256 -0.92151700 -0.2860281082 0.0000000
## Simulation-Shooter       -0.90976337 -1.23235266 -0.5871740760 0.0000000
## Sports-Shooter           -0.43463001 -0.71249028 -0.1567697438 0.0000208
## Strategy-Shooter         -1.89188227 -2.23378985 -1.5499746878 0.0000000
## Sports-Simulation         0.47513335  0.18408248  0.7661842326 0.0000063
## Strategy-Simulation      -0.98211890 -1.33482994 -0.6294078607 0.0000000
## Strategy-Sports          -1.45725226 -1.76957830 -1.1449262146 0.0000000
```

Ukoliko pogledamo parove žanrova koje smo prethodno izdvojili kao potencijalne kandidate, naime Platform i Shooter, dobivamo sljedeće:

(koristimo :alpha: = 0.05)

```
# Filtriramo samo parove s Platform ili Shooter
p_foo <- function(x) grepl("Platform", x)
platform_pairs <- Filter(p_foo, attr(tukey$Genre, "dimnames")[[1]])
s_foo <- function(x) grepl("Shooter", x)
shooter_pairs <- Filter(s_foo, attr(tukey$Genre, "dimnames")[[1]])

p_lim <- 0.05

platform_p_values <- c()
for (pair in platform_pairs) {
  platform_p_values <- c(platform_p_values, tukey$Genre[pair,][4])
}

shooter_p_values <- c()
for (pair in shooter_pairs) {
  shooter_p_values <- c(shooter_p_values, tukey$Genre[pair,][4])
}

print(paste("Za žanr 'Platform' ukupno", sum(platform_p_values > p_lim), "para nisu postigli značajno ra
```

```
## [1] "Za žanr 'Platform' ukupno 2 para nisu postigli značajno različit rezultat."
```

```
print(paste("Za žanr 'Shooter' ukupno", sum(shooter_p_values > p_lim), "para nisu postigli značajno razl
```

```
## [1] "Za žanr 'Shooter' ukupno 2 para nisu postigli značajno različit rezultat."
```

Koristeći Tukey-ov test vidimo da je žanr 'Platform' značajno popularniji od 9 drugih žanrova. Značajnost razlike srednjih prodaja nije postignuta jedino za parove 'Shooter-Platform' i 'Racing-Platform'.

Dobivena p-vrijednost pri usporedbi Platform i Racing je 0.0773562 (ova brojka će biti relevantna u dodatku).

Za drugi kandidat ('Shooter') Tukey-ov test također kaže da je značajno popularniji od 9 drugih žanrova. Uz 'Shooter-Platform', značajnost se još ne postiže jedino za par 'Shooter-Racing'.

Dodatak: ANOVA u slučaju nehomogenosti varijance

U ovom dodatku ilustriramo pristup provođenju usporedbe srednjih vrijednosti ukoliko nije zadovoljena pretpostavka homogenosti varijance. Koristi se Welch-ova ANOVA.

Opet testiramo:

$$H_0 : \mu_1 = \mu_2 = ... = \mu_k$$
$$H_1 : \neg H_0$$

Welchovu ANOVA-u provodimo koristeći oneway.test() funkciju, uz parametar var.equal = FALSE:

```
eps <- 0.005
rezultat_aov <- oneway.test(log(NA_Sales + eps) ~ Genre, data = vgsales_p3, var.equal = FALSE)
print(rezultat_aov)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  log(NA_Sales + eps) and Genre
## F = 135.32, num df = 11, denom df = 3842, p-value < 2.2e-16
```

Kao i kod obične ANOVA-e, vidimo da zaista postoji značajna razlika u popularnosti žanrova u NA, no opet ANOVA nam ne kaže točno *koji* su žanrovi popularniji od drugih.

Za to opet trebamo provesti post-hoc test koji radi parne usporedbe srednjih vrijednosti žanrova.

Za post-hoc test ne možemo koristiti Tukeyov test, jer on (isto kao one-way ANOVA) pretpostavlja homogenost varijanci. No možemo koristiti Games-Howell test iz paketa *rstatix*:

```
require(rstatix)
```

```
## Loading required package: rstatix
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
# Transformiramo potpune podatke, samo dodajemo konstantu veličine 1/2 preciznosti
# svim podacima, da izbjegnemo log(0)
transformirani <- data.frame(vgsales_p3)
transformirani$NA_Sales <- log(transformirani$NA_Sales + 0.005)

games_howell <- games_howell_test(transformirani, NA_Sales ~ Genre)
games_howell
```

```
## # A tibble: 66 x 8
##    .y.      group1 group2      estimate conf.low conf.high    p.adj p.adj.signif
##  * <chr>    <chr>  <chr>          <dbl>    <dbl>     <dbl>    <dbl> <chr>
##  1 NA_Sales Action Adventure     -1.55    -1.79   -1.32    0        ****
##  2 NA_Sales Action Fighting      -0.322   -0.638  -0.00675 4   e- 2 *
##  3 NA_Sales Action Misc          -0.465   -0.702  -0.228   8.62e- 9 ****
##  4 NA_Sales Action Platform       0.804    0.506   1.10    1.99e-13 ****
##  5 NA_Sales Action Puzzle        -0.410   -0.700  -0.121   2.41e- 4 ***
##  6 NA_Sales Action Racing         0.475    0.223   0.726   5.3 e- 8 ****
##  7 NA_Sales Action Role-Playi~   -0.744   -0.985  -0.504   0        ****
```

```
##  8 NA_Sales Action Shooter        0.544    0.276   0.812   2.79e- 9 ****
##  9 NA_Sales Action Simulation    -0.366   -0.648  -0.0842  1   e- 3 ***
## 10 NA_Sales Action Sports         0.109   -0.128   0.347   9.4 e- 1 ns
## # ... with 56 more rows
```

Ukoliko pogledamo iste parove žanrova kao u pravoj analizi, naime 'Platform' i 'Shooter', dobivamo:

```
p_lim <- 0.05

plat_usporedbe <- games_howell[games_howell$group1 == "Platform" | games_howell$group2 == "Platform",]
shoot_usporedbe <- games_howell[games_howell$group1 == "Shooter" | games_howell$group2 == "Shooter",]

print(paste("Za žanr 'Platform' ukupno", sum(plat_usporedbe$p.adj > p_lim), "par nije postigao značajno
```

```
## [1] "Za žanr 'Platform' ukupno 1 par nije postigao značajno različit rezultat."
```

```
print(paste("Za žanr 'Shooter' ukupno", sum(shoot_usporedbe$p.adj > p_lim), "para nisu postigli značajn
```

```
## [1] "Za žanr 'Shooter' ukupno 2 para nisu postigli značajno različit rezultat."
```

```
plat_usporedbe
```

```
## # A tibble: 11 x 8
##    .y.      group1    group2   estimate conf.low conf.high    p.adj p.adj.signif
##    <chr>    <chr>     <chr>       <dbl>    <dbl>     <dbl>    <dbl> <chr>
##  1 NA_Sales Action    Platform    0.804    0.506   1.10    1.99e-13 ****
##  2 NA_Sales Adventure Platform    2.36     2.05    2.67    8.09e-14 ****
##  3 NA_Sales Fighting  Platform    1.13     0.752   1.50    4.08e-13 ****
##  4 NA_Sales Misc      Platform    1.27     0.958   1.58    0        ****
##  5 NA_Sales Platform  Puzzle     -1.21    -1.57   -0.862   8.28e-13 ****
##  6 NA_Sales Platform  Racing     -0.330   -0.652  -0.00717 4   e- 2 *
##  7 NA_Sales Platform  Role-Pl~   -1.55    -1.86   -1.23    7.48e-13 ****
##  8 NA_Sales Platform  Shooter    -0.261   -0.596   0.0750  3.14e- 1 ns
##  9 NA_Sales Platform  Simulat~   -1.17    -1.52   -0.824   0        ****
## 10 NA_Sales Platform  Sports     -0.695   -1.01   -0.383   3.46e-11 ****
## 11 NA_Sales Platform  Strategy   -2.15    -2.50   -1.81    3.36e-13 ****
```

```
shoot_usporedbe
```

```
## # A tibble: 11 x 8
##    .y.      group1    group2   estimate conf.low conf.high    p.adj p.adj.signif
##    <chr>    <chr>     <chr>       <dbl>    <dbl>     <dbl>    <dbl> <chr>
##  1 NA_Sales Action    Shooter    0.544     0.276    0.812  2.79e- 9 ****
##  2 NA_Sales Adventure Shooter    2.10      1.82     2.38   2.35e-12 ****
##  3 NA_Sales Fighting  Shooter    0.866     0.515    1.22   0        ****
##  4 NA_Sales Misc      Shooter    1.01      0.726    1.29   1.5 e-11 ****
##  5 NA_Sales Platform  Shooter   -0.261    -0.596    0.0750 3.14e- 1 ns
##  6 NA_Sales Puzzle    Shooter    0.954     0.626    1.28   2.55e-13 ****
##  7 NA_Sales Racing    Shooter    0.0691   -0.226    0.364  1   e+ 0 ns
##  8 NA_Sales Role-Pla~ Shooter    1.29      1.00     1.57   1.56e-11 ****
##  9 NA_Sales Shooter   Simulat~  -0.910    -1.23    -0.589  0        ****
## 10 NA_Sales Shooter   Sports    -0.435    -0.718   -0.152  3.55e- 5 ****
## 11 NA_Sales Shooter   Strategy  -1.89     -2.21    -1.57   0        ****
```

Dobivamo drugačiji zaključak od onoga u pravoj ("pogrešnoj") analizi - žanr 'Platform' je značajno popularniji od 10 drugih žanrova, dok je 'Shooter' značajno popularniji od 9.

Možemo vidjeti gdje je razlika ukoliko se referiramo natrag na p-vrijednosti 'Platform-Racing'.

Sada je p-vrijednost pri usporedbi Platform i Racing 0.040, dok je prije bila 0.0773562. Zbog toga je žanr ‘Platform’ “dobio” značajnost nad još jednim žanrom (koristimo :alpha: = 0.05)

Ovaj primjer dobro ilustrira kako ne poštivanje pretpostavki testa može dovesti do krivog zaključka.

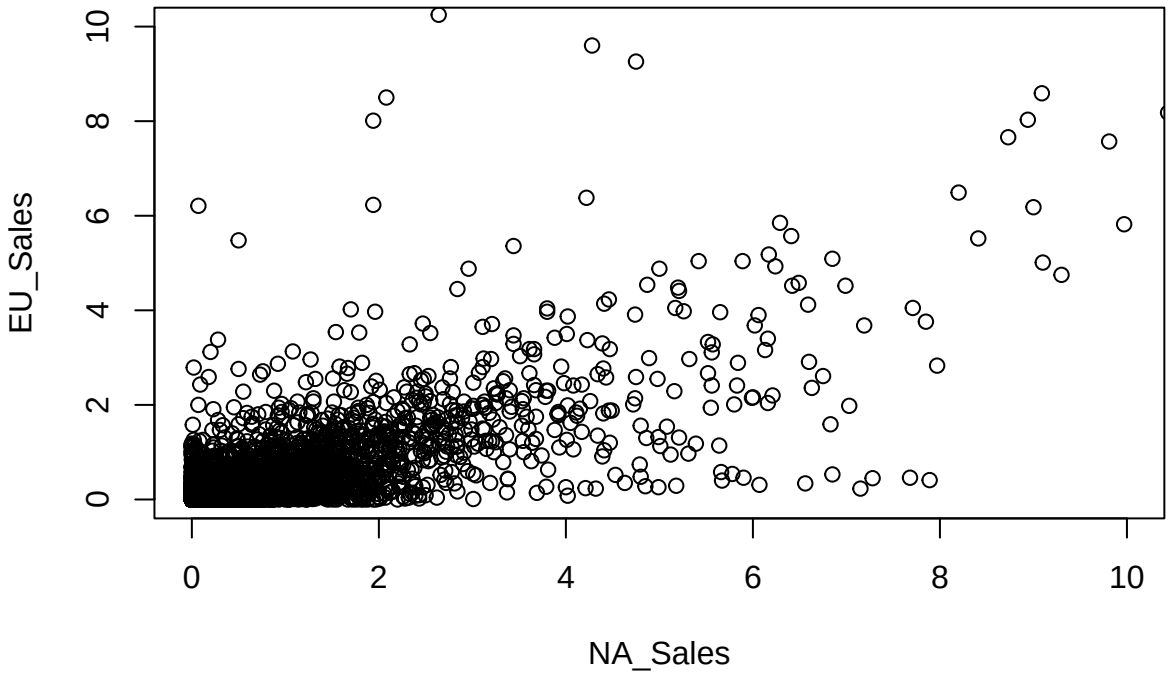
Pitanje 6: Možemo li danim varijablama predviđati prodaju videoigara

Kako bi odgovorili na pitanje, služimo se alatom linerne regresije. Specifično, pokušavamo napraviti model koji dobro objašnjava prodaju igara u Europi.

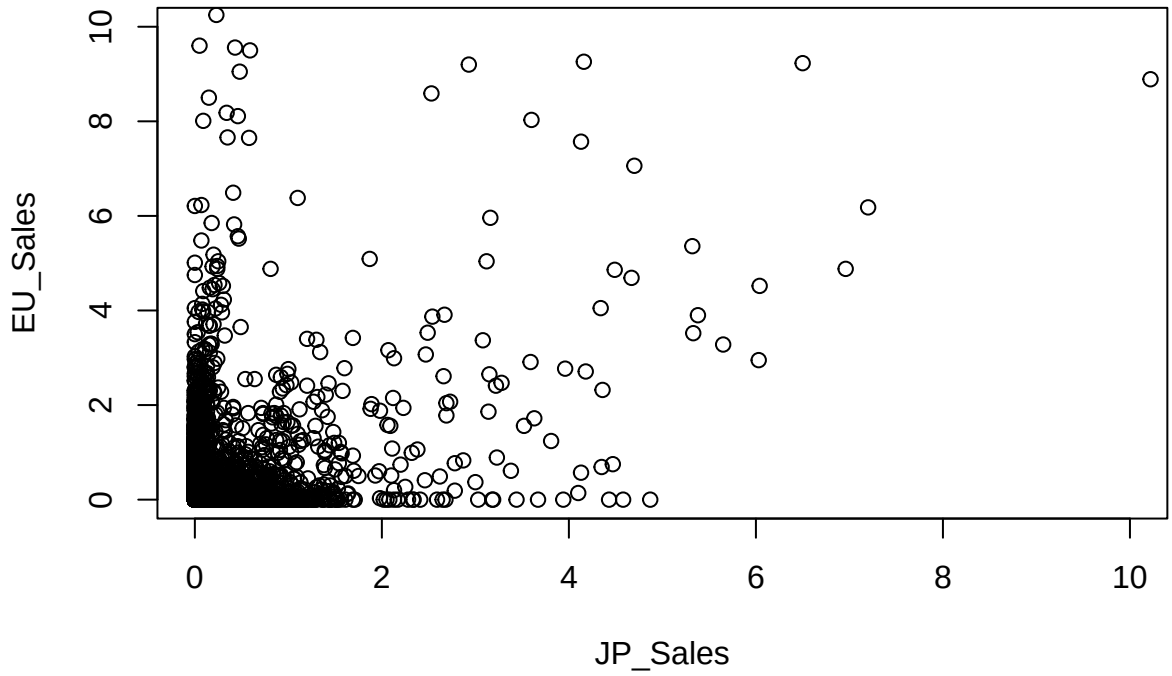
```
vgsales.lr = vgsales.grouped
```

Prvo promotrimo kakvi su kandidati za regresore NA_Sales i JP_Sales Mozemo crtati točkaste dijagrame kako bi dobili grubu sliku njihovog mogućeg odnosa prema EU_Sales.

```
plot(vgsales.lr$NA_Sales, vgsales.lr$EU_Sales, xlab = "NA_Sales", ylab = "EU_Sales",
     ylim=c(0,10), xlim=c(0,10))
```



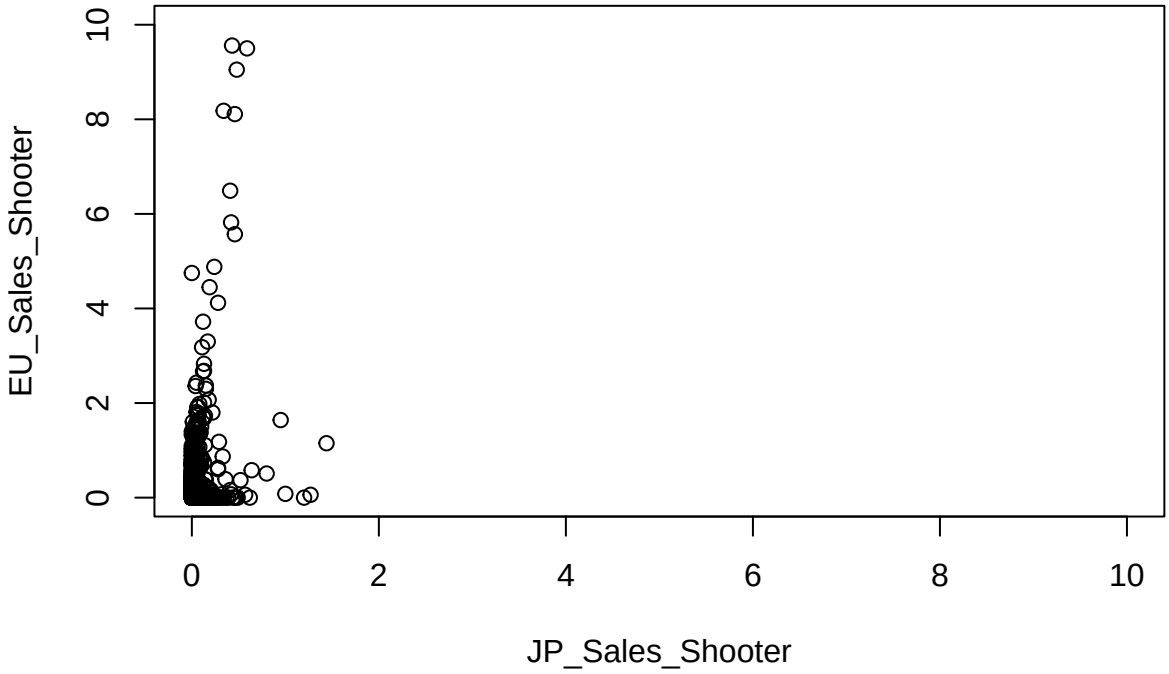
```
plot(vgsales.lr$JP_Sales, vgsales.lr$EU_Sales, xlab = "JP_Sales", ylab = "EU_Sales",
     ylim=c(0,10), xlim=c(0,10))
```



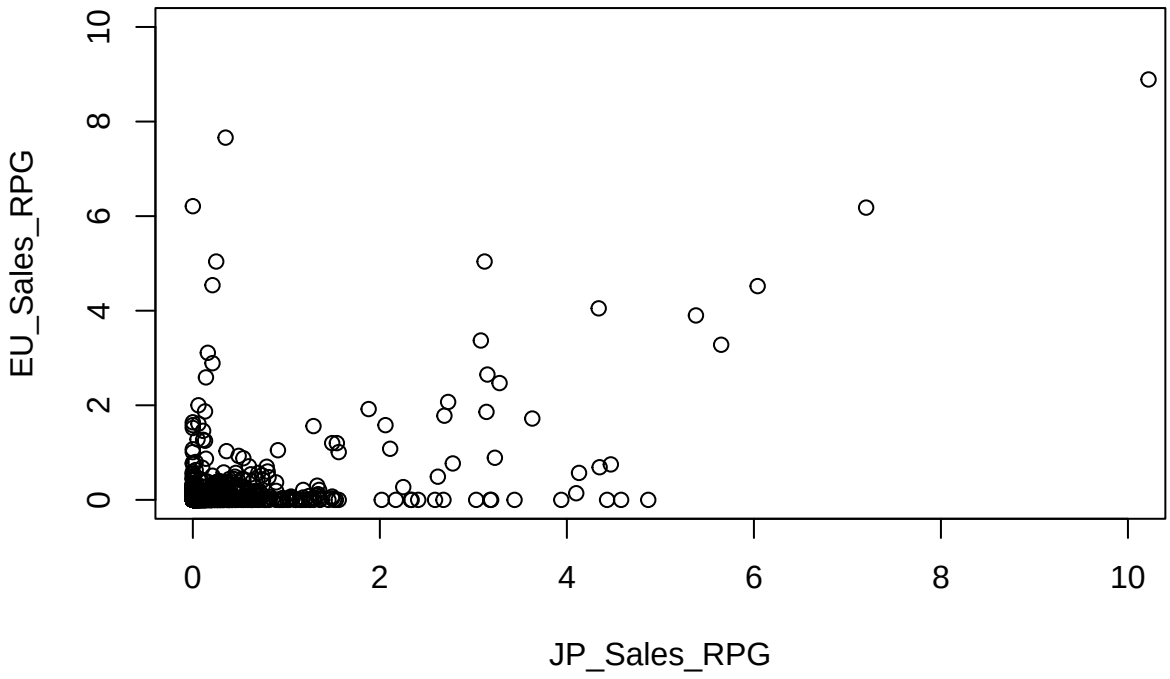
Vidimo da i NA_Sales i JP_Sales možda imaju pozitivan utjecaj na EU_Sales. Također, ovdje dolazi do izražaja razlika u popularnosti pojedinih žanrova izmedu Europe i Japana.

Nacrtajmo opet plotove, ali ovoga puta grupirajmo po žanrovima.

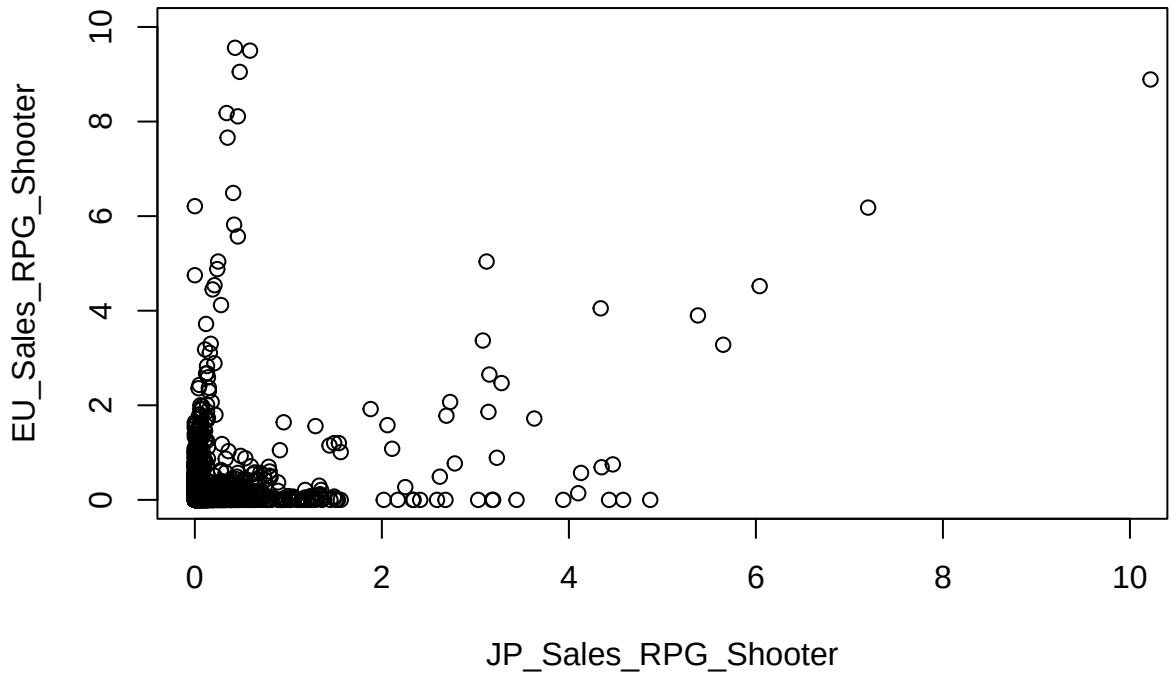
```
# shooter se skoro pa ne prodaje u Japanu, a dominira u EU
eu.jp.shooter = vgsales.lr[ vgsales.lr["Genre"] == "Shooter",]

plot(eu.jp.shooter$JP_Sales, eu.jp.shooter$EU_Sales, xlab = "JP_Sales_Shooter",
     ylab = "EU_Sales_Shooter", ylim=c(0,10), xlim=c(0,10))
```



```
# Role playing dominira u Japanu, a manje u EU
eu.jp.rpg = vgsales.lr[ vgsales.lr["Genre"] == "Role-Playing",]
plot(eu.jp.rpg$JP_Sales, eu.jp.rpg$EU_Sales, xlab = "JP_Sales_RPG",
     ylab = "EU_Sales_RPG", ylim=c(0,10), xlim=c(0,10))
```



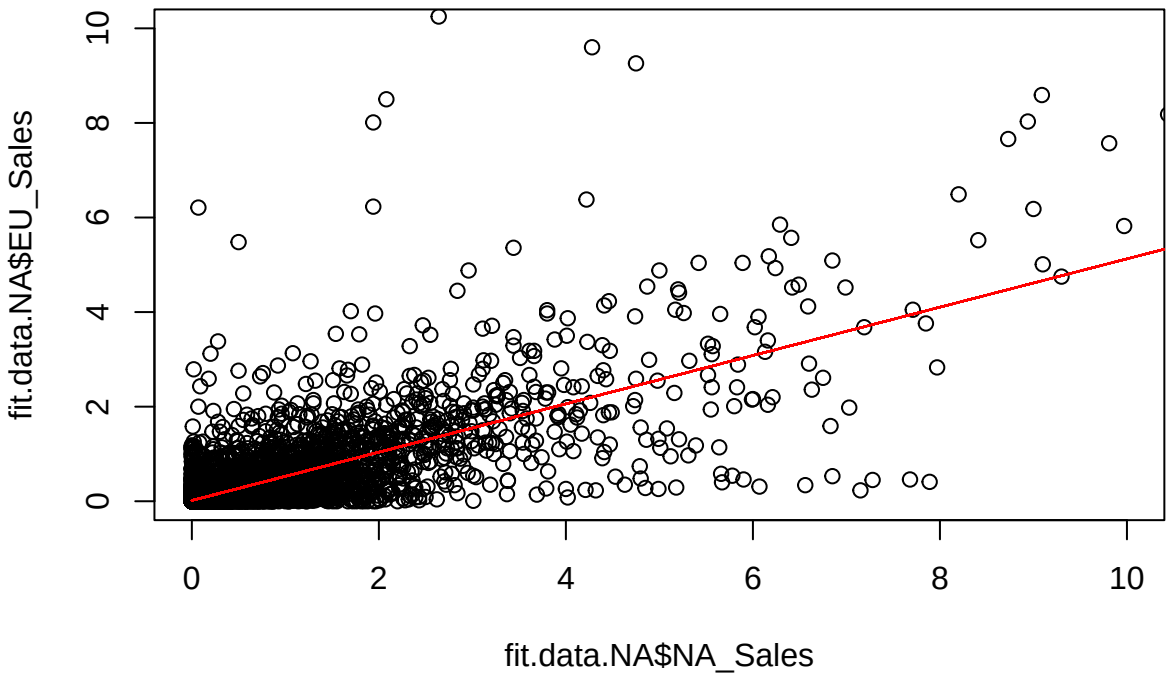
```
# Unija ovih dvaju daju otprilike onaj prvi
eu.jp.rpg.shooter = rbind( eu.jp.shooter, eu.jp.rpg)
plot(eu.jp.rpg.shooter$JP_Sales, eu.jp.rpg.shooter$EU_Sales, xlab = "JP_Sales_RPG_Shooter",
     ylab = "EU_Sales_RPG_Shooter", ylim=c(0,10), xlim=c(0,10))
```



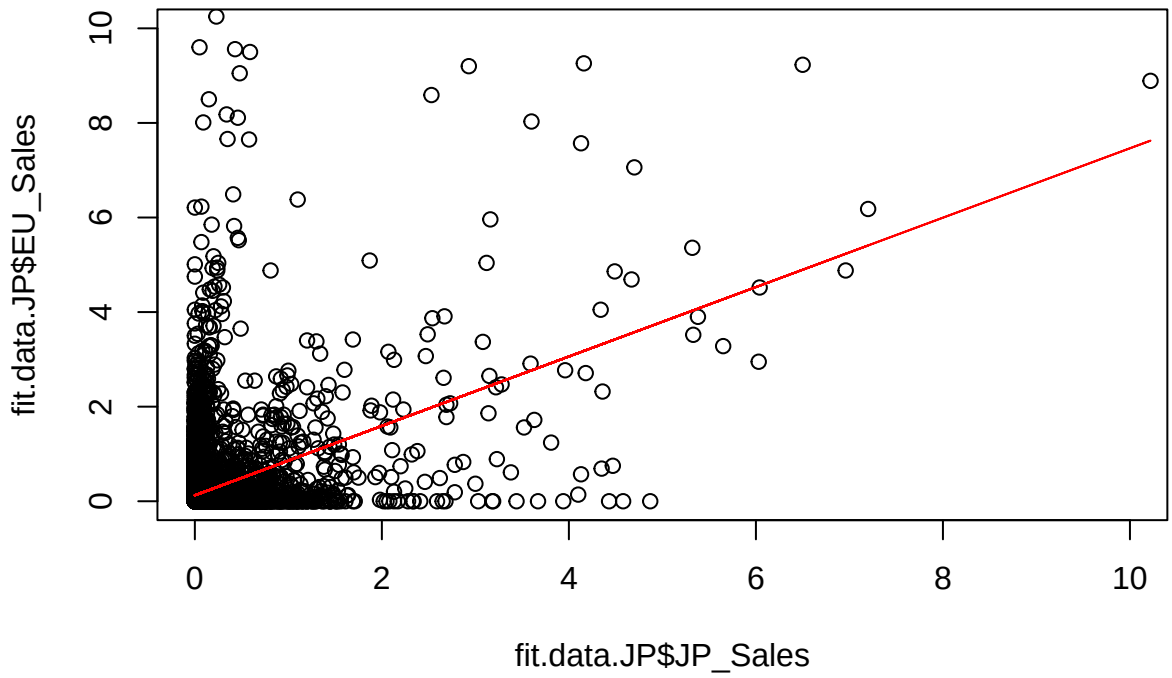
Dakle, JP_Sales direktno nije dobar regresor za EU_Sales. Bolje bi bilo odvojiti po žanrovima pa gledati njihov linearni doprinos.

Zbog navedenih razloga prepostavljamo da će NA_Sales biti bolji regresor za EU_Sales.

```
fit.data.NA = vgsales.lr
fit.data.JP = vgsales.lr

#Model sa NA_Sales kao jedinim regresorom
fit.NA_Sales = lm(EU_Sales~ NA_Sales, data = fit.data.NA)

#Model sa JP_Sales kao jedinim regresorom
fit.JP_Sales = lm(EU_Sales~JP_Sales, data = fit.data.JP)

#Nacrtajmo regresijske pravce na prijašnjim tockastim dijagramima
plot(fit.data.NA$NA_Sales, fit.data.NA$EU_Sales, ylim=c(0,10), xlim=c(0,10))
lines(fit.data.NA$NA_Sales, fit.NA_Sales$fitted.values, col = "red")
```




```
plot(fit.data.JP$JP_Sales, fit.data.JP$EU_Sales, ylim=c(0,10), xlim=c(0,10))
lines(fit.data.JP$JP_Sales, fit.JP_Sales$fitted.values, col = "red")
```



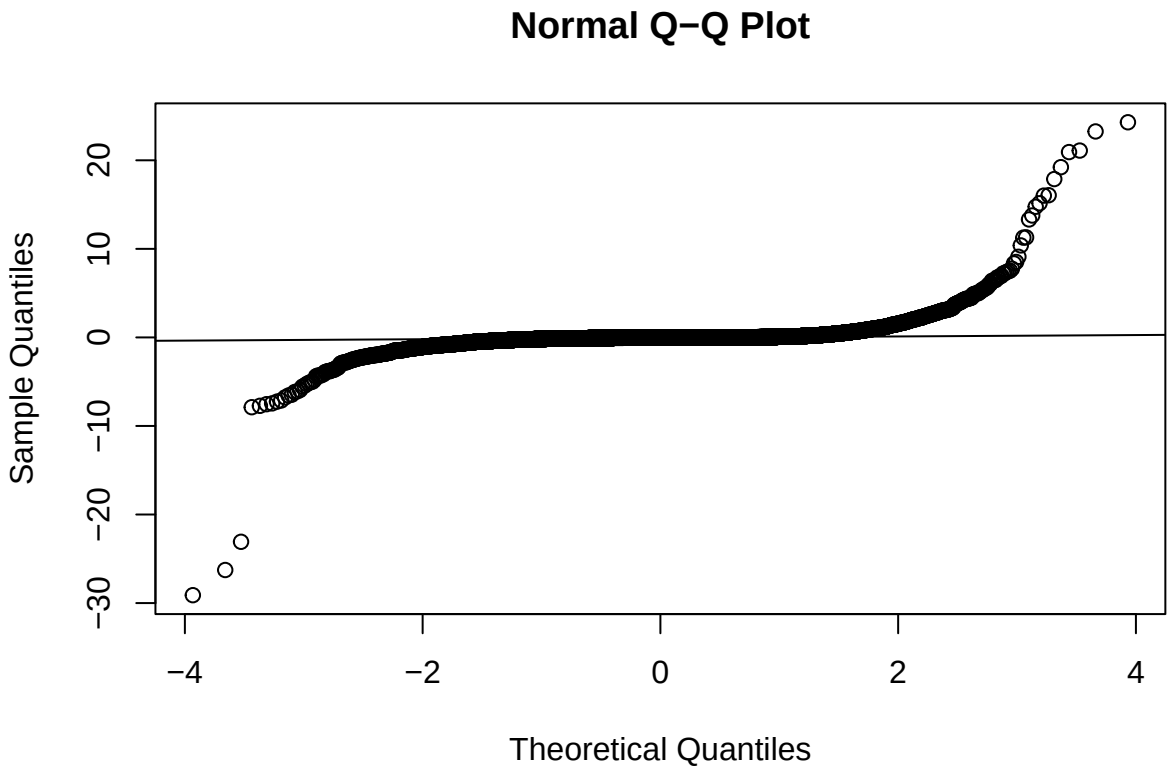
Ovim nagibima pravaca možemo potvrditi prethodnu tvrdnju da i NA_Sales i JP_Sales imaju pozitivan utjecaj na EU_Sales.

Definiramo pomoćnu funkciju koja će za model testirati njene reziduale.

```
test.residuals <- function(selected.model){

  #QQPlot standardiziranih reziduala
  qqnorm(rstandard(selected.model))
  qqline(rstandard(selected.model))

  print(ks.test(rstandard(selected.model), 'pnorm'))

  print(lillie.test(rstandard(selected.model)))
}
```

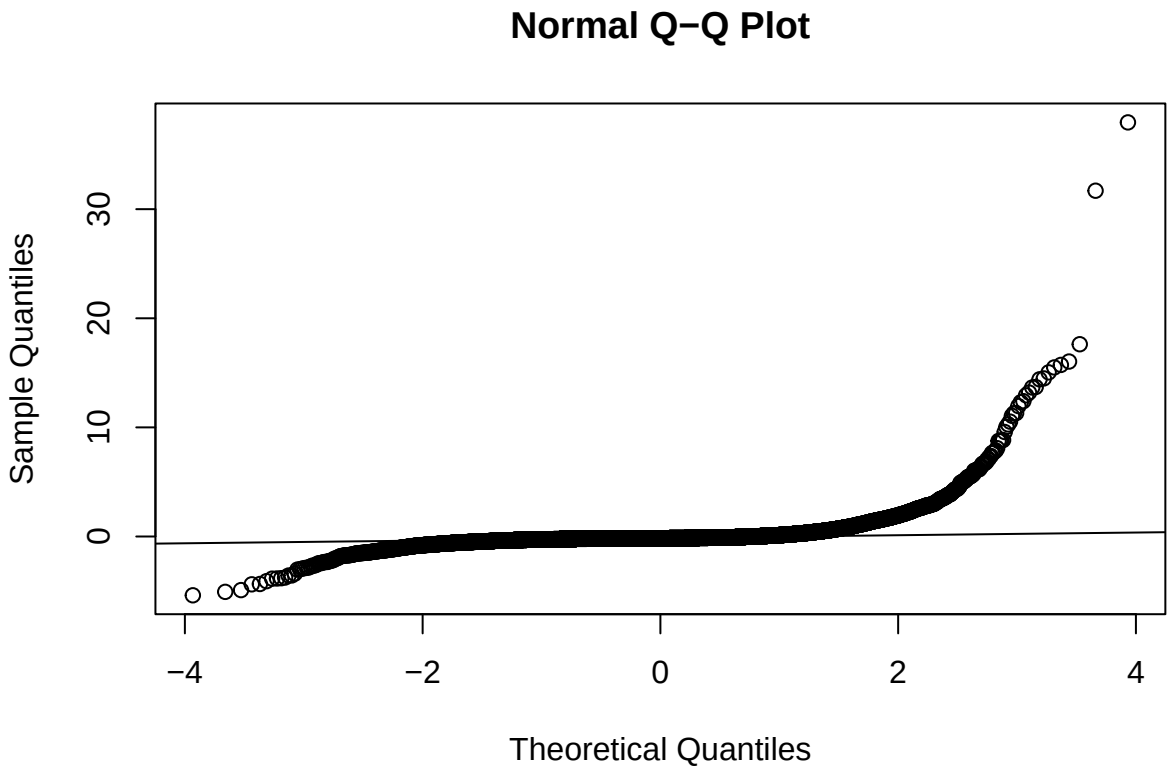
Testirajmo sada reziduale prijašnjih modela

```
test.residuals(fit.NA_Sales)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.32501, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.3254, p-value < 2.2e-16
```

```
test.residuals(fit.JP_Sales)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.2943, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.29435, p-value < 2.2e-16
```

Ne može se reći da su rezidualni normalno distribuirani. Testirajmo sada same modele.

```
summary(fit.NA_Sales)
```

```
##
## Call:
## lm(formula = EU_Sales ~ NA_Sales, data = fit.data.NA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1470  -0.0465  -0.0159   0.0024  11.0361
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01588    0.00445    3.57 0.000359 ***
## NA_Sales     0.51100    0.00371  137.73  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4623 on 11918 degrees of freedom
## Multiple R-squared:  0.6141, Adjusted R-squared:  0.6141
## F-statistic: 1.897e+04 on 1 and 11918 DF,  p-value: < 2.2e-16
```

```
summary(fit.JP_Sales)
```

```
##
## Call:
## lm(formula = EU_Sales ~ JP_Sales, data = fit.data.JP)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6989 -0.1467 -0.1147 -0.0321 26.1284
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.124716   0.006589   18.93   <2e-16 ***
## JP_Sales    0.733927   0.016846   43.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6913 on 11918 degrees of freedom
## Multiple R-squared:  0.1374, Adjusted R-squared:  0.1373
## F-statistic:  1898 on 1 and 11918 DF,  p-value: < 2.2e-16
```

Vidimo da NA_Sales ima značajniji utjecaj na EU_Sales nego JP_Sales, ali i dalje nismo sigurni da je utjecaj JP_Sales zanemariv.

Isprobajmo onda sada višestruki model kombinirajući NA_Sales i JP_Sales.

```
fit.multi.na.jp.data = vgsales.lr
```

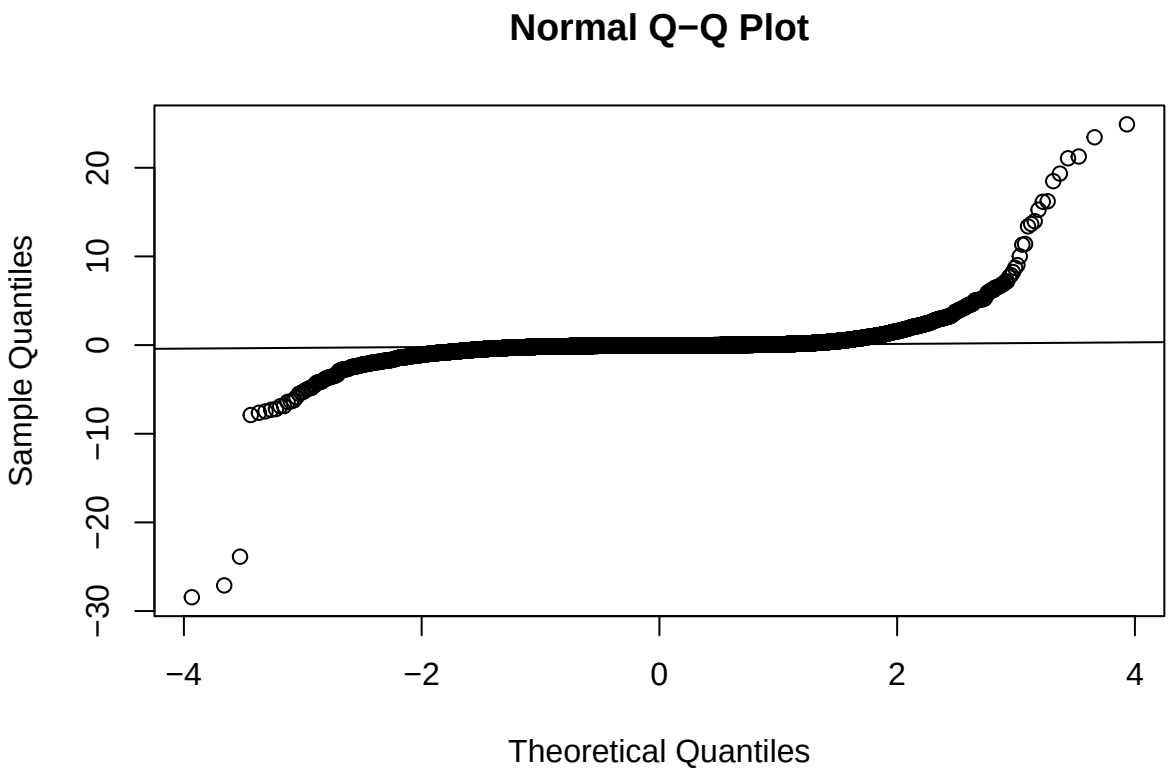
Ispitajmo prvo korelaciju izmedu NA_Sales i JP_Sales, jer ne želimo da oni budi "previše" korelirani.

```
cor(fit.multi.na.jp.data$NA_Sales, fit.multi.na.jp.data$JP_Sales)
```

```
## [1] 0.4091538
```

Dobivamo srednju korelaciju, što je u redu i možemo nastaviti sa izradom modela.

```
#Model sa NA_Sales i JP_Sales kao regresorima
fit.multi.na.jp = lm(EU_Sales ~ NA_Sales + JP_Sales, fit.multi.na.jp.data)
#Testiramo reziduale
test.residuals(fit.multi.na.jp)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.32025, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.32077, p-value < 2.2e-16
```

```
#Testiramo model
summary(fit.multi.na.jp)
```

```
##
## Call:
## lm(formula = EU_Sales ~ NA_Sales + JP_Sales, data = fit.multi.na.jp.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7419  -0.0485  -0.0160   0.0064  11.2537
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.008910   0.004491   1.984   0.0473 *
```

```
## NA_Sales    0.494976   0.004050 122.205   <2e-16 ***
## JP_Sales    0.118915   0.012300   9.668   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4605 on 11917 degrees of freedom
## Multiple R-squared:  0.6172, Adjusted R-squared:  0.6171
## F-statistic:  9605 on 2 and 11917 DF,  p-value: < 2.2e-16
```

Dobivamo otprilike iste rezultate kao i kod jednostrukog modela sa NA_Sales pa možemo zaključiti da JP_Sales u ovakvome obliku nema veliki utjecaj na model.

Testirajmo je li žanr možda značajan regresor. Stvaramo dummy varijable od Genre s tim da se radi o kategorijskoj varijabli i radimo model.

```
fit.multi.na.jp.data = vgsales.lr

#Izrada dummy varijabli
require(fastDummies)
```

```
## Loading required package: fastDummies
```

```
vgsales2.d = dummy_cols(fit.multi.na.jp.data, select_columns = "Genre")

#Promjenimo ime žanra Role Playing igri
names(vgsales2.d)[names(vgsales2.d) == "Genre_Role-Playing"] <- "Genre_Role_Playing"

#Stvaramo model sa dummy varijablama
fit.Genre.d = lm(EU_Sales~Genre_Action + Genre_Adventure + Genre_Fighting + Genre_Misc + Genre_Platform
summary(fit.Genre.d)
```

```
##
## Call:
## lm(formula = EU_Sales ~ Genre_Action + Genre_Adventure + Genre_Fighting +
##     Genre_Misc + Genre_Platform + Genre_Puzzle + Genre_Racing +
##     Genre_Role_Playing + Genre_Shooter + Genre_Simulation + Genre_Sports,
##     data = vgsales2.d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3660 -0.2341 -0.1441 -0.0522 28.7561
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.07582    0.03022   2.509   0.0121 *
## Genre_Action        0.18827    0.03446   5.463 4.78e-08 ***
## Genre_Adventure    -0.01671    0.03763  -0.444   0.6570
## Genre_Fighting      0.08225    0.04201   1.958   0.0503 .
## Genre_Misc          0.08057    0.03617   2.228   0.0259 *
## Genre_Platform      0.25636    0.04257   6.021 1.78e-09 ***
## Genre_Puzzle        0.02240    0.04437   0.505   0.6137
## Genre_Racing        0.21885    0.03985   5.492 4.05e-08 ***
## Genre_Role_Playing  0.07238    0.03665   1.975   0.0483 *
## Genre_Shooter       0.29015    0.03938   7.368 1.85e-13 ***
## Genre_Simulation    0.07719    0.04062   1.900   0.0574 .
## Genre_Sports        0.18808    0.03599   5.226 1.76e-07 ***
```

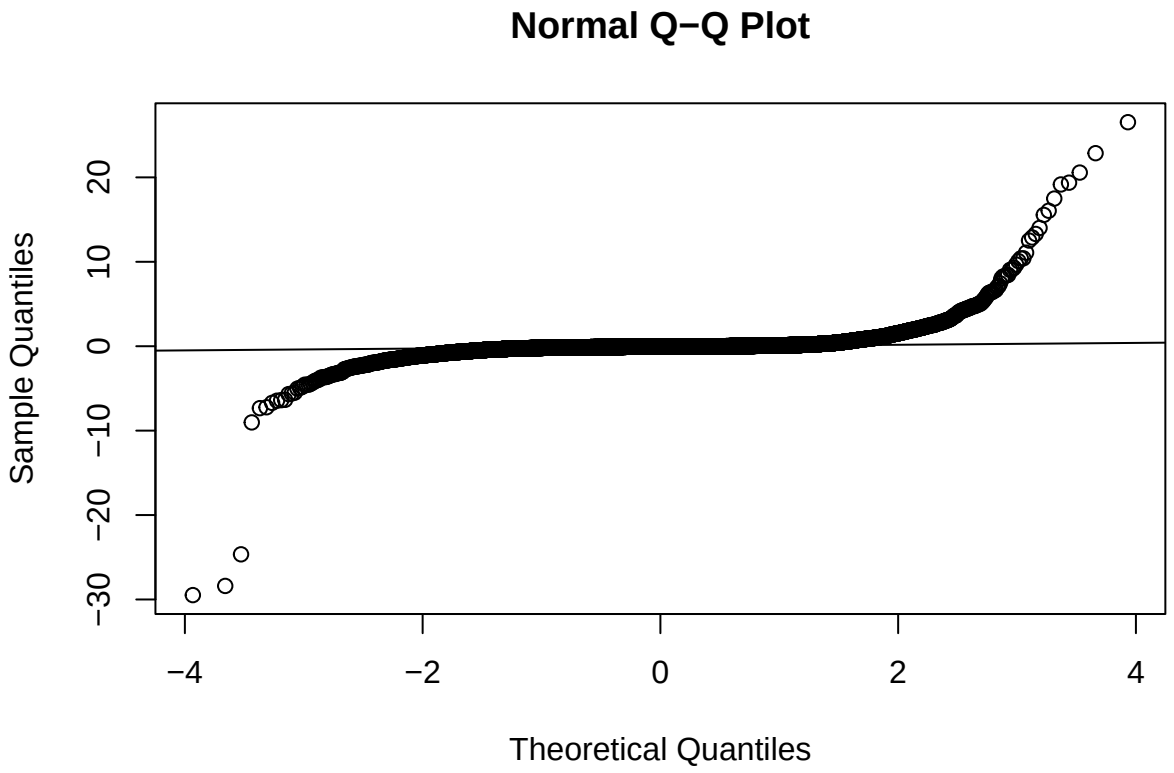
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7389 on 11908 degrees of freedom
## Multiple R-squared:  0.01517,    Adjusted R-squared:  0.01426
## F-statistic: 16.67 on 11 and 11908 DF,  p-value: < 2.2e-16
```

Neke dummy varijable nemaju veliki utjecaj na model pa njih možemo izbaciti.

Zbog tog razloga u model uključujemo samo Genre_Racing, Genre_Sports, Genre_Shooter, Genre_Platform i Genre_Action.

Dodat ćemo i regresore tipa Genre_!*JP_Sales jer želimo rastaviti JP_Sales na komponente ovisno o žanru.

```
fit.multi.d = lm(Global_Sales~NA_Sales + JP_Sales + Genre_Racing + Genre_Sports
                 + Genre_Shooter + Genre_Platform + Genre_Action
                 + I(JP_Sales*Genre_Shooter) + I(JP_Sales*Genre_Role_Playing),
                 data = vgsales2.d)
#Testiramo reziduale dobivenog modela
test.residuals(fit.multi.d)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.31951, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
```

```
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.32017, p-value < 2.2e-16
```

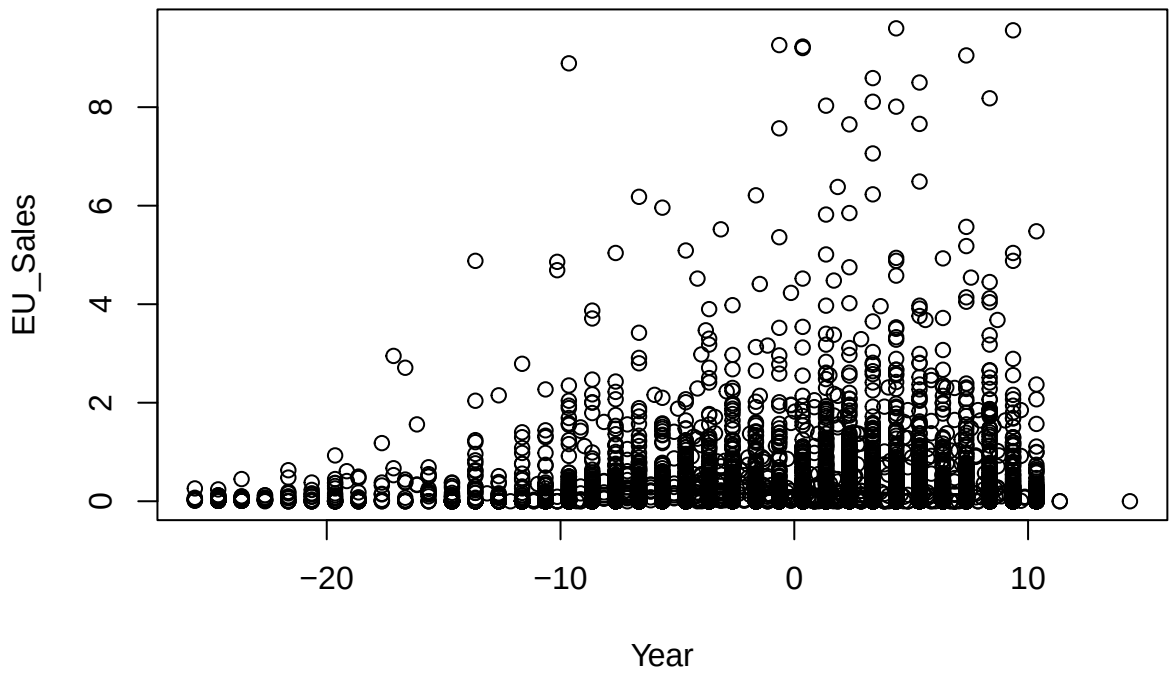
Reziduali opet nisu normalno distribuirani.

```
#Pogledajmo model
summary(fit.multi.d)
```

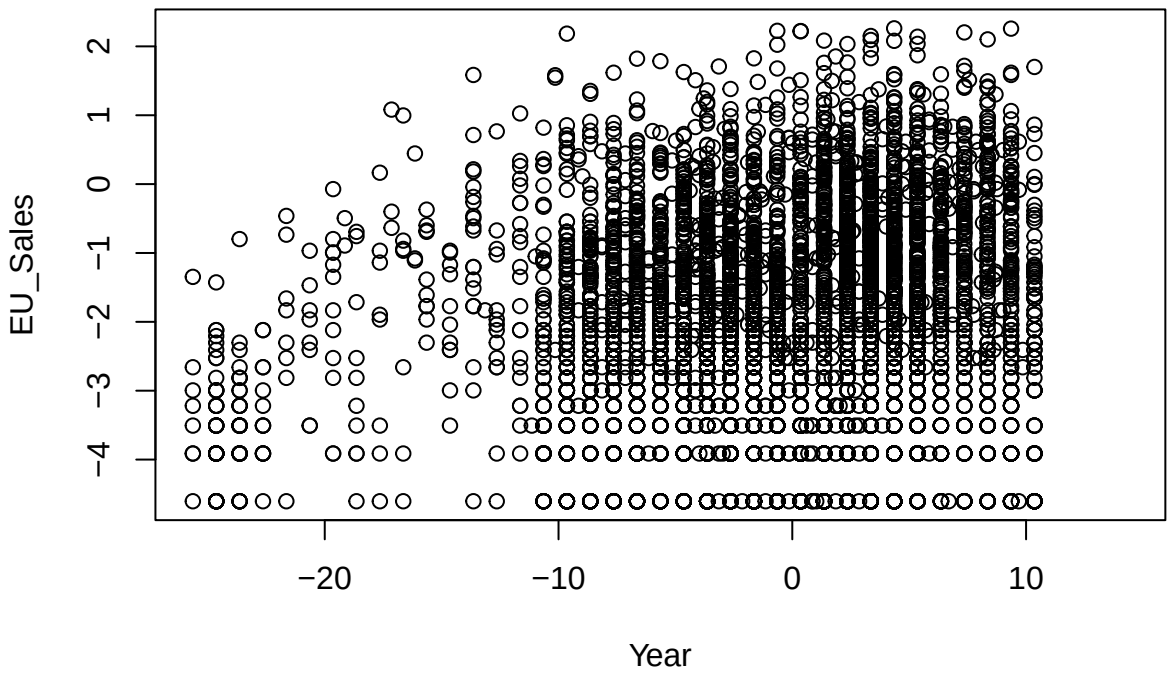
```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales + JP_Sales + Genre_Racing +
##     Genre_Sports + Genre_Shooter + Genre_Platform + Genre_Action +
##     I(JP_Sales * Genre_Shooter) + I(JP_Sales * Genre_Role_Playing),
##     data = vgsales2.d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9472  -0.0747  -0.0175   0.0138  15.4064
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -0.0004276  0.0077284  -0.055 0.955880
## NA_Sales                         1.6597946  0.0054360 305.334  < 2e-16 ***
## JP_Sales                         1.0895768  0.0210670  51.720  < 2e-16 ***
## Genre_Racing                     0.0915382  0.0222050   4.122 3.77e-05 ***
## Genre_Sports                     0.0348907  0.0174727   1.997 0.045861 *
## Genre_Shooter                   -0.0079746  0.0228404  -0.349 0.726985
## Genre_Platform                  -0.0870932  0.0254140  -3.427 0.000612 ***
## Genre_Action                     0.0607318  0.0153539   3.955 7.68e-05 ***
## I(JP_Sales * Genre_Shooter)      0.9367028  0.1581091   5.924 3.22e-09 ***
## I(JP_Sales * Genre_Role_Playing) 0.0190131  0.0291587   0.652 0.514376
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5922 on 11910 degrees of freedom
## Multiple R-squared:  0.9266, Adjusted R-squared:  0.9265
## F-statistic: 1.669e+04 on 9 and 11910 DF,  p-value: < 2.2e-16
```

Dodajmo godinu kao regresor. Gaming prije nije bio toliko popularan kao danas, pa očekujemo da modernije igre imaju veći sales.

```
fit.data.Year = vgsales.lr[ vgsales.lr["EU_Sales"] < 10,]
# bolje je kada regresor nije za nekoliko magnituda veci od varijable koju predviđamo
fit.data.Year$Year = scale( fit.data.Year$Year, center = TRUE, scale = FALSE)

plot(fit.data.Year$Year, fit.data.Year$EU_Sales, xlab = "Year", ylab = "EU_Sales")
```




```
# eksponencijalan rast?
plot(fit.data.Year$Year, log( fit.data.Year$EU_Sales), xlab = "Year", ylab = "EU_Sales")
```



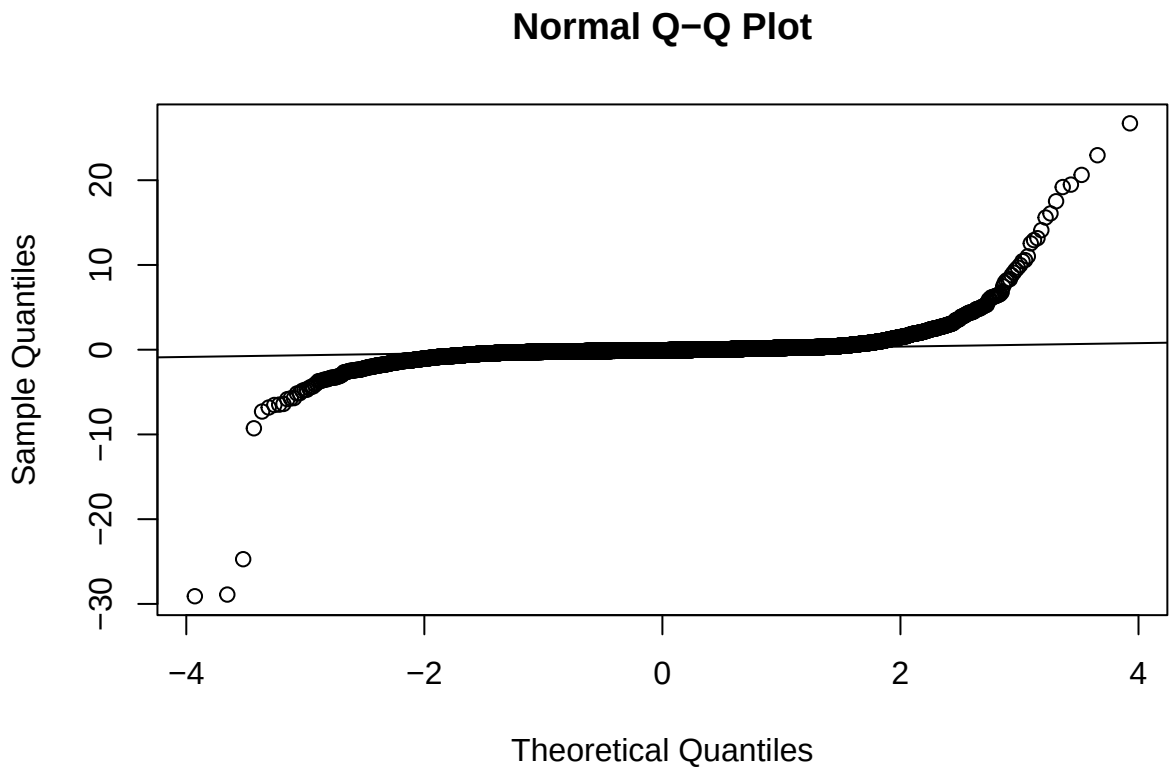
Zbog mogućeg eksponencijalnog rasta dodajmo jos transformirani regresor u novi model.

```
#Dodajemo uz Year i log(Year) kao regresor
fit.Year = lm(EU_Sales~Year + log(Year), data = fit.data.Year)

summary( fit.Year)
```

```
##
## Call:
## lm(formula = EU_Sales ~ Year + log(Year), data = fit.data.Year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3078 -0.2046 -0.1633 -0.0970  9.3954
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.147419   0.014286  10.319   <2e-16 ***
## Year        0.008796   0.006256   1.406    0.160
## log(Year)   0.012807   0.018244   0.702    0.483
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6374 on 6791 degrees of freedom
##   (5115 observations deleted due to missingness)
## Multiple R-squared:  0.002976,   Adjusted R-squared:  0.002682
## F-statistic: 10.13 on 2 and 6791 DF,  p-value: 4.027e-05
```

Najvjerojatnije godina nema značajan utjecaj, ali možemo još provjeriti.

```
#Prijašnji model sa dodanim regresorima Year i log(Year)
fit.final.v1 = lm(Global_Sales~NA_Sales + JP_Sales + Genre_Racing + Genre_Sports
                  + Genre_Action + I(JP_Sales*Genre_Shooter)
                  + I(JP_Sales*Genre_Role_Playing) + Year + log(Year),
                  data = vgsales2.d)
#Testiramo reziduale
test.residuals(fit.final.v1)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.28323, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.28456, p-value < 2.2e-16
```

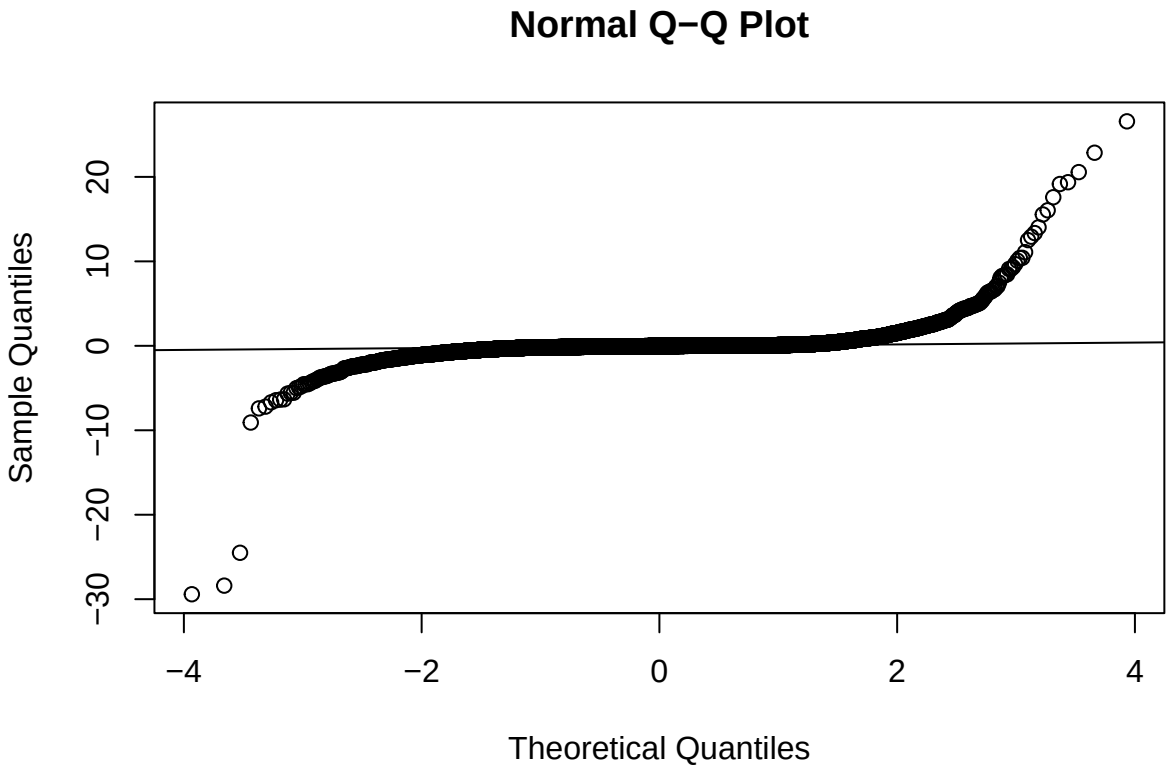
```
summary(fit.final.v1)
```

```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales + JP_Sales + Genre_Racing +
```

```
##     Genre_Sports + Genre_Action + I(JP_Sales * Genre_Shooter) +
##     I(JP_Sales * Genre_Role_Playing) + Year + log(Year), data = vgsales2.d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4729  -0.1034  -0.0373   0.0572  15.3196
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -2.105e+04  4.876e+03  -4.317 1.60e-05 ***
## NA_Sales                         1.657e+00  5.452e-03 303.983  < 2e-16 ***
## JP_Sales                         1.135e+00  2.112e-02  53.767  < 2e-16 ***
## Genre_Racing                     1.286e-01  2.207e-02   5.827 5.80e-09 ***
## Genre_Sports                     6.769e-02  1.724e-02   3.926 8.70e-05 ***
## Genre_Action                     6.010e-02  1.513e-02   3.971 7.19e-05 ***
## I(JP_Sales * Genre_Shooter)      1.062e+00  1.508e-01   7.040 2.03e-12 ***
## I(JP_Sales * Genre_Role_Playing) -5.196e-03  2.885e-02  -0.180    0.857
## Year                            -1.574e+00  3.688e-01  -4.269 1.98e-05 ***
## log(Year)                        3.183e+03  7.385e+02   4.310 1.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5853 on 11675 degrees of freedom
##   (235 observations deleted due to missingness)
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9276
## F-statistic: 1.664e+04 on 9 and 11675 DF,  p-value: < 2.2e-16
```

Rezultati nisu značajno bolji od prijašneg modela pa možemo odbaciti godinu kao regresora zbog jednostavnosti modela.

```
fit.final.v2 = lm(Global_Sales~NA_Sales + JP_Sales + Genre_Racing + Genre_Sports
                  + Genre_Action + I(JP_Sales*Genre_Shooter)
                  + I(JP_Sales*Genre_Role_Playing), data = vgsales2.d)
test.residuals(fit.final.v2)
```




```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  rstandard(selected.model)
## D = 0.32431, p-value < 2.2e-16
## alternative hypothesis: two-sided
##
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  rstandard(selected.model)
## D = 0.32495, p-value < 2.2e-16
```

```
summary(fit.final.v2)
```

```
##
## Call:
## lm(formula = Global_Sales ~ NA_Sales + JP_Sales + Genre_Racing +
##     Genre_Sports + Genre_Action + I(JP_Sales * Genre_Shooter) +
##     I(JP_Sales * Genre_Role_Playing), data = vgsales2.d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9180  -0.0744  -0.0150   0.0123  15.4402
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                      -0.007659   0.007052  -1.086   0.2775
## NA_Sales                          1.658623   0.005410 306.605  < 2e-16 ***
## JP_Sales                          1.084406   0.020922  51.831  < 2e-16 ***
## Genre_Racing                      0.099653   0.021955   4.539 5.71e-06 ***
## Genre_Sports                      0.043173   0.017147   2.518   0.0118 *
## Genre_Action                      0.068897   0.014984   4.598 4.31e-06 ***
## I(JP_Sales * Genre_Shooter)       0.947449   0.150831   6.282 3.47e-10 ***
## I(JP_Sales * Genre_Role_Playing)  0.028370   0.029035   0.977   0.3286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5924 on 11912 degrees of freedom
## Multiple R-squared:  0.9265, Adjusted R-squared:  0.9264
## F-statistic: 2.144e+04 on 7 and 11912 DF,  p-value: < 2.2e-16
```

Primjećujemo da, unatoč naizgled "kvalitetnim" modelima koje smo dobili, reziduali tih modela nikad nisu normalno distribuirani. Tijekom testiranja i izrade projekta nikakve transformacije podataka nisu značajno poboljšale taj rezultat. Zaključujemo da naši modeli ne objašnjavaju dobro sve trendove u skupu podataka.

Pitanje 7: Zamislite da radite videoigru. Kakve karakteristike bi ta igra trebala imati ako želite da ona bude što prodavanija u određenoj regiji.

Općenito, kad bismo htjeli određivati karakteristike igre razmatrali bismo one regresore koji najbolje objašnjavaju regresiju. Međutim, naši podaci pretežno se sastoje od prodaje u određenoj regiji (NA_Sales, EU_Sales. . .) koje nemamo na raspolaganju kada tek radimo videoigru. Trebaju nam podaci kao što su dob igrača, cijena proizvodnje igre, kupovna moć ljudi, single-player ili multi-player, informacije o uređajima na kojima će se igrica izvodit (jači uređaj može podržat bolju grafiku i veći svijet) itd. Bez toga možemo reći samo u grubo. Ako radimo igru za tržište Japana, htjeli bismo da je ona RPG i da ju objavi Nintendo.