How to Judge an App By Its Cover

Chi Ho Tam

12/6/2020

Introduction

The question that all users have is, “which app is better?” This project will help give a better understanding on how to determine if a certain app is considered better or well-rounded compared to others. This decision will be based on the data the users retrieve from the app’s descriptions in the Play Store (e.g. ratings, downloads, price, etc.) This measurement will be a good standard guideline on whether the app the user is looking into is good or not. There will be a measurement for every genre in the app store as well as an overall app rating.

Clients: App creators and App users

Data: Data from the Play Store

DataSets

- **Googleplaystore.csv:**
  A table that consists of ten thousand apps from Google Play Store with data of ratings, reviews, size of the app, number of installs, price, content rating, and genre. I plan to use most if not all of the variables provided to determine the measurement. I plan to use the mean or the most common of each category to find what an ideal app for their genre would be.
- **Googleplaystore_user_reviews.csv:**
  A table that consists of over sixty thousand comments posted on Google Play Store for each app. The table has data on whether the comment is positive, negative, or neutral. I plan on using the data and help determine if there is a correlation between the ratio of positive and negative to the ratings, which will be used in determining the measurement.

Data Cleaning

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#Load files
googleplaystore <- read.csv("googleplaystore.csv")
#Remove the entire row of data if it contains NaN in googleplaystore
googleplaystore <- googleplaystore %>% filter(!grepl(NaN, Rating))
googleplaystore <- googleplaystore %>% filter(!grepl(NaN, Current.Ver))
#Remove the entire row of data if it contains ; in the column genres in googleplaystore
googleplaystore <- googleplaystore %>% filter(!grepl(";", Genres))
#googleplaystore
#Remove the entire row of data if it contains varies with device in column size in googleplaysto
re
googleplaystore <- googleplaystore %>% filter(!grepl("Varies with device", Size    ))
#Remove the last 3 columns because it won't be used in googleplaystore
googleplaystore <- select(googleplaystore, -c(Last.Updated,Current.Ver,Android.Ver))
#Remove all the $ signs in the price column in googleplaystore
googleplaystore$Price <- ifelse(grepl("\\$", googleplaystore$Price), gsub("\\$", "", googleplays
tore$Price), googleplaystore$Price)
```

- In the code above, I cleaned up some of the data in Googleplaystore.csv so it was ready to be used. I first imported dplyr since that’s the tool I’ll be using to clean my data up. I then opened the csv while setting it to a variable so it’s easy to use. I removed all rows that contained NaN, since it is incomplete data. After that, I removed all rows that contains a semi colon in the column, genres, because I wanted to use specific genres and not mixed. I also removed all the rows that contained varies with device in the column, size, since I wanted all numbers and varies with device is not usable. In addition, I removed the last 3 columns of the csv file because I didn’t plan on using any of those data in my project. Lastly, I removed all the dollar signs in the column, price, so I can use the numbers while not having to worry about problems with the dollar sign being in the way.

```
#Load files
googleplaystore_reviews <- read.csv("googleplaystore_user_reviews.csv")
#Remove the entire row of data if it contains NaN in googleplaystore_reviews
googleplaystore_reviews <- googleplaystore_reviews %>% filter(!grepl(NaN, Translated_Review))
googleplaystore_reviews <- googleplaystore_reviews %>% filter(!grepl(NaN, Sentiment_Polarity))
googleplaystore_reviews <- googleplaystore_reviews %>% filter(!grepl(NaN, Sentiment_Subjectivit
y))
#Remove the last column because it won't be used in googleplaystore_reviews
googleplaystore_reviews <- select(googleplaystore_reviews, -c(Sentiment_Subjectivity))
```

- In the code above, I cleaned up some of the data in Googleplaystore_user_reviews.csv so it was ready to be used. I first opened the csv while setting it to a variable so it’s easy to use.

Then, I removed all rows that contained NaN, since it is incomplete data. Lastly, I removed the last column of the csv file because I didn’t plan on using any of those data in my project.

Data analysis

```
#Summary of googleplaystore
summary(googleplaystore)
```

```
##      App              Category             Rating         Reviews       
##  Length:7310        Length:7310        Min.   : 1.000   Length:7310       
##  Class :character   Class :character   1st Qu.: 4.000   Class :character  
##  Mode  :character   Mode  :character   Median : 4.300   Mode  :character  
##                                        Mean   : 4.169                     
##                                        3rd Qu.: 4.500                     
##                                        Max.   :19.000                     
##      Size             Installs             Type              Price          
##  Length:7310        Length:7310        Length:7310        Length:7310       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##  Content.Rating        Genres         
##  Length:7310        Length:7310       
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
## 
```

```
#Summary of googleplaystore_reviews
summary(googleplaystore_reviews)
```

```
##      App            Translated_Review   Sentiment         Sentiment_Polarity
##  Length:37432       Length:37432       Length:37432       Min.   :-1.0000   
##  Class :character   Class :character   Class :character   1st Qu.: 0.0000   
##  Mode  :character   Mode  :character   Mode  :character   Median : 0.1500   
##                                                           Mean   : 0.1821   
##                                                           3rd Qu.: 0.4000   
##                                                           Max.   : 1.0000
```

- In the code above, I did a summary for both csv files.

```
#Head of googleplaystore
head(googleplaystore)
```

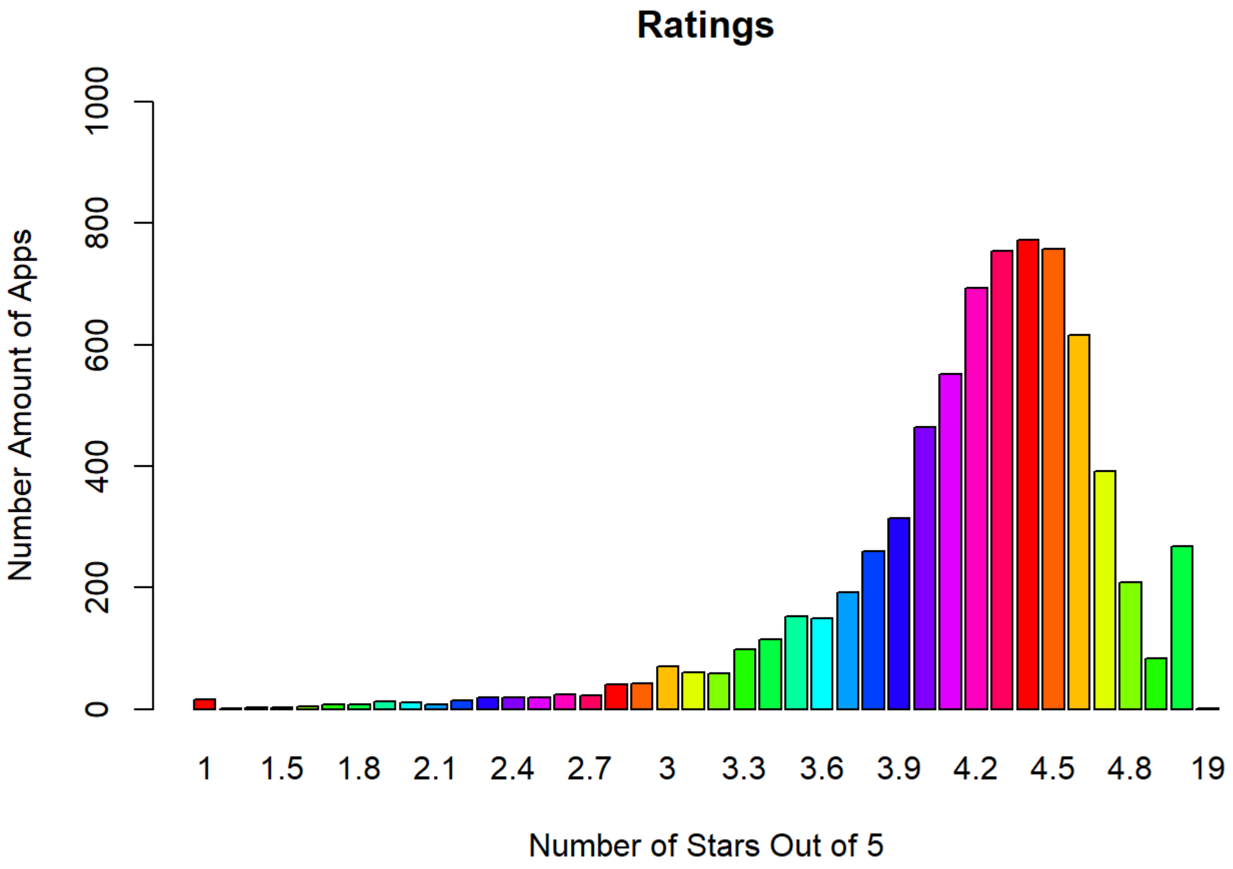
```
##                                                     App       Category Rating
## 1       Photo Editor & Candy Camera & Grid & ScrapBook ART_AND_DESIGN    4.1
## 2 U Launcher Lite â\200“ FREE Live Cool Themes, Hide Apps ART_AND_DESIGN    4.7
## 3                                Sketch - Draw & Paint ART_AND_DESIGN    4.5
## 4                           Paper flowers instructions ART_AND_DESIGN    4.4
## 5              Smoke Effect Photo Maker - Smoke Editor ART_AND_DESIGN    3.8
## 6                                     Infinite Painter ART_AND_DESIGN    4.1
##   Reviews Size    Installs Type Price Content.Rating       Genres
## 1     159  19M     10,000+ Free     0       Everyone Art & Design
## 2   87510 8.7M  5,000,000+ Free     0       Everyone Art & Design
## 3  215644  25M 50,000,000+ Free     0           Teen Art & Design
## 4     167 5.6M     50,000+ Free     0       Everyone Art & Design
## 5     178  19M     50,000+ Free     0       Everyone Art & Design
## 6   36815  29M  1,000,000+ Free     0       Everyone Art & Design
```

```
#Head of googleplaystore_reviews
head(googleplaystore_reviews)
```

```
##                     App
## 1 10 Best Foods for You
## 2 10 Best Foods for You
## 3 10 Best Foods for You
## 4 10 Best Foods for You
## 5 10 Best Foods for You
## 6 10 Best Foods for You
##
Translated_Review
## 1 I like eat delicious food. That's I'm cooking food myself, case "10 Best Foods" helps lot,
also "Best Before (Shelf Life)"
## 2                                                                            This help eating
healthy exercise regular basis
## 3                                                                                 Works great
especially going grocery store
## 4
Best idea us
## 5
Best way
## 6
Amazing
##   Sentiment Sentiment_Polarity
## 1  Positive               1.00
## 2  Positive               0.25
## 3  Positive               0.40
## 4  Positive               1.00
## 5  Positive               1.00
## 6  Positive               0.60
```

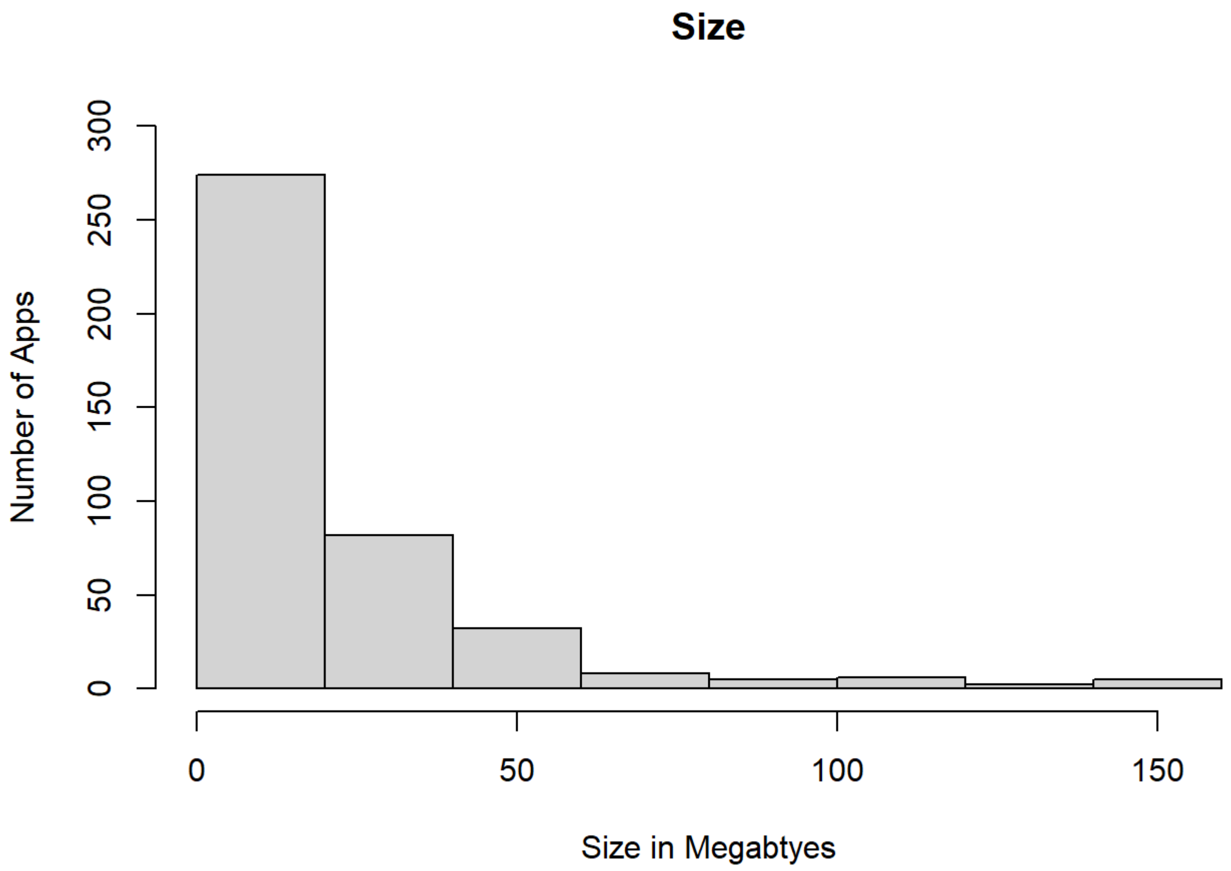
- In the code above, I did the head for both csv files.

```
#Bargraph of googleplaystore ratings
total <- table(googleplaystore$Rating)
barplot(total, main = "Ratings", xlab = "Number of Stars Out of 5", ylab = "Number Amount of App
s", ylim = c(0,1000), col = rainbow(total))
```



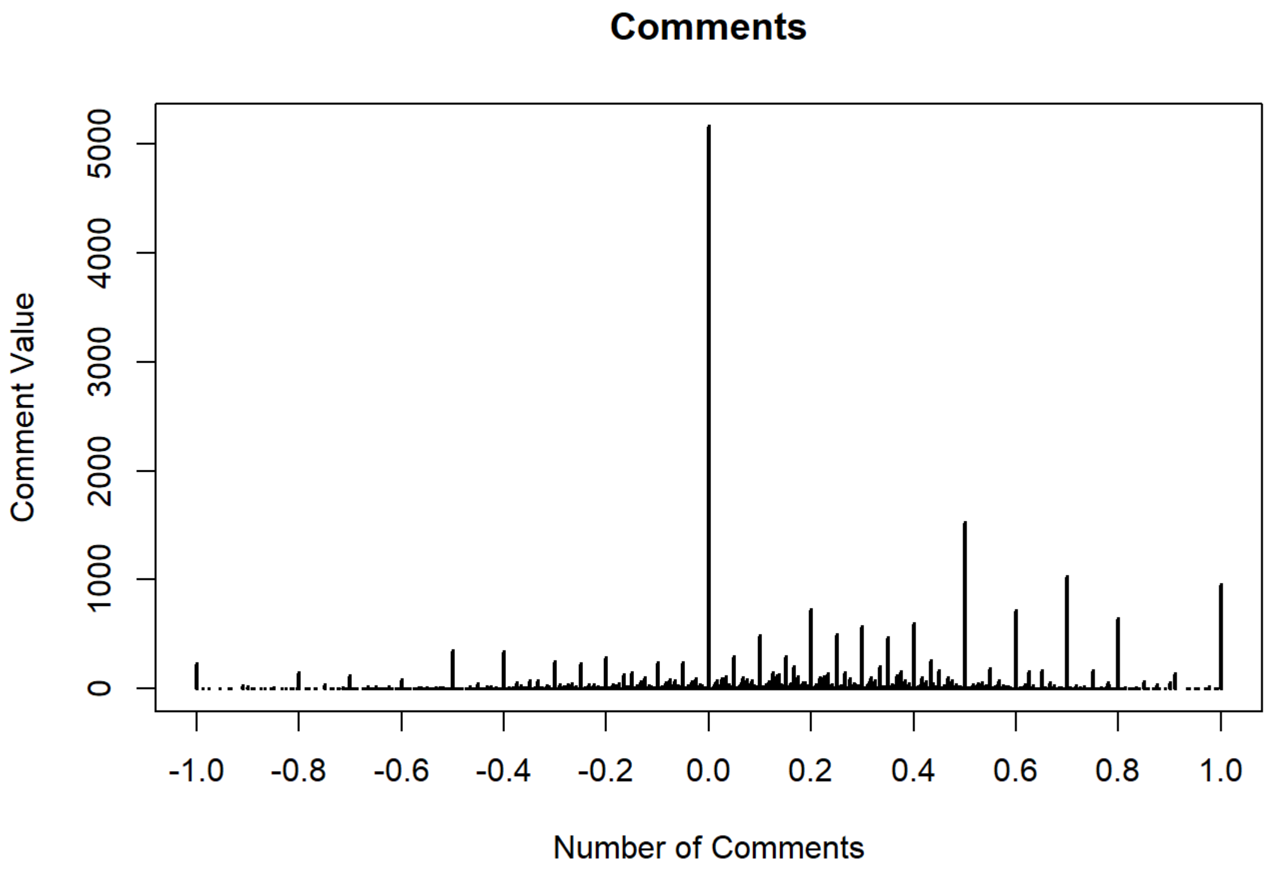
- In the code above, I made a bargraph showing the ratings of all the apps.

```
#Histogram of googleplaystore size
total <- table(googleplaystore$Size)
hist(total, main = "Size", xlab = "Size in Megabtyes", ylab = "Number of Apps", ylim = c(0,300))
```



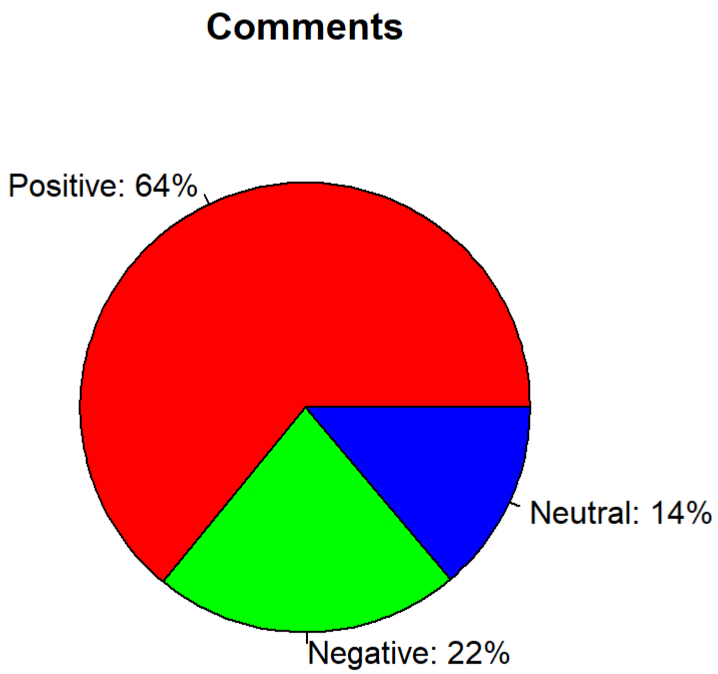
- In the code above, I made a histogram showing the sizes of all the apps.

```
#Plot of googleplaystore_reviews comment value
total <- table(googleplaystore_reviews$Sentiment_Polarity)
plot(total, main = "Comments", xaxt = "n", xlab = "Number of Comments", ylab = "Comment Value")
axis(side = 1, at=seq(-1,1, by = 0.2))
```



- In the code above, I made a plot showing the comment values of all the apps.

```
#PieChart of googleplaystore_reviews comment type
positive <- sum(googleplaystore_reviews$Sentiment == "Positive")
negative <- sum(googleplaystore_reviews$Sentiment == "Negative")
neutral <- sum(googleplaystore_reviews$Sentiment == "Neutral")
parts <- c(positive, negative, neutral)
names <- c("Positive", "Negative", "Neutral")
percent <- round(parts/sum(parts)*100)
names <- paste(names, sep = ": ", percent)
names <- paste(names, "%", sep = "")
pie(parts, main = "Comments", labels = names, col = rainbow(length(names)))
```



- In the code above, I made a pie chart showing the comment types of all the apps.

Modeling

```
#Linear Regression of rating and price in googleplaystore
lm_plot = lm(Rating ~ Price, data = googleplaystore)
summary(lm_plot)
```

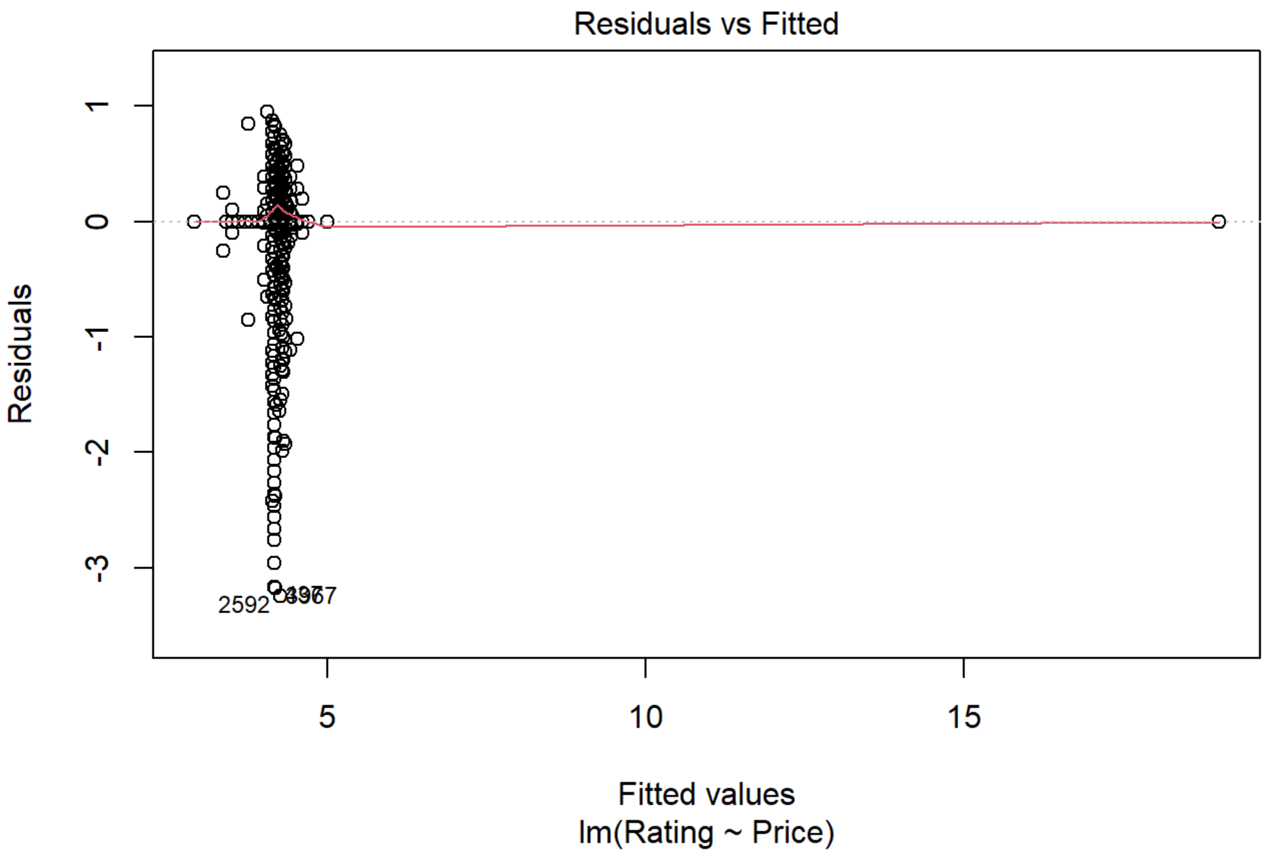
```
##
## Call:
## lm(formula = Rating ~ Price, data = googleplaystore)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2479 -0.1619  0.1381  0.3381  0.9500
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.16192    0.00672 619.301   <2e-16 ***
## Price0.99     0.12917    0.05555   2.325   0.0201 *
## Price1.00     0.28808    0.39189   0.735   0.4623
## Price1.20     0.03808    0.55417   0.069   0.9452
## Price1.29    -0.06192    0.55417  -0.112   0.9110
## Price1.49     0.01308    0.10494   0.125   0.9008
## Price1.50     0.03808    0.55417   0.069   0.9452
## Price1.61     0.03808    0.55417   0.069   0.9452
## Price1.70     0.13808    0.39189   0.352   0.7246
## Price1.75     0.83808    0.55417   1.512   0.1305
## Price1.76     0.33808    0.55417   0.610   0.5418
## Price1.97     0.33808    0.55417   0.610   0.5418
## Price1.99     0.13808    0.08111   1.702   0.0887 .
## Price10.00    0.43808    0.32000   1.369   0.1710
## Price10.99   -0.66192    0.39189  -1.689   0.0913 .
## Price11.99    0.13808    0.27715   0.498   0.6183
## Price12.99    0.26308    0.27715   0.949   0.3425
## Price13.99    0.13808    0.55417   0.249   0.8032
## Price14.00    0.43808    0.55417   0.791   0.4293
## Price14.99    0.13808    0.17536   0.787   0.4311
## Price15.46   -0.76192    0.55417  -1.375   0.1692
## Price15.99    0.53808    0.55417   0.971   0.3316
## Price16.99    0.03808    0.32000   0.119   0.9053
## Price17.99   -0.41192    0.39189  -1.051   0.2932
## Price18.99    0.43808    0.55417   0.791   0.4293
## Price19.40    0.53808    0.55417   0.971   0.3316
## Price19.99    0.35808    0.24791   1.444   0.1487
## Price2.00     0.23808    0.55417   0.430   0.6675
## Price2.49     0.11808    0.14323   0.824   0.4098
## Price2.56    -0.76192    0.55417  -1.375   0.1692
## Price2.59     0.53808    0.55417   0.971   0.3316
## Price2.90     0.03808    0.55417   0.069   0.9452
## Price2.99     0.08602    0.06520   1.319   0.1871
## Price24.99    0.21808    0.24791   0.880   0.3791
## Price29.99   -0.11192    0.22632  -0.495   0.6210
## Price299.99  -0.36192    0.55417  -0.653   0.5137
## Price3.02     0.03808    0.55417   0.069   0.9452
## Price3.04     0.83808    0.55417   1.512   0.1305
## Price3.08     0.23808    0.55417   0.430   0.6675
## Price3.28    -0.26192    0.55417  -0.473   0.6365
## Price3.49     0.12141    0.22632   0.536   0.5917
## Price3.88     0.43808    0.55417   0.791   0.4293
## Price3.99     0.16475    0.10139   1.625   0.1042
## Price33.99    -0.66192    0.39189  -1.689   0.0913 .
## Price37.99     0.03808    0.55417   0.069   0.9452
## Price379.99   -1.26192    0.55417  -2.277   0.0228 *
## Price389.99   -0.56192    0.55417  -1.014   0.3106
## Price39.99    -0.16192    0.55417  -0.292   0.7702
## Price399.99   -0.15283    0.16721  -0.914   0.3608
## Price4.29      0.43808    0.55417   0.791   0.4293
## Price4.49      0.25237    0.20955   1.204   0.2285
## Price4.60     -0.76192    0.55417  -1.375   0.1692
## Price4.77     -0.36192    0.55417  -0.653   0.5137
## Price4.84     -0.06192    0.55417  -0.112   0.9110
## Price4.99     -0.03785    0.07571  -0.500   0.6172
## Price400.00   -0.56192    0.55417  -1.014   0.3106
## Price5.49      0.43808    0.39189   1.118   0.2637
## Price5.99      0.10731    0.15384   0.698   0.4855
## Price6.49     -0.76192    0.55417  -1.375   0.1692
## Price6.99      0.02697    0.18483   0.146   0.8840
## Price7.49      0.03808    0.55417   0.069   0.9452
## Price7.99      0.18094    0.20955   0.863   0.3879
## Price79.99     0.43808    0.39189   1.118   0.2637
## Price8.49     -0.46192    0.55417  -0.834   0.4046
## Price8.99     -0.81192    0.39189  -2.072   0.0383 *
## Price9.00      0.03808    0.39189   0.097   0.9226
## Price9.99      0.07558    0.13870   0.545   0.5858
## PriceEveryone 14.83808    0.55417  26.775   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5541 on 7242 degrees of freedom
## Multiple R-squared:  0.09751,    Adjusted R-squared:  0.08916
## F-statistic: 11.68 on 67 and 7242 DF,  p-value: < 2.2e-16
```

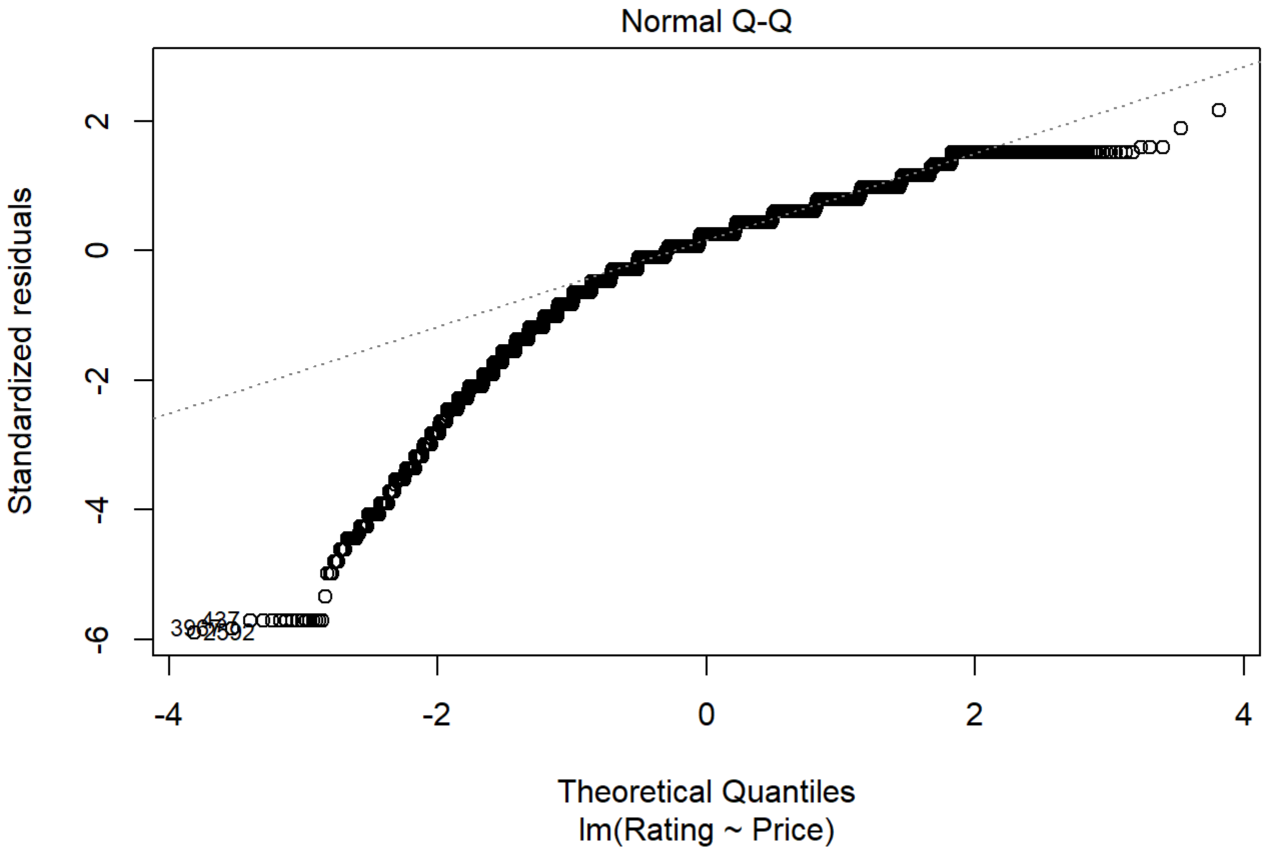
```
coefficients(lm_plot)
```

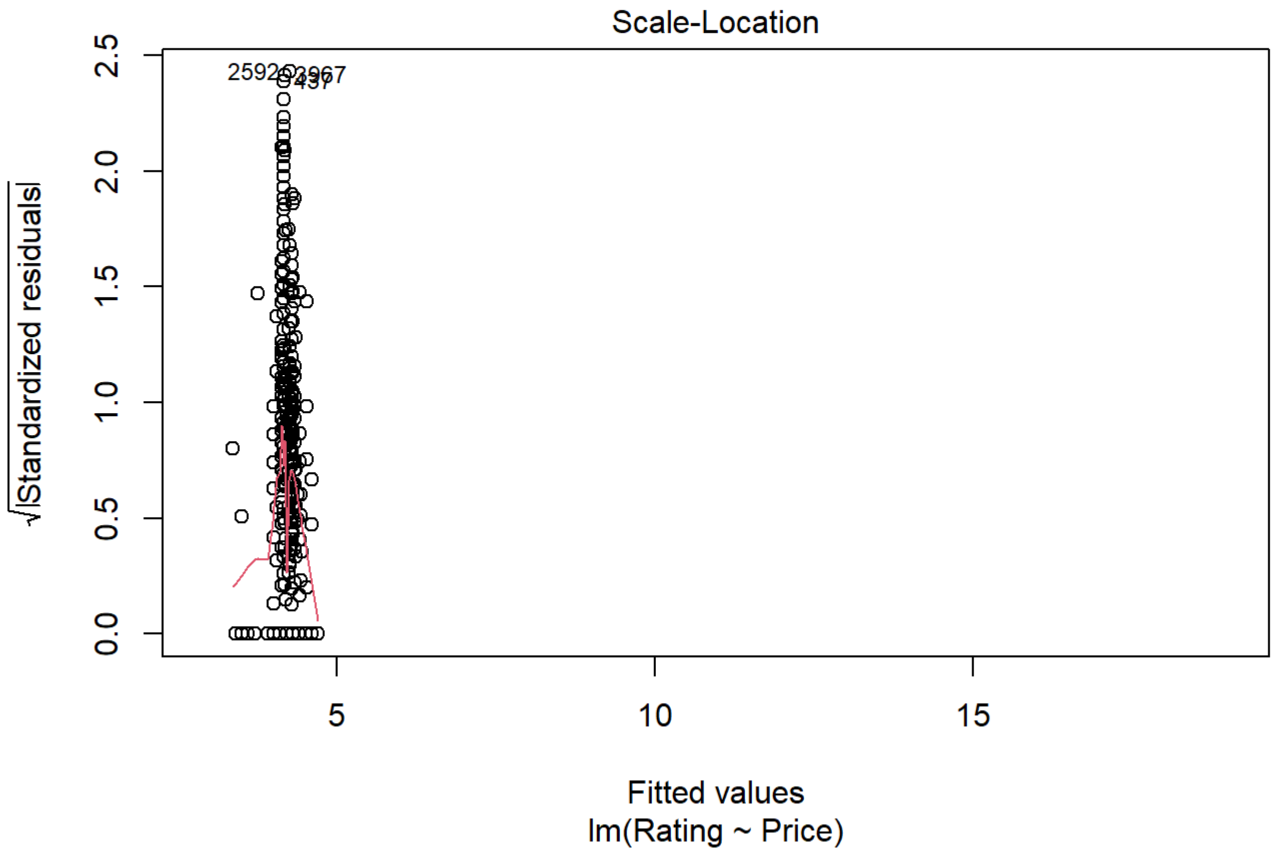
```
##   (Intercept)     Price0.99     Price1.00     Price1.20     Price1.29
##    4.16192087    0.12916824    0.28807913    0.03807913   -0.06192087
##     Price1.49     Price1.50     Price1.61     Price1.70     Price1.75
##    0.01307913    0.03807913    0.03807913    0.13807913    0.83807913
##     Price1.76     Price1.97     Price1.99    Price10.00    Price10.99
##    0.33807913    0.33807913    0.13807913    0.43807913   -0.66192087
##    Price11.99    Price12.99    Price13.99    Price14.00    Price14.99
##    0.13807913    0.26307913    0.13807913    0.43807913    0.13807913
##    Price15.46    Price15.99    Price16.99    Price17.99    Price18.99
##   -0.76192087    0.53807913    0.03807913   -0.41192087    0.43807913
##    Price19.40    Price19.99     Price2.00     Price2.49     Price2.56
##    0.53807913    0.35807913    0.23807913    0.11807913   -0.76192087
##     Price2.59     Price2.90     Price2.99    Price24.99    Price29.99
##    0.53807913    0.03807913    0.08602433    0.21807913   -0.11192087
##   Price299.99     Price3.02     Price3.04     Price3.08     Price3.28
##   -0.36192087    0.03807913    0.83807913    0.23807913   -0.26192087
##     Price3.49     Price3.88     Price3.99    Price33.99    Price37.99
##    0.12141246    0.43807913    0.16474580   -0.66192087    0.03807913
##   Price379.99   Price389.99    Price39.99   Price399.99     Price4.29
##   -1.26192087   -0.56192087   -0.16192087   -0.15282996    0.43807913
##     Price4.49     Price4.60     Price4.77     Price4.84     Price4.99
##    0.25236484   -0.76192087   -0.36192087   -0.06192087   -0.03784680
##   Price400.00     Price5.49     Price5.99     Price6.49     Price6.99
##   -0.56192087    0.43807913    0.10730990   -0.76192087    0.02696802
##     Price7.49     Price7.99    Price79.99     Price8.49     Price8.99
##    0.03807913    0.18093627    0.43807913   -0.46192087   -0.81192087
##     Price9.00     Price9.99 PriceEveryone
##    0.03807913    0.07557913   14.83807913
```

```
plot(lm_plot)
```

```
## Warning: not plotting observations with leverage one:
##   1413, 2619, 2924, 3510, 3512, 3521, 3958, 4224, 4298, 4537, 4862, 5542, 5868, 5885, 6663, 7
079, 7275
```

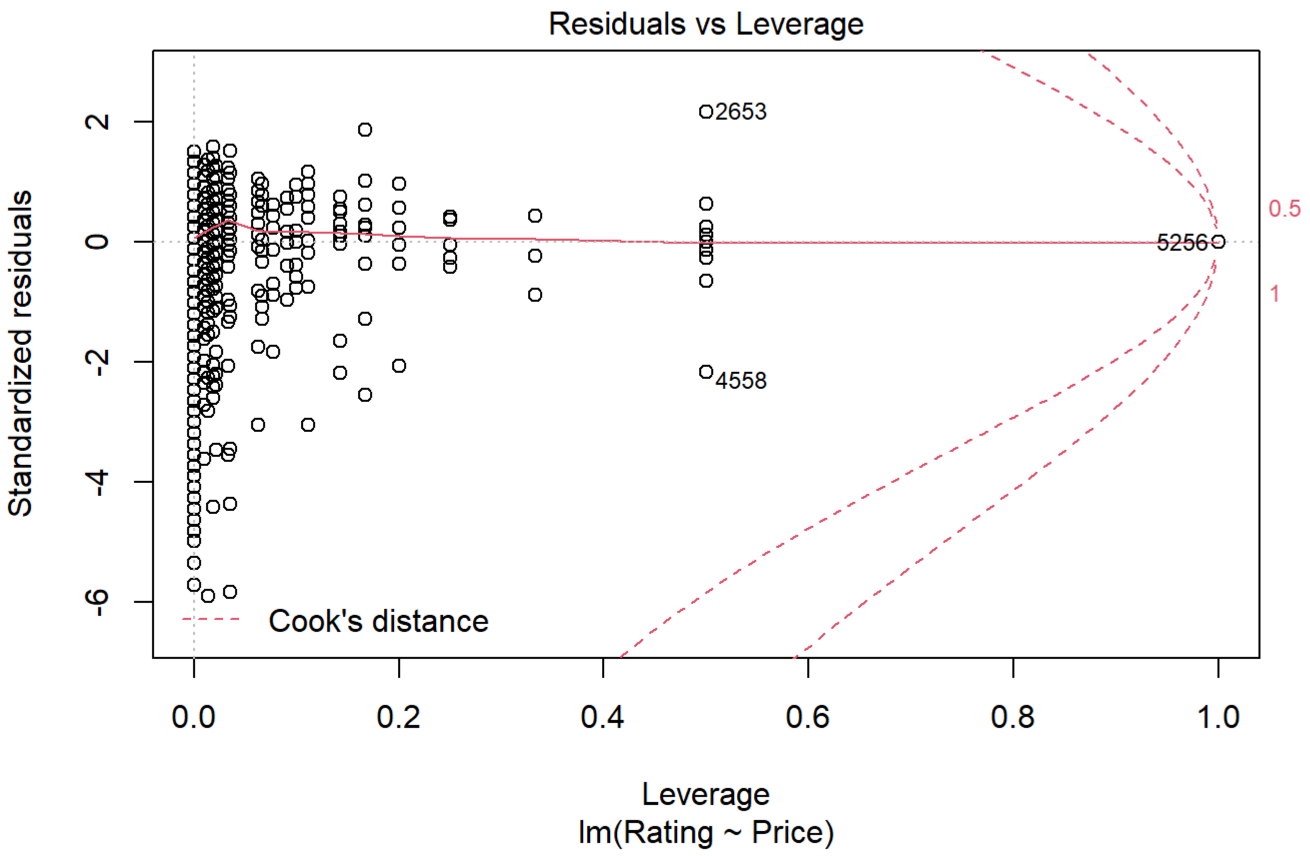

Residuals vs Fitted
Residuals
1
0
-1
-2
-3
2592
3967
437
5
10
15
Fitted values
lm(Rating ~ Price)


Normal Q-Q
Standardized residuals
2
0
-2
-4
-6
3967
437
2592
-4
-2
0
2
4
Theoretical Quantiles
lm(Rating ~ Price)



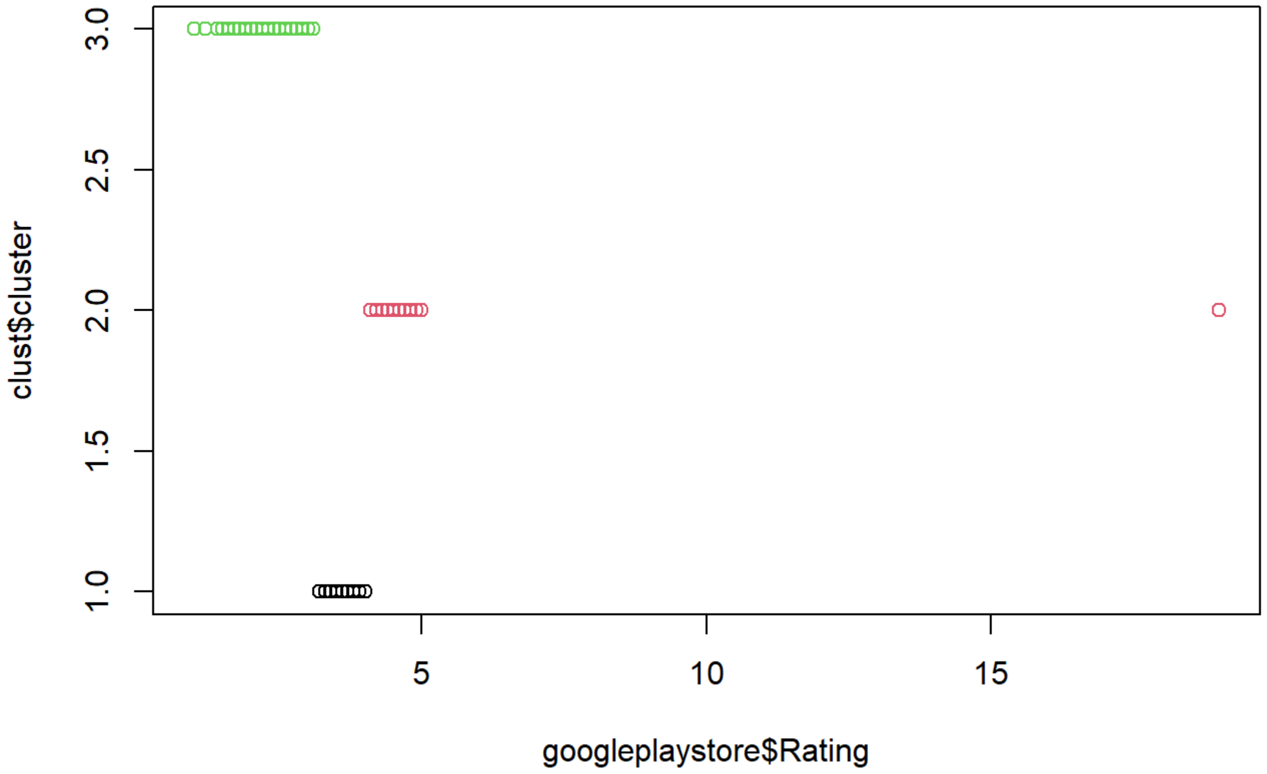
```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

```
## Warning in sqrt(crit * p * (1 - hh)/hh): NaNs produced
```

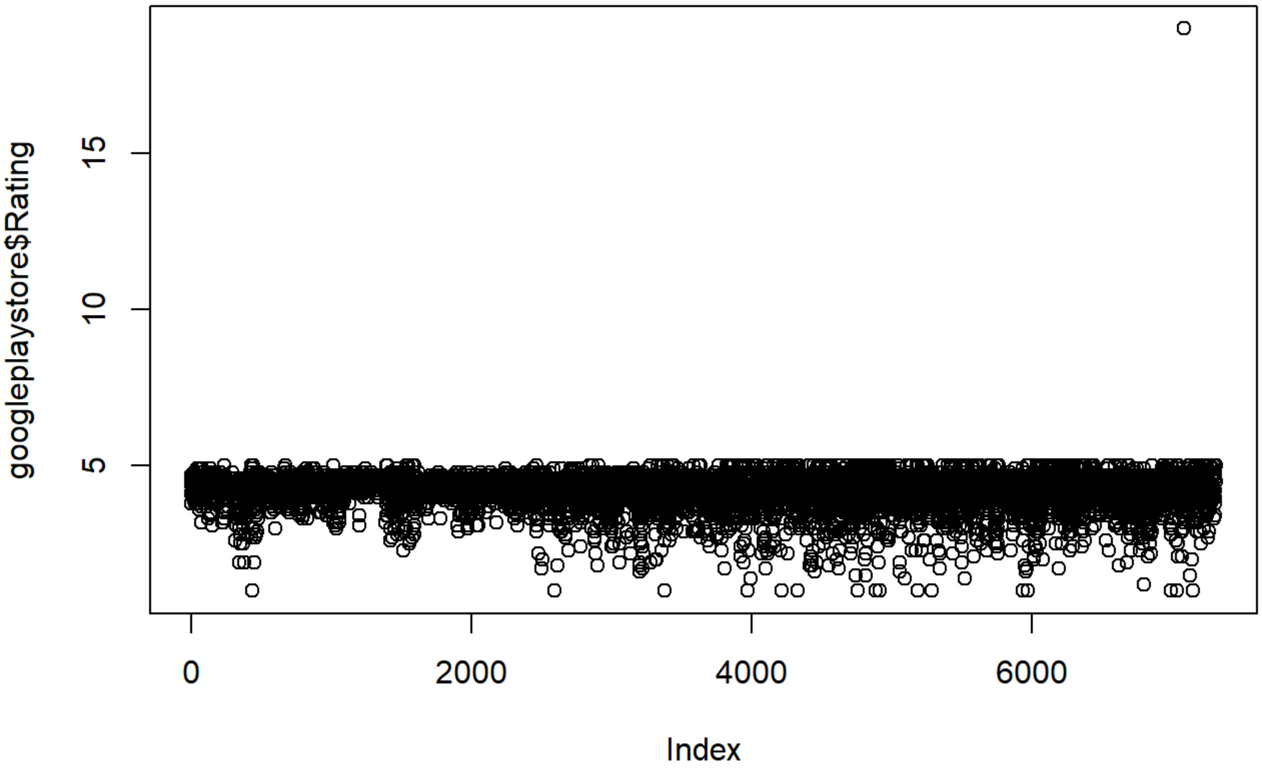


- In the code above, I used linear regression for the ratings and prices of all the apps to see their correlation.

```
#K-Means Cluster of rating in googleplaystore
clust <- kmeans(googleplaystore$Rating, centers = 3)
plot(googleplaystore$Rating, clust$cluster, col = factor(clust$cluster))
```



```
plot(googleplaystore$Rating)
```



- In the code above, I used K_Means Clustering for the ratings of all the apps to see the huge range of ratings in the dataset.

Data Product

```
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 4.0.3
```

```
ui <- fluidPage(
  titlePanel("Rating Comparison With All The Apps In the AppStore"),
  sidebarPanel(
    numericInput('clusters', 'Ratings', 4,
                 min = 1, max = 5),
    numericInput('clusters', 'Total', 50000,
                 min = 0, max = 100000)
  ),
  mainPanel(plotOutput('plot'))
)

server <- function(input, output){
  clusters <- reactive({
    kmeans(googleplaystore$Rating, centers = 3)
  })

  output$plot <- renderPlot({
    plot(googleplaystore$Rating,
         col = clusters()$cluster,
         pch = 20, cex = 3)
  })
}

shinyApp(ui, server)
```

Shiny applications not supported in static R Markdown documents

- In the code above, I used r shiny to create a web application that displays the relationship of app ratings with other apps. I made two user interfaces where the user can change the rating and the amount of apps to see the interaction between the two variables.