IST 687

Final Project

Group 3:

Gina Clark

Yea Rin Kim

Shayan Orellana

Andrew Dobkowski

Elizabeth Westbrook

Table of Contents

Introduction

Group 3 initially decided between two datasets found on Kaggle, one about Olympic athletes and the other about movies on streaming services. We decided the streaming service data would likely be more fruitful, as the host of variables seemed ripe for analyzing.

The data contains such variables as movie title, release year, movie ratings from IMDb and Rotten Tomatoes, whether a movie is being hosted on any of four streaming platforms (Netflix, Hulu, Prime Video and Disney+), movie genres, languages in which the film is available, and total runtime.

Our initial (simple) business questions were developed early in the process. We eventually each selected a business question to answer. In most cases, our initial business question led us to develop more interesting, advanced, and useful questions.

Business Question 1

What platforms have the longest runtimes, and can we make any business interpretations from our results?

We wanted to look at movies across different streaming platforms to compare/contrast the different runtime strategies taken by each platform. As a team, we downloaded a streaming platform dataset from Kaggle as a Microsoft Excel document. I imported this dataset into R using the following strategy:

- library(readxl)
- data <- read_excel('C:\\Users\\andre\\OneDrive\\Documents\\IST-687\\NetflixGroup.xlsx')

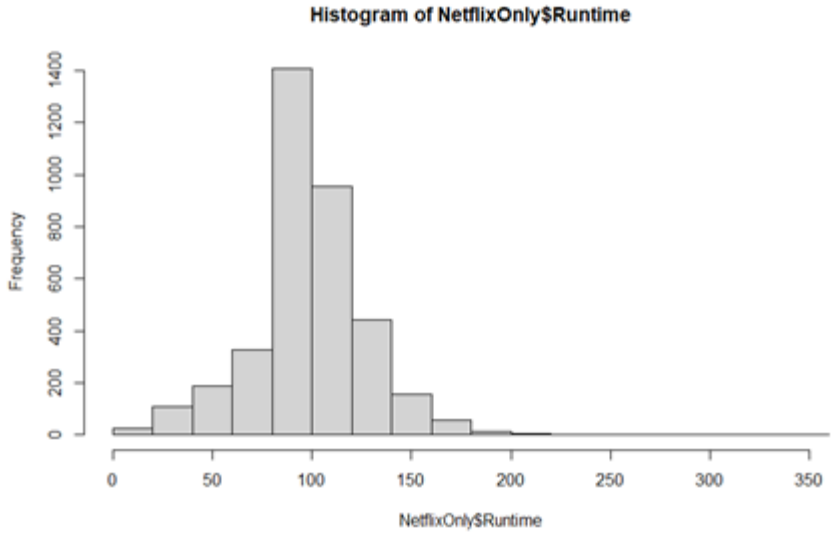
The next step I took was creating a data frame. Then, I had to clean the dataset to ensure all values were in the numeric format and replaced all "NAs" with the mean value as described in class:

- df <- data.frame(data)
- df$Runtime <- as.numeric(df$Runtime)
- df <- df[,-27:-35]
- is.na(df$Runtime)
- df$Runtime[is.na(df$Runtime)] <- mean(df$Runtime, na.rm=TRUE)

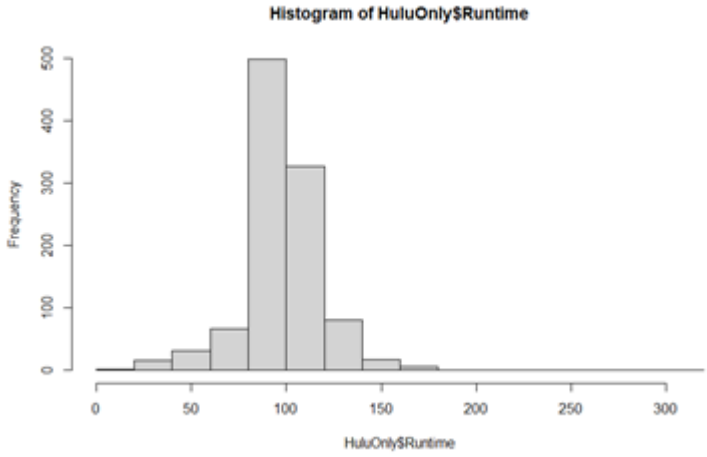
Next, I observed the structure of the data frame. While reviewing the data frame, I noticed that I needed to divide my data frame into subsets for each streaming platform. This allowed me to properly compare the differences between each of the platforms.

The histograms of each of the data frames allow us to recognize the distributions of each data frame and the descriptive statistics allow us to review each of the structures.

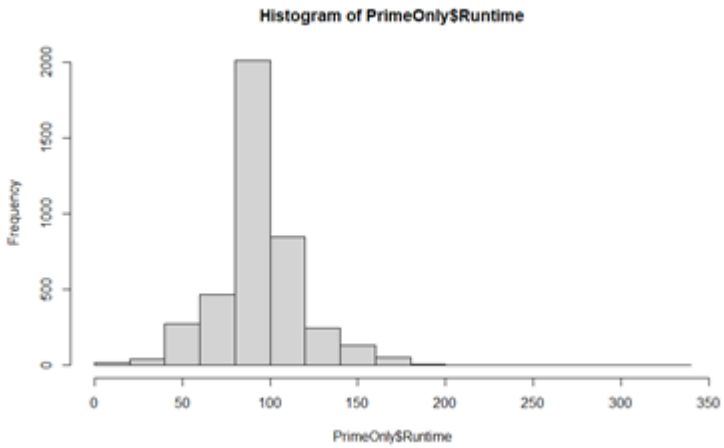
- Netflix
    - Mean = 99.52 min
    - Median = 99.519 min
    - Min = 3 min
    - Max = 359 min



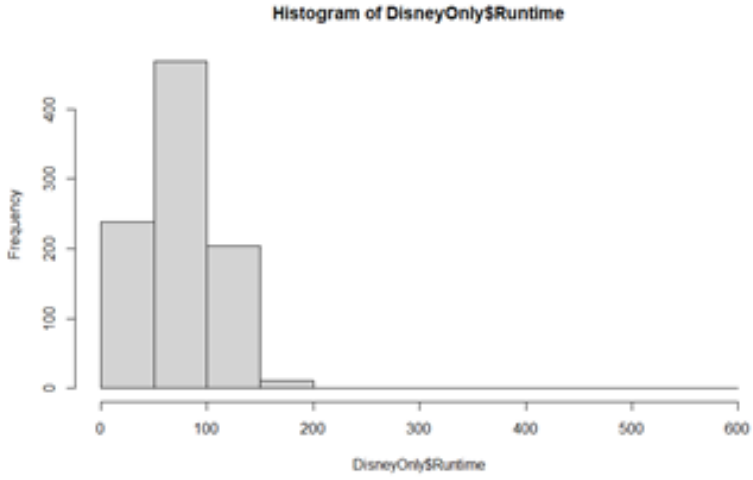
- Hulu
    - Mean = 98.32 min
    - Median = 98 min
    - Min = 5 min
    - Max = 317 min



- Amazon Prime
    - Mean = 95.22 min
    - Median = 94 min
    - Min = 2 min
    - Max = 328 min



- Disney+
  - Mean = 75.77 min
  - Median = 85 min
  - Min = 1 min
  - Max = 566 min

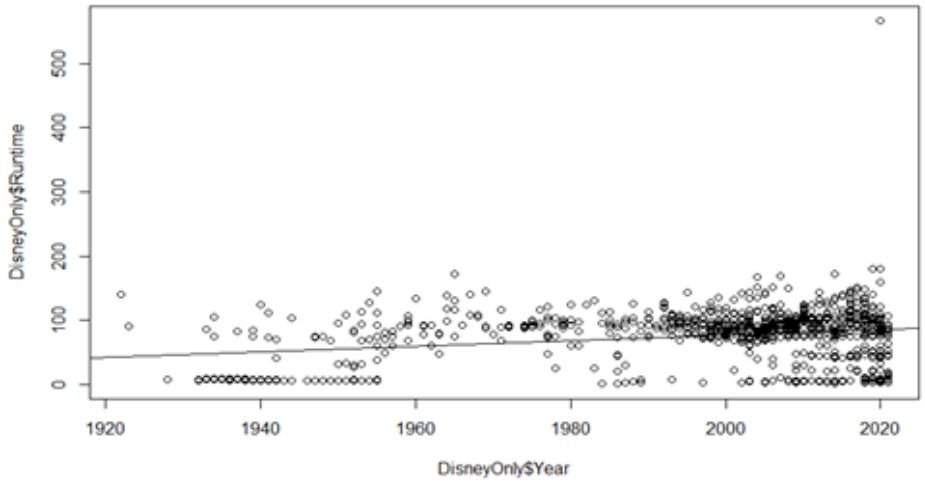


There is a clear difference between the runtimes of Disney+ and the other platforms. The mean for the Disney platform is 75 minutes; meanwhile, the competing platforms' average runtimes are in the mid to high 90s. The reason for the difference seems likely to be Disney's consideration of its target audience. Disney's audience is primarily children with shorter attention spans than adults. Therefore, it makes complete sense that Disney will have shorter runtimes than their competitors.

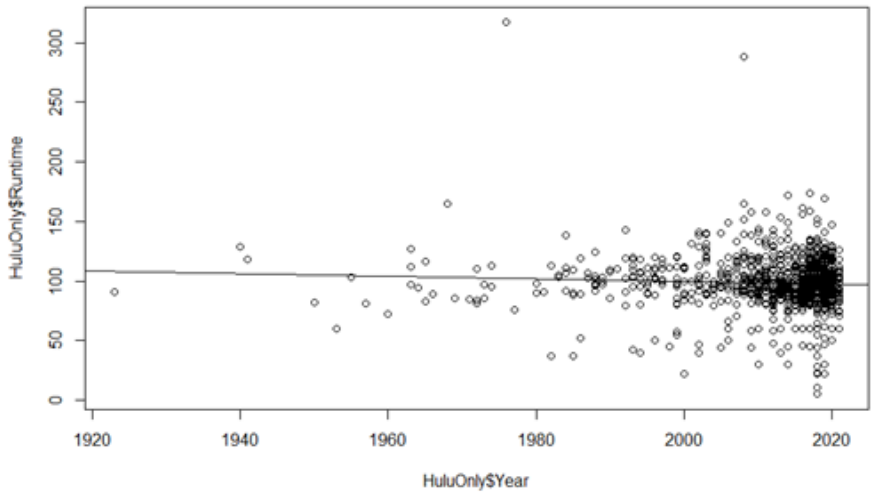
The next topic I wanted to analyze was runtimes over time. Have movies been getting longer or shorter over time? I decided to plot each platform with their runtime over time. Then, I created a linear model for each platform to predict the average runtime for each platform for a given year in the future:

- plot(DisneyOnly$Year, DisneyOnly$Runtime)

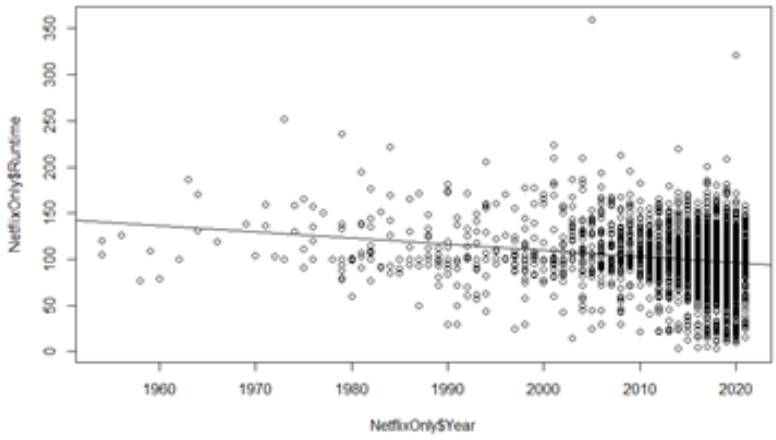
- LM1 <- lm(Runtime ~ Year, data = DisneyOnly)
- plot(DisneyOnly$Year, DisneyOnly$Runtime)
- summary(LM1)
- abline(LM1)
- predict(LM1)
- #2025 = 82.84



- Plot(HuluOnly$Year, HuluOnly$Runtime)
- LM2 <- lm(Runtime ~ Year, data = HuluOnly)
- plot(HuluOnly$Year, HuluOnly$Runtime)
- summary(LM2)
- abline(LM2)
- predict(LM2)
- #2022 =98.08

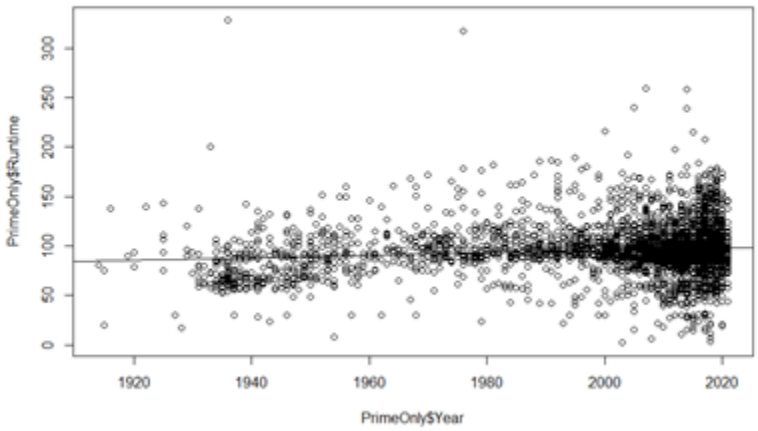


- plot(NetflixOnly$Year, NetflixOnly$Runtime)
- LM3 <- lm(Runtime ~ Year, data = NetflixOnly)
- plot(NetflixOnly$Year, NetflixOnly$Runtime)
- summary(LM3)
- abline(LM3)
- predict(LM3)
- #2026 = 98.69



- plot(PrimeOnly$Year, PrimeOnly$Runtime)
- LM4 <- lm(Runtime ~ Year, data = PrimeOnly)
- plot(PrimeOnly$Year, PrimeOnly$Runtime)

- summary(LM4)
- abline(LM4)
- predict(LM4)
- #2024 = 90.59



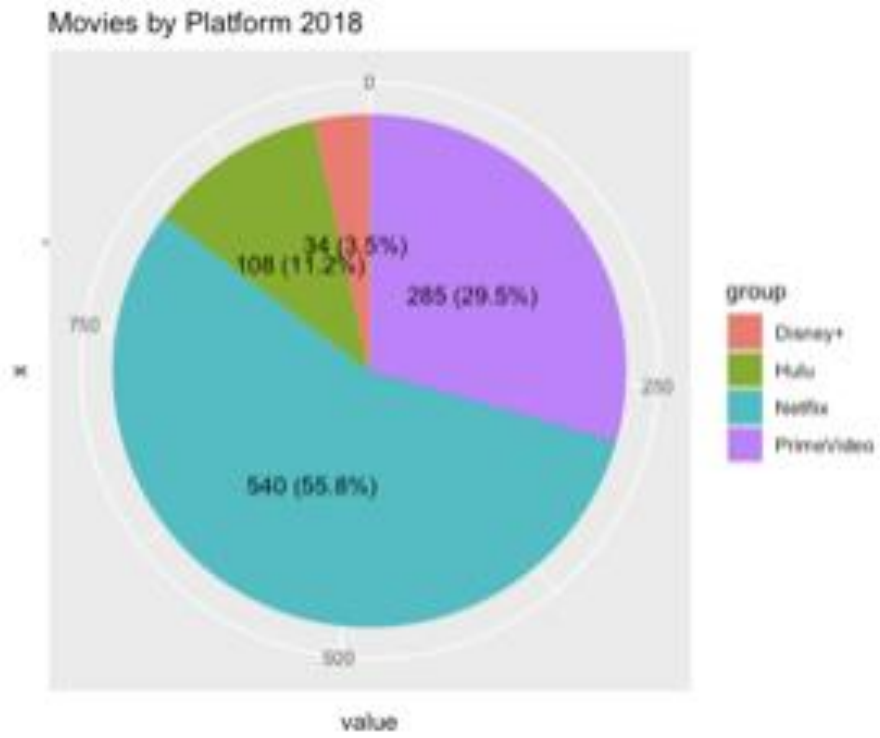
The linear models clearly illustrate that all platforms are trying to keep their movies under 100 minutes and all trendlines point towards around 90 minutes moving forward.

Business Question 2

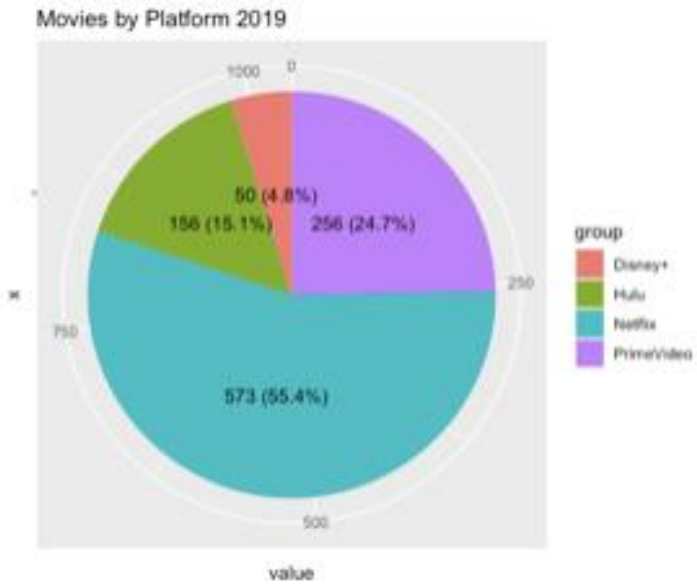
How many movies from the years 2018-2021 do different streaming services have?

We wanted to take the number of movies on different streaming services and explore if we could gain insights into market shares that each service has. Currently, Netflix has more than 200 million global subscribers. However, as the streaming war becomes more fierce with other competitors emerging into the market, we dived into the data to see how many recent movies (2018-2021) each service streams. There is an assumption made in the study, however. We assumed that the market shares are purely based on having the most number of movies that were released in the year.

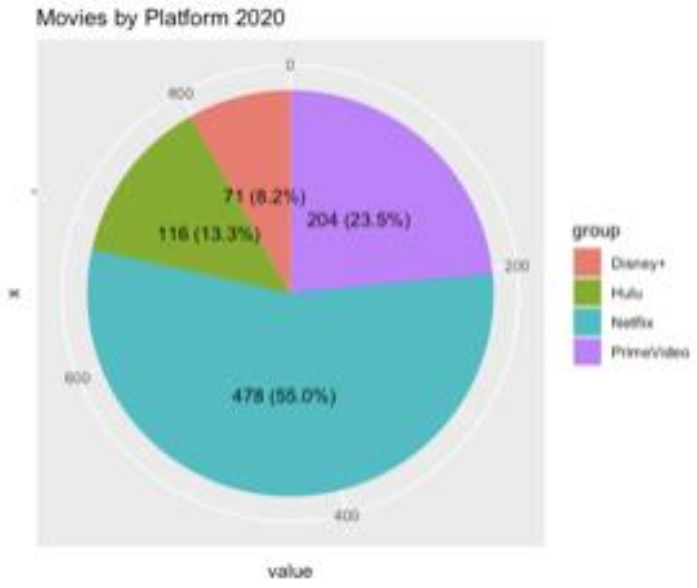
For movies released in 2018, Netflix had the largest share of it by having 540 of them. Amazon Prime had the second largest, and Disney Plus had the least which was 3.5% of the total.



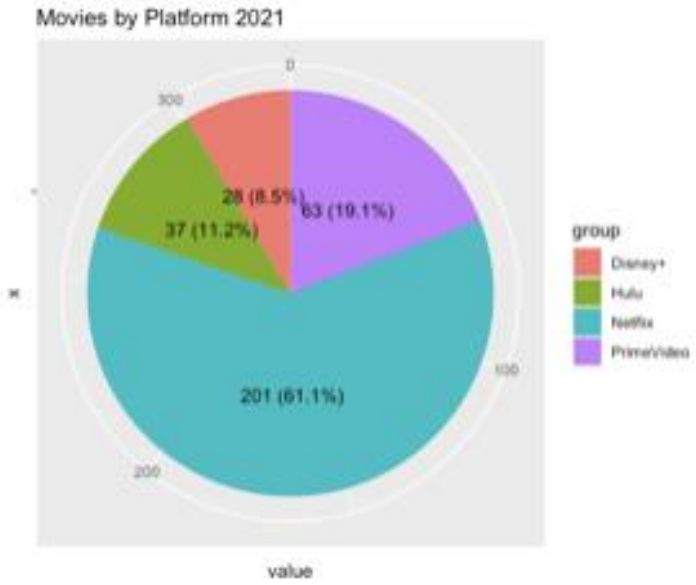
For movies released in 2019, Netflix again took the first place by having 573 movies on its service. It's worth making a note that Disney is gradually increasing shares in the market, as 2019 is the year that Disney Plus was launched, and also acquired full control over Hulu. The company started serving its customers on their own platform instead of collaborating with other streaming partners. Amazon Prime kept its second place, followed by Hulu.



What was most surprising about our study was year 2020. With the entire world going into a shelter-in-place, we held an assumption that the pandemic halted film production and release for 2020. Amazon Prime, Hulu, and Disney Plus all expanded their shares by having more TV shows and movies on their service, but there was a slight decrease in number of movies that Netflix deployed. This could well be because of an intensified competition created by different streaming services.



In year 2021, number of movies launched in general were reduced by half but market shares by different platforms remained the same. This makes sense because the data was generated by the mid-point of the year.



We also ran a multiple regression based on the hypothesis that if one streaming service has a vastly greater number of movies than the other streaming services over the past 3 years the other streaming services will necessarily have fewer because there are a limited number of movies. Results showed all of the streaming services had significant p-value in the multiple regression with the model's accuracy of about 89%. P-value also indicates that there is less than 1% that this happens by coincidence because p-value of the model is less than 0.01.

```
> #multiple regression
> fit<-lm(Netflix ~ Hulu + PrimeVideo + `Disney+`, data=df2)
> summary(fit)

Call:
lm(formula = Netflix ~ Hulu + PrimeVideo + `Disney+`, data = df2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.12973 -0.04973 -0.04973  0.02953  2.73653

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.970473   0.002670   363.4   <2e-16 ***
Hulu        -0.840743   0.005546  -151.6   <2e-16 ***
PrimeVideo  -0.920746   0.003624  -254.1   <2e-16 ***
`Disney+`   -0.945512   0.005927  -159.5   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1615 on 9511 degrees of freedom
Multiple R-squared:  0.8903,    Adjusted R-squared:  0.8903
F-statistic: 2.573e+04 on 3 and 9511 DF,  p-value: < 2.2e-16
```

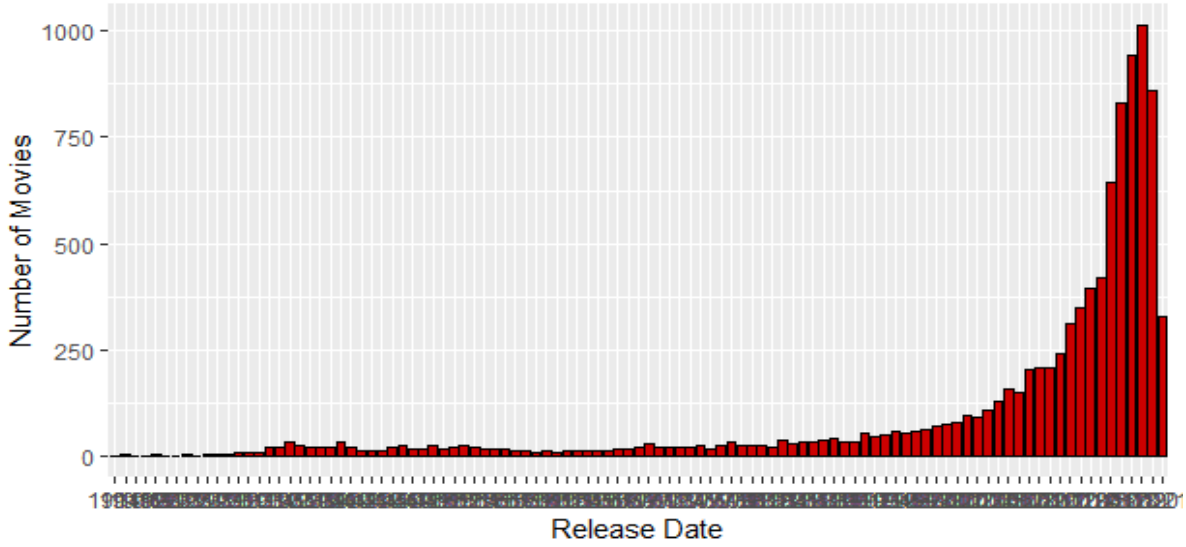
Based on the patterns observed, Netflix has a strongest footing in the market even though there is a fierce competition going on. Other competitors like Amazon and Disney+ continue to

invest heavily in streaming and because of the competition, now consumers have a lower-cost alternatives to Netflix. Netflix can only stay afloat in the market by securing more movies. It'd be also interesting to see pricing structures of different platforms to see how they play in gaining shares in the market.

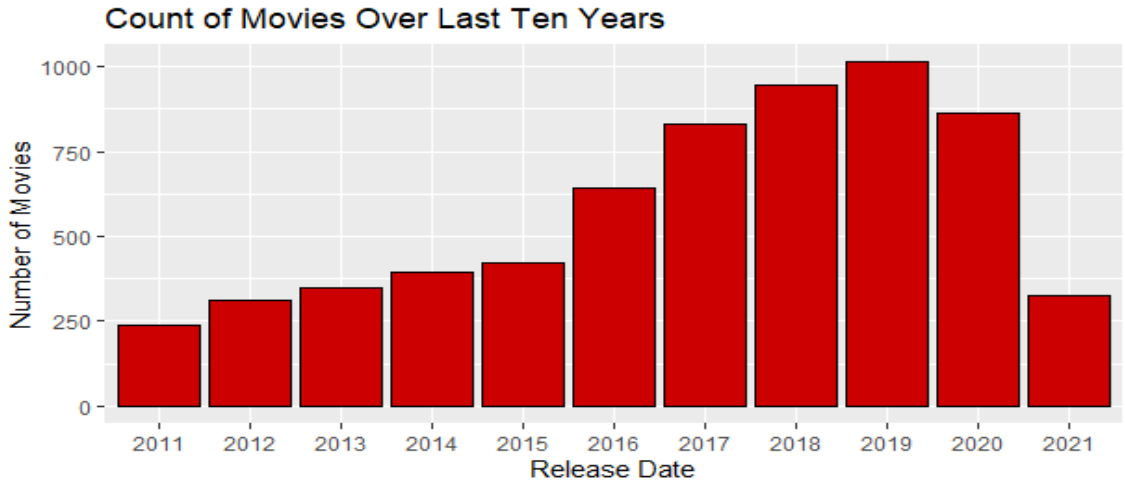
Business Question 3

Our third business question asked which release year was the most frequent across all the streaming services. It was hypothesized that, as a result of the COVID 19 pandemic, the movie industry experienced a decline in movies produced in the years following 2020, which in turn affected the number of movies released on the streaming services in 2020 and 2021.

Looking at the dataset provided by Kaggle spanning over the last 107 years, there is a sharp increase in the number of movies released in recent years. However, this is such a large scope that is hard to even view the years on the chart below.

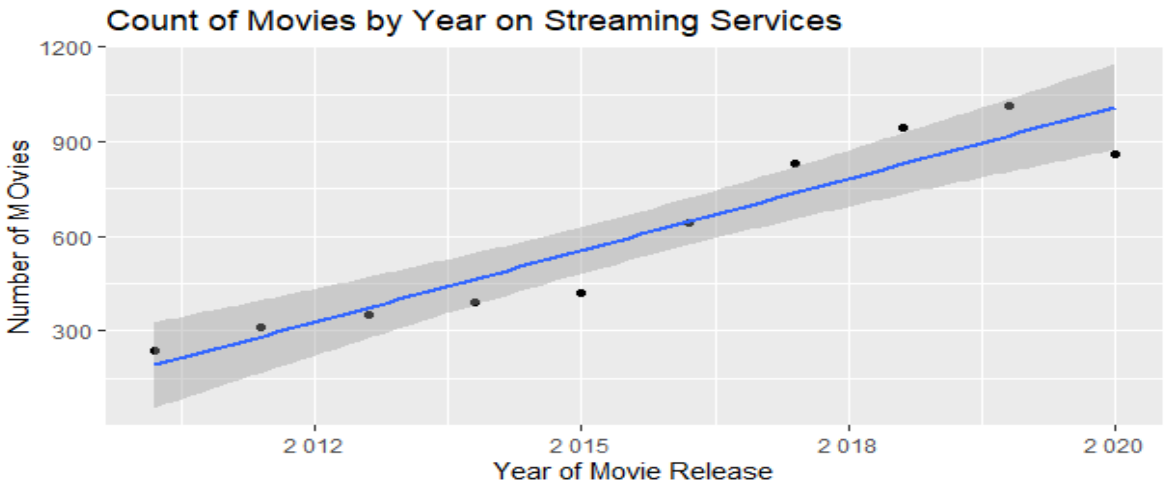


Therefore, the scope was narrowed to the frequency of releases over the last ten years.



Now it is easier to visualize the steady increase in the count of movies released from 2011 to 2019, and the decline in movie production in 2020 and 2021. Although it cannot be concluded from this graph what caused the decline, there is not sufficient evidence to dismiss the hypothesis that the COVID 19 pandemic contributed to a decline in movie production and release in 2020 and thus far in 2021.

This spurs the question, what kinds of numbers of movie releases should we have expected to see going forward? Assuming that movies are still being produced for the remainder of 2021 and that the streaming services will continue to add movies with release dates in 2021, predict the number of movies to be released in that year. To answer this, the lm function was used in R and the count of movies are plotted on a point chart and a best fit line drawn on the graph below.



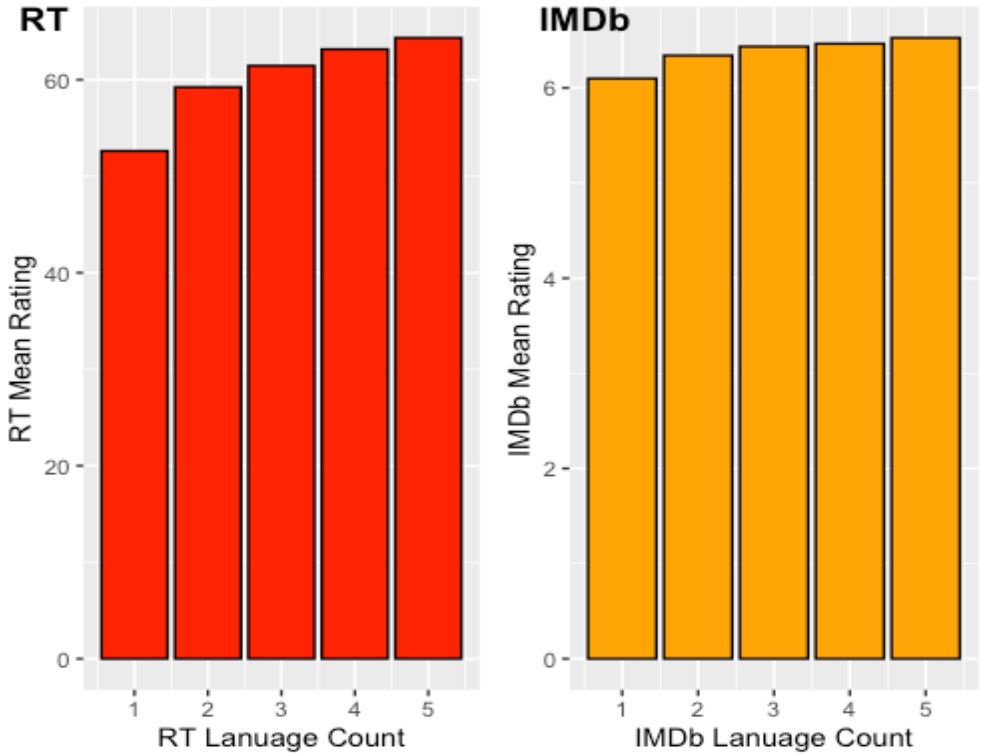
This lm model predicts a total of 1,010 movies are estimated to be released in 2021. Currently, the number of movies released in 2021 on the streaming sites totals 327. This is a deficit of 683 movies. It will be interesting to run this analysis again in five years to see if the number of movies featured with 2021 release dates grows to reach the predicted total or even reaches the lower limit generated by this prediction of 874 movies.

Business Question 4

The fourth business question called for us to explore the relationship between review aggregation ratings and languages. In our dataset, the main language consisted of the original movie language (OML) to be defined as the language of production. The secondary to *nth* language (up to 14 languages were possible) used were either completely dubbed with additional native speech actors or using text subtitles.

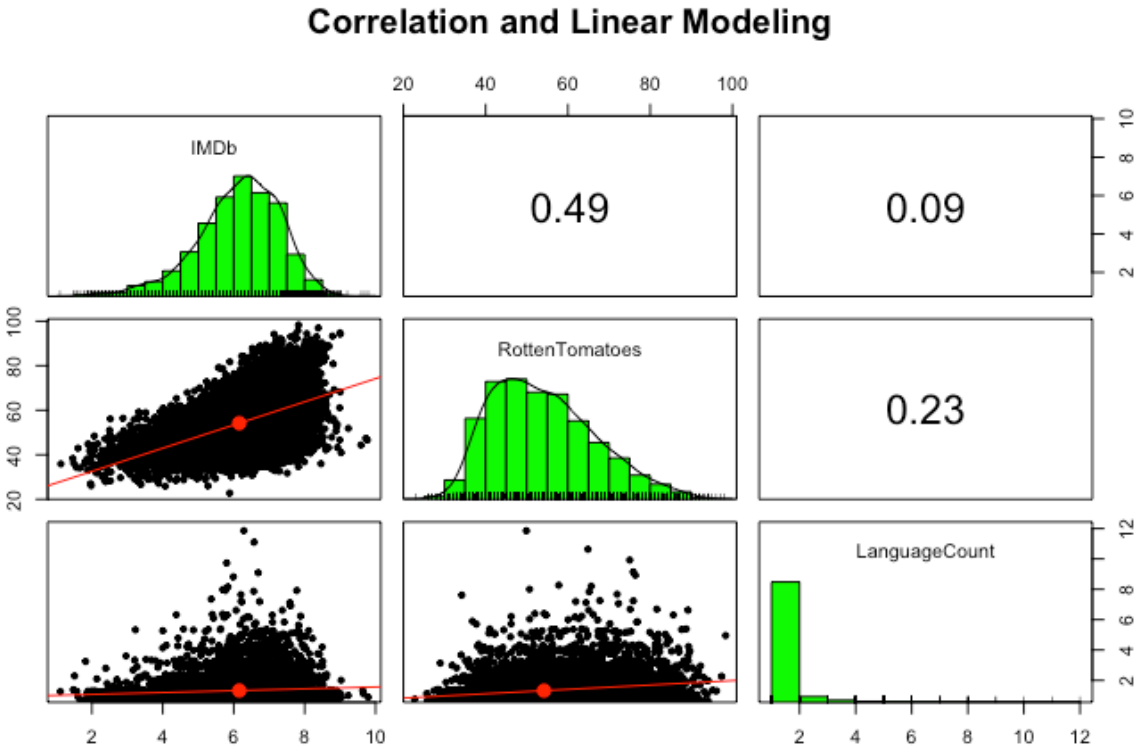
We wanted to test a hypothesis that if a movie had a larger number of languages would mean higher average movie rating. This hypothesis tested if having a larger language catalog equated to more popularity, thus leading to a higher *quality* of film which would ultimately receive a higher rating from users and critics.

At first glance below, the overview of language count relative to mean ratings appears to have a positive correlation. In order to definitively consider this significance to be substantial, we must take a more statistical approach rather than a top-down view on the hypothesis at-hand.

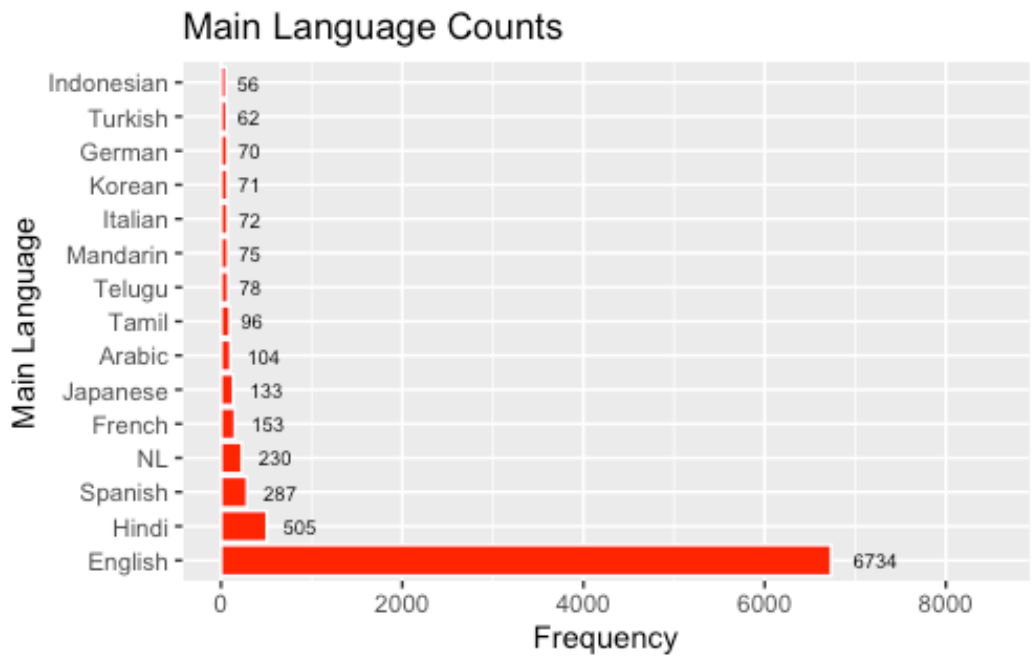


We used the psych library to plot the correlation coefficients along with the linear fit of the attributes. Ratings from IMDb and Rotten Tomatoes have a correlation coefficient of 0.49, which is perceived to be relatively strong.

The correlation coefficient for both IMDb and Rotten Tomatoes were both below values of 0.25, with Rotten Tomatoes and IMDb having a correlation score of 0.23 and score of 0.09 to language count, respectively. According to Pearson's and Spearman's coefficient, these values are considered to indicate an absence of correlation. Therefore, our hypothesis of a significant correlation between language count and aggregate ratings from IMDb and Rotten Tomatoes is rejected.



For the second part of the question, we wanted to create a classification model of movies with English as the original movie language (OML) compared to those in other languages. When exploring the language attribute in the dataset, we found an overwhelming number of OMLs in English.



We posed the question as to how well the dataset can predict whether a movie has an English OML or not. For this example, Naïve Bayes classification fit our model best with the large number of numerical and categorical attributes. Using an 'ifelse' statement to assign a 1 to English OMLs and a 0 to non-English OMLs, the Naïve Bayes model returned with a 98% accuracy in depicting the classifier.

```
> confusionMatrix(p, test$EnglishOrNot)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  721    7
         1   42 1995

               Accuracy : 0.9823
                 95% CI : (0.9766, 0.9869)
```

Business Question 5

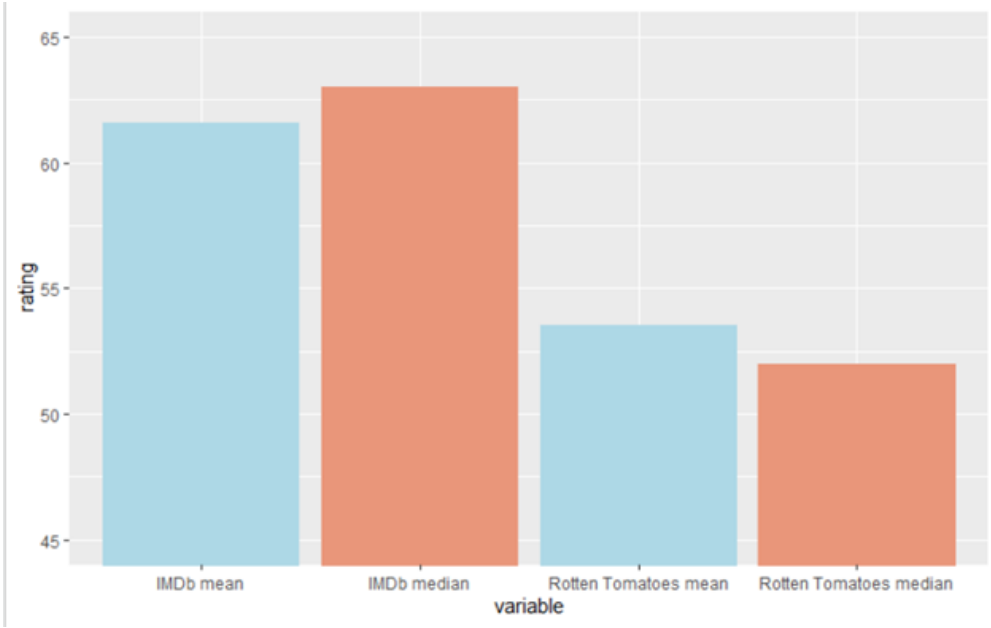
Which movie-rating platform rates movies higher? IMDb or Rotten Tomatoes?

We wanted to look at which ratings platform rated movies higher. IMDb polls users for their movie ratings, and movies are rated on a scale from 1 to 10 stars. Rotten Tomatoes, on the other hand, is known for its critical reviews; professional movie critics rate films on the "tomatometer," from 1-100%. Would critical ratings be higher or lower than general user ratings? We looked at two measures of central tendency, mean and median.

The initial tasks here involved munging the data so we could actually compare apples to apples (well, tomatoes, I suppose).

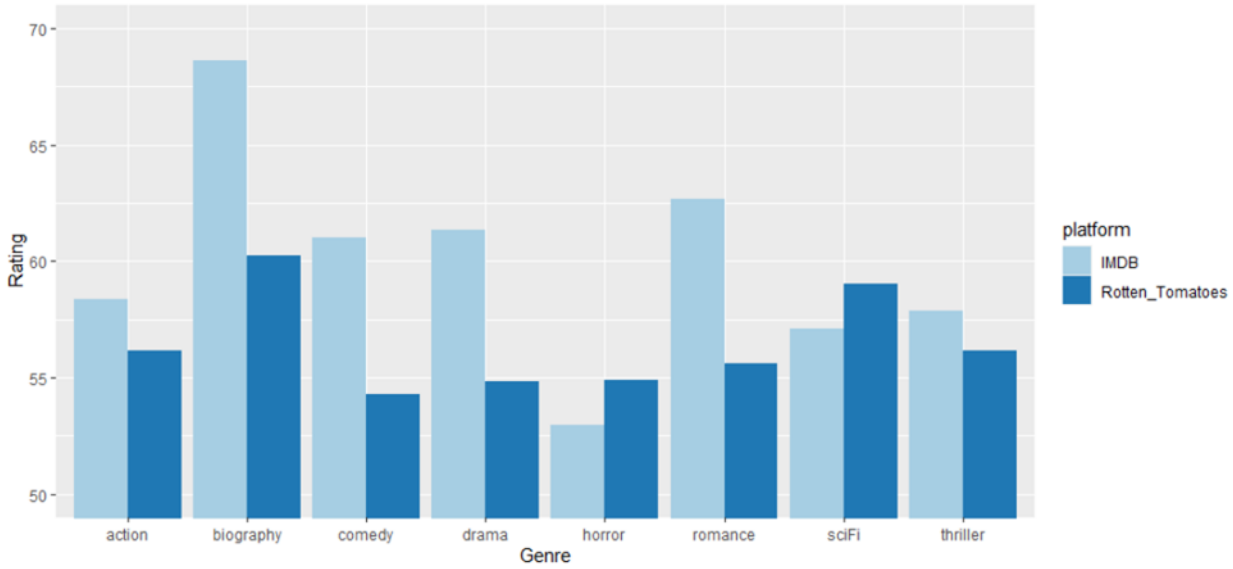
```
sData$IMDb <- gsub ("/10", "", sData$IMDb)
sData$RottenTomatoes <- gsub ("/100", "", sData$RottenTomatoes)
Numberize <-function (inputVector) {
  inputVector <-gsub (" ", "", inputVector)
  return (as.numeric(inputVector))
}
sData$IMDb <- Numberize(sData$IMDb)
sData$IMDb <- sData$IMDb * 10
sData$RottenTomatoes <- Numberize(sData$RottenTomatoes)
```

Then means and medians were calculated across both platforms, removing NAs in the process. After creating a data frame, this is the resulting visualization.



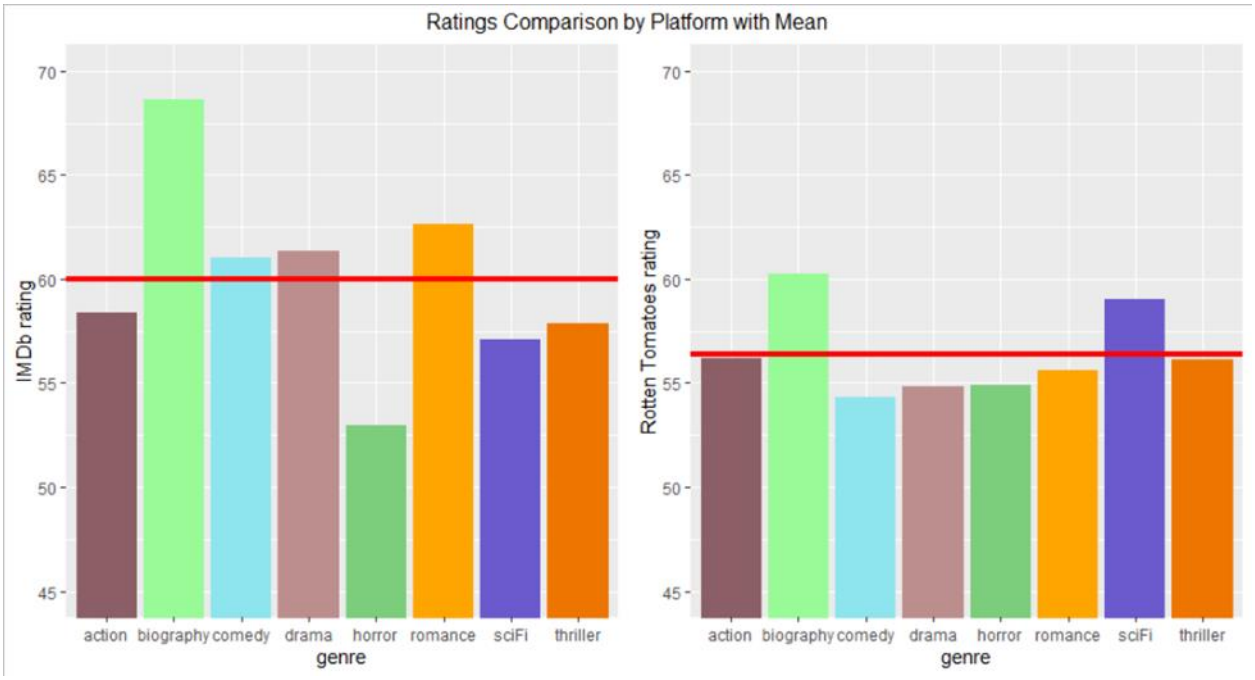
We can see that IMDB users rate movies higher than Rotten Tomatoes users. But does this trend hold true across genres?

We decided that looking at ALL ratings—for all genres of film—was quite broad. We decided that a genre-by-genre comparison of movie ratings would help us determine if Rotten Tomatoes users rate all movies lower than IMDb users. To answer this question, we selected eight common movie genres and compared the average rating for each genre across both platforms.



Here we can see ratings by genre and platform. We can see that in eight genres, IMDB users rate movies higher than Rotten Tomatoes critics. However, Rotten Tomatoes critics rate horror and sci-fi higher than IMDb users.

This led to another questions: which ratings platform is most consistent in its ratings?



Putting the graphs side by side, as above, shows less variance in the Rotten Tomatoes ratings. The smaller standard deviation among Rotten Tomatoes ratings shows more consistency among the critics giving the ratings.

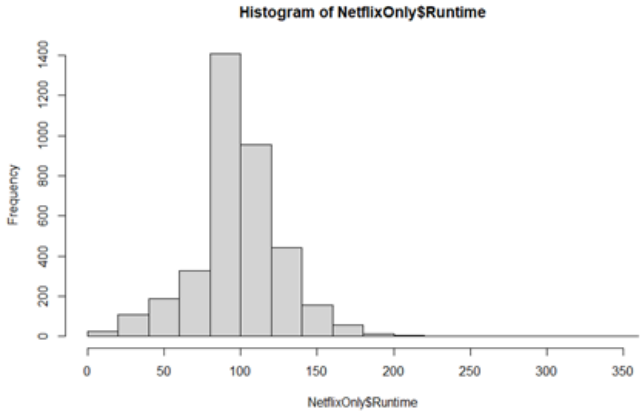
```
> mean(ratings2$IMDB)
[1] 59.99281
> sd(ratings2$IMDB)
[1] 4.623158
> mean(ratings2$Rotten_Tomatoes)
[1] 56.40997
> sd(ratings2$Rotten_Tomatoes)
[1] 2.113434
```

These findings are most useful for those deciding which ratings platform they'd prefer—or trust—more. For serious film buffs, Rotten Tomatoes ratings might be more in line with their own views. However, casual movie watchers might just want to know how the average Joe rated a movie; in that case, IMDb might be the preferred rating system.

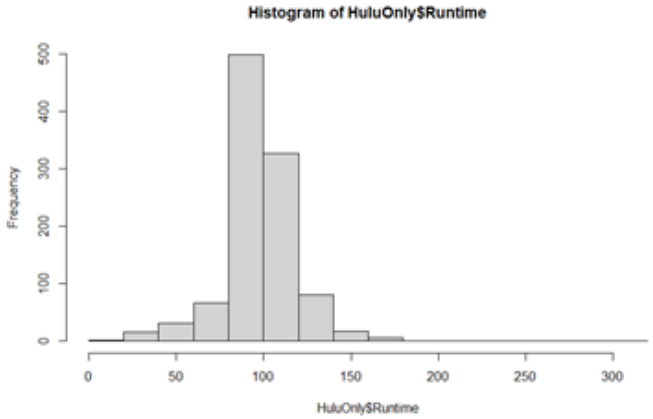
Appendix 1 Code: Business Question 1

```
> #Business Question 1: What platforms have the longest Run times and are there any trends?
> #Import the Data from Excel
> library(readxl)
> data <- read_excel('C:\\Users\\andre\\OneDrive\\Documents\\IST-687\\NetflixGroup.xlsx')
> #Create a data frame
> df <- data.frame(data)
> #Clean the data Set
> df$Runtime <- as.numeric(df$Runtime)
> df <- df[,-27:-35]
> is.na(df$Runtime)
> df$Runtime[is.na(df$Runtime)] <- mean(df$Runtime, na.rm=TRUE)
> str(df$Runtime)
 num [1:9515] 209 161 139 83 152 52 99 224 94 139 ...
> summary(df$Runtime)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0    85.0    95.2    95.2   108.0   566.0
> min(df$Runtime)
[1] 1
> max(df$Runtime)
[1] 566
> median(df$Runtime)
[1] 95.19943
> mean(df$Runtime)
[1] 95.19943
> #Create Data Frame for Netflix Only movies and clean it
> NetflixOnly <- data.frame(data)
> #Create Data Frame for Netflix Only movies and clean it
> NetflixOnly <- data.frame(data)
> NetflixOnly <- NetflixOnly[order(NetflixOnly$Netflix),]
> NetflixOnly <- NetflixOnly[-1:-5820,]
> NetflixOnly$Runtime[is.na(NetflixOnly$Runtime)] <- mean(NetflixOnly$Runtime, na.rm=TRUE)
> #look at the structure of the Netflix only data
> str(NetflixOnly$Runtime)
 num [1:3695] 209 161 83 52 99 224 94 120 118 133 ...
> summary(NetflixOnly$Runtime)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00   88.00   99.52   99.52  114.00  359.00
> min(NetflixOnly$Runtime)
[1] 3
> max(NetflixOnly$Runtime)
[1] 359
> median(NetflixOnly$Runtime)
```

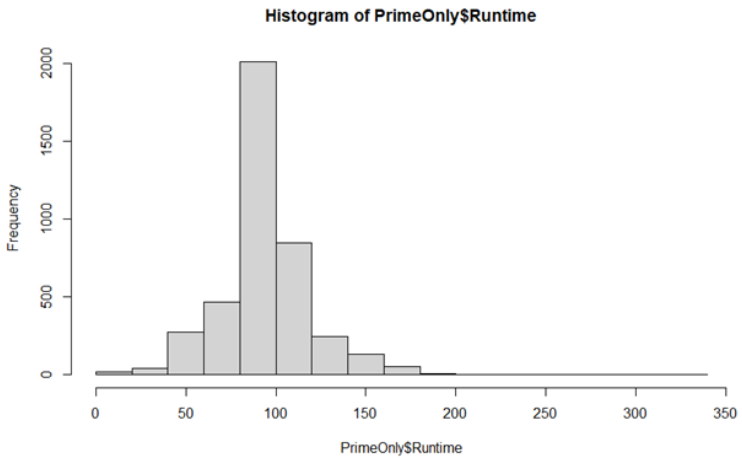
```
[1] 99.51952
> mean(NetflixOnly$Runtime)
[1] 99.51952
> hist(NetflixOnly$Runtime)
```



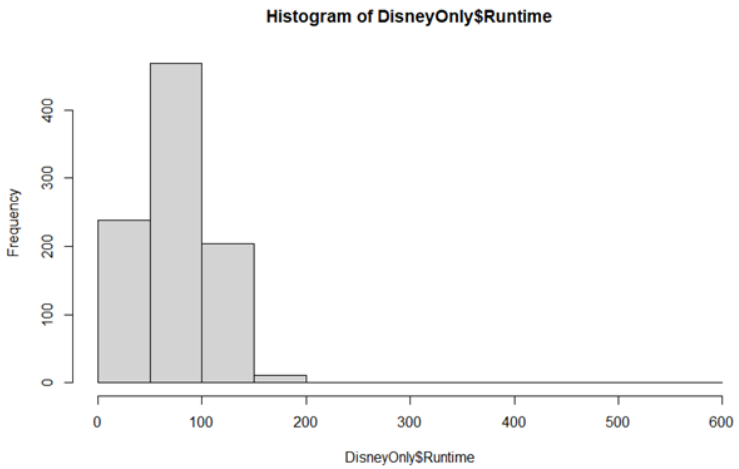

```
> #Create Data Frame for Hulu only movies and clean it
> HuluOnly <- data.frame(data)
> HuluOnly <- HuluOnly[order(HuluOnly$Hulu),]
> HuluOnly <- HuluOnly[-1:-8467,]
> HuluOnly$Runtime[is.na(HuluOnly$Runtime)] <- mean(HuluOnly$Runtime, na.rm=TRUE)
> #look at the structure of the Hulu only data
> str(HuluOnly$Runtime)
 num [1:1048] 98.3 152 93 140 132 ...
> summary(HuluOnly$Runtime)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   89.00   98.00   98.32  108.00  317.00
> min(HuluOnly$Runtime)
[1] 5
> max(HuluOnly$Runtime)
[1] 317
> median(HuluOnly$Runtime)
[1] 98
> mean(HuluOnly$Runtime)
[1] 98.31818
> hist(HuluOnly$Runtime)
```




```
> #Create Data Frame for Prime only movies and clean it
> PrimeOnly <- data.frame(data)
> PrimeOnly <- PrimeOnly[order(PrimeOnly$Prime.Video),]
> PrimeOnly <- PrimeOnly[-1:-5402,]
> PrimeOnly$Runtime[is.na(PrimeOnly$Runtime)] <- mean(PrimeOnly$Runtime, na.rm=TRUE)
> #look at the structure of the Prime only data
> str(PrimeOnly$Runtime)
 num [1:4113] 139 104 154 153 119 94 95 117 138 120 ...
> summary(PrimeOnly$Runtime)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00   85.00   94.00   95.22  105.00  328.00
> min(PrimeOnly$Runtime)
[1] 2
> max(PrimeOnly$Runtime)
[1] 328
> median(PrimeOnly$Runtime)
[1] 94
> mean(PrimeOnly$Runtime)
[1] 95.22477
> hist(PrimeOnly$Runtime)
```




```
> #Create Data Frame for Disney only movies and clean it
> DisneyOnly <- data.frame(data)
> DisneyOnly <- DisneyOnly[order(DisneyOnly$Disney.),]
> DisneyOnly <- DisneyOnly [-1:-8593,]
> DisneyOnly$Runtime[is.na(DisneyOnly$Runtime)] <- mean(DisneyOnly$Runtime, na.rm=TRUE)
> #look at the structure of the Disney only data
> str(DisneyOnly$Runtime)
 num [1:922] 139 181 149 121 121 124 100 88 143 96 ...
> summary(DisneyOnly$Runtime)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   47.00   85.00   75.77  100.00  566.00
> min(DisneyOnly$Runtime)
[1] 1
> max(DisneyOnly$Runtime)
[1] 566
> median(DisneyOnly$Runtime)
[1] 85
> mean(DisneyOnly$Runtime)
[1] 75.77294
> hist(DisneyOnly$Runtime)
```



#Plot each platform to visualize runtime over time
> plot(DisneyOnly$Year, DisneyOnly$Runtime)
> plot(HuluOnly$Year, HuluOnly$Runtime)
> plot(NetflixOnly$Year, NetflixOnly$Runtime)
> plot(PrimeOnly$Year, PrimeOnly$Runtime)
> #Create linear model for Runtime over time per platform & predict the outyears
> LM1 <- lm(Runtime ~ Year, data = DisneyOnly)
> plot(DisneyOnly$Year, DisneyOnly$Runtime)
> summary(LM1)

Call:
lm(formula = Runtime ~ Year, data = DisneyOnly)

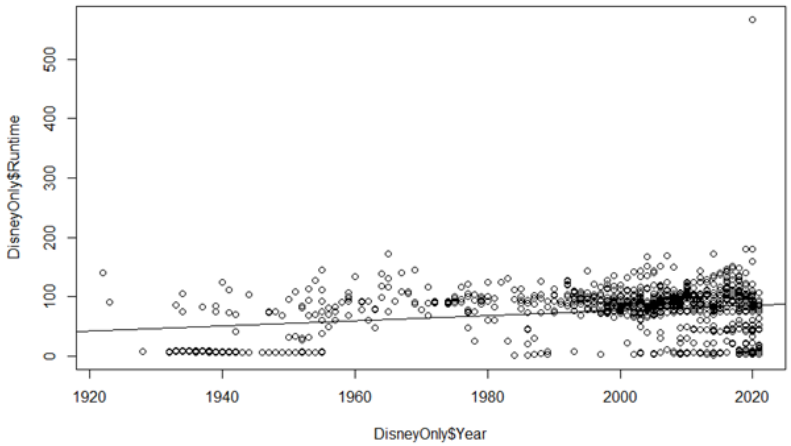
Residuals:
  Min   1Q Median   3Q   Max
-83.14 -31.93   7.17 24.01 480.29

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -787.88566 107.88965 -7.303 6.10e-13 ***
Year   0.43247   0.05402   8.006 3.57e-15 ***

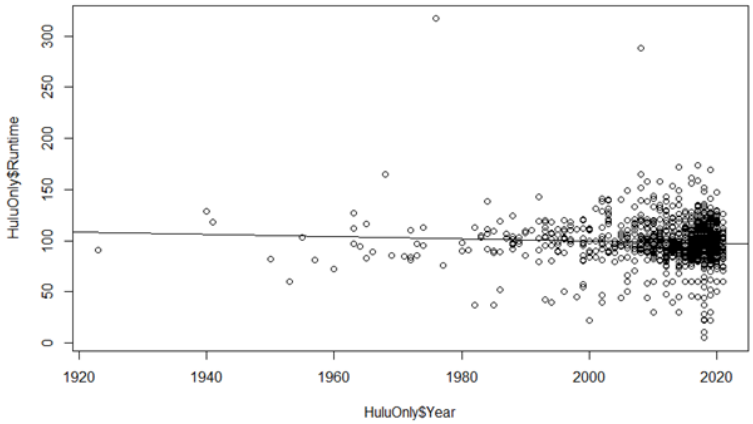
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 40.82 on 920 degrees of freedom
Multiple R-squared: 0.06513,   Adjusted R-squared: 0.06411
F-statistic: 64.09 on 1 and 920 DF, p-value: 3.572e-15

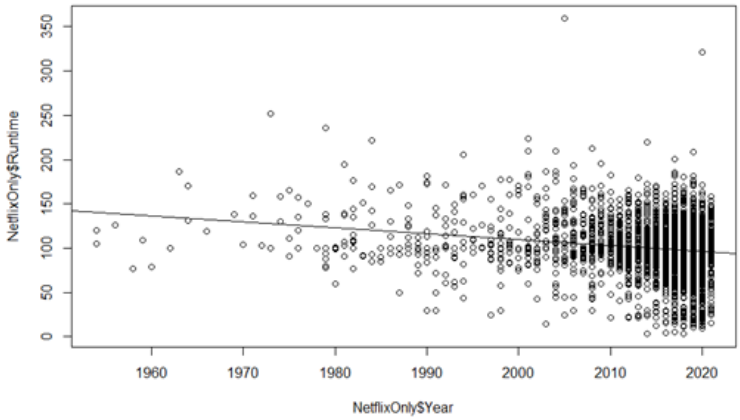
> abline(LM1)



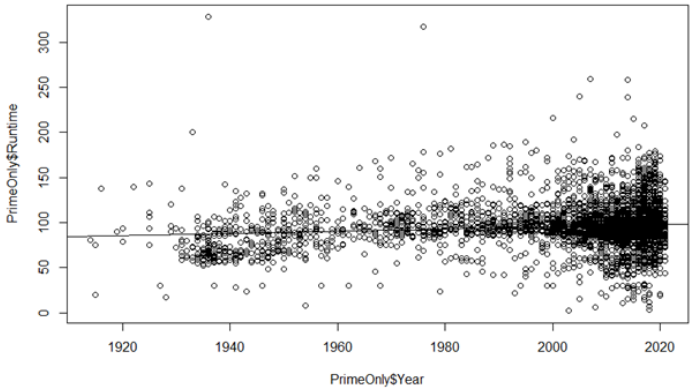

```
> LM2 <- lm(Runtime ~ Year, data = HuluOnly)
> plot(HuluOnly$Year, HuluOnly$Runtime)
> summary(LM2)

Call:
lm(formula = Runtime ~ Year, data = HuluOnly)

Residuals:
   Min     1Q  Median     3Q    Max
-92.648  -9.554  -0.540   9.676 214.811

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 315.84416  116.67198   2.707   0.0069 **
Year         -0.10812    0.05799  -1.864   0.0625 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.91 on 1046 degrees of freedom
Multiple R-squared:  0.003312,  Adjusted R-squared:  0.002359
F-statistic: 3.476 on 1 and 1046 DF,  p-value: 0.06254

> abline(LM2)
```




```
> LM3 <- lm(Runtime ~ Year, data = NetflixOnly)
> plot(NetflixOnly$Year, NetflixOnly$Runtime)
> summary(LM3)

Call:
lm(formula = Runtime ~ Year, data = NetflixOnly)

Residuals:
   Min     1Q  Median     3Q    Max
-97.053 -12.334   0.666  14.346 252.830

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1469.00845  118.08281   12.44   <2e-16 ***
Year          -0.67972    0.05861  -11.60   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 28.62 on 3693 degrees of freedom
Multiple R-squared:  0.03514,    Adjusted R-squared:  0.03488
F-statistic: 134.5 on 1 and 3693 DF,  p-value: < 2.2e-16

> abline(LM3)
```




```
> LM4 <- lm(Runtime ~ Year, data = PrimeOnly)
> plot(PrimeOnly$Year, PrimeOnly$Runtime)
> summary(LM4)

Call:
lm(formula = Runtime ~ Year, data = PrimeOnly)

Residuals:
   Min     1Q  Median     3Q    Max
-94.110 -10.992  -2.004   9.771 240.602

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -141.91870   34.22474  -4.147 3.44e-05 ***
Year           0.11845    0.01709   6.929 4.88e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 25.09 on 4111 degrees of freedom
Multiple R-squared:  0.01155,    Adjusted R-squared:  0.0113
F-statistic: 48.02 on 1 and 4111 DF,  p-value: 4.883e-12

> abline(LM4)
```

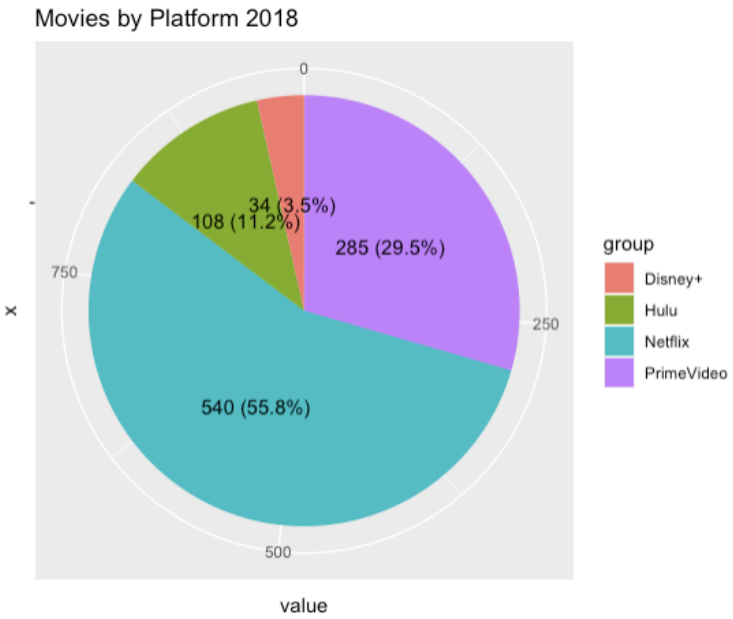

PrimeOnly$Runtime
0
50
100
150
200
250
300
1920
1940
1960
1980
2000
2020
PrimeOnly$Year

Appendix 2 Code: Business Question 2

```
Business Question 2: Comparing number of movies on different streaming services
#Storing dataset into a data frame
library(readxl)
> NetflixDS2 <- read_excel("Desktop/IST 687 Intro to DS/NetflixDS2.xlsx")
> View(NetflixDS2)
> NetflixDS2<-data.frame(NetflixDS2)
> colnames(NetflixDS2) <- c("Title", "Year", "Netflix", "Hulu","PrimeVideo","Disney+")
> attach(NetflixDS2)
> #Number of Movies in year 2018 by Platform
> tapply(Year, list(Netflix == 1, Year ==2018),length )
      FALSE TRUE
FALSE  5415  405
TRUE   3155  540
> #Number of Movies in year 2018 by Platform
> tapply(Year, list(Hulu == 1, Year ==2018),length )
      FALSE TRUE
FALSE  7631  837
TRUE    939  108
> #Number of Movies in year 2018 by Platform
> tapply(Year, list(PrimeVideo == 1, Year ==2018),length )
      FALSE TRUE
FALSE  4742  660
TRUE   3828  285
> #Number of Movies in year 2018 by Platform
> tapply(Year, list(`Disney+` == 1, Year ==2018),length )
      FALSE TRUE
FALSE  7682  911
TRUE    888   34
> #Number of Movies in year 2019 by Platform
> tapply(Year, list(Netflix == 1, Year ==2019),length )
      FALSE TRUE
FALSE  5379  441
TRUE   3122  573
> #Number of Movies in year 2019 by Platform
> tapply(Year, list(Hulu == 1, Year ==2019),length )
      FALSE TRUE
FALSE  7610  858
TRUE    891  156
> #Number of Movies in year 2019 by Platform
```

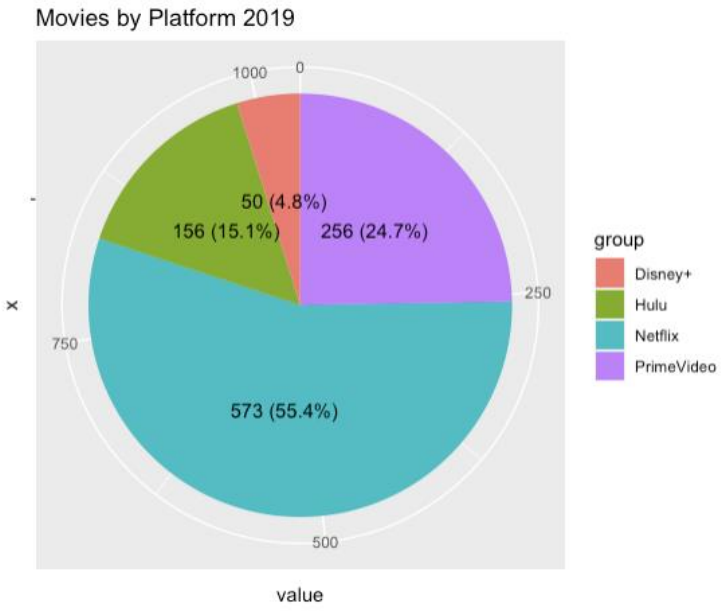
```
> tapply(Year, list(PrimeVideo == 1, Year ==2019),length )
    FALSE TRUE
FALSE  4644  758
TRUE   3857  256
> #Number of Movies in year 2019 by Platform
> tapply(Year, list(`Disney+` == 1, Year ==2019),length )
    FALSE TRUE
FALSE  7629  964
TRUE    872   50
> #Number of Movies in year 2020 by Platform
> tapply(Year, list(Netflix == 1, Year ==2020),length )
    FALSE TRUE
FALSE  5436  384
TRUE   3217  478
> #Number of Movies in year 2020 by Platform
> tapply(Year, list(Hulu == 1, Year ==2020),length )
    FALSE TRUE
FALSE  7722  746
TRUE    931  116
> #Number of Movies in year 2020 by Platform
> tapply(Year, list(PrimeVideo == 1, Year ==2020),length )
    FALSE TRUE
FALSE  4744  658
TRUE   3909  204
> #Number of Movies in year 2020 by Platform
> tapply(Year, list(`Disney+` == 1, Year ==2020),length )
    FALSE TRUE
FALSE  7802  791
TRUE    851   71
> #Number of Movies in year 2021 by Platform
> tapply(Year, list(Netflix == 1, Year ==2021),length )
    FALSE TRUE
FALSE  5694  126
TRUE   3494  201
> #Number of Movies in year 2021 by Platform
> tapply(Year, list(Hulu == 1, Year ==2021),length )
    FALSE TRUE
FALSE  8178  290
TRUE   1010   37
> #Number of Movies in year 2021 by Platform
> tapply(Year, list(PrimeVideo == 1, Year ==2021),length )
    FALSE TRUE
```

```
FALSE  5138  264
TRUE   4050   63
> #Number of Movies in year 2021 by Platform
> tapply(Year, list(`Disney+` == 1, Year ==2021),length )
     FALSE TRUE
FALSE  8294  299
TRUE    894   28

> #Create Data for Movies in 2018 by Streaming Services
> df1 <- data.frame(
+   group = c("Netflix", "Hulu", "PrimeVideo","Disney+"),
+   value = c(540, 108, 285, 34)
+ )
> head(df)
1 function (x, df1, df2, ncp, log = FALSE)
2 {
3     if (missing(ncp))
4         .Call(C_df, x, df1, df2, log)
5     else .Call(C_dnf, x, df1, df2, ncp, log)
6 }

#piechart for Movies in 2018 by Streaming Services
> pie <- bp1 + coord_polar("y", start=0)
> pie
# Barplot for Movies in 2018 by Streaming Services
bp1<- ggplot(df1, aes(x="", y=value, fill=group))+
  geom_bar(width = 1, stat = "identity")
bp1
#piechart for Movies in 2018 by Streaming Services
pie <- bp1 + coord_polar("y", start=0)+ggtitle("Movies by Platform 2018")
pie

pie +
  geom_text(aes(label = paste0(value,
                               " (",
                               scales::percent(value / sum(value)),
                               ")")),
            position = position_stack(vjust = 0.5))
```



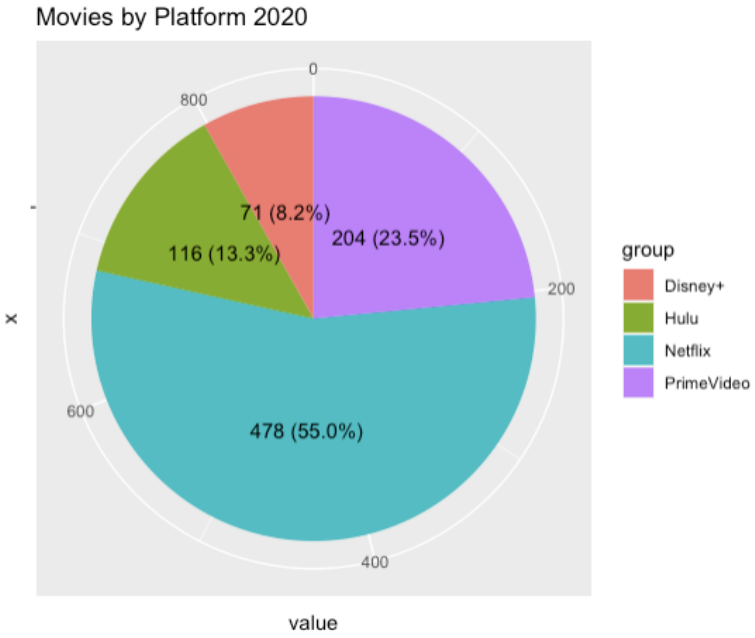

```
+ #Create Data for Movies in 2019 by Streaming Services
> df2 <- data.frame(
+   group = c("Netflix", "Hulu", "PrimeVideo","Disney+"),
+   value = c(573, 156, 256, 50)
+ )
> head(df)

1 function (x, df1, df2, ncp, log = FALSE)
2 {
3     if (missing(ncp))
4         .Call(C_df, x, df1, df2, log)
5     else .Call(C_dnf, x, df1, df2, ncp, log)
6 }
> # Barplot for Movies in 2019 by Streaming Services
> bp2<- ggplot(df1, aes(x="", y=value, fill=group))+
+   geom_bar(width = 1, stat = "identity")
> bp2
> #piechart for Movies in 2019 by Streaming Services
> pie2 <- bp2 + coord_polar("y", start=0)+ggtitle("Movies by Platform 2019")
> pie2
>
> pie2 +
+   geom_text(aes(label = paste0(value,
+                                " (",
+                                scales::percent(value / sum(value)),
+                                ")")),
```

```
+           position = position_stack(vjust = 0.5))
```



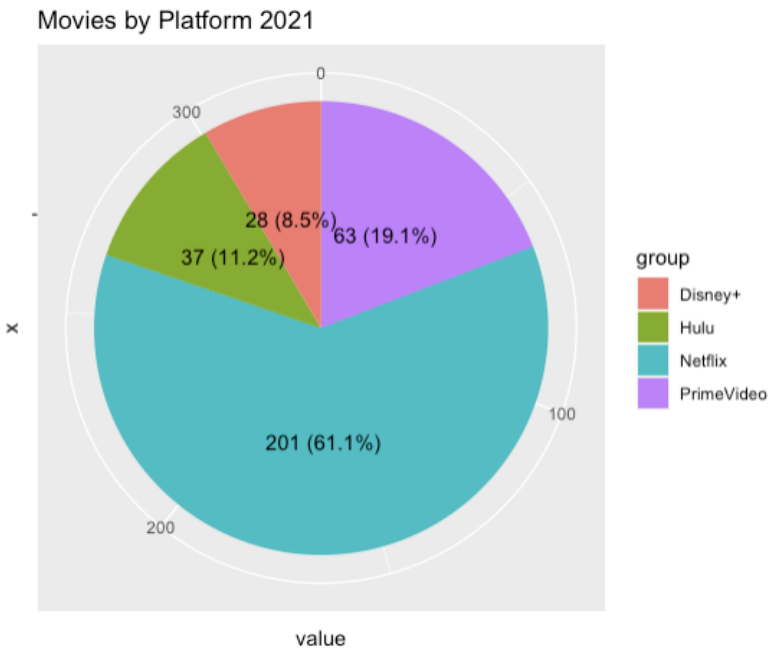

```
> #Create Data for Movies in 2020 by Streaming Services
+   df3 <- data.frame(
"#Create Data for Movies in 2020 by Streaming Services
  df3"
>     group = c("Netflix", "Hulu", "PrimeVideo","Disney+"),
>     value = c(478, 116, 204, 71)
> head(df)

1 function (x, df1, df2, ncp, log = FALSE)
2 {
3     if (missing(ncp))
4         .Call(C_df, x, df1, df2, log)
5     else .Call(C_dnf, x, df1, df2, ncp, log)
6 }
>
> # Barplot for Movies in 2020 by Streaming Services
> bp3<- ggplot(df3, aes(x="", y=value, fill=group))+
+   geom_bar(width = 1, stat = "identity")
> bp3
> #piechart for Movies in 2020 by Streaming Services
> pie3 <- bp3 + coord_polar("y", start=0)+ggtitle("Movies by Platform 2020")
> pie3
>
>
> pie3 +
+   geom_text(aes(label = paste0(value,
```

```
+                     " (",
+                     scales::percent(value / sum(value)),
+                     ")")),
+         position = position_stack(vjust = 0.5))
```




```
#Create Data for Movies in 2021 by Streaming Services
> df4 <- data.frame(
+   group = c("Netflix", "Hulu", "PrimeVideo","Disney+"),
+   value = c(201,37,63,28)
+ )
> head(df)
1 function (x, df1, df2, ncp, log = FALSE)
2 {
3     if (missing(ncp))
4         .Call(C_df, x, df1, df2, log)
5     else .Call(C_dnf, x, df1, df2, ncp, log)
6 }
>
> # Barplot for Movies in 2021 by Streaming Services
> bp4<- ggplot(df4, aes(x="", y=value, fill=group))+
+   geom_bar(width = 1, stat = "identity")
> bp4
> #piechart for Movies in 2021 by Streaming Services
> pie4 <- bp4 + coord_polar("y", start=0)+ggtitle("Movies by Platform 2021")
> pie4
>
```

```
> pie4 +
+   geom_text(aes(label = paste0(value,
+                                " (",
+                                scales::percent(value / sum(value)),
+                                ")")),
+             position = position_stack(vjust = 0.5))
>
```




```
> #multiple regression
> fit<-lm(Netflix ~ Hulu + PrimeVideo + `Disney+`, data=df2)
> summary(fit)
Call:
lm(formula = Netflix ~ Hulu + PrimeVideo + `Disney+`, data = df2)
Residuals:    Min      1Q  Median      3Q     Max
          -0.12973 -0.04973 -0.04973  0.02953  2.73653
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.970473   0.002670   363.4   <2e-16 ***
Hulu        -0.840743   0.005546  -151.6   <2e-16 ***
PrimeVideo  -0.920746   0.003624  -254.1   <2e-16 ***
`Disney+`   -0.945512   0.005927  -159.5   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.1615 on 9511 degrees of freedom
Multiple R-squared:  0.8903,      Adjusted R-squared:  0.8903
F-statistic: 2.573e+04 on 3 and 9511 DF,  p-value: < 2.2e-16
```

Appendix 3 Code: Business Question 3

R Code Unexecuted:

```
# Business Question #3: Which year is the most common for a movie (was there a slowdown because of the pandemic)
#Hypothesis: As a result of the COVID 19 pandemic the movie industry experienced a decline in movies produced in the years 2020 proceeding, which in turn
#affected the number of movies featured on the streaming services that were released in the years 2021 and 2021.

library(ggplot2)

#Download the csv file from Kaggle.com
#https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney?select=MoviesOnStreamingPlatforms_updated.csv

#Import the csv file from my hard drive before cleaning the dataset
StreamingService <- read.csv("C:\\Users\\gclark\\OneDrive - Ag Business Solutions\\Desktop\\IST 687\\IST 687 Group 3 Data Qs - Netflix.csv")

#Clean the dataset
StreamingService <- StreamingService[ ,-16:-21]
StreamingService <- StreamingService[ ,-18:-29]
StreamingService <- StreamingService[,-11]

colnames(StreamingService) <- c("ID", "Title", "Year", "Age", "IMDb", "Rotten Tomatoes", "Netflix","Hulu", "Prime Video", "Disney+", "Directors",  "Primary Genre", "Secondary Genre", "Tertiary Genre", "Film Language", "Other Language Option", "Runtime")

# look at structure of the data
str(StreamingService)
tapply(StreamingService$Year, StreamingService$Year, length)
min(StreamingService$Year)
median(StreamingService$Year)
mean(StreamingService$Year)
max(StreamingService$Year)


YearCount <- tapply(StreamingService$Year, StreamingService$Year, length)
str(YearCount) # 103 years is too large a scope

#Reduce scope by looking at last ten years and total movies featured
```

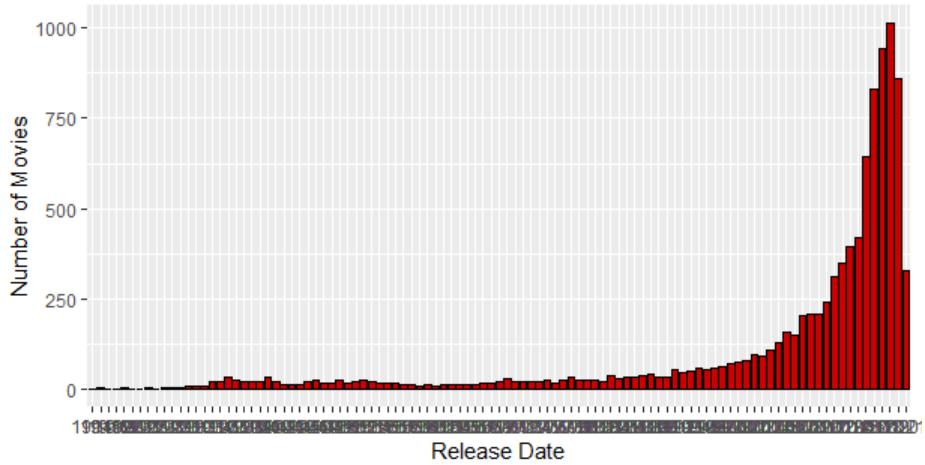
```
attach(StreamingService)
StreamingService <- StreamingService[order(Year),]

 AllMovies <- tapply(StreamingService$ID, StreamingService$Year, length)
AllMoviesDF <- data.frame(AllMovies)
GraphAll <- ggplot(AllMoviesDF, aes(x=rownames(AllMoviesDF), y=AllMovies)) +
geom_bar(stat="identity", color = "black", fill = "red3") + xlab("Release Date") + ylab("Number of
Movies")
GraphAll

LastTenYears <- StreamingService[-1:-3176,]
MovieCount <- tapply(LastTenYears$ID, LastTenYears$Year, length)

# Graph the Data
MovieCount
MovieCountDF <- data.frame(MovieCount)
MovieCountDF
str(MovieCountDF)
library(ggplot2)
length(LastTenYears$Year)
bargraph <- ggplot(MovieCountDF, aes(x=rownames(MovieCountDF), y=MovieCount)) +
geom_bar(stat="identity", color = "black", fill = "red3") + xlab("Release Date") + ylab("Number of
Movies")
bargraph <- bargraph + ggtitle("Count of Movies Over Last Ten Years")
bargraph
# There is a decline in movie production in 2020 and 2021. Although the exact cause of the decline is not
apparent.
#There is not sufficient evidence to dismiss the hypothesis that the pandemic caused a decline in movie
production.
# Predict number of movies in 2021

MovieCounts <- MovieCountDF[1:10,]
Years <- c(2011,2012,2013,2014,2015,2016,2017,2018,2019,2020)
MovieCountDF2 <- data.frame(MovieCounts, Years)

m <- lm(formula = MovieCounts ~ Years, data = MovieCountDF2)
summary(m)

g <- ggplot(MovieCountDF2, aes(x=Years, y=MovieCounts)) + geom_point()
g + ggtitle("Count of Movies by Year on Streaming Services") + xlab("Year of Movie Release") +
ylab("Number of MOvies") + scale_x_continuous(labels = scales::number_format(accuracy = 1)) +
stat_smooth(method = "lm")
```

```
#Assuming that movies released in 2021 are still being added to the streaming Services,
# Predict the number of movies we'd expect to be featured with a release date in the year 2021

NextYear <- data.frame(Dates = c(2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021))
predict(m,newdata = NextYear, type = "response")
predict(m,newdata = NextYear, interval  = 'confidence')
# The prediction for the featured movies released in 2021 is 1,010.78 or roughly 1,010 movies.
# Currently, 2021 is at 327 movies, which is a variance of 683 movies. The year isn't over yet, and movies
may be added
# proceeding years, but it would be interesting to run this analysis again in 5 years to see if the number
grew to the predicted
# value or even fell somewhere within the lower limit of 874 and upper limit of 1,146 movies.
```

R Code Executed:

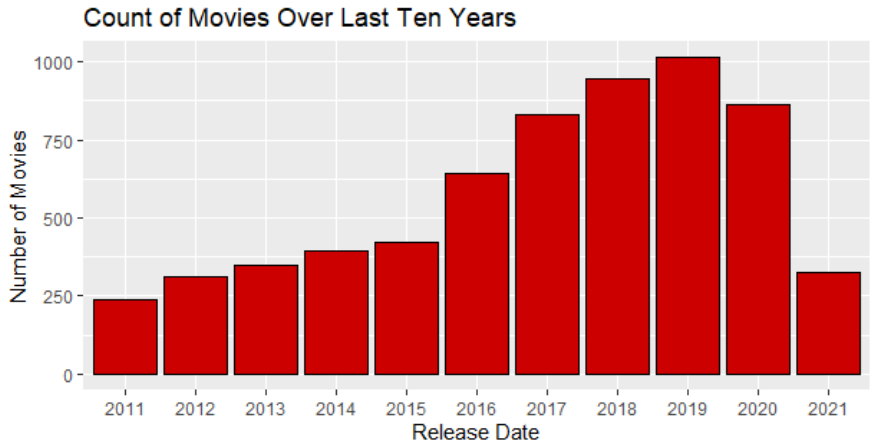
```
> library(ggplot2)
> #Import the csv file from my hard drive before cleaning the dataset
> StreamingService <- read.csv("C:\\Users\\gclark\\OneDrive - Ag Business Solutions\\Desktop\\IST
687\\IST 687 Group 3 Data Qs - Netflix.csv")
> #Clean the dataset
> StreamingService <- StreamingService[ ,-16:-21]
> StreamingService <- StreamingService[ ,-18:-29]
> StreamingService <- StreamingService[,-11]
> colnames(StreamingService) <- c("ID", "Title", "Year", "Age", "IMDb", "Rotten Tomatoes",
"Netflix","Hulu", "Prime Video", "Disney+", "Directors",  "Primary Genre", "Secondary Genre", "Tertiary
Genre", "Film Language", "Other Language Option", "Runtime")
> # look at structure of the data
> min(StreamingService$Year)
[1] 1914
> median(StreamingService$Year)
[1] 2015
> mean(StreamingService$Year)
[1] 2007.422
> max(StreamingService$Year)
[1] 2021
> YearCount <- tapply(StreamingService$Year, StreamingService$Year, length)
> str(YearCount) # 103 years is too large a scope
 int [1:103(1d)] 1 2 1 1 2 1 1 5 1 2 ...
 - attr(*, "dimnames")=List of 1
  ..$ : chr [1:103] "1914" "1915" "1916" "1919" ...
> #Reduce scope by looking at last ten years and total movies featured
> attach(StreamingService)
> StreamingService <- StreamingService[order(Year),]
```

```
> AllMovies <- tapply(StreamingService$ID, StreamingService$Year, length)
> AllMoviesDF <- data.frame(AllMovies)
> GraphAll <- ggplot(AllMoviesDF, aes(x=rownames(AllMoviesDF), y=AllMovies)) +
geom_bar(stat="identity", color = "black", fill = "red3") + xlab("Release Date") + ylab("Number of
Movies")
> GraphAll
```



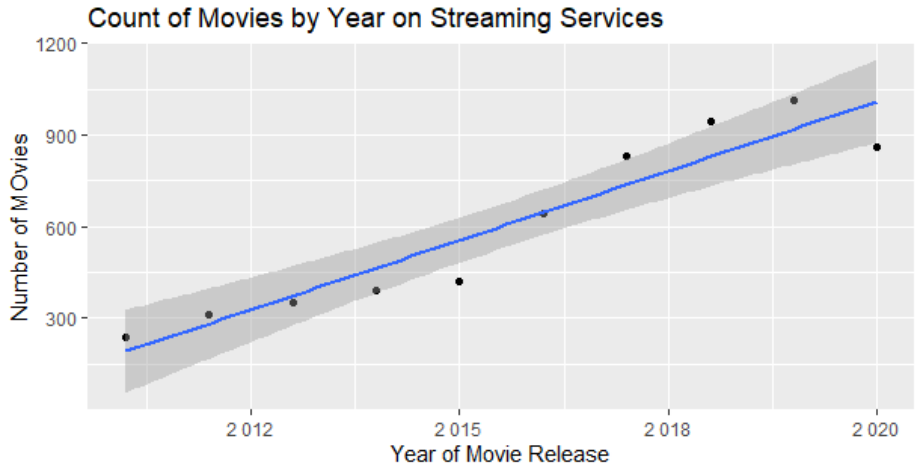

```
> LastTenYears <- StreamingService[-1:-3176,]
> MovieCount <- tapply(LastTenYears$ID, LastTenYears$Year, length)
> # Graph the Data
> MovieCount
2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021
 240  313  350  393  420  644  831  945 1014  862  327
> MovieCountDF <- data.frame(MovieCount)
> MovieCountDF
     MovieCount
2011        240
2012        313
2013        350
2014        393
2015        420
2016        644
2017        831
2018        945
2019       1014
2020        862
2021        327
> str(MovieCountDF)
'data.frame':      11 obs. of  1 variable:
 $ MovieCount: int  240 313 350 393 420 644 831 945 1014 862 ...
> library(ggplot2)
```

```
> length(LastTenYears$Year)
[1] 6339
> bargraph <- ggplot(MovieCountDF, aes(x=rownames(MovieCountDF), y=MovieCount)) +
geom_bar(stat="identity", color = "black", fill = "red3") + xlab("Release Date") + ylab("Number of
Movies")
> bargraph <- bargraph + ggtitle("Count of Movies Over Last Ten Years")
> bargraph
```




```
> MovieCounts <- MovieCountDF[1:10,]
> Years <- c(2011,2012,2013,2014,2015,2016,2017,2018,2019,2020)
> MovieCountDF2 <- data.frame(MovieCounts, Years)
> m <- lm(formula = MovieCounts ~ Years, data = MovieCountDF2)
> summary(m)

Call:
lm(formula = MovieCounts ~ Years, data = MovieCountDF2)

Residuals:
   Min     1Q  Median     3Q    Max
-148.78  -59.67   13.83   82.05  116.25

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -182845.95   22274.40  -8.209 3.63e-05 ***
Years           91.02      11.05   8.236 3.54e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 100.4 on 8 degrees of freedom
Multiple R-squared:  0.8945,      Adjusted R-squared:  0.8813
F-statistic: 67.83 on 1 and 8 DF,  p-value: 3.541e-05
```

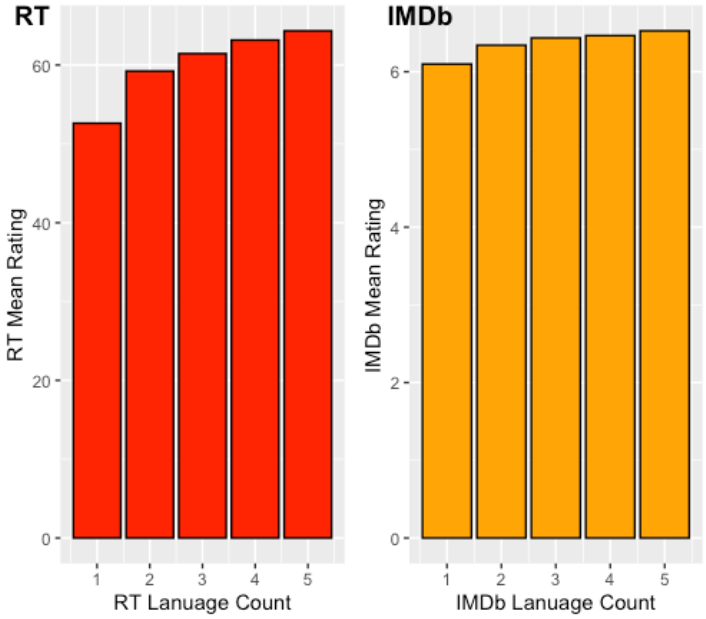
```
> g <- ggplot(MovieCountDF2, aes(x=Years, y=MovieCounts)) + geom_point()
> g + ggtitle("Count of Movies by Year on Streaming Services") + xlab("Year of Movie Release") +
ylab("Number of MOvies") + scale_x_continuous(labels = scales::number_format(accuracy = 1)) +
stat_smooth(method = "lm")
```



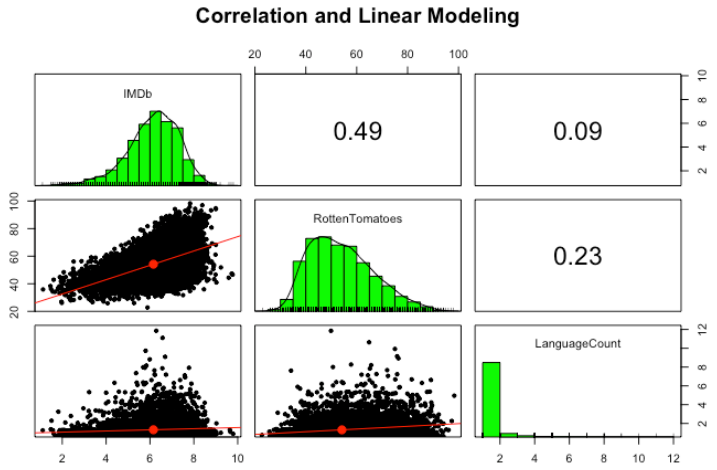

```
`geom_smooth()` using formula 'y ~ x'
> NextYear <- data.frame(Dates = c(2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021))
> predict(m,newdata = NextYear, type = "response")
      1       2       3       4       5       6       7       8       9      10
 191.6182  282.6364  373.6545  464.6727  555.6909  646.7091  737.7273  828.7455  919.7636
1010.7818
> predict(m,newdata = NextYear, interval  = 'confidence')
       fit     lwr      upr
1   191.6182  55.5661  327.6703
2   282.6364 167.2486  398.0241
3   373.6545 276.6110  470.6981
4   464.6727 382.0922  547.2532
5   555.6909 481.3903  629.9915
6   646.7091 572.4085  721.0097
7   737.7273 655.1468  820.3078
8   828.7455 731.7019  925.7890
9   919.7636 804.3759 1035.1514
10 1010.7818 874.7297 1146.8339
```

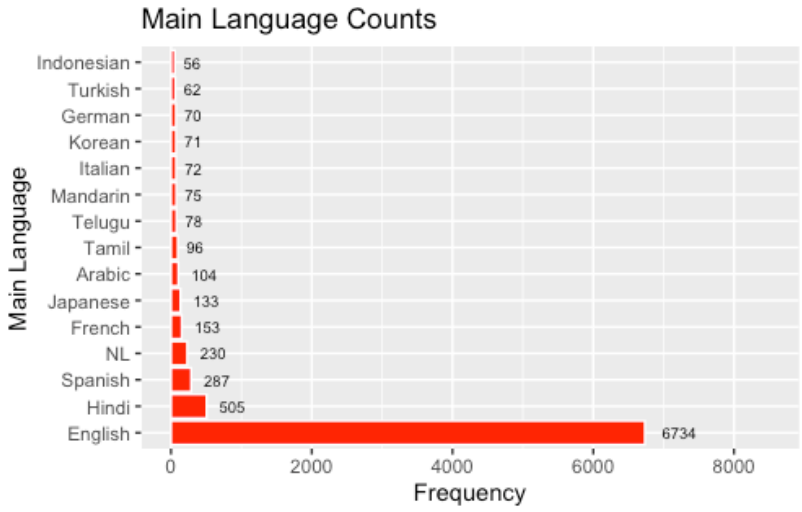
Appendix 4 Code: Business Question 4

```
library(cowplot)
df1$LanguageCount <- count.fields(textConnection(df1$Language), sep = ',')
##group by language count and return summarization of statistical readings
langcount_rt <- df1 %>%
  select(RottenTomatoes) %>%
  group_by(df1$LanguageCount) %>%
  summarise(mean(RottenTomatoes),
            length(RottenTomatoes), sd(RottenTomatoes))

colnames(langcount_rt) <- c('language_count','mean_rating','length','sd')
rt_bar <- ggplot(langcount_rt[1:5,], aes(x = language_count, y = mean_rating)) +
  geom_bar(stat = 'identity', color='black', fill='red') +
  labs(x = 'RT Lanuage Count', y = 'RT Mean Rating')

langcount_imdb <- df1 %>%
  select(IMDb) %>%
  group_by(df1$LanguageCount) %>%
  summarise(mean(IMDb),
            length(IMDb), sd(IMDb))

colnames(langcount_imdb) <- c('language_count','mean_rating','length','sd')

imdb_bar <- ggplot(langcount_imdb[1:5,], aes(x = language_count, y = mean_rating)) +
  geom_bar(stat = 'identity', color='black', fill='orange') +
  labs(x = 'IMDb Lanuage Count', y = 'IMDb Mean Rating')

plot_grid(rt_bar, imdb_bar, labels = c('RT','IMDb'))
```




```
Library(psych)
df3$LanguageCount <- as.factor(df3$LanguageCount)
pairs.panels(df3, lm = T, hist.col = 'green', show.points = T,
             jiggle = T, rug = T,
             main = 'Correlation and Linear Modeling')
```





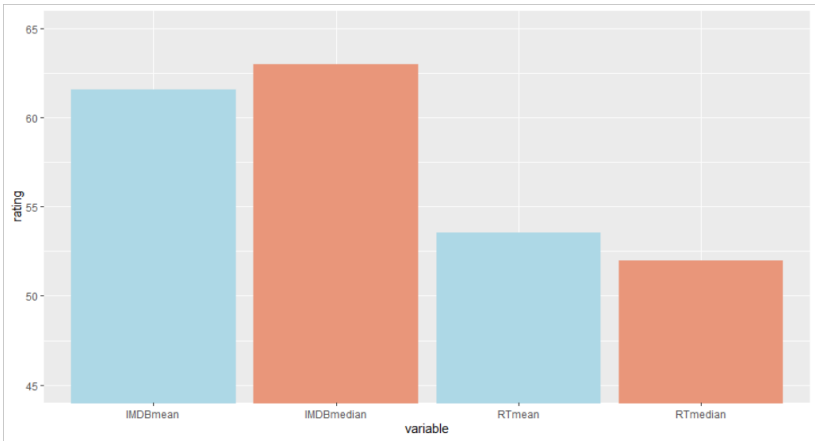

```
ind <- sample(2, nrow(df4), replace = T, prob = c(0.7,0.3))
train <- df4[ind == 1,]
test <- df4[ind == 2,]
nb_model <- naiveBayes(EnglishOrNot ~ ., data = train)
p <- predict(nb_model, test)
tab1 <- table(p, test$EnglishOrNot)
confusionMatrix(p, test$EnglishOrNot)

> confusionMatrix(p, test$EnglishOrNot)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0  721    7
         1   42 1995
               Accuracy : 0.9823
                 95% CI : (0.9766, 0.9869)
    No Information Rate : 0.7241
    P-Value [Acc > NIR] : < 2.2e-16
                  Kappa : 0.955
Mcnemar's Test P-Value : 1.191e-06
            Sensitivity : 0.9450
            Specificity : 0.9965
         Pos Pred Value : 0.9904
         Neg Pred Value : 0.9794
             Prevalence : 0.2759
         Detection Rate : 0.2608
   Detection Prevalence : 0.2633
```

Balanced Accuracy : 0.9707
'Positive' Class : 0

Appendix 5 Code: Business Question 5

```
> library(readxl)
> library(sqldf)
> library(ggplot2)
> library(reshape2)
> sData <- read_excel ("C:\\Users\\Elizabeth's\\Documents\\IST_687\\streamingData3.xlsx")
> View(sData)
> #remove /10
> sData$IMDb <- gsub ("/10", "", sData$IMDb)
> #remove ' ' around original column heading; remove /100
> colnames(sData) [6] <- "RottenTomatoes"
> colnames(sData) [23] <- "SciFi"
> sData$RottenTomatoes <- gsub ("/100", "", sData$RottenTomatoes)
> #numberize function to read data as numbers and remove decimal
> Numberize <-function (inputVector) {
+   inputVector <-gsub (" ", "", inputVector)
+   return (as.numeric(inputVector))
+ }
>
> sData$IMDb <- Numberize(sData$IMDb)
> sData$IMDb <- sData$IMDb * 10
> sData$RottenTomatoes <- Numberize(sData$RottenTomatoes)
>
> #new data frame with select attributes
> nData <- sData [ c("Title", "IMDb", "RottenTomatoes", "Comedy", "Thriller", "SciFi",
+                    "Romance", "Drama", "Action", "Biography", "Horror", "Genre 1", "Genre 2", "Genre 3",
"Genre 4")]
> View(nData)
>
> IMDBmean <- mean(nData$IMDb, na.rm=TRUE)
> RTmean <- mean(nData$RottenTomatoes, na.rm=TRUE)
> IMDBmedian <- median(nData$IMDb, na.rm=TRUE)
> RTmedian <-median(nData$RottenTomatoes, na.rm=TRUE)
> #NEED A DATAFRAME FOR THE ABOVE INFO SO I CAN PLOT IT
> mm <- as.numeric(c (IMDBmean, RTmean, IMDBmedian, RTmedian))
> description <- c("IMDBmean", "RTmean", "IMDBmedian", "RTmedian")
> meansMed <- data.frame (description, mm)
> #let's plot it
> mm.plot4 <- ggplot(meansMed, aes(x=description, y=mm)) + geom_col(fill=c("lightblue", "lightblue",
"darksalmon", "darksalmon"))
> mm.plot4 <- mm.plot4 + coord_cartesian(ylim = c(45, 65))  + labs(x="variable", y= "rating")
```

```
> mm.plot4
```

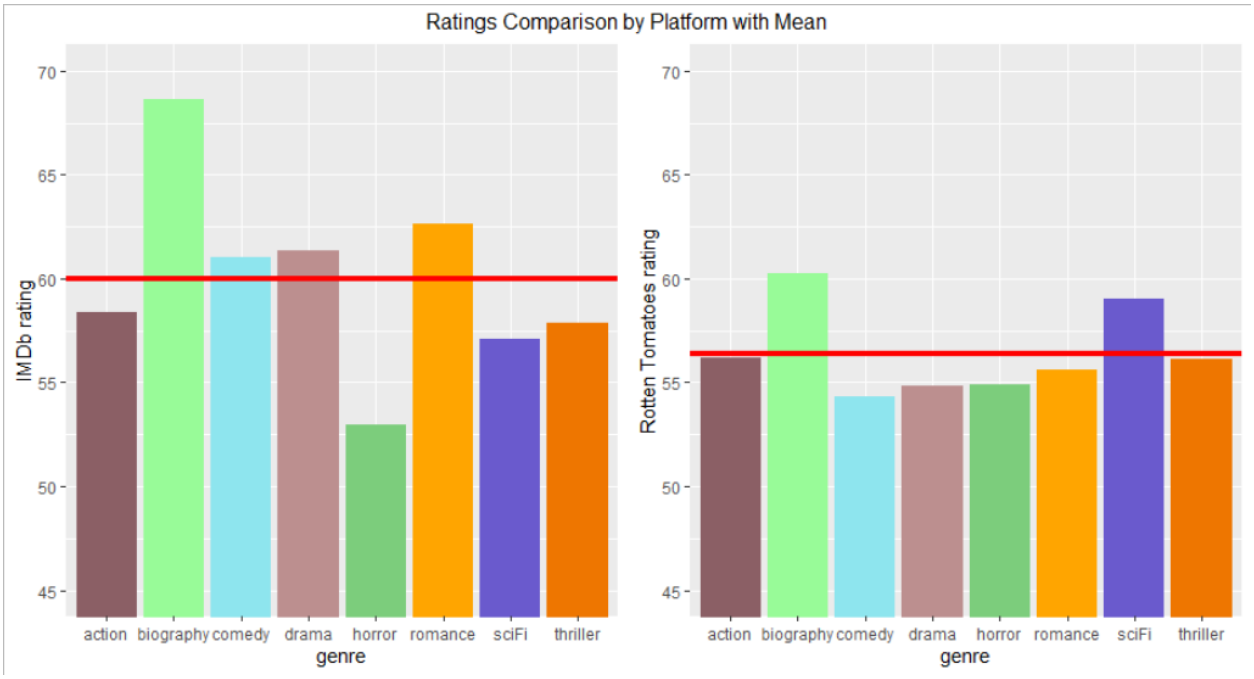



```
> #comedy ratings and counts
> comedyRatIMDB <- sqldf("select AVG(IMDb) from nData where Comedy = 1")
> comedyRatIMDB
  AVG(IMDb)
1  61.03133
> comedyRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Comedy = 1")
> comedyRatRT
  AVG(RottenTomatoes)
1           54.30841
>
> #thriller ratings and counts
> thrillerRatIMDB <- sqldf("select AVG(IMDb) from nData where Thriller = 1")
> thrillerRatIMDB
  AVG(IMDb)
1  57.86808
> thrillerratRT <- sqldf("select AVG(RottenTomatoes) from nData where Thriller = 1")
> thrillerratRT
  AVG(RottenTomatoes)
1           56.15594
>
> #Sci-Fi ratings
> sfRatIMDB <-sqldf("select AVG(IMDb) from nData where SciFi = 1")
> sfRatIMDB
  AVG(IMDb)
1  57.08478
> sfRatRT <- sqldf("select AVG(RottenTomatoes) from nData where SciFi = 1")
> sfRatRT
  AVG(RottenTomatoes)
1           59.00192
>
```

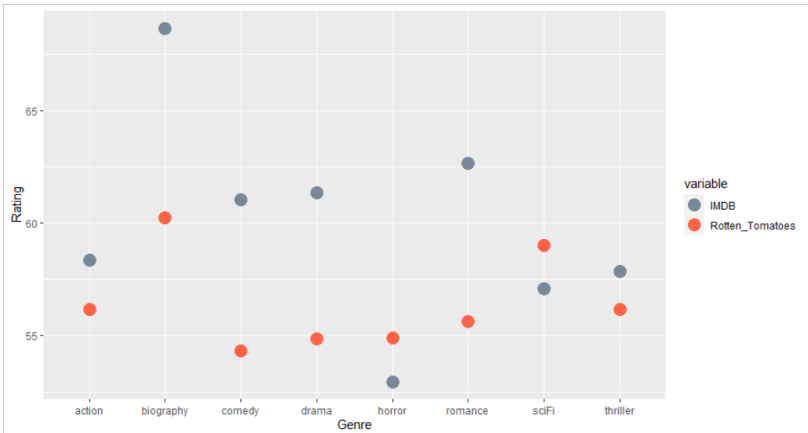
```
> #Romance ratings
> romRatIMDB <- sqldf("select AVG(IMDb) from nData where Romance = 1")
> romRatIMDB
  AVG(IMDb)
1  61.34108
> romRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Romance = 1")
> romRatRT
  AVG(RottenTomatoes)
1           54.86054
>
> #Drama ratings
> dramaRatIMDB <- sqldf("select AVG(IMDb) from nData where Drama = 1")
> dramaRatIMDB
  AVG(IMDb)
1  62.65887
> dramaRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Drama = 1")
> dramaRatRT
  AVG(RottenTomatoes)
1           55.63002
>
> #Action ratings
> actionRatIMDB <- sqldf("select AVG(IMDb) from nData where Action = 1")
> actionRatIMDB
  AVG(IMDb)
1  58.36742
> actionRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Action = 1")
> actionRatRT
  AVG(RottenTomatoes)
1           56.15892
>
> #Biography
> bioRatIMDB <- sqldf("select AVG(IMDb) from nData where Biography = 1")
> bioRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Biography = 1")
> bioRatIMDB
  AVG(IMDb)
1  68.62839
> bioRatRT
  AVG(RottenTomatoes)
1           60.25104
>
> #Horror
> horRatIMDB <- sqldf("select AVG(IMDb) from nData where Horror = 1")
```

```
> horRatRT <- sqldf("select AVG(RottenTomatoes) from nData where Horror = 1")
> horRatIMDB
  AVG(IMDb)
1  52.96255
> horRatRT
  AVG(RottenTomatoes)
1           54.91294
> #let's do something with these averages--maybe a plot!
> IMDB <- as.numeric(c ( comedyRatIMDB, thrillerRatIMDB, sfRatIMDB, dramaRatIMDB, romRatIMDB,
actionRatIMDB, bioRatIMDB, horRatIMDB))
> Rotten_Tomatoes <- as.numeric (c(comedyRatRT, thrillerratRT, sfRatRT, dramaRatRT, romRatRT,
actionRatRT, bioRatRT, horRatRT))
> genre2 <- c("comedy", "thriller", "sciFi",
+            "romance", "drama", "action", "biography", "horror")
> #create the data frame
> ratings2 <- data.frame (genre2, IMDB, Rotten_Tomatoes)
> View(ratings2)
>
> #look at mean/sd for just the eight selected genres(rather than all ratings)
> mean(ratings2$IMDB)
[1] 59.99281
> sd(ratings2$IMDB)
[1] 4.623158
> mean(ratings2$Rotten_Tomatoes)
[1] 56.40997
> sd(ratings2$Rotten_Tomatoes)
[1] 2.113434
>
> #just choosing some stupid colors because I can't get brewer to work
> cpal <- c("cadetblue2", "darkorange2" ,"slateblue", "orange1", "rosybrown", "lightpink4", "palegreen",
"palegreen3")
>
> IM.plot <-ggplot(ratings2, aes(x=genre2, y=IMDB)  )+ geom_col(fill=cpal)+ geom_hline(yintercept
=mean(IMDB), color="red", size=1.5 )
> IM.plot <- IM.plot + coord_cartesian(ylim=c(45,70)) + labs(x="genre", y= "IMDb rating")
> IM.plot
>
> RT.plot <-ggplot(ratings2, aes(x=genre2, y=Rotten_Tomatoes)  )+ geom_col(fill=cpal) +
geom_hline(yintercept =mean(Rotten_Tomatoes), color="red", size=1.5 )
> RT.plot <- RT.plot + coord_cartesian(ylim=c(45,70)) + labs(x="genre", y= "Rotten Tomatoes rating")
> RT.plot
>
```

> grid.arrange(IM.plot, RT.plot, ncol=2, nrow=1, top="Ratings Comparison by Platform with Mean")

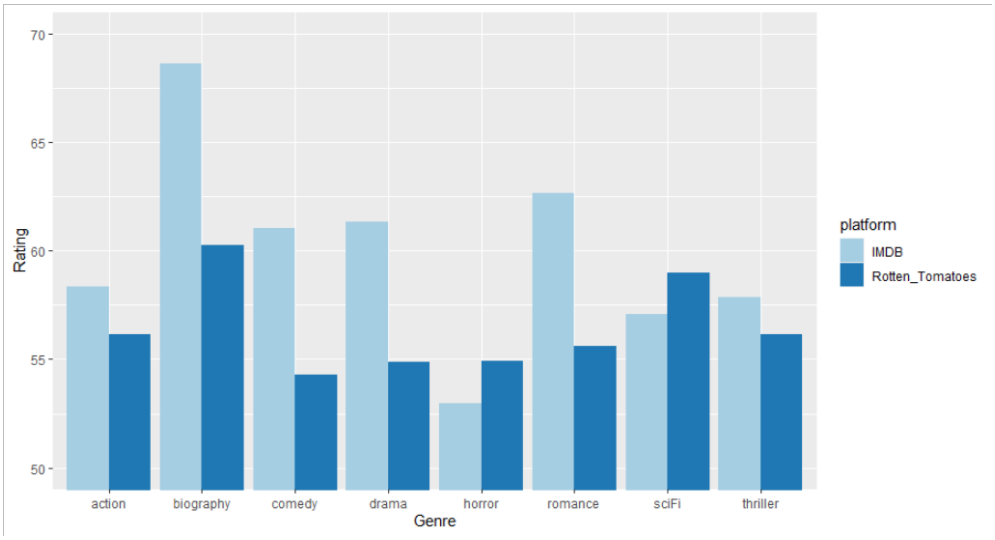


> #MELT
> ratings2Long <-melt(ratings2, id="genre2")
> View(ratings2Long)
>
> #a decent scatter plot, alternative plot
> gg1 <- ggplot(ratings2Long, aes(x=genre2, y=value, color=variable)) + geom_point(size=5)
> gg1<- gg1 + scale_color_manual(values = c("lightslategray", "tomato"))+ labs(x="Genre", y= "Rating", fill= "platform")
> gg1



> #side by side columns ratings by genre and platform, which is better than the above scatter
> gg <- ggplot(ratings2Long, aes(x=genre2, y=value, fill=variable)) + geom_col(position = position_dodge())

```
> gg<- gg + coord_cartesian(ylim = c(50, 70))  + labs(x="Genre", y= "Rating", fill= "platform" )+
scale_fill_brewer(palette="Paired")
> gg
> #138 lines of code of 433 lines written
```



References

Bhatia, R. (July 2021). *Movies on Netflix, Prime Video, Hulu and Disney+* (Version 3). Kaggle. https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney?select=MoviesOnStreamingPlatforms_updated.csv