Data Collection and Analysis

Bryan Persaud

5/11/2021

```
# Libraries
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## 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
```

```
library(ggplot2)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.0.5
```

Two datasets were collected from Kaggle. The first one shows different streaming services, Netflix, Hulu, Amazon Prime Video, and Disney+. The second one shows the number of subscribers for Netflix.

Data Exploration

Let's explore these datasets a little to see what they contain.

```
shows <- read.csv("https://raw.githubusercontent.com/bpersaud104/Data698/main/Data%20Collection%20and%2
summary(shows)
```

```
##        X           Title                Year          Age
##  Min.   :   0   Length:5611        Min.   :1901   Length:5611
##  1st Qu.:1402   Class :character   1st Qu.:2010   Class :character
##  Median :2805   Mode  :character   Median :2015   Mode  :character
##  Mean   :2805                      Mean   :2011
##  3rd Qu.:4208                      3rd Qu.:2017
##  Max.   :5610                      Max.   :2020
##
##       IMDb       Rotten.Tomatoes       Netflix            Hulu
##  Min.   :1.000   Length:5611        Min.   :0.0000   Min.   :0.0000
##  1st Qu.:6.600   Class :character   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :7.300   Mode  :character   Median :0.0000   Median :0.0000
##  Mean   :7.113                      Mean   :0.3441   Mean   :0.3126
##  3rd Qu.:7.900                      3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :9.600                      Max.   :1.0000   Max.   :1.0000
##  NA's   :1161
##   Prime.Video        Disney.             type
##  Min.   :0.0000   Min.   :0.00000   Min.   :1
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1
##  Median :0.0000   Median :0.00000   Median :1
##  Mean   :0.3821   Mean   :0.03208   Mean   :1
##  3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:1
##  Max.   :1.0000   Max.   :1.00000   Max.   :1
##
```

```
head(shows, 50)
```

```
##     X                           Title Year Age IMDb Rotten.Tomatoes Netflix
## 1   0                     Breaking Bad 2008 18+  9.5             96%       1
## 2   1                 Stranger Things 2016 16+  8.8             93%       1
## 3   2                      Money Heist 2017 18+  8.4             91%       1
## 4   3                         Sherlock 2010 16+  9.1             78%       1
## 5   4                Better Call Saul 2015 18+  8.7             97%       1
## 6   5                      The Office 2005 16+  8.9             81%       1
## 7   6                    Black Mirror 2011 18+  8.8             83%       1
## 8   7                    Supernatural 2005 16+  8.4             93%       1
## 9   8                   Peaky Blinders 2013 18+  8.8             92%       1
## 10  9       Avatar: The Last Airbender 2005  7+  9.2            100%       1
## 11 10                The Walking Dead 2010 18+  8.2             81%       1
## 12 11                            Dark 2017 16+  8.7             94%       1
## 13 12                          Ozark 2017 18+  8.4             81%       1
## 14 13                 Attack on Titan 2013 16+  8.8             94%       1
## 15 14                          Narcos 2015 18+  8.8             89%       1
## 16 15 Fullmetal Alchemist: Brotherhood 2009 18+  9.1            100%       1
## 17 16                       Community 2009  7+  8.5             88%       1
```

```
## 18 17                       Mindhunter 2017 18+  8.6            96%       1
## 19 18             Parks and Recreation 2009 16+  8.6            93%       1
## 20 19                           Dexter 2006 18+  8.6            72%       1
## 21 20               Marvel's Daredevil 2015 18+  8.6            92%       1
## 22 21                      The Witcher 2019 18+  8.3            67%       1
## 23 22                       Twin Peaks 1990 18+  8.8            89%       1
## 24 23                    One-Punch Man 2015 16+  8.8           100%       1
## 25 24                        Outlander 2014 18+  8.4            91%       1
## 26 25                    House of Cards 2013 18+  8.7            78%       1
## 27 26                        Shameless 2011 18+  8.6            85%       1
## 28 27                    The Good Place 2016 16+  8.2            97%       1
## 29 28                      The Haunting 2018 18+  8.7            93%       1
## 30 29                     The Blacklist 2013 16+  8.0            91%       1
## 31 30                        The Flash 2014  7+  7.7            89%       1
## 32 31                  The Last Kingdom 2015 18+  8.4            91%       1
## 33 32                           Mad Men 2007 16+  8.6            94%       1
## 34 33                           Lucifer 2016 16+  8.2            87%       1
## 35 34          Orange Is the New Black 2013 18+  8.1            90%       1
## 36 35                    Grey's Anatomy 2005 16+  7.6            83%       1
## 37 36     The End of the F***ing World 2017 18+  8.1            93%       1
## 38 37              Arrested Development 2003 16+  8.7            75%       1
## 39 38               The Vampire Diaries 2009  7+  7.7            85%       1
## 40 39                           The Crown 2016 18+  8.7            89%       1
## 41 40                             The 100 2014 16+  7.7            92%       1
## 42 41                  When They See Us 2019 18+  8.9            96%       1
## 43 42      How to Get Away with Murder 2014 16+  8.1            88%       1
## 44 43                          After Life 2019 18+  8.5            70%       1
## 45 44                               Elite 2018 18+  7.6            97%       1
## 46 45                  BoJack Horseman 2014 18+  8.7            93%       1
## 47 46                 Never Have I Ever 2020 16+  8.0            97%       1
## 48 47                    Penny Dreadful 2014 18+  8.2            91%       1
## 49 48  Marvel's Agents of S.H.I.E.L.D. 2013 16+  7.5            94%       1
## 50 49                          Dead to Me 2019 18+  8.1            91%       1
##    Hulu Prime.Video Disney. type
## 1     0           0       0    1
## 2     0           0       0    1
## 3     0           0       0    1
## 4     0           0       0    1
## 5     0           0       0    1
## 6     0           0       0    1
## 7     0           0       0    1
## 8     0           0       0    1
## 9     0           0       0    1
## 10    0           0       0    1
## 11    0           0       0    1
## 12    0           0       0    1
## 13    0           0       0    1
## 14    1           0       0    1
## 15    0           0       0    1
## 16    1           0       0    1
## 17    1           0       0    1
## 18    0           0       0    1
## 19    1           1       0    1
## 20    0           0       0    1
```

```
## 21    0           0       0    1
## 22    0           0       0    1
## 23    1           0       0    1
## 24    1           0       0    1
## 25    0           0       0    1
## 26    0           0       0    1
## 27    0           0       0    1
## 28    1           0       0    1
## 29    0           0       0    1
## 30    0           0       0    1
## 31    0           0       0    1
## 32    0           0       0    1
## 33    0           0       0    1
## 34    0           0       0    1
## 35    0           0       0    1
## 36    1           0       0    1
## 37    0           0       0    1
## 38    1           0       0    1
## 39    0           0       0    1
## 40    0           0       0    1
## 41    0           0       0    1
## 42    0           0       0    1
## 43    1           0       0    1
## 44    0           0       0    1
## 45    0           0       0    1
## 46    0           0       0    1
## 47    0           0       0    1
## 48    0           0       0    1
## 49    0           0       0    1
## 50    0           0       0    1
```

The first dataset has each observation as a different TV show, there are 5611 different observation. Each of them is shown the year the show came out, the IMDb and Rotten Tomatoes rating, and whether the show is in the specified streaming service. A value of 1 means that the show is included in the streaming service and a value of 0 means that the show is not in the streaming service.

```
Netflix_subs <- read.csv("https://raw.githubusercontent.com/bpersaud104/Data698/main/Data%20Collection%
head(Netflix_subs, 40)
```

```
##                               Area     Years Subscribers
## 1         United States and Canada Q1 - 2018    60909000
## 2  Europe,  Middle East and Africa Q1 - 2018    29339000
## 3                    Latin America Q1 - 2018    21260000
## 4                     Asia-Pacific Q1 - 2018     7394000
## 5         United States and Canada Q2 - 2018    61870000
## 6  Europe,  Middle East and Africa Q2 - 2018    31317000
## 7                    Latin America Q2 - 2018    22795000
## 8                     Asia-Pacific Q2 - 2018     8372000
## 9         United States and Canada Q3 - 2018    63010000
## 10 Europe,  Middle East and Africa Q3 - 2018    33836000
## 11                   Latin America Q3 - 2018    24115000
## 12                    Asia-Pacific Q3 - 2018     9461000
## 13        United States and Canada Q4 - 2018    64757000
```

```
## 14 Europe,  Middle East and Africa Q4 - 2018    37818000
## 15                   Latin America Q4 - 2018    26077000
## 16                    Asia-Pacific Q4 - 2018    10607000
## 17        United States and Canada Q1 - 2019    66633000
## 18 Europe,  Middle East and Africa Q1 - 2019    42542000
## 19                   Latin America Q1 - 2019    27547000
## 20                    Asia-Pacific Q1 - 2019    12141000
## 21        United States and Canada Q2 - 2019    66501000
## 22 Europe,  Middle East and Africa Q2 - 2019    44229000
## 23                   Latin America Q2 - 2019    27890000
## 24                    Asia-Pacific Q2 - 2019    12942000
## 25        United States and Canada Q3 - 2019    67114000
## 26 Europe,  Middle East and Africa Q3 - 2019    47355000
## 27                   Latin America Q3 - 2019    29380000
## 28                    Asia-Pacific Q3 - 2019    14485000
## 29        United States and Canada Q4 - 2019    67662000
## 30 Europe,  Middle East and Africa Q4 - 2019    51778000
## 31                   Latin America Q4 - 2019    31417000
## 32                    Asia-Pacific Q4 - 2019    16233000
## 33        United States and Canada Q1 - 2020    69969000
## 34 Europe,  Middle East and Africa Q1 - 2020    58734000
## 35                   Latin America Q1 - 2020    34318000
## 36                    Asia-Pacific Q1 - 2020    19835000
## 37        United States and Canada Q2 - 2020    72904000
## 38 Europe,  Middle East and Africa Q2 - 2020    61483000
## 39                   Latin America Q2 - 2020    36068000
## 40                    Asia-Pacific Q2 - 2020    22492000
```

The second dataset has each observation as a certain area for each quarter from the year 2018 to 2020. There are 40 observations showing 4 quarters for the year 2018 and 2019, and the first two quarters for the year 2020. The number of subscribers are shown for each area by the quarter of that year.

Data Analysis

```
Netflix_data <- shows %>%
  select(Title, Year, Age, IMDb, Rotten.Tomatoes, Netflix) %>%
  filter(Netflix == 1) %>%
  group_by(Title)
Netflix_data
```

```
## # A tibble: 1,931 x 6
## # Groups:   Title [1,925]
##    Title                 Year Age    IMDb Rotten.Tomatoes Netflix
##    <chr>                <int> <chr> <dbl> <chr>              <int>
##  1 Breaking Bad          2008 18+     9.5 96%                    1
##  2 Stranger Things       2016 16+     8.8 93%                    1
##  3 Money Heist           2017 18+     8.4 91%                    1
##  4 Sherlock              2010 16+     9.1 78%                    1
##  5 Better Call Saul      2015 18+     8.7 97%                    1
##  6 The Office            2005 16+     8.9 81%                    1
##  7 Black Mirror          2011 18+     8.8 83%                    1
```

```
##  8 Supernatural                2005 16+     8.4 93%                   1
##  9 Peaky Blinders              2013 18+     8.8 92%                   1
## 10 Avatar: The Last Airbender  2005 7+      9.2 100%                  1
## # ... with 1,921 more rows
```

```
Hulu_data <- shows %>%
  select(Title, Year, Age, IMDb, Rotten.Tomatoes, Hulu) %>%
  filter(Hulu == 1) %>%
  group_by(Title)
Hulu_data
```

```
## # A tibble: 1,754 x 6
## # Groups:   Title [1,739]
##    Title                             Year Age    IMDb Rotten.Tomatoes  Hulu
##    <chr>                            <int> <chr> <dbl> <chr>           <int>
##  1 Attack on Titan                   2013 16+     8.8 94%                 1
##  2 Fullmetal Alchemist: Brotherhood  2009 18+     9.1 100%                1
##  3 Community                         2009 7+      8.5 88%                 1
##  4 Parks and Recreation              2009 16+     8.6 93%                 1
##  5 Twin Peaks                        1990 18+     8.8 89%                 1
##  6 One-Punch Man                     2015 16+     8.8 100%                1
##  7 The Good Place                    2016 16+     8.2 97%                 1
##  8 Grey's Anatomy                    2005 16+     7.6 83%                 1
##  9 Arrested Development              2003 16+     8.7 75%                 1
## 10 How to Get Away with Murder       2014 16+     8.1 88%                 1
## # ... with 1,744 more rows
```

```
Prime_video_data <- shows %>%
  select(Title, Year, Age, IMDb, Rotten.Tomatoes, Prime.Video) %>%
  filter(Prime.Video == 1) %>%
  group_by(Title)
Prime_video_data
```

```
## # A tibble: 2,144 x 6
## # Groups:   Title [2,138]
##    Title                             Year Age    IMDb Rotten.Tomatoes Prime.Video
##    <chr>                            <int> <chr> <dbl> <chr>                <int>
##  1 Parks and Recreation              2009 16+     8.6 "93%"                    1
##  2 Star Trek: The Next Generation    1987 7+      8.6 "89%"                    1
##  3 The Good Wife                     2009 16+     8.3 "94%"                    1
##  4 Schitt's Creek                    2015 16+     8.4 "50%"                    1
##  5 Burn Notice                       2007 7+      7.9 "88%"                    1
##  6 American Horror Story             2011 18+     8   ""                       1
##  7 Star Trek                         1966 7+      8.3 "80%"                    1
##  8 Mushi-Shi                         2005 16+     8.5 "100%"                   1
##  9 Star Trek: Deep Space Nine        1993 7+      7.9 "90%"                    1
## 10 Law & Order: Special Victims U~   1999 16+     8   ""                       1
## # ... with 2,134 more rows
```

```
Disney_plus_data <- shows %>%
  select(Title, Year, Age, IMDb, Rotten.Tomatoes, Disney.) %>%
  filter(Disney. == 1) %>%
```

```
  group_by(Title)
Disney_plus_data
```

```
## # A tibble: 180 x 6
## # Groups:   Title [179]
##    Title                          Year Age    IMDb Rotten.Tomatoes Disney.
##    <chr>                         <int> <chr> <dbl> <chr>             <int>
##  1 Lab Rats                       2012 7+      6.6 ""                    1
##  2 America's Funniest Home Videos  1989 7+      6.2 ""                    1
##  3 Brain Games                    2011 7+      8.3 ""                    1
##  4 Jessie                         2011 all     5.9 ""                    1
##  5 PJ Masks                       2015 all     5.6 ""                    1
##  6 Best Friends Whenever          2015 all     5.5 ""                    1
##  7 The Simpsons                   1989 7+      8.7 "85%"                 1
##  8 Gravity Falls                  2012 7+      8.9 "100%"                1
##  9 Marvel's Runaways              2017 16+     7   "87%"                 1
## 10 Star vs. the Forces of Evil    2015 7+      8   ""                    1
## # ... with 170 more rows
```

From sorting the data we see what shows are available for each streaming services, Netflix, Hulu, Prime video, and Disney+. We see that Prime Video contains more shows than the other streaming services while Disney+ contains the least amount. From the data we can see that Netflix contains more shows that have higher ratings. This is based on the IMDb and Rotten Tomatoes scores.

```
ggplot(Netflix_data, aes(x = Title, y = IMDb)) + geom_point()
```

```
## Warning: Removed 120 rows containing missing values (geom_point).
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x14
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

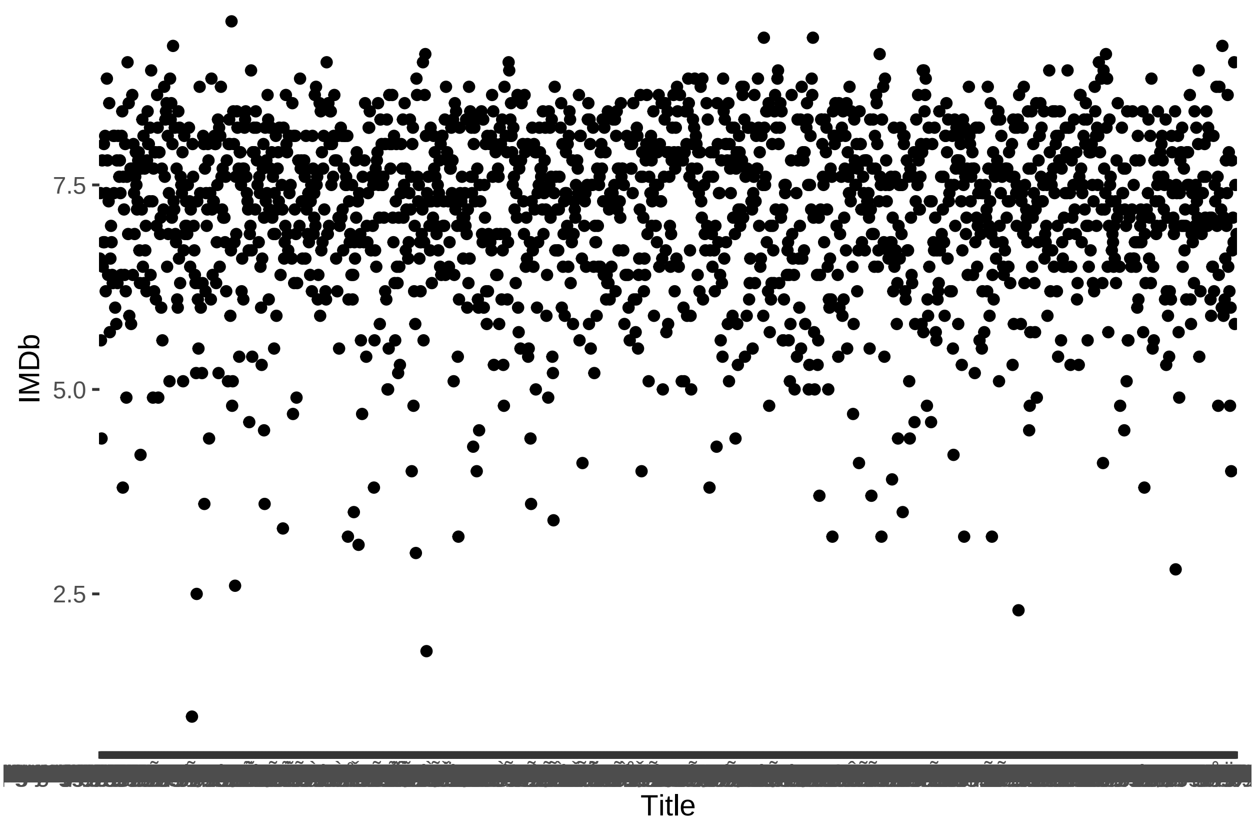
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0xe
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x1d

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
```

```
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x4

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x81

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x4

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x81

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x81
```




```
ggplot(Hulu_data, aes(x = Title, y = IMDb)) + geom_point()
```

```
## Warning: Removed 237 rows containing missing values (geom_point).
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

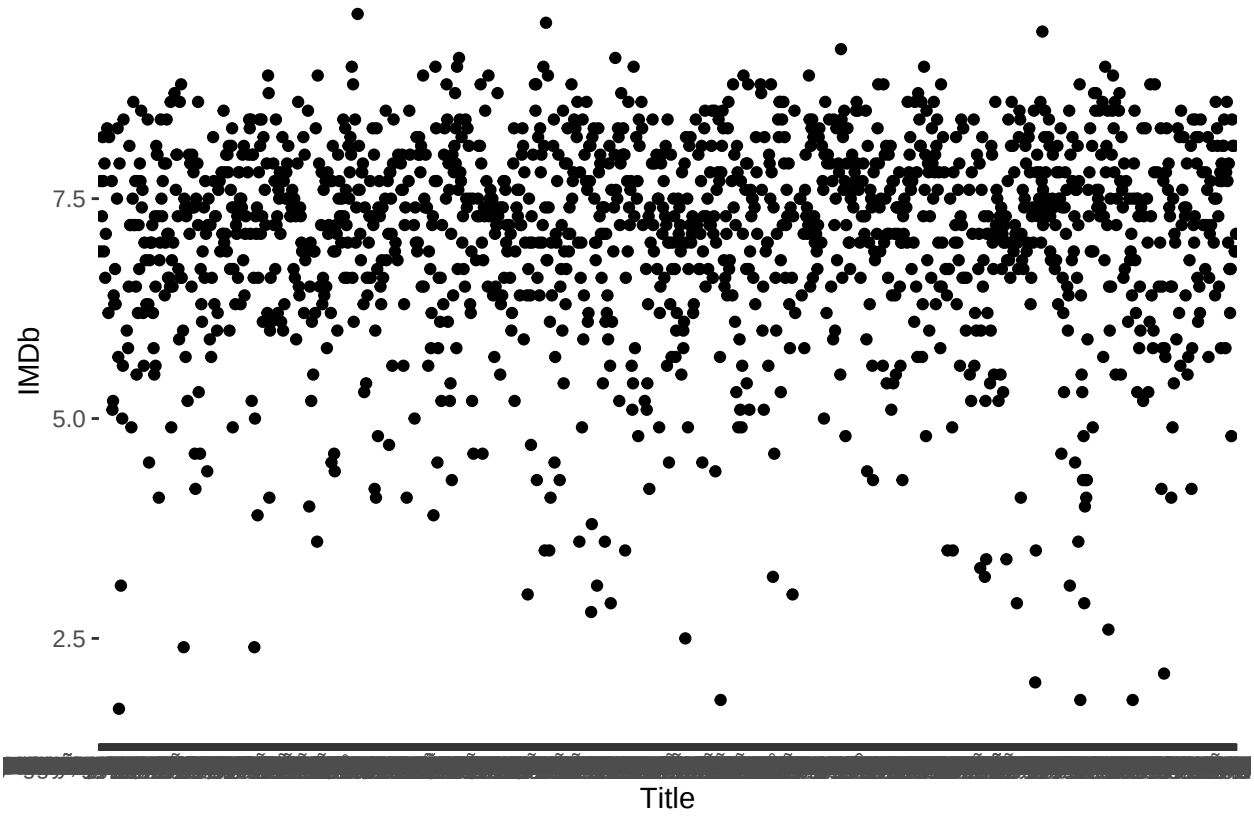
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x81

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x81

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x81
```

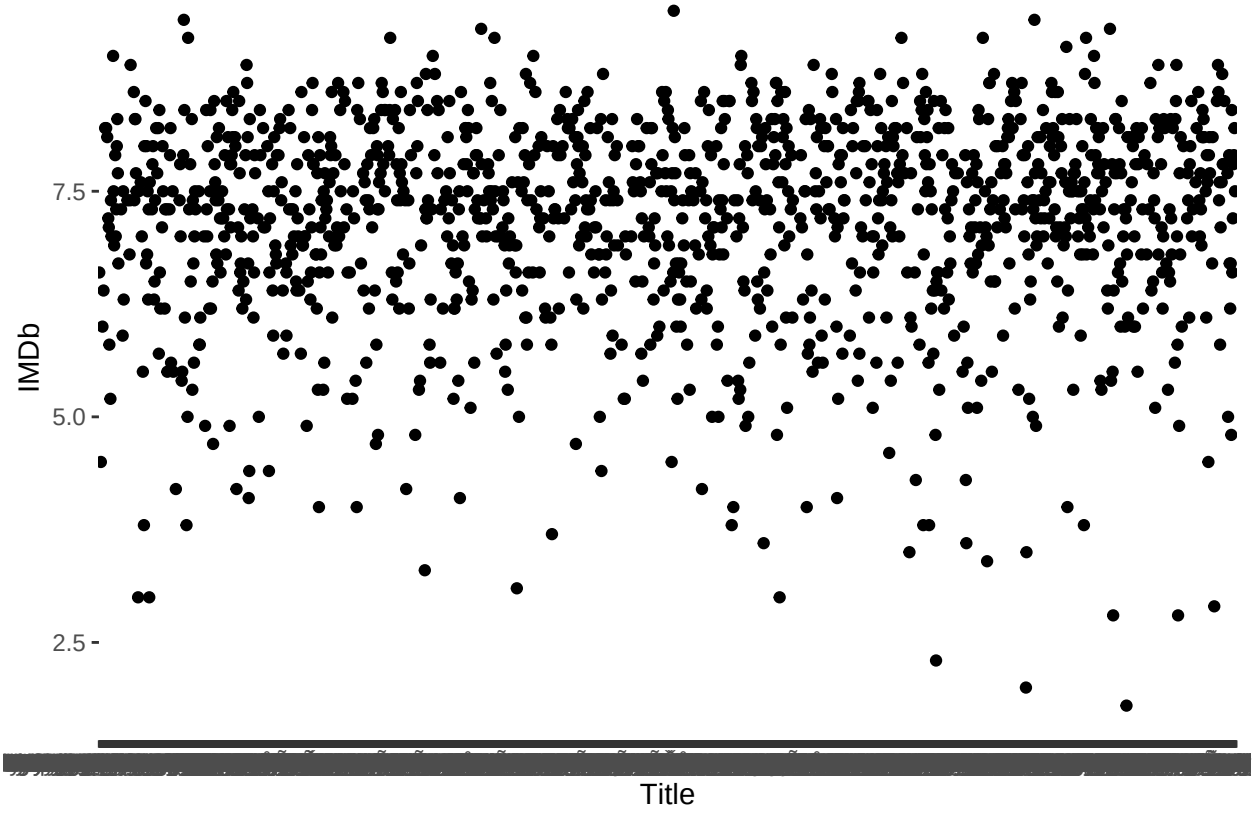



```
ggplot(Prime_video_data, aes(x = Title, y = IMDb)) + geom_point()
```

```
## Warning: Removed 837 rows containing missing values (geom_point).

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90
```

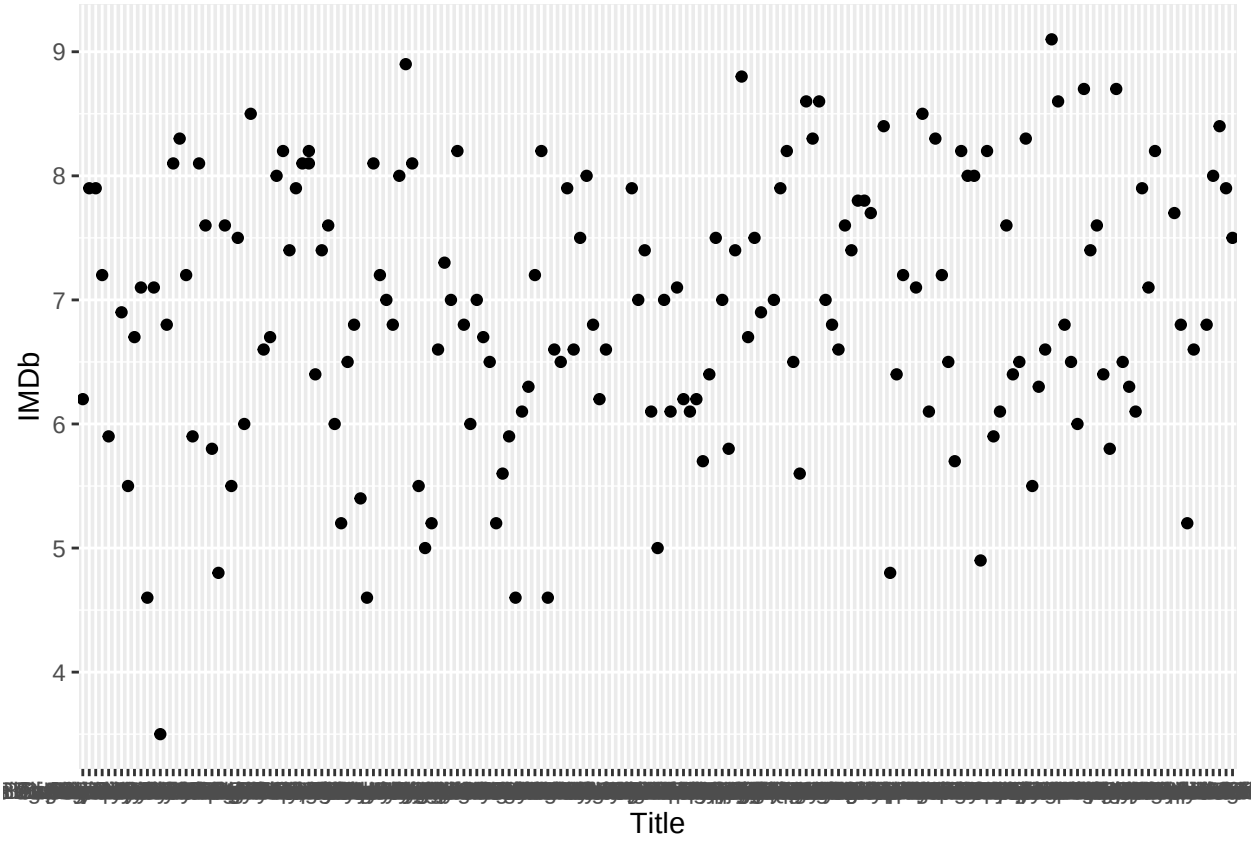
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x90

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x90

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x90
```




```
ggplot(Disney_plus_data, aes(x = Title, y = IMDb)) + geom_point()
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```



Since Netflix and Amazon Prime Video seem to be two of the biggest streaming services let us look into the number of people who have subscribed to these services over the years. This will give us an insight on whether people are using these services and how much they have grown. I will be using data I found on Netflix subscriptions to show this.

For the second dataset since the data is split by different regions and shows the number of subscribers from the year 2018 to the first half of 2020 a shiny app was created to visualize the data. For this visualization a shiny app was created. It can be viewed here: https://bpersaud104.shinyapps.io/Netflix_Analysis/

From the shiny app we can see the number of subscriptions for Netflix divided by four areas, United States and Canada, Europe, Middle East, and Africa, Latin American, and Asia-Pacific. For all four areas the number of subscribers have increased since 2018. North America and Canada has the most subscribers going from around sixty million in 2018 to around 72 million in the first half of 2020. Europe, Middle East, and Africa has seen the most rise in subscribers going from around 29 million in 2018 to around 61 million in the first half of 2020.

Model Building

Two models will be made to show the relationship between the rating of a show and whether a person is subscribed to a streaming service or not. One model will be a count regression model and the other one will be a random forest.

```
model1 <- lm(IMDb ~ Netflix + Hulu + Prime.Video + Disney., data = shows)
summary(model1)
```

```
##
```

```
## Call:
## lm(formula = IMDb ~ Netflix + Hulu + Prime.Video + Disney., data = shows)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1456 -0.5456  0.1825  0.7825  2.5825
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.912859   0.064577 107.049  < 2e-16 ***
## Netflix      0.232791   0.064223   3.625 0.000292 ***
## Hulu         0.104682   0.061036   1.715 0.086398 .
## Prime.Video  0.239362   0.062738   3.815 0.000138 ***
## Disney.     -0.008604   0.103832  -0.083 0.933961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.129 on 4445 degrees of freedom
##   (1161 observations deleted due to missingness)
## Multiple R-squared:  0.005946,   Adjusted R-squared:  0.005051
## F-statistic: 6.647 on 4 and 4445 DF,  p-value: 2.493e-05
```

The first model shows a p-value that is very high. This large value tells me that whether a streaming service has high ratings shows does not affect whether someone subscribes or not.

```
sapply(shows, class)
```

```
##               X           Title            Year             Age            IMDb
##       "integer"     "character"       "integer"     "character"       "numeric"
## Rotten.Tomatoes         Netflix            Hulu     Prime.Video         Disney.
##     "character"       "integer"       "integer"       "integer"       "integer"
##            type
##       "integer"
```

We can see above that some columns are not accurately. Le us transform the data to change the type of Netflix, Hulu, Prime.Video, and Disney.

```
ratings <- transform(shows, X = as.integer(X), Title = as.character(Title), Year = as.integer(Year), Ag
ratings <- na.omit(ratings)
summary(ratings)
```

```
##        X           Title                Year          Age
##  Min.   :   0   Length:4450        Min.   :1934   Length:4450
##  1st Qu.:1114   Class :character   1st Qu.:2009   Class :character
##  Median :2344   Mode  :character   Median :2014   Mode  :character
##  Mean   :2392                      Mean   :2011
##  3rd Qu.:3693                      3rd Qu.:2017
##  Max.   :5602                      Max.   :2020
##       IMDb       Rotten.Tomatoes    Netflix  Hulu     Prime.Video Disney.
##  Min.   :1.000   Length:4450        0:2639   0:2933   0:3143      0:4281
##  1st Qu.:6.600   Class :character   1:1811   1:1517   1:1307      1: 169
##  Median :7.300   Mode  :character
```

```
##  Mean   :7.113
##  3rd Qu.:7.900
##  Max.   :9.600
##       type
##  Min.   :1
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
```

A train/test split will be set up to use for the random forest model.

```
set.seed(17)
sample <- sample.split(ratings$IMDb, SplitRatio = 0.80)
train <- subset(ratings, sample == TRUE)
test <- subset(ratings, sample == FALSE)
dim(train)
```

```
## [1] 3560   11
```

```
dim(test)
```

```
## [1] 890  11
```

```
model2 <- randomForest(IMDb ~ Netflix + Hulu + Prime.Video + Disney., data = train)
model2
```

```
##
## Call:
##  randomForest(formula = IMDb ~ Netflix + Hulu + Prime.Video +      Disney., data = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 1.284291
##                     % Var explained: 0.34
```

A model was created using random forest on the train dataset.

```
pred <- predict(model2, newdata = test[-12])
confusion_matrix <- table(observed = test[,11], predicted = pred)
confusion_matrix
```

```
##         predicted
## observed 6.8126414859897 6.98745155839828 6.99685537548213 7.03498048718308
##        1               2                1               22              257
##         predicted
## observed 7.06747450385424 7.10156756077865 7.14302750808382 7.14961256431749
##        1                5                8              331               23
##         predicted
## observed 7.15991846907594 7.19734861586613 7.19865178247932
##        1              211                5               25
```

Here we can see a confusion matrix set up using the model created. It shows that for the train dataset most of the data lies in ratings between 6.8 and 7.1. I would say that this model does not show a good relationship between whether a rating of a show affects if someone subscribes to a streaming service.