Lab2 - Team Turquoise

Samantha Williams, Kate Kostelc, Mick Rejniak, Sean Baughman

3/20/2022

Contents

0.1 Proposal

We will conduct an analysis of IMDB movie data that clarifies how different movie attributes affect the gross revenue of a movie. The examination will include factors such as director, actor(s), genre, etc.

0.2 Dataset

Our project will utilize the IMDB Movies Dataset. The data can be found here as part of Kaggle.com's library of datasets. It contains 1,000 records with 16 columns that describe the top 1,000 highest rated movies by IMDb, an online entertainment database. The dataset includes the following columns that were scraped from IMDB:

- Poster_Link - Link of the poster that imdb using
- Series_Title = Name of the movie
- Released_Year - Year at which that movie released
- Certificate - Certificate earned by that movie
- Runtime - Total runtime of the movie
- Genre - Genre of the movie
- IMDB_Rating - Rating of the movie at IMDB site
- Overview - mini story/ summary
- Meta_score - Score earned by the movie
- Director - Name of the Director
- Star1,Star2,Star3,Star4 - Name of the Stars
- Noofvotes - Total number of votes
- Gross - Money earned by that movie

```
## Rows: 1,000
## Columns: 16
## $ Poster_Link   <chr> "https://m.media-amazon.com/images/M/MV5BMDFkYTc0MGEtZmN~
## $ Series_Title  <chr> "The Shawshank Redemption", "The Godfather", "The Dark K~
## $ Released_Year <chr> "1994", "1972", "2008", "1974", "1957", "2003", "1994", ~
## $ Certificate   <chr> "A", "A", "UA", "A", "U", "U", "A", "A", "UA", "A", "U",~
## $ Runtime       <chr> "142 min", "175 min", "152 min", "202 min", "96 min", "2~
## $ Genre         <chr> "Drama", "Crime, Drama", "Action, Crime, Drama", "Crime,~
## $ IMDB_Rating   <dbl> 9.3, 9.2, 9.0, 9.0, 9.0, 8.9, 8.9, 8.9, 8.8, 8.8, 8.8, 8~
## $ Overview      <chr> "Two imprisoned men bond over a number of years, finding~
## $ Meta_score    <dbl> 80, 100, 84, 90, 96, 94, 94, 94, 74, 66, 92, 82, 90, 87,~
## $ Director      <chr> "Frank Darabont", "Francis Ford Coppola", "Christopher N~
## $ Star1         <chr> "Tim Robbins", "Marlon Brando", "Christian Bale", "Al Pa~
## $ Star2         <chr> "Morgan Freeman", "Al Pacino", "Heath Ledger", "Robert D~
## $ Star3         <chr> "Bob Gunton", "James Caan", "Aaron Eckhart", "Robert Duv~
## $ Star4         <chr> "William Sadler", "Diane Keaton", "Michael Caine", "Dian~
## $ No_of_Votes   <dbl> 2343110, 1620367, 2303232, 1129952, 689845, 1642758, 182~
## $ Gross         <dbl> 28341469, 134966411, 534858444, 57300000, 4360000, 37784~
```

```
## Rows: 1,000
## Columns: 19
## $ Poster_Link   <chr> "https://m.media-amazon.com/images/M/MV5BMDFkYTc0MGEtZmN~
## $ Series_Title  <chr> "The Shawshank Redemption", "The Godfather", "The Dark K~
## $ Released_Year <chr> "1994", "1972", "2008", "1974", "1957", "2003", "1994", ~
## $ Certificate   <chr> "A", "A", "UA", "A", "U", "U", "A", "A", "UA", "A", "U",~
## $ Runtime       <chr> "142", "175", "152", "202", "96", "201", "154", "195", "~
## $ Runtime_Units <chr> "min", "min", "min", "min", "min", "min", "min", "min", ~
## $ Genre1        <chr> "Drama", "Crime", "Action", "Crime", "Crime", "Action", ~
```

```
## $ Genre2        <chr> NA, "Drama", "Crime", "Drama", "Drama", "Adventure", "Dr~
## $ Genre3        <chr> NA, NA, "Drama", NA, NA, "Drama", NA, "History", "SciFi"~
## $ IMDB_Rating   <dbl> 9.3, 9.2, 9.0, 9.0, 9.0, 8.9, 8.9, 8.9, 8.8, 8.8, 8.8, 8~
## $ Overview      <chr> "Two imprisoned men bond over a number of years, finding~
## $ Meta_score    <dbl> 80, 100, 84, 90, 96, 94, 94, 94, 74, 66, 92, 82, 90, 87,~
## $ Director      <chr> "Frank Darabont", "Francis Ford Coppola", "Christopher N~
## $ Star1         <chr> "Tim Robbins", "Marlon Brando", "Christian Bale", "Al Pa~
## $ Star2         <chr> "Morgan Freeman", "Al Pacino", "Heath Ledger", "Robert D~
## $ Star3         <chr> "Bob Gunton", "James Caan", "Aaron Eckhart", "Robert Duv~
## $ Star4         <chr> "William Sadler", "Diane Keaton", "Michael Caine", "Dian~
## $ No_of_Votes   <dbl> 2343110, 1620367, 2303232, 1129952, 689845, 1642758, 182~
## $ Gross         <dbl> 28341469, 134966411, 534858444, 57300000, 4360000, 37784~
```

0.3 Core Questions/Investigations:

We work for a start-up streaming service company and want to maximize viewer subscriptions to our new platform by offering the most watched films.

Does a commercially successful movie become a highly rated movie? #Does a highly rated movie predict higher numbers of views?

-Output: Meta_score -Beta: Gross -Beta2: IMBD_Rating -Beta3: No_of_Votes -Beta4: Runtime

0.4 General EDA:

```
#Range of $1,305 to $936,662,225
#There are missing fields: 169
library(scales) #should make neat commas of the lower plot, may need to use ggplot
```

```
##
## Attaching package: 'scales'
## The following object is masked from 'package:purrr':
##
##     discard
## The following object is masked from 'package:readr':
##
##     col_factor
```

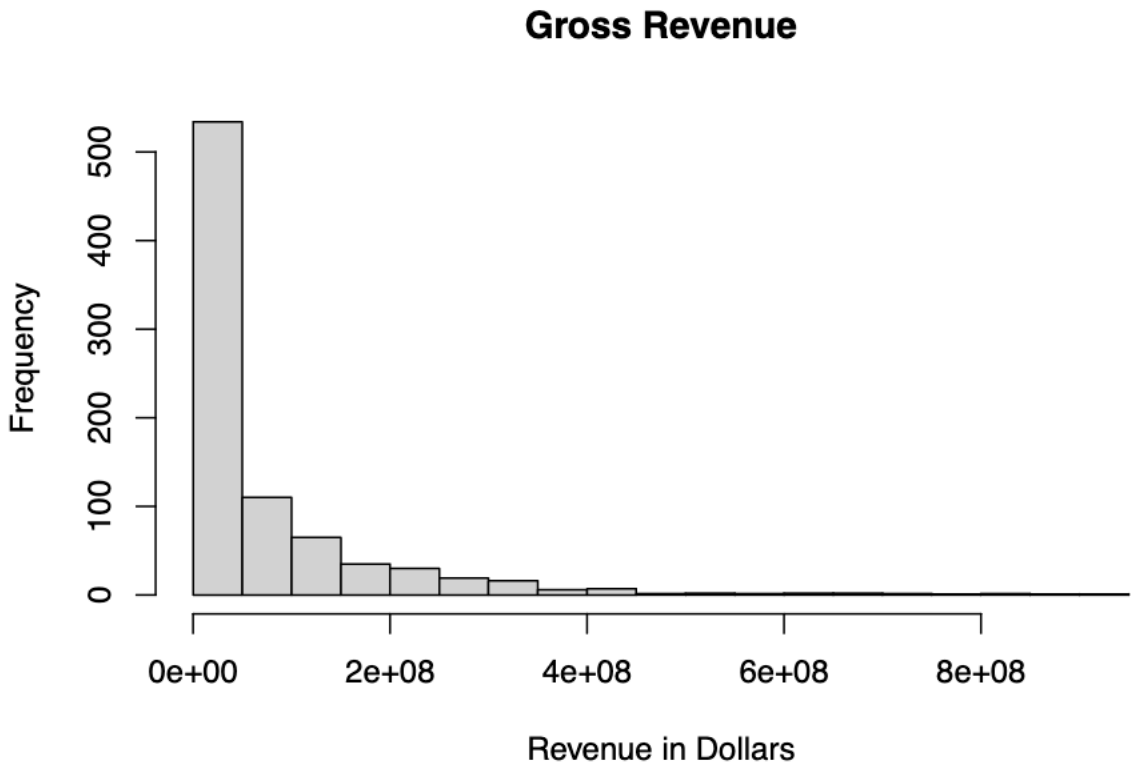
```
summary(movies$Gross)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##      1305   3253559  23530892  68034751  80750894 936662225       169
```

```
sum(is.na(movies$Gross))
```

```
## [1] 169
```

```
hist(movies$Gross, labels=comma, main='Gross Revenue', xlab="Revenue in Dollars", breaks = 30)
```



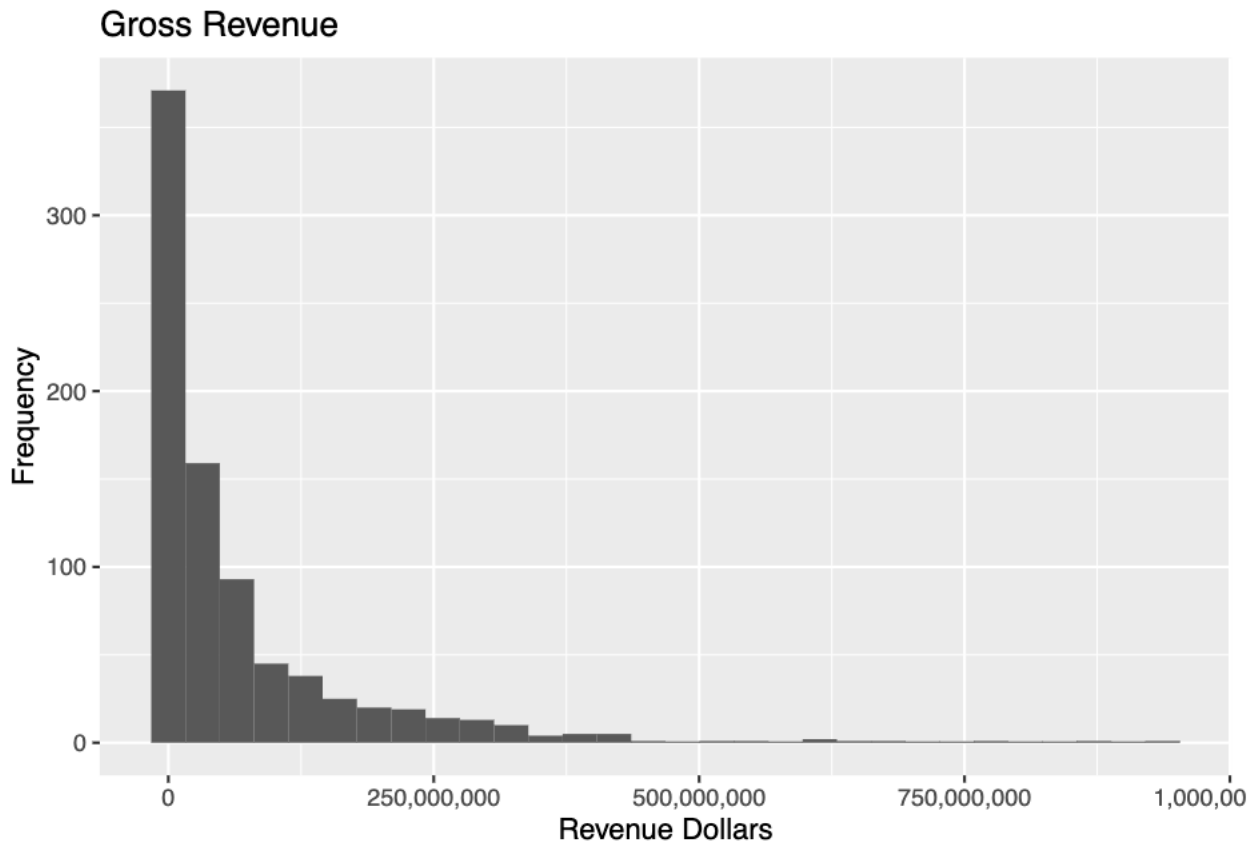

```
p <- ggplot(movies, aes(x=Gross)) +
  geom_histogram(bindwidth=1) +
  scale_x_continuous(labels = comma) +
  labs(title = 'Gross Revenue', x = "Revenue Dollars", y = "Frequency")
```

```
## Warning: Ignoring unknown parameters: bindwidth
```

```
p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 169 rows containing non-finite values (stat_bin).
```




```
library(dplyr)
#Looking at who has directed the most movies

#table(movies$Director, useNA = "ifany")
maximum <- movies %>%
  count(Director, sort = TRUE) %>%
  slice_max(n)
maximum
```

```
## # A tibble: 1 x 2
##   Director              n
##   <chr>             <int>
## 1 Alfred Hitchcock    14
```

```
#Looking at the runtimes of movies
table(movies$Runtime, useNA = "ifany")
```

```
##
## 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119
##  23  22  18  14  15  16  15  16  19  11  20  10  15  22  11  13  12  16  17  16
## 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139
##  17  15  20  11  13  12  15  16  12  22  23   9  13  10  14   9  11  12  15   8
## 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
##   7   7   6   9  12   5   7   7   3   4   3   3   6   8   5   5   6   6   4   4
## 160 161 162 163 164 165 166 167 168 169 170 171 172 174 175 177 178 179 180 181
##   5   6   5   3   2   5   1   3   1   4   7   1   3   1   1   1   5   2   4   3
## 183 184 185 186 188 189 191 192 193 194 195 196 197 201 202 204 205 207 209 210
```

```
##   2   1   1   2   3   3   1   1   1   1   1   1   2   2   2   1   1   1   1   1
## 212 220 224 228 229 238 242 321  45  64  67  68  69  70  71  72  75  76  78  79
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   2   2   3   1   1
##  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
##   6   4   2   2   3   5   6   8   8   7  10   9   9  11  14  13  17   9  11  14
```

```
# One film with 321 minutes and the shortest film is 45 minutes. Do we want to classify what a full len

#runtime is all listed in minutes
table(movies$Runtime_Units, useNA = "ifany")
```

```
##
##  min
## 1000
```

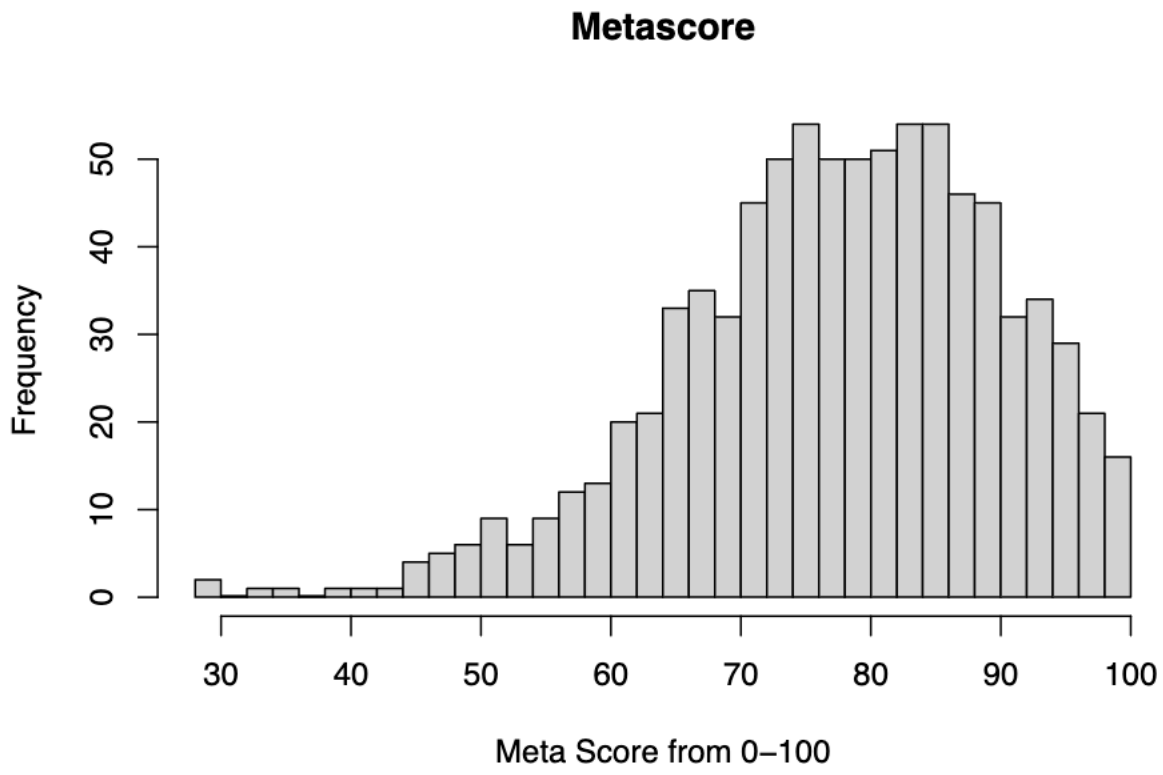
```
summary(movies$Meta_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   28.00   70.00   79.00   77.97   87.00  100.00     157
```

```
#157 film titles have missing values. Lowest is 28 and highest is 100.
sum(is.na(movies$Meta_score))
```

```
## [1] 157
```

```
hist(movies$Meta_score, labels=comma, main='Metascore', xlab="Meta Score from 0-100", breaks = 30)
```




```
summary(movies$IMDB_Rating)
```

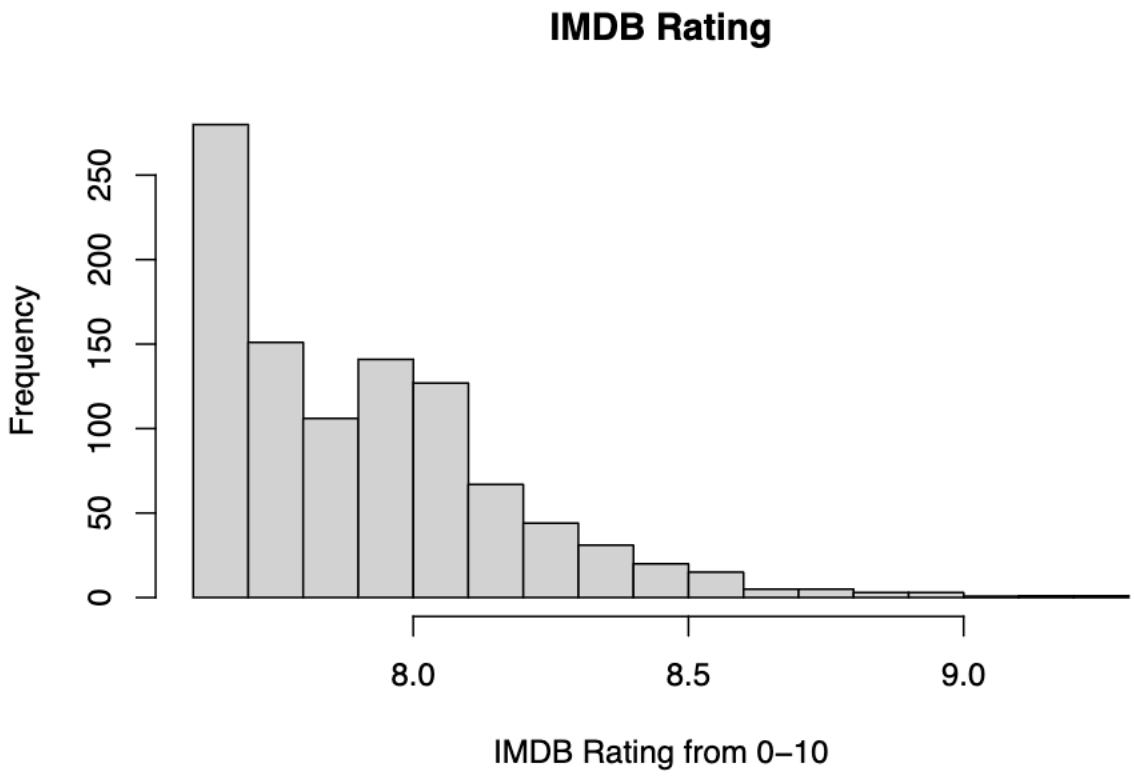
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##   7.600   7.700   7.900   7.949   8.100   9.300
```

```
#no missing values with the lowest rating being a 7.6 and the highest being 9.3.
sum(is.na(movies$IMDB_Rating))
```

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

```
hist(movies$IMDB_Rating, labels=comma, main='IMDB Rating', xlab="IMDB Rating from 0-10", breaks = 20)
```




```
colSums(is.na(movies))
```

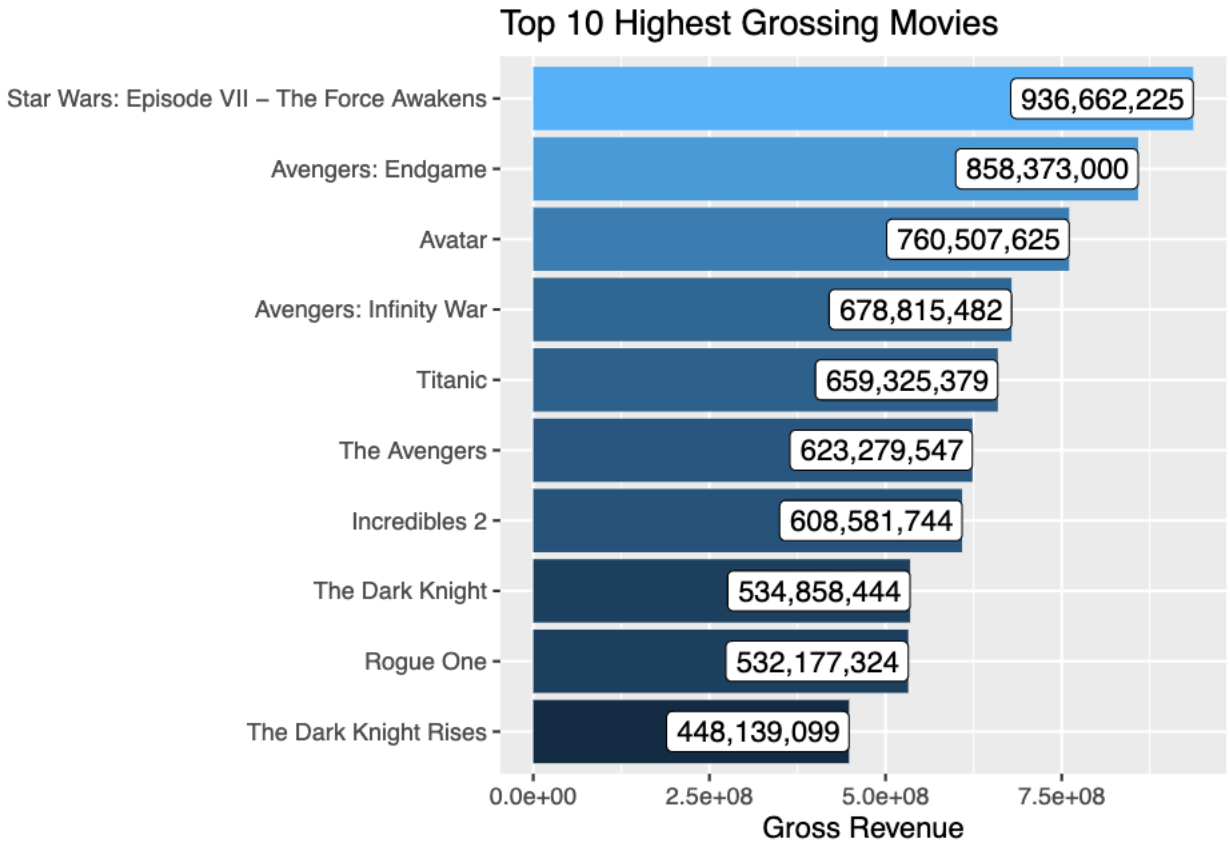
```
##   Poster_Link  Series_Title Released_Year   Certificate       Runtime
##             0             0             0           101             0
## Runtime_Units        Genre1        Genre2        Genre3   IMDB_Rating
##             0             0           105           354             0
##      Overview    Meta_score      Director         Star1         Star2
##             0           157             0             0             0
##         Star3         Star4   No_of_Votes         Gross
##             0             0             0           169
```

```
highest_gross <- movies[order(movies$Gross,decreasing = T),][1:10,]
highest_gross
```

```
## # A tibble: 10 x 19
##    Poster_Link      Series_Title Released_Year Certificate Runtime Runtime_Units
##    <chr>            <chr>        <chr>         <chr>       <chr>   <chr>
##  1 https://m.media~ Star Wars: ~ 2015          U           138     min
##  2 https://m.media~ Avengers: E~ 2019          UA          181     min
##  3 https://m.media~ Avatar       2009          UA          162     min
```

```
##  4 https://m.media~ Avengers: I~ 2018          UA          149     min
##  5 https://m.media~ Titanic      1997          UA          194     min
##  6 https://m.media~ The Avengers 2012          UA          143     min
##  7 https://m.media~ Incredibles~ 2018          UA          118     min
##  8 https://m.media~ The Dark Kn~ 2008          UA          152     min
##  9 https://m.media~ Rogue One    2016          UA          133     min
## 10 https://m.media~ The Dark Kn~ 2012          UA          164     min
## # ... with 13 more variables: Genre1 <chr>, Genre2 <chr>, Genre3 <chr>,
## #   IMDB_Rating <dbl>, Overview <chr>, Meta_score <dbl>, Director <chr>,
## #   Star1 <chr>, Star2 <chr>, Star3 <chr>, Star4 <chr>, No_of_Votes <dbl>,
## #   Gross <dbl>
```

```
#Interseting that the Franchise films seem to be doing the best in the top 10 of highest grossing films
ggplot(highest_gross, aes(x = Gross,y = reorder(Series_Title, Gross))) +
  geom_col(aes(fill = Gross), show.legend = F) +
  labs(title = "Top 10 Highest Grossing Movies", x = "Gross Revenue", y = NULL) +
  geom_label(aes(label = comma(Gross)), hjust = 1)
```




```
highest_meta <- movies[order(movies$Meta_score,decreasing = T),][1:10,]
highest_meta
```

```
## # A tibble: 10 x 19
##    Poster_Link      Series_Title Released_Year Certificate Runtime Runtime_Units
##    <chr>            <chr>        <chr>         <chr>       <chr>   <chr>
##  1 https://m.media~ The Godfath~ 1972          A           175     min
##  2 https://m.media~ Casablanca   1942          U           102     min
##  3 https://m.media~ Rear Window  1954          U           112     min
```

```
##  4 https://m.media~ Lawrence of~ 1962          U           228     min
##  5 https://m.media~ Vertigo      1958          A           128     min
##  6 https://m.media~ Citizen Kane 1941          UA          119     min
##  7 https://m.media~ Trois coule~ 1994          U           99      min
##  8 https://m.media~ Fanny och A~ 1982          A           188     min
##  9 https://m.media~ Il conformi~ 1970          UA          113     min
## 10 https://m.media~ Sweet Smell~ 1957          Approved    96      min
## # ... with 13 more variables: Genre1 <chr>, Genre2 <chr>, Genre3 <chr>,
## #   IMDB_Rating <dbl>, Overview <chr>, Meta_score <dbl>, Director <chr>,
## #   Star1 <chr>, Star2 <chr>, Star3 <chr>, Star4 <chr>, No_of_Votes <dbl>,
## #   Gross <dbl>
```

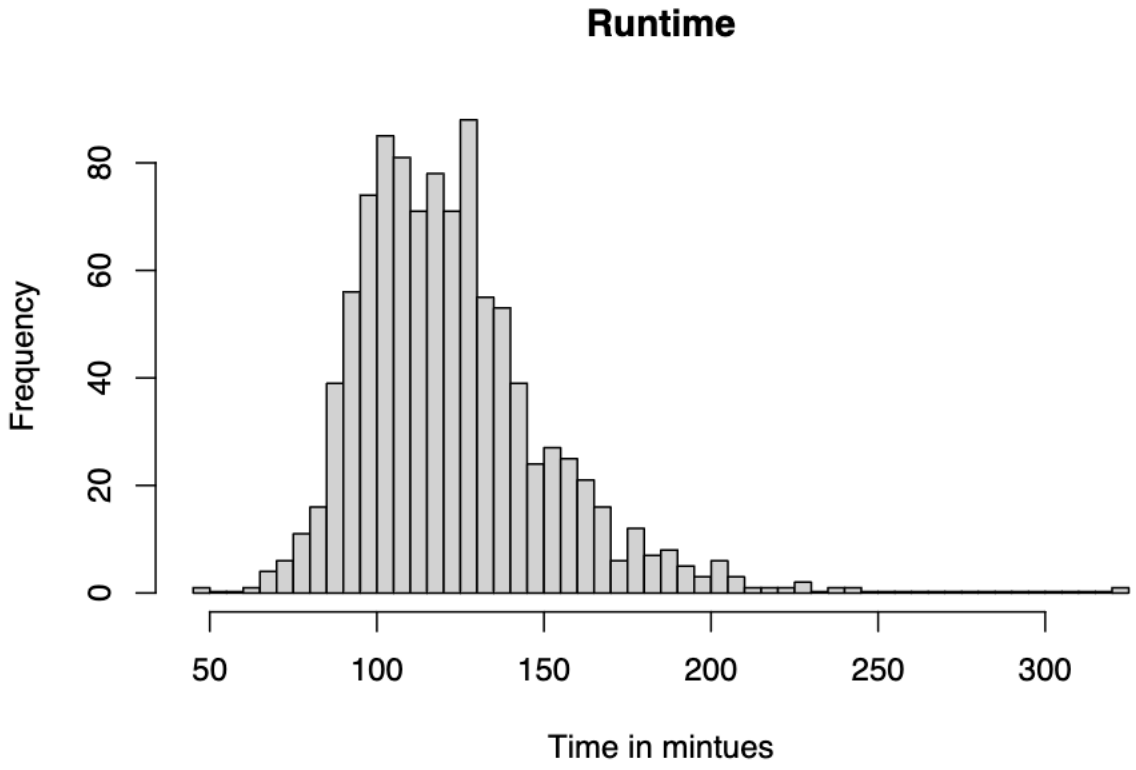
```
highest_imdb <- movies[order(movies$IMDB_Rating,decreasing = T),][1:10,]
highest_imdb
```

```
## # A tibble: 10 x 19
##    Poster_Link      Series_Title Released_Year Certificate Runtime Runtime_Units
##    <chr>            <chr>        <chr>         <chr>       <chr>   <chr>
##  1 https://m.media~ The Shawsha~ 1994          A           142     min
##  2 https://m.media~ The Godfath~ 1972          A           175     min
##  3 https://m.media~ The Dark Kn~ 2008          UA          152     min
##  4 https://m.media~ The Godfath~ 1974          A           202     min
##  5 https://m.media~ 12 Angry Men 1957          U           96      min
##  6 https://m.media~ The Lord of~ 2003          U           201     min
##  7 https://m.media~ Pulp Fiction 1994          A           154     min
##  8 https://m.media~ Schindler's~ 1993          A           195     min
##  9 https://m.media~ Inception    2010          UA          148     min
## 10 https://m.media~ Fight Club   1999          A           139     min
## # ... with 13 more variables: Genre1 <chr>, Genre2 <chr>, Genre3 <chr>,
## #   IMDB_Rating <dbl>, Overview <chr>, Meta_score <dbl>, Director <chr>,
## #   Star1 <chr>, Star2 <chr>, Star3 <chr>, Star4 <chr>, No_of_Votes <dbl>,
## #   Gross <dbl>
```

```
#is there a sweet spot of runtime that maximizes gross?
#as.numeric(movies$Runtime)#this may need to move higher up in the EDA
highest_runtime <- movies[order(as.numeric(movies$Runtime),decreasing = T),][1:10,]
highest_runtime
```

```
## # A tibble: 10 x 19
##    Poster_Link      Series_Title Released_Year Certificate Runtime Runtime_Units
##    <chr>            <chr>        <chr>         <chr>       <chr>   <chr>
##  1 https://m.media~ Gangs of Wa~ 2012          A           321     min
##  2 https://m.media~ Hamlet       1996          PG-13       242     min
##  3 https://m.media~ Gone with t~ 1939          U           238     min
##  4 https://m.media~ Once Upon a~ 1984          A           229     min
##  5 https://m.media~ Lawrence of~ 1962          U           228     min
##  6 https://m.media~ Lagaan: Onc~ 2001          U           224     min
##  7 https://m.media~ The Ten Com~ 1956          U           220     min
##  8 https://m.media~ Ben-Hur      1959          U           212     min
##  9 https://m.media~ Swades: We,~ 2004          U           210     min
## 10 https://m.media~ The Irishman 2019          R           209     min
## # ... with 13 more variables: Genre1 <chr>, Genre2 <chr>, Genre3 <chr>,
## #   IMDB_Rating <dbl>, Overview <chr>, Meta_score <dbl>, Director <chr>,
## #   Star1 <chr>, Star2 <chr>, Star3 <chr>, Star4 <chr>, No_of_Votes <dbl>,
## #   Gross <dbl>
```

```
hist(as.numeric(movies$Runtime), labels=comma, main='Runtime', xlab="Time in mintues", breaks = 40)
```

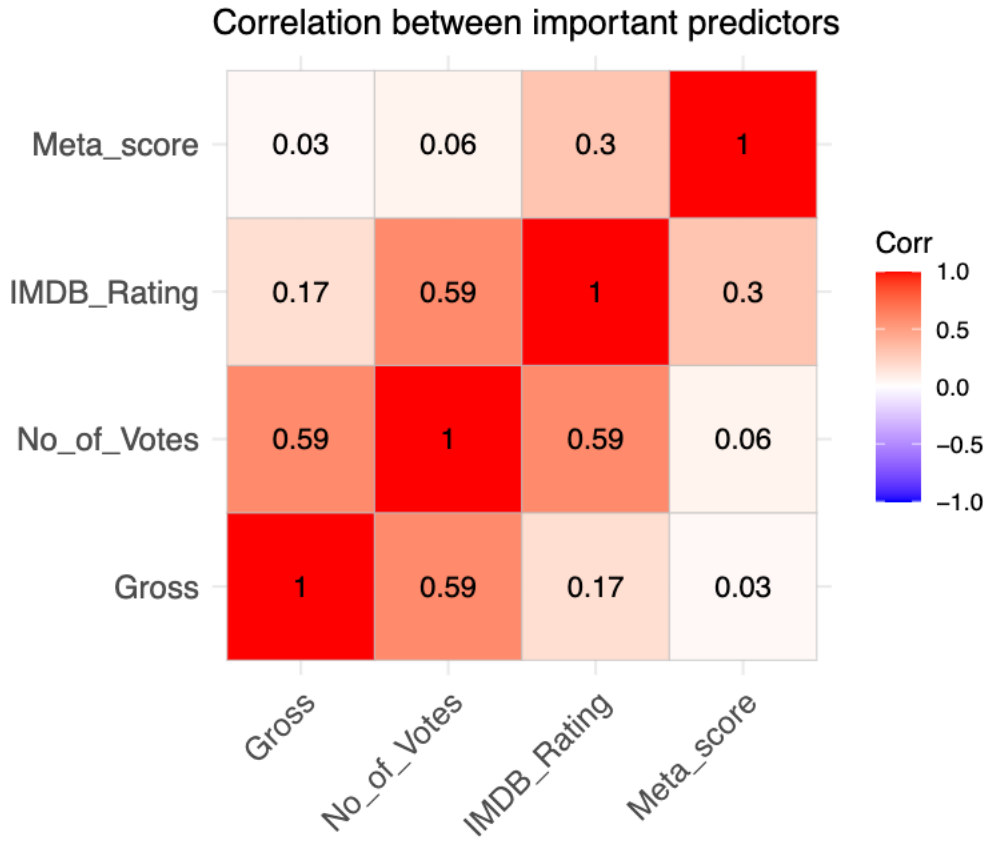



```
#What values do we want on here as we will need to make a new table: Gross, IMDB_Rating, Meta_Score, Ru
install.packages("ggcorrplot")
```

```
## Installing package into '/opt/r'
## (as 'lib' is unspecified)
```

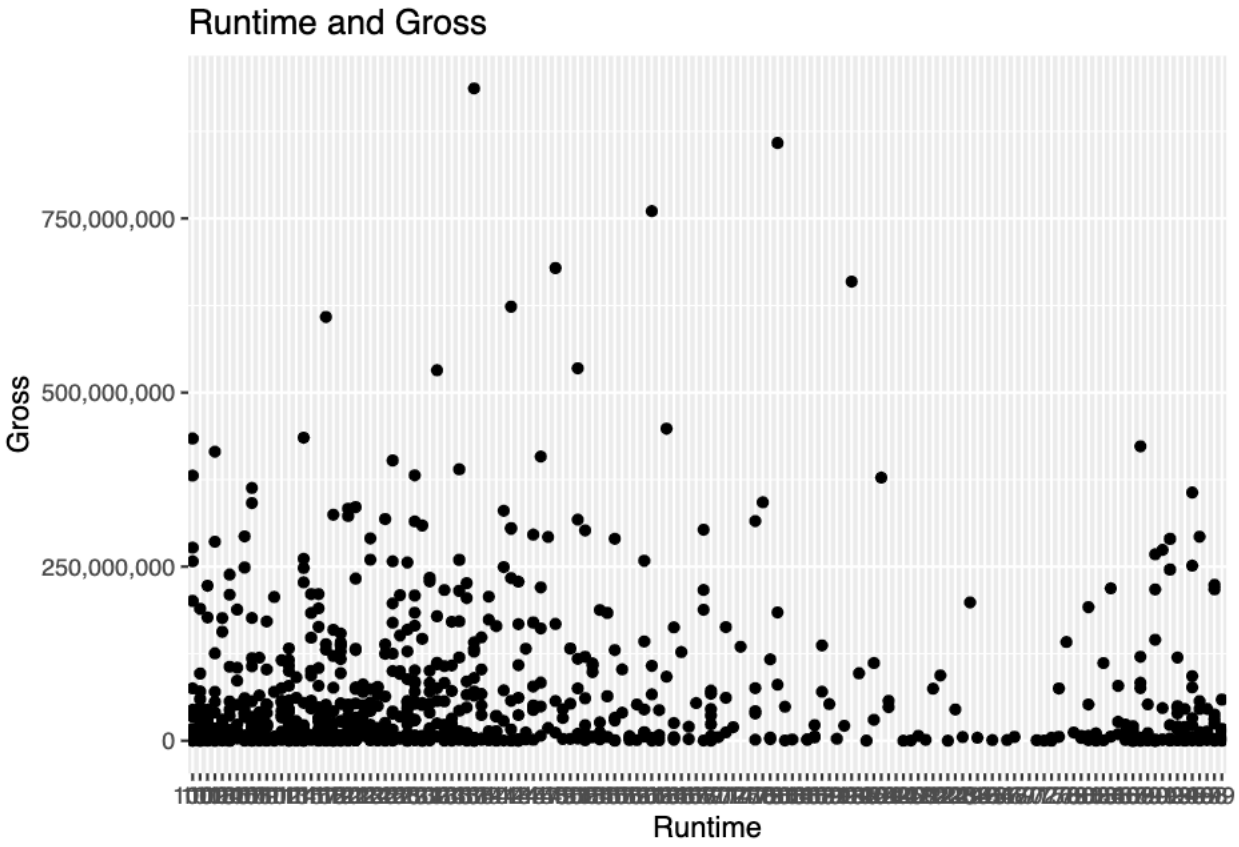
```
library(ggcorrplot)

#this does not include Runtime as it needs to be changed to a numeric number.
movies_corr <- na.omit(movies)
corr_matrix <- cor(movies_corr[,c('Gross','IMDB_Rating','Meta_score','No_of_Votes')])
ggcorrplot(corr_matrix,hc.order = TRUE,lab=TRUE) + ggtitle("Correlation between important predictors")
```




```
#need help cleaning up our x to make it readable.
ggplot(movies,aes(x=Runtime,y=Gross)) +
  geom_point(binwidth=1) +
  scale_y_continuous(labels = comma) +
  geom_smooth() +
  labs(title='Runtime and Gross',x='Runtime',y='Gross')
```

```
## Warning: Ignoring unknown parameters: binwidth
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## Warning: Removed 169 rows containing non-finite values (stat_smooth).
## Warning: Removed 169 rows containing missing values (geom_point).
```


Runtime and Gross
750,000,000
500,000,000
250,000,000
0
Gross
Runtime