IMDB

2023-12-02

```
myData <- read.csv("movie_metadata_original.csv")
movie_data <- read.csv("movie_metadata_original.csv")
head(myData)
```

```
##       director_name                                     movie_title
## 1     James Cameron                                      Avatar\xa0
## 2    Gore Verbinski Pirates of the Caribbean: At World's End\xa0
## 3        Sam Mendes                                     Spectre\xa0
## 4 Christopher Nolan                    The Dark Knight Rises\xa0
## 5    Andrew Stanton                                  John Carter\xa0
## 6         Sam Raimi                                 Spider-Man 3\xa0
##   num_critic_for_reviews duration director_facebook_likes     actor_2_name
## 1                    723      178                      0 Joel David Moore
## 2                    302      169                    563    Orlando Bloom
## 3                    602      148                      0     Rory Kinnear
## 4                    813      164                  22000   Christian Bale
## 5                    462      132                    475  Samantha Morton
## 6                    392      156                      0     James Franco
##   actor_1_facebook_likes     gross                           genres
## 1                   1000 760505847 Action|Adventure|Fantasy|Sci-Fi
## 2                  40000 309404152        Action|Adventure|Fantasy
## 3                  11000 200074175       Action|Adventure|Thriller
## 4                  27000 448130642                 Action|Thriller
## 5                    640  73058679         Action|Adventure|Sci-Fi
## 6                  24000 336530303        Action|Adventure|Romance
##      actor_1_name num_voted_users cast_total_facebook_likes
## 1     CCH Pounder          886204                      4834
## 2     Johnny Depp          471220                     48350
## 3 Christoph Waltz          275868                     11700
## 4       Tom Hardy         1144337                    106759
## 5    Daryl Sabara          212204                      1873
## 6    J.K. Simmons          383056                     46055
##   facenumber_in_poster
## 1                    0
## 2                    0
## 3                    1
## 4                    0
## 5                    1
## 6                    0
##                                                 plot_keywords
## 1                          avatar|future|marine|native|paraplegic
## 2     goddess|marriage ceremony|marriage proposal|pirate|singapore
## 3                            bomb|espionage|sequel|spy|terrorist
## 4 deception|imprisonment|lawlessness|police officer|terrorist plot
```

```
## 5               alien|american civil war|male nipple|mars|princess
## 6                        sandman|spider man|symbiote|venom|villain
##   num_user_for_reviews language country    budget title_year
## 1                 3054  English     USA 237000000       2009
## 2                 1238  English     USA 300000000       2007
## 3                  994  English      UK 245000000       2015
## 4                 2701  English     USA 250000000       2012
## 5                  738  English     USA 263700000       2012
## 6                 1902  English     USA 258000000       2007
##   actor_2_facebook_likes actors_facebook_likes    profits imdb_score
## 1                    936                 1936  523505847        7.9
## 2                   5000                45000    9404152        7.1
## 3                    393                11393  -44925825        6.8
## 4                  23000                50000  198130642        8.5
## 5                    632                 1272 -190641321        6.6
## 6                  11000                35000   78530303        6.2
##   movie_facebook_likes
## 1                33000
## 2                    0
## 3                85000
## 4               164000
## 5                24000
## 6                    0
```

```
sum(is.na(myData))
```

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

```
nrow(myData)
```

```
## [1] 3853
```

```
library(ggplot2)
```

#Summary Statistics

```
summary(movie_data)
```

```
##  director_name      movie_title        num_critic_for_reviews    duration
##  Length:3853        Length:3853        Min.   :  1.0          Min.   : 34
##  Class :character   Class :character   1st Qu.: 74.0          1st Qu.: 95
##  Mode  :character   Mode  :character   Median :136.0          Median :106
##                                        Mean   :164.5          Mean   :110
##                                        3rd Qu.:222.0          3rd Qu.:120
##                                        Max.   :813.0          Max.   :330
##  director_facebook_likes actor_2_name       actor_1_facebook_likes
##  Min.   :    0.0         Length:3853        Min.   :     0
##  1st Qu.:   10.0         Class :character   1st Qu.:   726
##  Median :   59.0         Mode  :character   Median :  1000
##  Mean   :  788.6                            Mean   :  7646
##  3rd Qu.:  230.0                            3rd Qu.: 12000
```

```
##  Max.   :23000.0                            Max.   :640000
##      gross              genres          actor_1_name       num_voted_users
##  Min.   :      162   Length:3853        Length:3853        Min.   :     40
##  1st Qu.:  7221458   Class :character   Class :character   1st Qu.:  17983
##  Median : 28734552   Mode  :character   Mode  :character   Median :  52055
##  Mean   : 51545582                                         Mean   : 103566
##  3rd Qu.: 66002004                                         3rd Qu.: 125109
##  Max.   :760505847                                         Max.   :1689764
##  cast_total_facebook_likes facenumber_in_poster plot_keywords
##  Min.   :     0            Min.   : 0.000       Length:3853
##  1st Qu.:  1852            1st Qu.: 0.000       Class :character
##  Median :  3944            Median : 1.000       Mode  :character
##  Mean   : 11359            Mean   : 1.378
##  3rd Qu.: 16098            3rd Qu.: 2.000
##  Max.   :656730            Max.   :43.000
##  num_user_for_reviews   language           country              budget
##  Min.   :   2.0       Length:3853        Length:3853        Min.   :2.180e+02
##  1st Qu.: 104.0       Class :character   Class :character   1st Qu.:1.000e+07
##  Median : 205.0       Mode  :character   Mode  :character   Median :2.500e+07
##  Mean   : 330.3                                            Mean   :4.559e+07
##  3rd Qu.: 394.0                                            3rd Qu.:5.000e+07
##  Max.   :5060.0                                            Max.   :1.222e+10
##    title_year   actor_2_facebook_likes actors_facebook_likes
##  Min.   :1927   Min.   :     0         Min.   :     0
##  1st Qu.:1999   1st Qu.:   370         1st Qu.:  1164
##  Median :2005   Median :   670         Median :  1928
##  Mean   :2003   Mean   :  1985         Mean   :  9631
##  3rd Qu.:2010   3rd Qu.:   973         3rd Qu.: 14230
##  Max.   :2016   Max.   :137000         Max.   :648000
##     profits             imdb_score    movie_facebook_likes
##  Min.   :-1.221e+10   Min.   :1.600   Min.   :     0
##  1st Qu.:-1.036e+07   1st Qu.:5.900   1st Qu.:     0
##  Median : 1.093e+06   Median :6.600   Median :   212
##  Mean   : 5.956e+06   Mean   :6.466   Mean   :  9210
##  3rd Qu.: 2.506e+07   3rd Qu.:7.200   3rd Qu.: 11000
##  Max.   : 5.235e+08   Max.   :9.300   Max.   :349000
```

#Data Structure

```
str(movie_data)
```

```
## 'data.frame':    3853 obs. of  24 variables:
##  $ director_name            : chr  "James Cameron" "Gore Verbinski" "Sam Mendes" "Christopher Nolan"
##  $ movie_title              : chr  "Avatar\xa0" "Pirates of the Caribbean: At World's End\xa0" "Spec
##  $ num_critic_for_reviews   : int  723 302 602 813 462 392 324 635 375 673 ...
##  $ duration                 : int  178 169 148 164 132 156 100 141 153 183 ...
##  $ director_facebook_likes  : int  0 563 0 22000 475 0 15 0 282 0 ...
##  $ actor_2_name             : chr  "Joel David Moore" "Orlando Bloom" "Rory Kinnear" "Christian Bale
##  $ actor_1_facebook_likes   : int  1000 40000 11000 27000 640 24000 799 26000 25000 15000 ...
##  $ gross                    : int  760505847 309404152 200074175 448130642 73058679 336530303 200807
##  $ genres                   : chr  "Action|Adventure|Fantasy|Sci-Fi" "Action|Adventure|Fantasy" "Acti
##  $ actor_1_name             : chr  "CCH Pounder" "Johnny Depp" "Christoph Waltz" "Tom Hardy" ...
##  $ num_voted_users          : int  886204 471220 275868 1144337 212204 383056 294810 462669 321795 3
```

```
##  $ cast_total_facebook_likes: int  4834 48350 11700 106759 1873 46055 2036 92000 58753 24450 ...
##  $ facenumber_in_poster     : int  0 0 1 0 1 0 1 4 3 0 ...
##  $ plot_keywords            : chr  "avatar|future|marine|native|paraplegic" "goddess|marriage ceremo
##  $ num_user_for_reviews     : int  3054 1238 994 2701 738 1902 387 1117 973 3018 ...
##  $ language                 : chr  "English" "English" "English" "English" ...
##  $ country                  : chr  "USA" "USA" "UK" "USA" ...
##  $ budget                   : num  2.37e+08 3.00e+08 2.45e+08 2.50e+08 2.64e+08 ...
##  $ title_year               : int  2009 2007 2015 2012 2012 2007 2010 2015 2009 2016 ...
##  $ actor_2_facebook_likes   : int  936 5000 393 23000 632 11000 553 21000 11000 4000 ...
##  $ actors_facebook_likes    : int  1936 45000 11393 50000 1272 35000 1352 47000 36000 19000 ...
##  $ profits                  : num  5.24e+08 9.40e+06 -4.49e+07 1.98e+08 -1.91e+08 ...
##  $ imdb_score               : num  7.9 7.1 6.8 8.5 6.6 6.2 7.8 7.5 7.5 6.9 ...
##  $ movie_facebook_likes     : int  33000 0 85000 164000 24000 0 29000 118000 10000 197000 ...
```

#Visualization

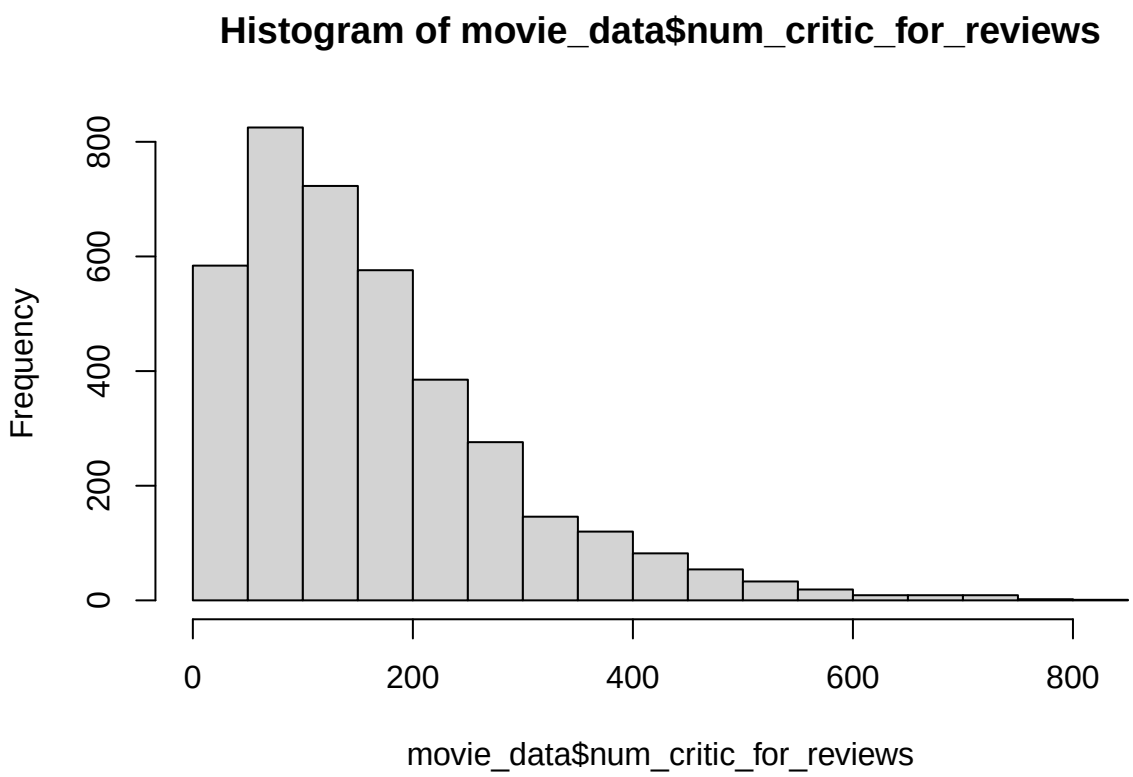
#Histograms

#Histogram num_critic_for_reviews #Saving Histogram to a PNG file

```
png("num_critic_for_reviews",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$num_critic_for_reviews)
```

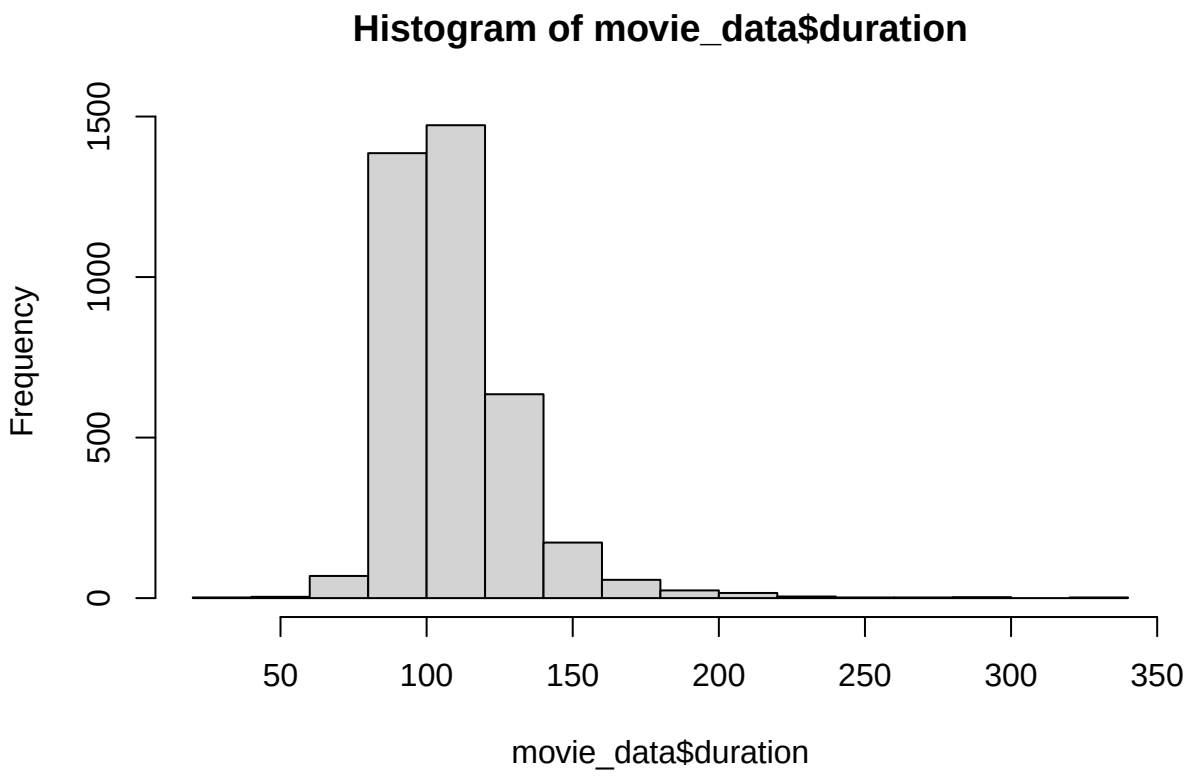


#Histogram duration #Saving Histogram to a PNG file

```
png("duration ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$duration)
```

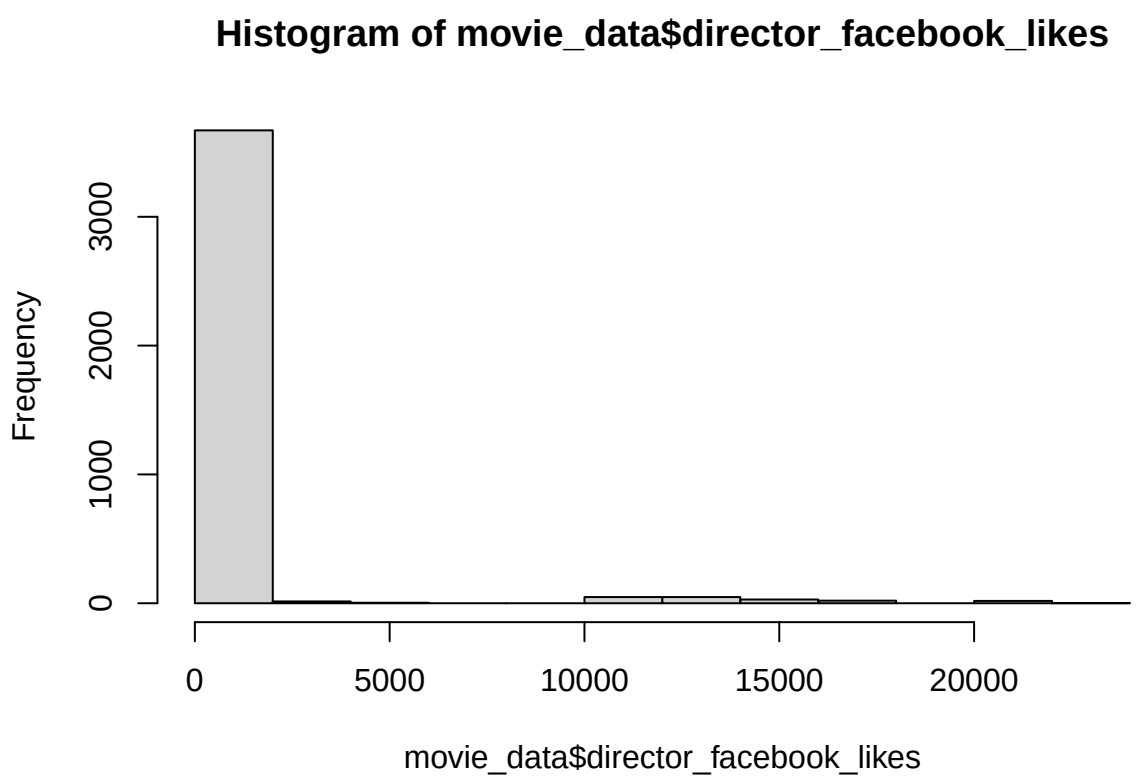


#Histogram director_facebook_likes #Saving Histogram to a PNG file

```
png("director_facebook_likes ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$director_facebook_likes)
```



#create histogram using ggplot2

```
histogram<-ggplot(movie_data,aes(x=duration))+
  geom_histogram(binwidth = 10,fill="white",color="black")+
  labs(x="Duration in Minutes", y="Frequency",title="Histogram of Movie Duration")
```

#Save the histogram as a PNG file

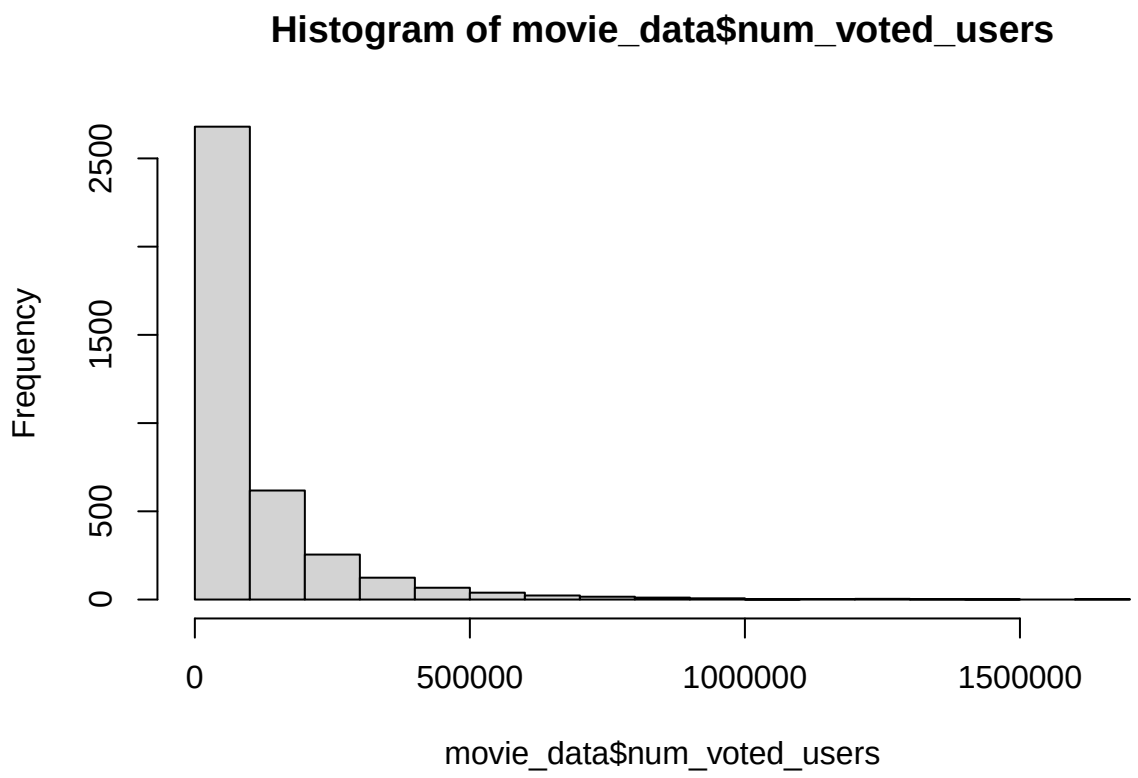
```
ggsave("durtion_histogram_ggplot.png",plot=histogram, width=8,height = 6, units = "in")
```

#Histogram num_voted_users #Saving Histogram to a PNG file

```
png("num_voted_users ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$num_voted_users)
```

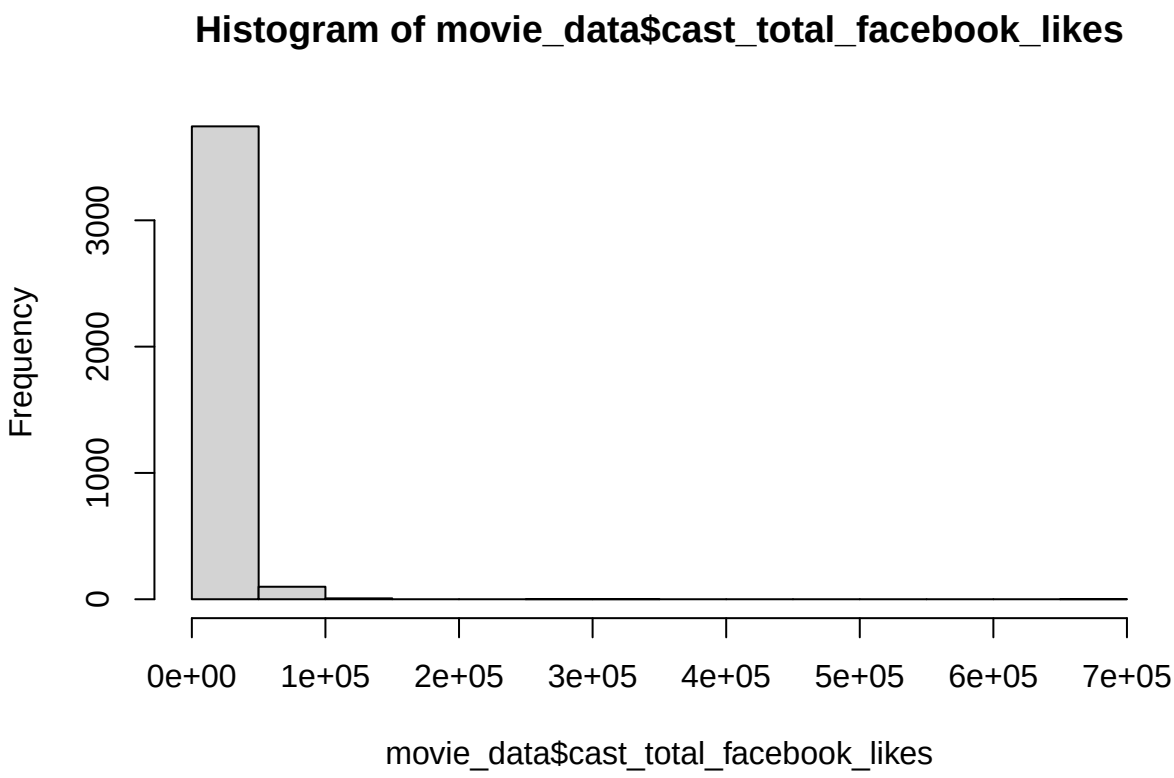


#Histogram cast_total_facebook_likes #Saving Histogram to a PNG file

```
png("cast_total_facebook_likes ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$cast_total_facebook_likes)
```

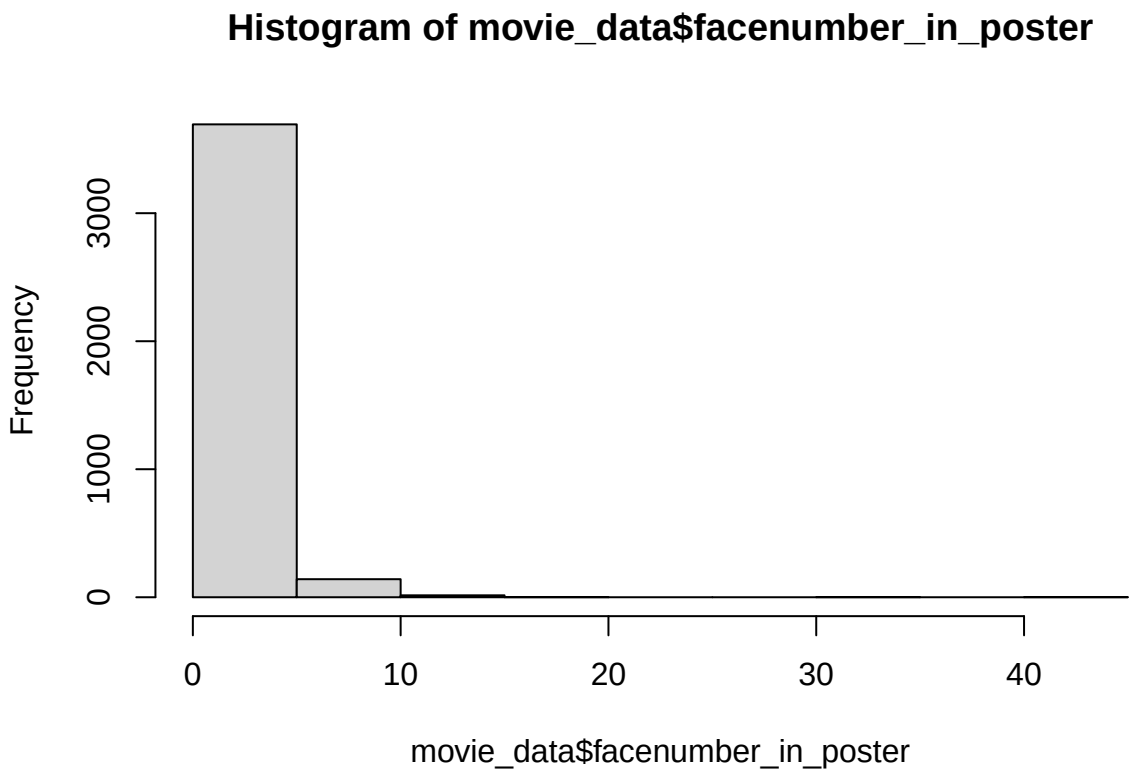


#Histogram facenumber_in_poster #Saving Histogram to a PNG file

```
png("facenumber_in_poster ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$facenumber_in_poster)
```

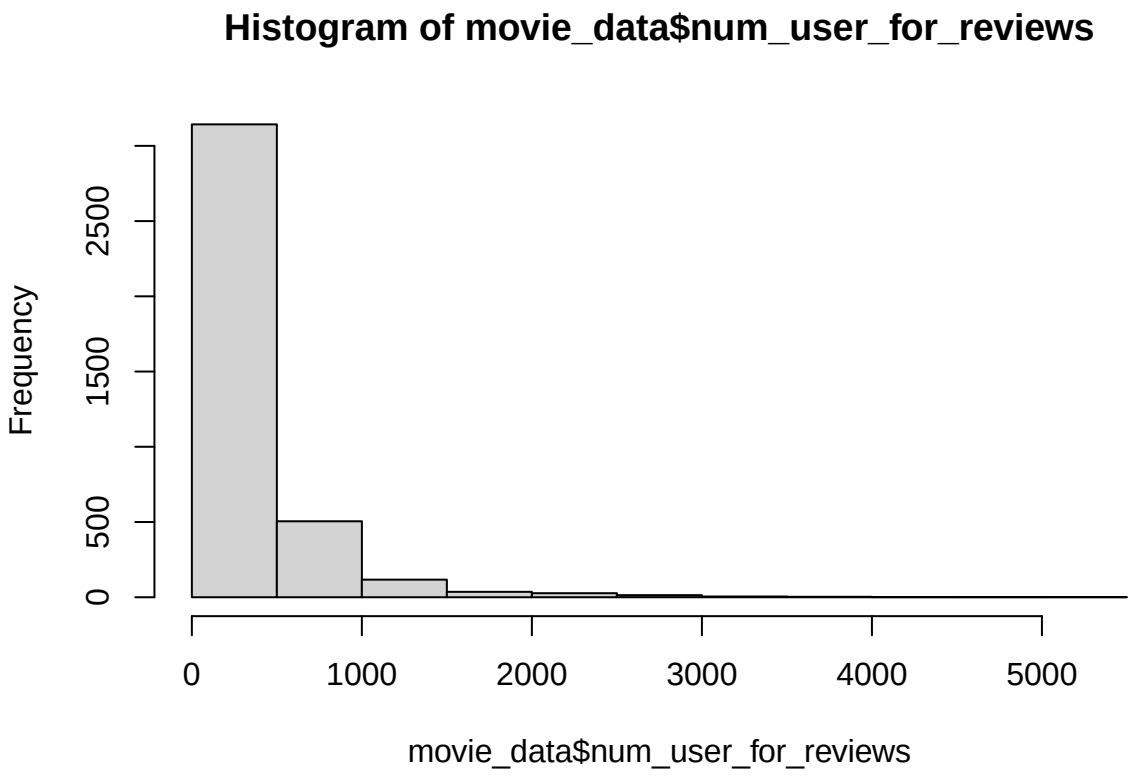


#Histogram num_user_for_reviews #Saving Histogram to a PNG file

```
png("num_user_for_reviews ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$num_user_for_reviews)
```

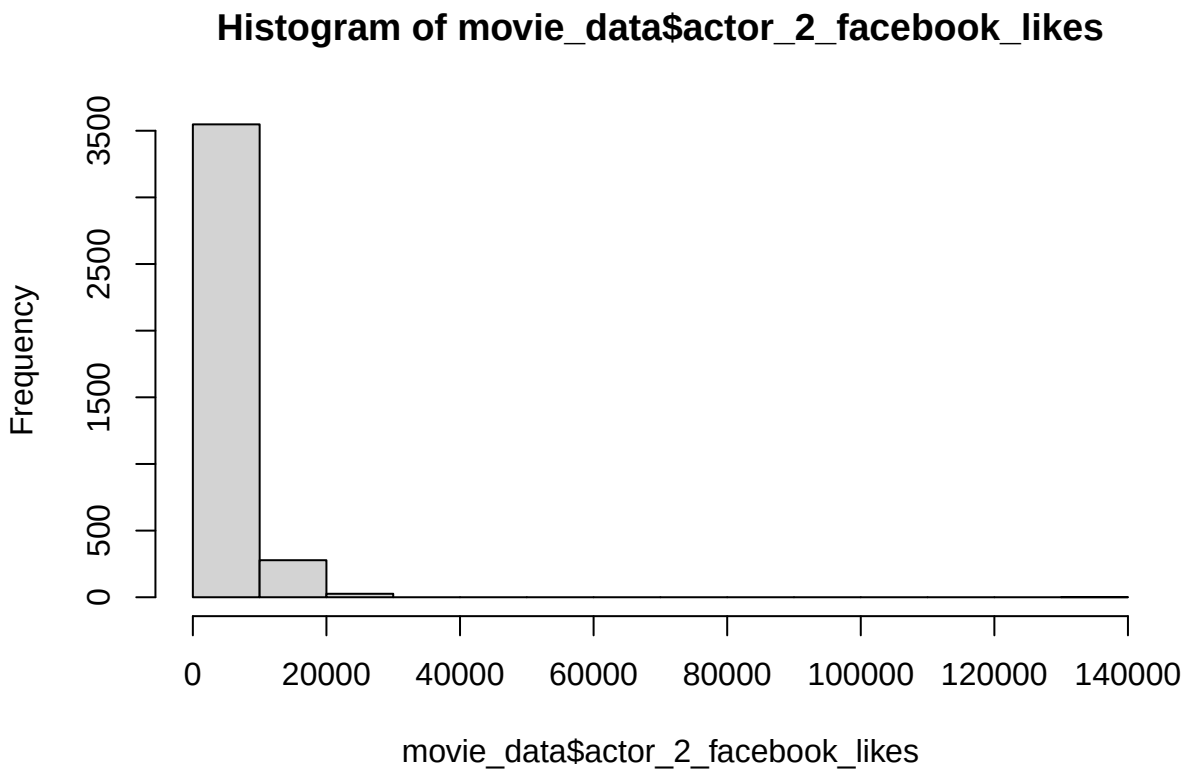


#Histogram actor_2_facebook_likes #Saving Histogram to a PNG file

```
png("actor_2_facebook_likes ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$actor_2_facebook_likes)
```



#Histogram imdb_score #Saving Histogram to a PNG file

```
png("imdb_score ",width = 800,height = 600)
```

#Create Histogram

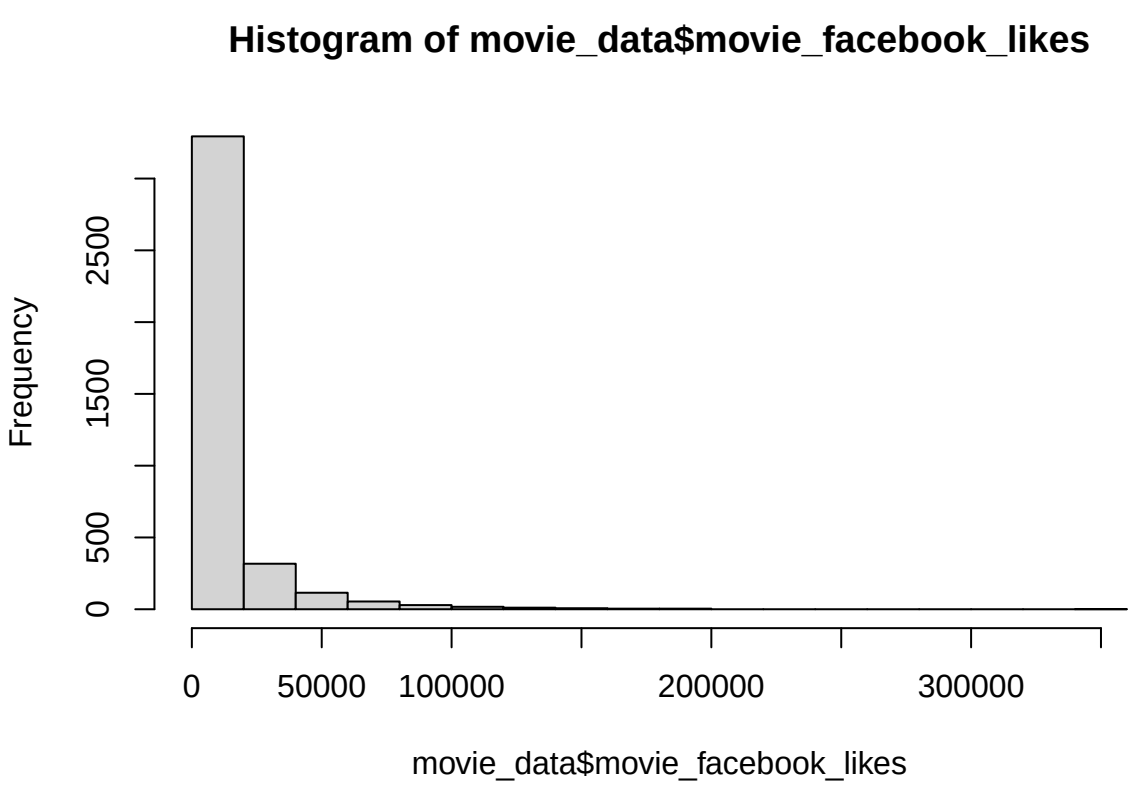
```
hist(movie_data$imdb_score)
```

#Histogram movie_facebook_likes #Saving Histogram to a PNG file

```
png("movie_facebook_likes ",width = 800,height = 600)
```

#Create Histogram

```
hist(movie_data$movie_facebook_likes)
```




```
numeric_cols <- sapply(myData, is.numeric)

detect_outliers <- function(x, na.rm = TRUE) {
  qnt <- quantile(x, probs=c(0.25,0.75), na.rm = na.rm)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  return(x < (qnt[1] - H) | x > (qnt[2] + H))
}

outliers_logical <- apply(myData[, numeric_cols], 2, detect_outliers)

rows_with_outliers <- apply(outliers_logical, 1, any)

myData_filtered <- myData[!rows_with_outliers, ]
```

```
nrow(myData_filtered)
```

```
## [1] 2042
```

```
set.seed(2)
sampled_data <- myData_filtered[sample(nrow(myData), 2101), ]
```

```
  sampled_data <- na.omit(sampled_data)
```

```r
str(sampled_data)
```

```
## 'data.frame':    1117 obs. of  24 variables:
##  $ director_name            : chr  "Jonathan Glazer" "Alex Proyas" "Bobby Farrelly" "John Moore" ...
##  $ movie_title              : chr  "Birth\xa0" "Dark City\xa0" "Fever Pitch\xa0" "Behind Enemy Lines"
##  $ num_critic_for_reviews   : int  167 222 124 131 143 191 8 44 189 86 ...
##  $ duration                 : int  100 111 104 106 90 96 81 139 80 130 ...
##  $ director_facebook_likes  : int  143 295 101 212 241 47 0 105 129 39 ...
##  $ actor_2_name             : chr  "Anne Heche" "William Hurt" "KaDee Strickland" "David Keith" ...
##  $ actor_1_facebook_likes   : int  829 3000 787 578 1000 3000 119 14000 375 2000 ...
##  $ gross                    : int  5005883 14337579 42071069 59068786 81517441 12570442 18195 316000
##  $ genres                   : chr  "Drama|Mystery|Romance|Thriller" "Action|Drama|Fantasy|Mystery|Sc
##  $ actor_1_name             : chr  "Cameron Bright" "Rufus Sewell" "Jimmy Fallon" "Joaquim de Almeid
##  $ num_voted_users          : int  29649 156929 36223 86902 111368 40514 336 34427 12241 16673 ...
##  $ cast_total_facebook_likes: int  2295 4696 1827 2544 3148 4053 154 16125 460 5162 ...
##  $ facenumber_in_poster     : int  1 1 1 2 1 0 0 1 1 0 ...
##  $ plot_keywords            : chr  "birthday|boy|dead husband|death|widow" "memory|murder|neo noir|r
##  $ num_user_for_reviews     : int  361 624 208 411 394 234 7 126 112 45 ...
##  $ language                 : chr  "English" "English" "English" "English" ...
##  $ country                  : chr  "USA" "Australia" "USA" "USA" ...
##  $ budget                   : num  2.0e+07 2.7e+07 3.0e+07 4.0e+07 5.0e+07 1.0e+07 2.5e+05 2.8e+07 2
##  $ title_year               : int  2004 1998 2005 2001 1999 2005 1997 1995 2008 2008 ...
##  $ actor_2_facebook_likes   : int  681 882 299 563 495 497 24 1000 44 898 ...
##  $ actors_facebook_likes    : int  1510 3882 1086 1141 1495 3497 143 15000 419 2898 ...
##  $ profits                  : num  -14994117 -12662421 12071069 19068786 31517441 ...
##  $ imdb_score               : num  6.1 7.7 6.2 6.4 7.1 6.8 6.3 6.9 7.1 7.3 ...
##  $ movie_facebook_likes     : int  0 14000 0 0 0 808 67 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:984] 3 9 10 11 13 15 16 17 20 24 ...
##    ..- attr(*, "names")= chr [1:984] "NA" "NA.1" "NA.2" "NA.3" ...
```

```r
# Install
#install.packages("tm")
#install.packages("wordcloud")
#install.packages("SnowballC") # for text stemming
#install.packages("RColorBrewer") # color palettes
library(tm)
```

```
## Warning: package 'tm' was built under R version 4.3.2
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```r
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.3.2
```

```
## Loading required package: RColorBrewer
```

```
library("SnowballC")
library("RColorBrewer")
```

Regression preparation

```
filtered_sampled_data <- subset(sampled_data, language == "English")
filtered_sampled_data <- subset(filtered_sampled_data, country == "USA")
View(filtered_sampled_data)
```

```
column_names <- colnames(filtered_sampled_data)
print(column_names)
```

```
##  [1] "director_name"             "movie_title"
##  [3] "num_critic_for_reviews"    "duration"
##  [5] "director_facebook_likes"   "actor_2_name"
##  [7] "actor_1_facebook_likes"    "gross"
##  [9] "genres"                    "actor_1_name"
## [11] "num_voted_users"           "cast_total_facebook_likes"
## [13] "facenumber_in_poster"      "plot_keywords"
## [15] "num_user_for_reviews"      "language"
## [17] "country"                   "budget"
## [19] "title_year"                "actor_2_facebook_likes"
## [21] "actors_facebook_likes"     "profits"
## [23] "imdb_score"                "movie_facebook_likes"
```

```
sampled_data_regression <- filtered_sampled_data[, !(colnames(filtered_sampled_data) %in% c('director_na
```

Linear Regression

```
Model1 <- lm(imdb_score ~ . , data =  sampled_data_regression)
summary(Model1)
```

```
##
## Call:
## lm(formula = imdb_score ~ ., data = sampled_data_regression)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99949 -0.47389  0.03986  0.48345  2.40637
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.915e+01  8.951e+00   3.256  0.00117 **
## num_critic_for_reviews      5.383e-04  5.310e-04   1.014  0.31098
```

```
## duration                   1.657e-02  1.841e-03   8.999  < 2e-16 ***
## director_facebook_likes    3.141e-04  2.072e-04   1.516  0.12988
## gross                     -2.539e-09  1.173e-09  -2.165  0.03070 *
## num_voted_users            8.871e-06  8.640e-07  10.268  < 2e-16 ***
## cast_total_facebook_likes -1.321e-04  2.669e-05  -4.949 8.96e-07 ***
## facenumber_in_poster      -3.314e-02  1.867e-02  -1.775  0.07623 .
## num_user_for_reviews      -6.988e-04  2.366e-04  -2.953  0.00324 **
## budget                    -1.132e-08  1.402e-09  -8.072 2.34e-15 ***
## title_year                -1.224e-02  4.472e-03  -2.737  0.00633 **
## actors_facebook_likes      1.387e-04  2.767e-05   5.015 6.45e-07 ***
## profits                           NA         NA      NA       NA
## movie_facebook_likes       1.894e-06  5.086e-06   0.372  0.70967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7087 on 857 degrees of freedom
## Multiple R-squared:  0.3007, Adjusted R-squared:  0.2909
## F-statistic: 30.71 on 12 and 857 DF,  p-value: < 2.2e-16
```

```
Model2 <- lm(imdb_score ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenu
summary(Model2)
```

```
##
## Call:
## lm(formula = imdb_score ~ num_critic_for_reviews + num_voted_users +
##     cast_total_facebook_likes + facenumber_in_poster + budget +
##     title_year + actors_facebook_likes + profits + movie_facebook_likes,
##     data = sampled_data_regression)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02330 -0.48526  0.03432  0.52127  2.30737
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.297e+01  8.927e+00   3.693 0.000236 ***
## num_critic_for_reviews     2.266e-04  4.990e-04   0.454 0.649869
## num_voted_users            7.975e-06  8.205e-07   9.720  < 2e-16 ***
## cast_total_facebook_likes -1.367e-04  2.788e-05  -4.902 1.14e-06 ***
## facenumber_in_poster      -2.039e-02  1.947e-02  -1.047 0.295479
## budget                    -1.186e-08  1.294e-09  -9.167  < 2e-16 ***
## title_year                -1.336e-02  4.468e-03  -2.990 0.002873 **
## actors_facebook_likes      1.490e-04  2.889e-05   5.157 3.11e-07 ***
## profits                   -2.851e-09  1.231e-09  -2.315 0.020824 *
## movie_facebook_likes       5.180e-06  5.239e-06   0.989 0.323056
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7443 on 860 degrees of freedom
## Multiple R-squared:  0.226,  Adjusted R-squared:  0.2179
## F-statistic:  27.9 on 9 and 860 DF,  p-value: < 2.2e-16
```

```r
Model5 <- lm(imdb_score ~ num_voted_users + cast_total_facebook_likes + facenumber_in_poster + title_yea
summary(Model5)
```

```
##
## Call:
## lm(formula = imdb_score ~ num_voted_users + cast_total_facebook_likes +
##     facenumber_in_poster + title_year + actors_facebook_likes +
##     profits + movie_facebook_likes, data = sampled_data_regression)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05121 -0.52987  0.06939  0.56833  2.43716
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.996e+01  8.184e+00   3.661 0.000267 ***
## num_voted_users             5.133e-06  6.506e-07   7.889 9.20e-15 ***
## cast_total_facebook_likes -1.686e-04  2.892e-05  -5.830 7.82e-09 ***
## facenumber_in_poster      -1.858e-02  2.027e-02  -0.917 0.359550
## title_year                -1.192e-02  4.088e-03  -2.917 0.003629 **
## actors_facebook_likes      1.783e-04  3.000e-05   5.943 4.06e-09 ***
## profits                    9.680e-10  1.207e-09   0.802 0.422764
## movie_facebook_likes       9.226e-06  5.331e-06   1.731 0.083856 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.779 on 862 degrees of freedom
## Multiple R-squared:  0.1501, Adjusted R-squared:  0.1432
## F-statistic: 21.75 on 7 and 862 DF,  p-value: < 2.2e-16
```

```r
Model3 <- lm(gross ~ . , data =  sampled_data_regression)
summary(Model3)
```

```
## Warning in summary.lm(Model3): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = gross ~ ., data = sampled_data_regression)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.363e-07 -5.580e-09 -2.300e-10  4.460e-09  7.985e-07
##
## Coefficients:
##                              Estimate Std. Error    t value Pr(>|t|)
## (Intercept)                -5.148e-07  4.851e-07 -1.061e+00 0.288873
## num_critic_for_reviews      9.582e-11  2.861e-11  3.349e+00 0.000848 ***
## duration                    3.390e-10  1.037e-10  3.268e+00 0.001127 **
## director_facebook_likes     8.195e-12  1.117e-11  7.330e-01 0.463532
## num_voted_users            -8.407e-13  4.931e-14 -1.705e+01  < 2e-16 ***
## cast_total_facebook_likes -2.400e-12  1.458e-12 -1.646e+00 0.100052
```

```
## facenumber_in_poster      -3.812e-11  1.007e-09 -3.800e-02 0.969825
## num_user_for_reviews       1.518e-12  1.281e-11  1.190e-01 0.905698
## budget                     1.000e+00  7.211e-17  1.387e+16  < 2e-16 ***
## title_year                 2.136e-10  2.419e-10  8.830e-01 0.377430
## actors_facebook_likes      2.313e-12  1.512e-12  1.530e+00 0.126406
## profits                    1.000e+00  6.334e-17  1.579e+16  < 2e-16 ***
## imdb_score                 3.997e-09  1.840e-09  2.173e+00 0.030076 *
## movie_facebook_likes      -5.494e-14  2.739e-13 -2.010e-01 0.841080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.817e-08 on 856 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.413e+31 on 13 and 856 DF,  p-value: < 2.2e-16
```

```
Model4 <- lm(gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenumber_
summary(Model4)
```

```
## Warning in summary.lm(Model4): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = gross ~ num_critic_for_reviews + num_voted_users +
##     cast_total_facebook_likes + facenumber_in_poster + budget +
##     title_year + actors_facebook_likes + profits + movie_facebook_likes,
##     data = sampled_data_regression)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -7.714e-07 -3.040e-09 -1.100e-10  2.790e-09  6.358e-07
##
## Coefficients:
##                              Estimate Std. Error    t value Pr(>|t|)
## (Intercept)                -2.457e-07  4.282e-07 -5.740e-01   0.5664
## num_critic_for_reviews     -5.132e-11  2.394e-11 -2.144e+00   0.0323 *
## num_voted_users             8.794e-14  3.936e-14  2.234e+00   0.0257 *
## cast_total_facebook_likes -3.917e-13  1.338e-12 -2.930e-01   0.7697
## facenumber_in_poster       -4.307e-10  9.342e-10 -4.610e-01   0.6449
## budget                      1.000e+00  6.208e-17  1.611e+16   <2e-16 ***
## title_year                  1.253e-10  2.143e-10  5.850e-01   0.5589
## actors_facebook_likes       1.436e-12  1.386e-12  1.036e+00   0.3003
## profits                     1.000e+00  5.907e-17  1.693e+16   <2e-16 ***
## movie_facebook_likes        1.069e-13  2.513e-13  4.250e-01   0.6708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.57e-08 on 860 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.285e+31 on 9 and 860 DF,  p-value: < 2.2e-16
```

```r
Model6 <- lm(gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenumber_
summary(Model6)
```

```
##
## Call:
## lm(formula = gross ~ num_critic_for_reviews + num_voted_users +
##     cast_total_facebook_likes + facenumber_in_poster + title_year +
##     actors_facebook_likes + movie_facebook_likes, data = sampled_data_regression)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -82145465 -14087362  -7192717  10757746  98919592
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.383e+08  2.941e+08   0.810 0.417986
## num_critic_for_reviews    -2.653e+04  1.640e+04  -1.618 0.105977
## num_voted_users            4.224e+02  2.289e+01  18.455  < 2e-16 ***
## cast_total_facebook_likes  3.125e+03  9.119e+02   3.427 0.000639 ***
## facenumber_in_poster      -5.693e+05  6.416e+05  -0.887 0.375167
## title_year                -1.124e+05  1.472e+05  -0.764 0.445276
## actors_facebook_likes     -3.239e+03  9.455e+02  -3.425 0.000643 ***
## movie_facebook_likes      -4.583e+02  1.720e+02  -2.665 0.007850 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24530000 on 862 degrees of freedom
## Multiple R-squared:  0.3792, Adjusted R-squared:  0.3742
## F-statistic: 75.23 on 7 and 862 DF,  p-value: < 2.2e-16
```

```r
#install.packages("lmtest")
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.3.2
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
lr_test <- lrtest(Model1, Model2, Model3, Model4, Model5, Model6)
print(lr_test)
```

```
## Likelihood ratio test
##
## Model 1: imdb_score ~ num_critic_for_reviews + duration + director_facebook_likes +
##     gross + num_voted_users + cast_total_facebook_likes + facenumber_in_poster +
```

```
##     num_user_for_reviews + budget + title_year + actors_facebook_likes +
##     profits + movie_facebook_likes
## Model 2: imdb_score ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes +
##     facenumber_in_poster + budget + title_year + actors_facebook_likes +
##     profits + movie_facebook_likes
## Model 3: gross ~ num_critic_for_reviews + duration + director_facebook_likes +
##     num_voted_users + cast_total_facebook_likes + facenumber_in_poster +
##     num_user_for_reviews + budget + title_year + actors_facebook_likes +
##     profits + imdb_score + movie_facebook_likes
## Model 4: gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes +
##     facenumber_in_poster + budget + title_year + actors_facebook_likes +
##     profits + movie_facebook_likes
## Model 5: imdb_score ~ num_voted_users + cast_total_facebook_likes + facenumber_in_poster +
##     title_year + actors_facebook_likes + profits + movie_facebook_likes
## Model 6: gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes +
##     facenumber_in_poster + title_year + actors_facebook_likes +
##     movie_facebook_likes
##   #Df   LogLik Df     Chisq Pr(>Chisq)
## 1  14   -928.3
## 2  11   -972.5 -3    88.339  < 2.2e-16 ***
## 3  15  13633.3  4 29211.579  < 2.2e-16 ***
## 4  11  13689.3 -4   112.053  < 2.2e-16 ***
## 5   9  -1013.2 -2 29405.007  < 2.2e-16 ***
## 6   9 -16034.0  0 30041.603  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
nrow(sampled_data_regression)
```

```
## [1] 870
```

```
TData <- sampled_data_regression[1:515,]
VData <- sampled_data_regression[516:858,]

Model11 <- lm(imdb_score ~ . , data =  TData)
Model22 <- lm(imdb_score ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenu
Model55 <- lm(imdb_score ~ num_voted_users + cast_total_facebook_likes + facenumber_in_poster + title_y

Model33 <- lm(gross ~ . , data =  TData)
Model44 <- lm(gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenumber
Model66 <- lm(gross ~ num_critic_for_reviews + num_voted_users + cast_total_facebook_likes + facenumber

Pred1 <- predict(Model11, VData)
Pred2 <- predict(Model22, VData)
Pred5 <- predict(Model55, VData)

print(sqrt(mean(VData$imdb_score-Pred1)^2))
```

```
## [1] 0.1160082
```

```
print(sqrt(mean(VData$imdb_score-Pred2)^2))
```

```
## [1] 0.1359774
```

```r
print(sqrt(mean(VData$imdb_score-Pred5)^2))
```

```
## [1] 0.1528253
```

```r
Pred3 <- predict(Model11, VData)
Pred4 <- predict(Model22, VData)
Pred6 <- predict(Model55, VData)

print(sqrt(mean(VData$imdb_score-Pred3)^2))
```

```
## [1] 0.1160082
```

```r
print(sqrt(mean(VData$imdb_score-Pred4)^2))
```

```
## [1] 0.1359774
```

```r
print(sqrt(mean(VData$imdb_score-Pred6)^2))
```

```
## [1] 0.1528253
```

Linear Regression

```r
Model <- lm(imdb_score ~ . , data =  sampled_data_regression)
summary(Model)
```

```
##
## Call:
## lm(formula = imdb_score ~ ., data = sampled_data_regression)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.99949 -0.47389  0.03986  0.48345  2.40637
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.915e+01  8.951e+00   3.256  0.00117 **
## num_critic_for_reviews      5.383e-04  5.310e-04   1.014  0.31098
## duration                    1.657e-02  1.841e-03   8.999  < 2e-16 ***
## director_facebook_likes     3.141e-04  2.072e-04   1.516  0.12988
## gross                      -2.539e-09  1.173e-09  -2.165  0.03070 *
## num_voted_users             8.871e-06  8.640e-07  10.268  < 2e-16 ***
## cast_total_facebook_likes -1.321e-04  2.669e-05  -4.949 8.96e-07 ***
## facenumber_in_poster       -3.314e-02  1.867e-02  -1.775  0.07623 .
## num_user_for_reviews       -6.988e-04  2.366e-04  -2.953  0.00324 **
## budget                     -1.132e-08  1.402e-09  -8.072 2.34e-15 ***
## title_year                 -1.224e-02  4.472e-03  -2.737  0.00633 **
## actors_facebook_likes       1.387e-04  2.767e-05   5.015 6.45e-07 ***
```

```
## profits                           NA         NA      NA       NA
## movie_facebook_likes       1.894e-06  5.086e-06   0.372  0.70967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7087 on 857 degrees of freedom
## Multiple R-squared:  0.3007, Adjusted R-squared:  0.2909
## F-statistic: 30.71 on 12 and 857 DF,  p-value: < 2.2e-16
```

```
Model <- lm(gross ~ . , data =  sampled_data_regression)
summary(Model)
```

```
## Warning in summary.lm(Model): essentially perfect fit: summary may be
## unreliable
```

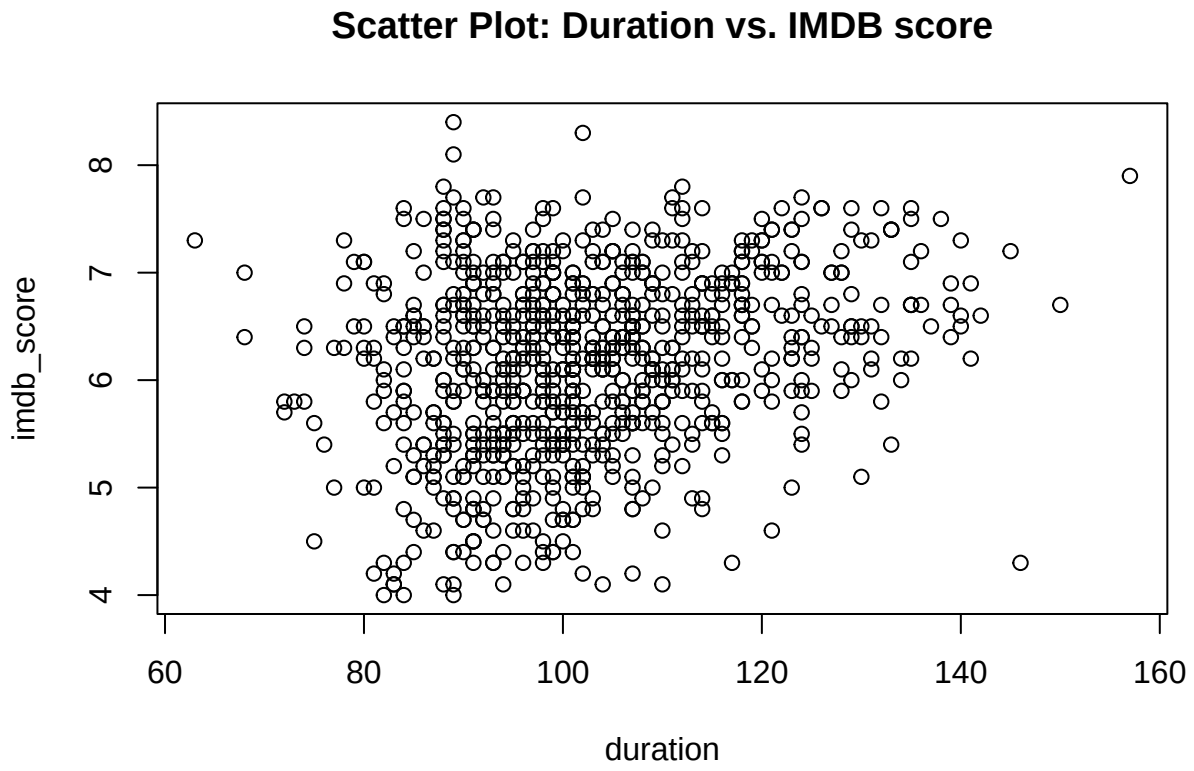
```
##
## Call:
## lm(formula = gross ~ ., data = sampled_data_regression)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.363e-07 -5.580e-09 -2.300e-10  4.460e-09  7.985e-07
##
## Coefficients:
##                              Estimate Std. Error    t value Pr(>|t|)
## (Intercept)                -5.148e-07  4.851e-07 -1.061e+00 0.288873
## num_critic_for_reviews      9.582e-11  2.861e-11  3.349e+00 0.000848 ***
## duration                    3.390e-10  1.037e-10  3.268e+00 0.001127 **
## director_facebook_likes     8.195e-12  1.117e-11  7.330e-01 0.463532
## num_voted_users            -8.407e-13  4.931e-14 -1.705e+01  < 2e-16 ***
## cast_total_facebook_likes -2.400e-12  1.458e-12 -1.646e+00 0.100052
## facenumber_in_poster      -3.812e-11  1.007e-09 -3.800e-02 0.969825
## num_user_for_reviews       1.518e-12  1.281e-11  1.190e-01 0.905698
## budget                     1.000e+00  7.211e-17  1.387e+16  < 2e-16 ***
## title_year                 2.136e-10  2.419e-10  8.830e-01 0.377430
## actors_facebook_likes      2.313e-12  1.512e-12  1.530e+00 0.126406
## profits                    1.000e+00  6.334e-17  1.579e+16  < 2e-16 ***
## imdb_score                 3.997e-09  1.840e-09  2.173e+00 0.030076 *
## movie_facebook_likes      -5.494e-14  2.739e-13 -2.010e-01 0.841080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.817e-08 on 856 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 4.413e+31 on 13 and 856 DF,  p-value: < 2.2e-16
```

```
library(ggplot2)
```

#Creating Scatter Plot between imdb score and variables highly correlated with IMDB score

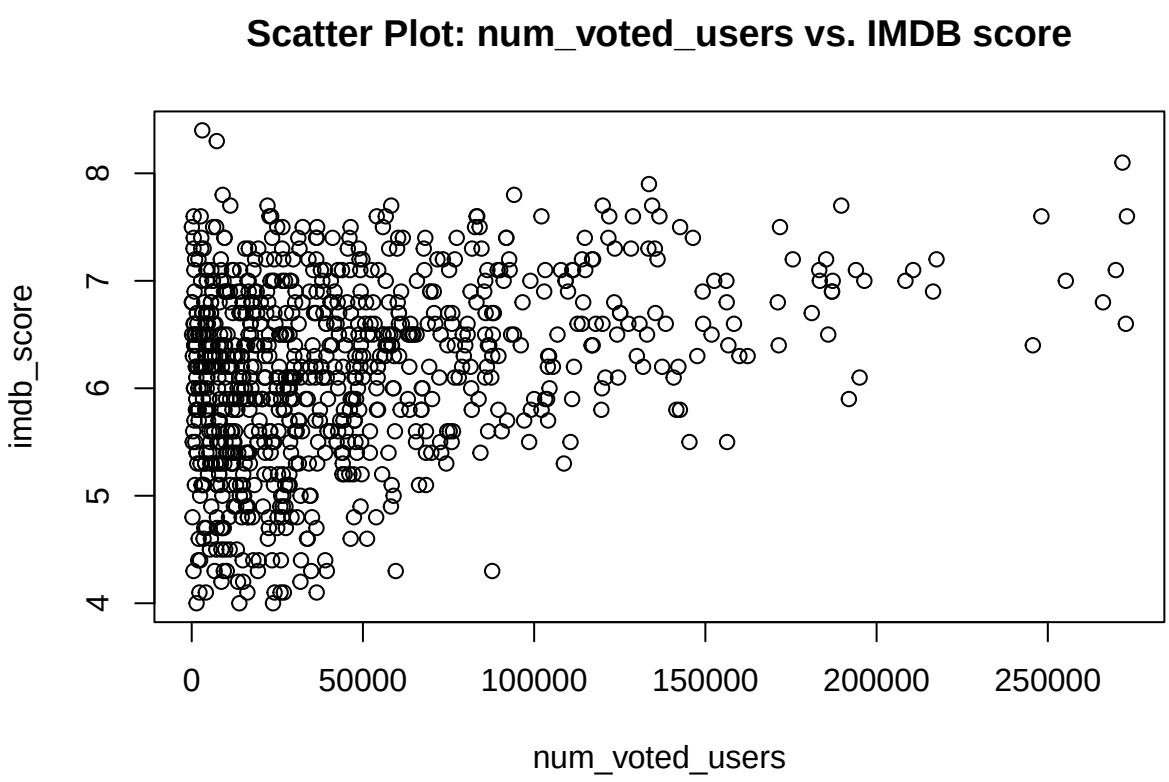
#scatter plot between duration and imdb score

```r
plot(sampled_data_regression$duration,sampled_data_regression$imdb_score,
     xlab = "duration", ylab = "imdb_score",
     main = "Scatter Plot: Duration vs. IMDB score")
```



#scatter plot between num_voted_users and imdb score

```r
plot(sampled_data_regression$num_voted_users,sampled_data_regression$imdb_score,
     xlab = "num_voted_users", ylab = "imdb_score",
     main = "Scatter Plot: num_voted_users vs. IMDB score")
```

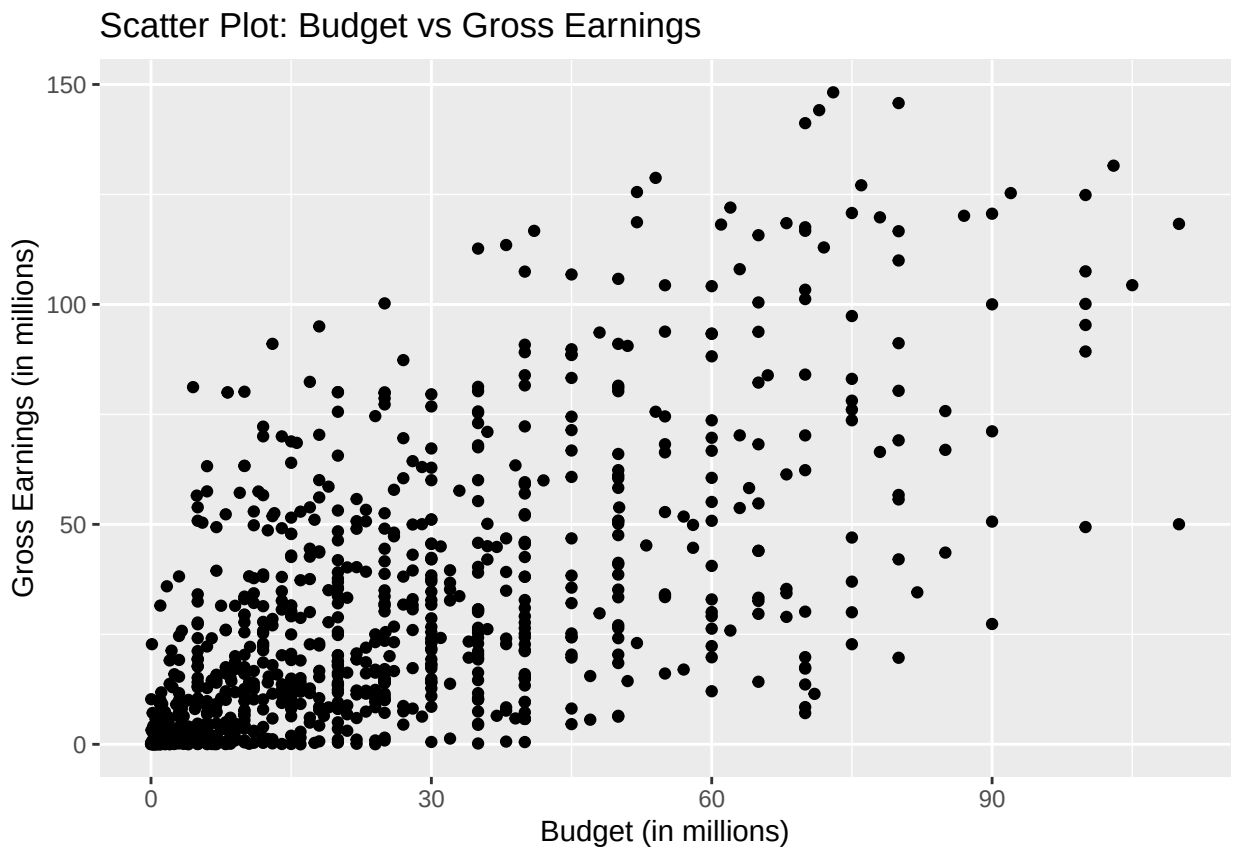



```
library(scales)
```

#Financial & Social Metrics #scatter plot between budget and gross

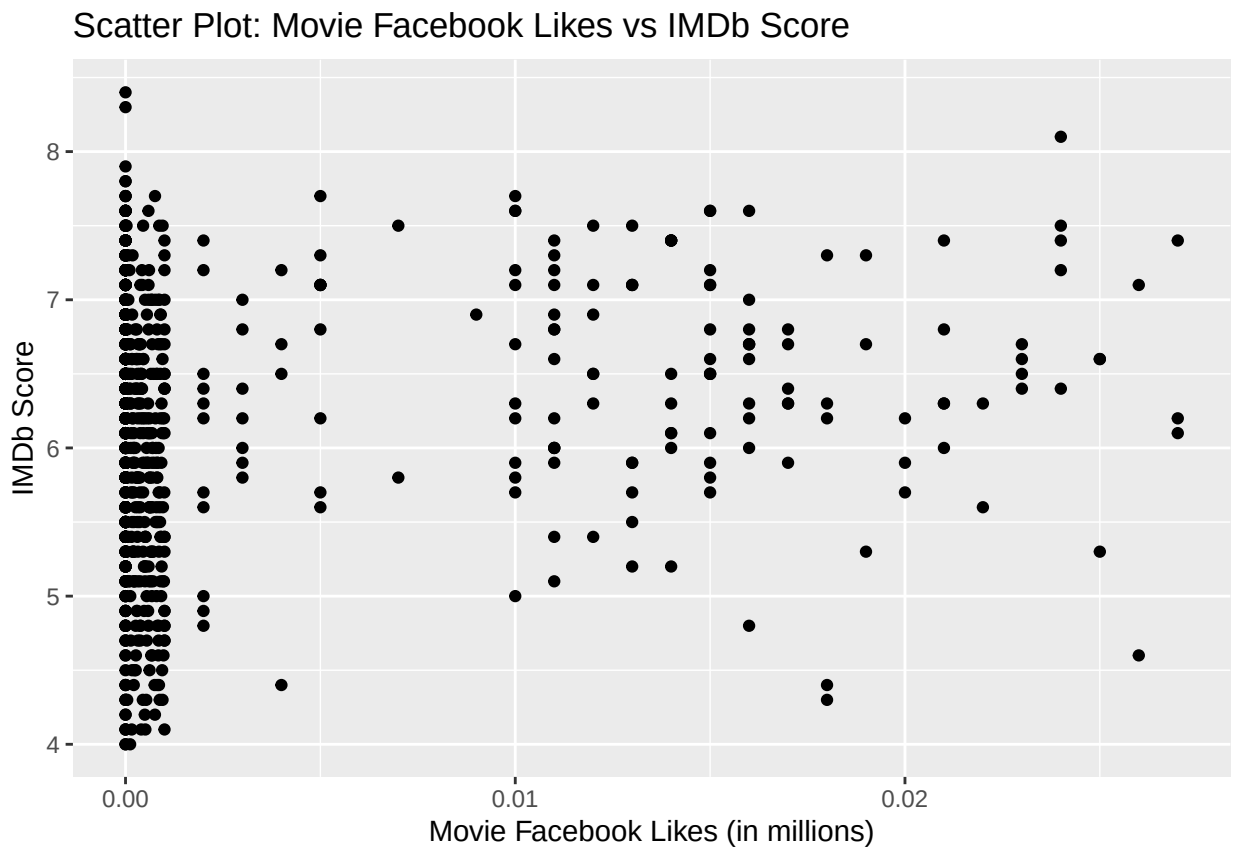
```
ggplot(sampled_data_regression, aes(x = budget / 1e6, y = gross / 1e6)) +
  geom_point() +
  labs(x = "Budget (in millions)", y = "Gross Earnings (in millions)",
       title = "Scatter Plot: Budget vs Gross Earnings") +
  scale_x_continuous(labels = scales::label_number_si()) +
  scale_y_continuous(labels = scales::label_number_si())
```

```
## Warning: ‘label_number_si()‘ was deprecated in scales 1.2.0.
## i Please use the ‘scale_cut‘ argument of ‘label_number()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Scatter plot between 'movie_facebook_likes' and 'imdb_score' with labels in millions

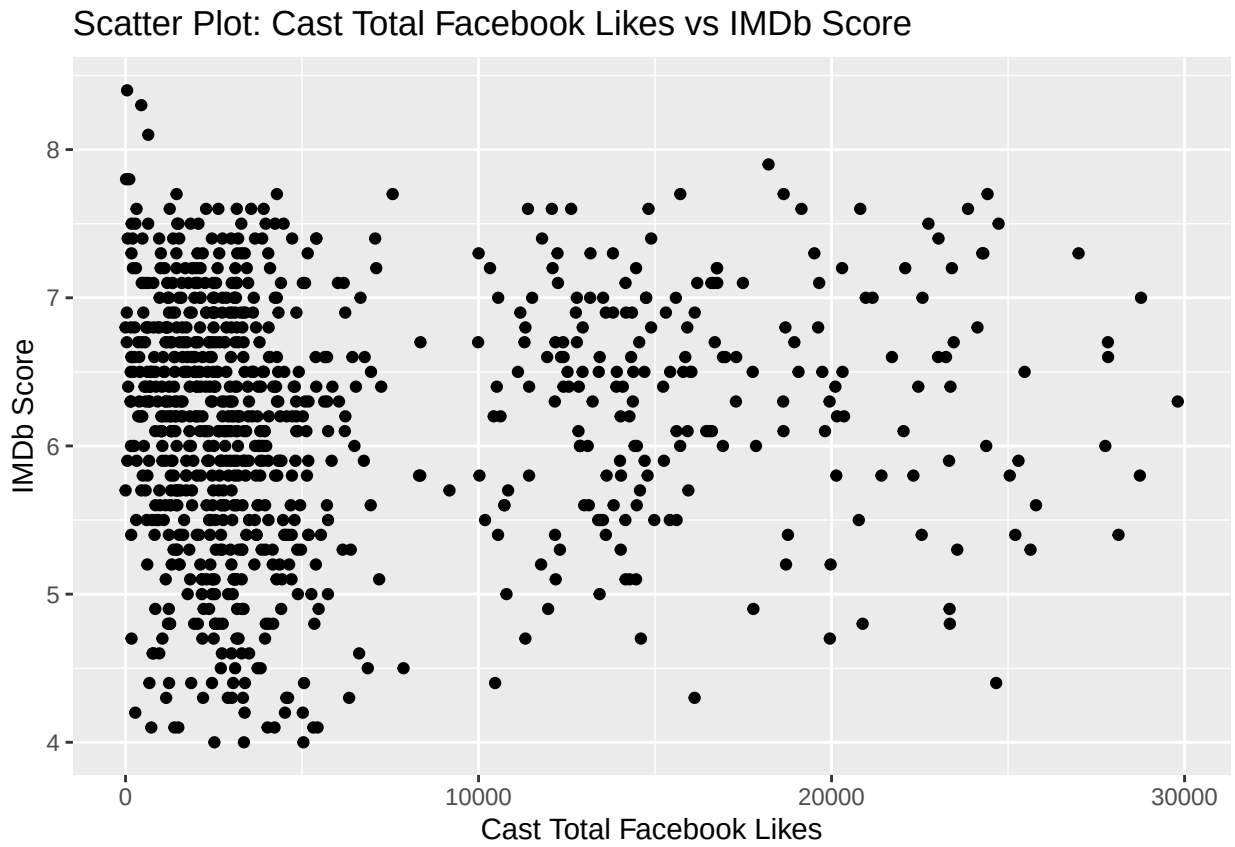
```
ggplot(sampled_data_regression, aes(x = movie_facebook_likes / 1e6, y = imdb_score)) +
  geom_point() +
  labs(x = "Movie Facebook Likes (in millions)", y = "IMDb Score",
       title = "Scatter Plot: Movie Facebook Likes vs IMDb Score") +
  scale_x_continuous(labels = scales::label_number_si())
```



Scatter plot between 'director_facebook_likes' and 'imdb_score'

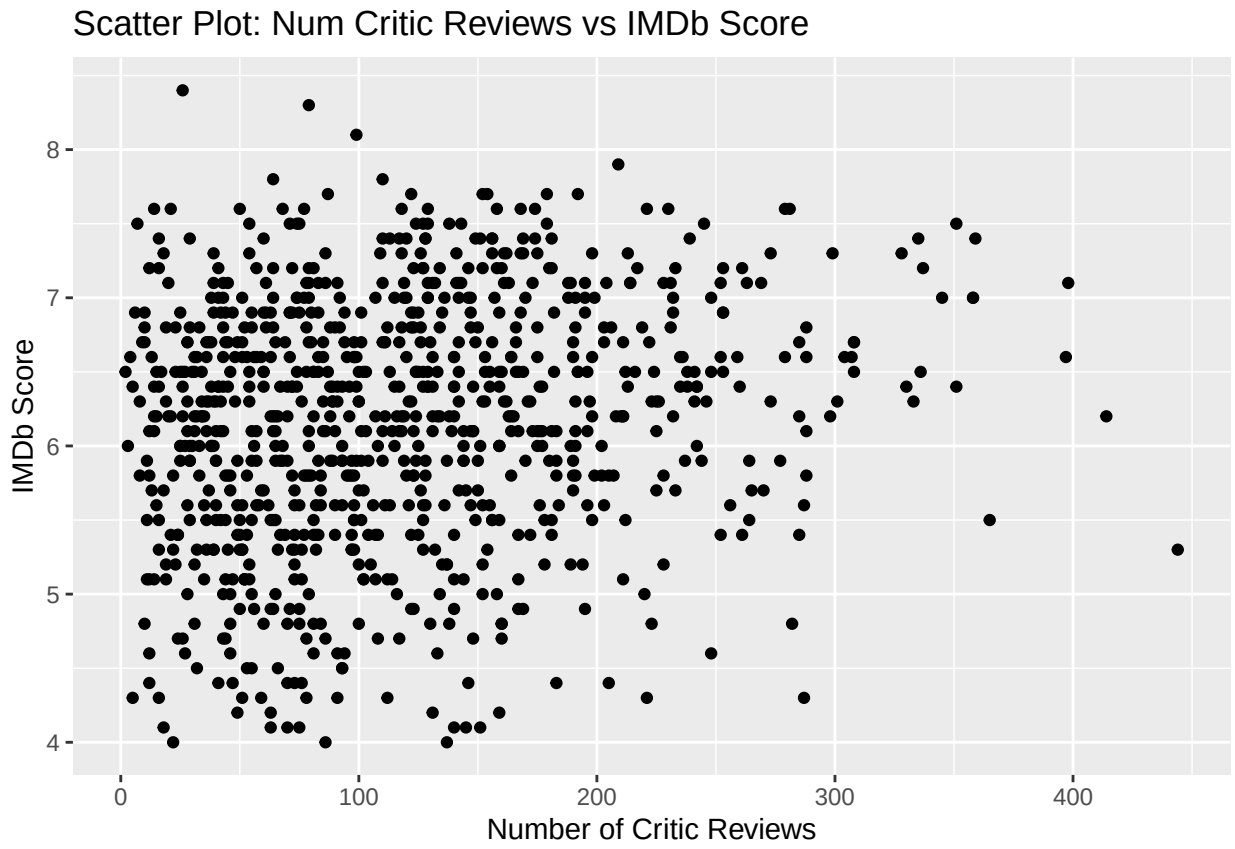
Scatter plot between 'cast_total_facebook_likes' and 'imdb_score'

```
ggplot(sampled_data_regression, aes(x = cast_total_facebook_likes, y = imdb_score)) +
  geom_point() +
  labs(x = "Cast Total Facebook Likes", y = "IMDb Score",
       title = "Scatter Plot: Cast Total Facebook Likes vs IMDb Score")
```



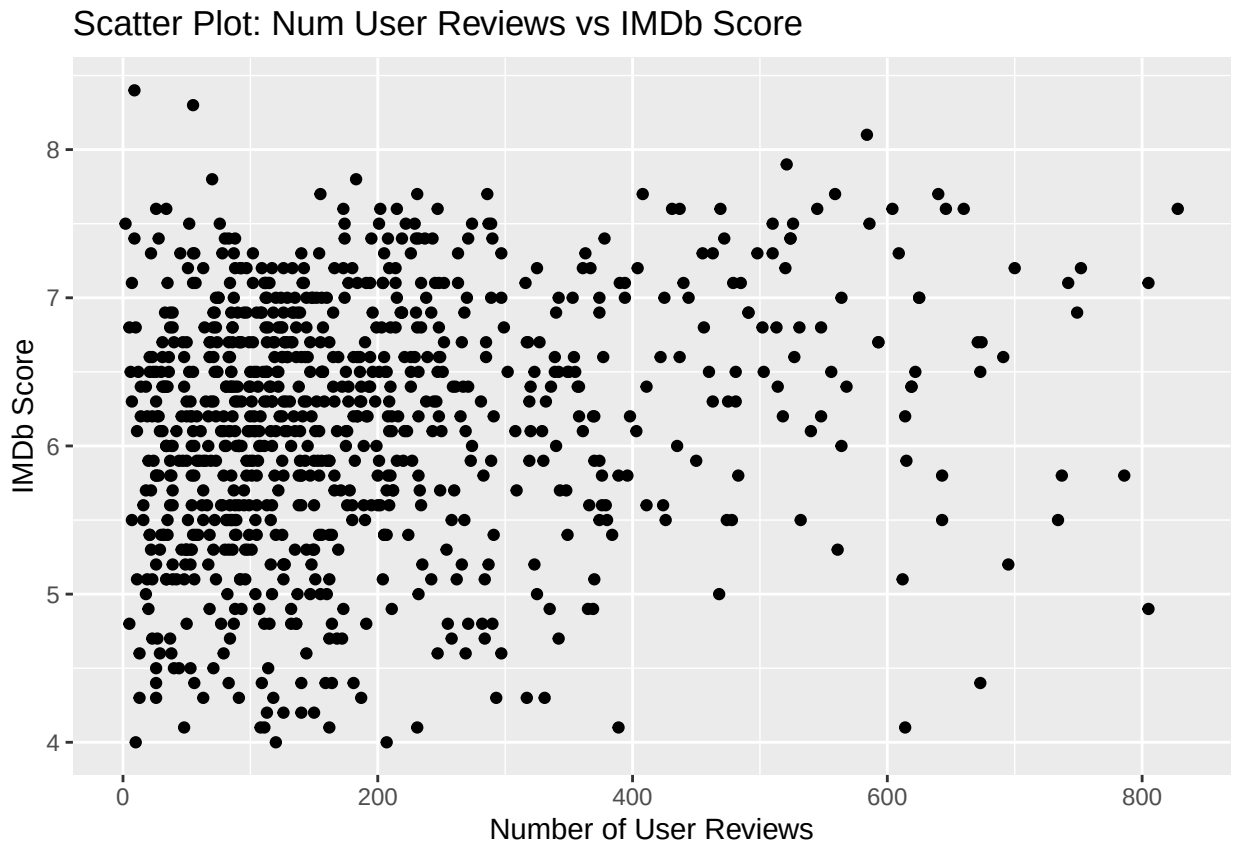
#Review & Reception Metrics # Scatter plot between 'num_critic_for_reviews' and 'imdb_score'

```
ggplot(sampled_data_regression, aes(x = num_critic_for_reviews, y = imdb_score)) +
  geom_point() +
  labs(x = "Number of Critic Reviews", y = "IMDb Score",
       title = "Scatter Plot: Num Critic Reviews vs IMDb Score")
```



Scatter plot between 'num_user_for_reviews' and 'imdb_score'

```
ggplot(sampled_data_regression, aes(x = num_user_for_reviews, y = imdb_score)) +
  geom_point() +
  labs(x = "Number of User Reviews", y = "IMDb Score",
       title = "Scatter Plot: Num User Reviews vs IMDb Score")
```



#R Code for Correlation Analysis # Calculate correlation between 'gross' and 'imdb_score'

```
correlation_result <- cor(sampled_data_regression$gross, sampled_data_regression$imdb_score)
```

Print the correlation coefficient

```
print(correlation_result)
```

```
## [1] 0.01028039
```

Interpretation based on the correlation coefficient

```
if (correlation_result > 0) {
  cat("There is a positive correlation between gross earnings and IMDb ratings.\n")
} else if (correlation_result < 0) {
  cat("There is a negative correlation between gross earnings and IMDb ratings.\n")
} else {
  cat("There is no linear relationship between gross earnings and IMDb ratings.\n")
}
```

```
## There is a positive correlation between gross earnings and IMDb ratings.
```

```
sample_index <- sample(nrow(sampled_data_regression), size=nrow(sampled_data_regression)*0.90, replace=F

train_valid <- sampled_data_regression[sample_index,]
test <- sampled_data_regression[-sample_index,]

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.2
```

```
## Loading required package: lattice
```

```
myCtrl <- trainControl(method="cv",
                       number=10)
myGrid <- expand.grid(.k=c(1:15))
set.seed(1)
knn_imdb <- train(as.numeric(imdb_score)~.,
                  data=train_valid,
                  method="knn",
                  trControl=myCtrl,
                  tuneGrid=myGrid,
                  preProc=c("center","scale"))

knn_imdb
```

```
## k-Nearest Neighbors
##
## 783 samples
##  13 predictor
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 705, 704, 705, 705, 705, 705, ...
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##    1  0.9419761  0.1388129  0.7449010
##    2  0.8389456  0.1678747  0.6668998
##    3  0.7948104  0.1966333  0.6310235
##    4  0.7768867  0.2080645  0.6116929
##    5  0.7547520  0.2336307  0.5981925
##    6  0.7470958  0.2381447  0.5935633
##    7  0.7453591  0.2393235  0.5923747
##    8  0.7427983  0.2404278  0.5895026
##    9  0.7451771  0.2370629  0.5967312
##   10  0.7430226  0.2406110  0.5950130
##   11  0.7481403  0.2311394  0.6000473
##   12  0.7441947  0.2413992  0.5972110
##   13  0.7451071  0.2419108  0.5981485
##   14  0.7441604  0.2443153  0.5979700
##   15  0.7453480  0.2428755  0.5977124
```

```
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 8.
```

```
predicted_imdb <- predict(knn_imdb,
                          newdata=test,
                          type='raw')

forecast::accuracy(predicted_imdb,test$imdb_score)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
##                   ME      RMSE       MAE       MPE     MAPE
## Test set 0.02025862 0.7418535 0.5920977 -1.205968 10.29737
```