UNIVERSITY OF MALAYA

EIE3001 STATISTICAL COMPUTING

TITLE:

AN ANALYSIS ON THE HOLLYWOOD TOTAL GROSSING BLOCKBUSTER FILM'S DATA FROM 1975 TO 2018 USING MULTIPLE LINEAR REGRESSION MODEL.

GROUP H

No	Member's Name	Matric. No
1	NORDIN BIN ZAHARI	17142100
2	NAVIN KUMAR A/L GANESAN	17060772
`3.	NURDIANA IEZATY BINTI MOHD NORDIN	17090719
4	NURSYAZLIN BINTI NORAZMI	17123156
5	OUMELLOUCHE IMANE	17203388

Lecturer's Name:

Dr. Diana Binti Abdul Wahab 😊

Table of Contents

1.0 Introduction

Hollywood is considered to be the oldest film industry where earliest film studios and production companies emerged, and is also the birthplace of various genres of cinemas, among them comedy, drama, action, the musical, romance, horror, science fiction, and the war epic, which had set an example for other national film industries. It is a true symbol of entertainment industry that creates many innovative and extraordinary movies which is apart from our thinking such as Avatar, Harry Potter and list goes on.

However, not all the movies produced are good to watch or worth watching in fact audience will always goes for the blockbuster hit movies where acted by popular stars and having nice plot of the story. Blockbuster hit movies refer to the movies produced using large budget and big stars usually would yield massive amount of gross in return. The term has also come to refer to any large-budget production, aimed at mass markets with associated merchandising, sometimes on a scale that meant the financial fortunes of a film studio or a distributor could depend on it.

Besides budget, actors, genre, plot story of the movies it's total gross was the crucial indicator in determining whether the movie is a blockbuster hit or not. There is no certain value of gross film used to perceive it as a blockbuster movies but by comparing total gross among other films and pick up the highest top 5 or 10 by language, country or genre we can identify blockbuster movies.

For that matter we have chosen to identify the variables that influence the total gross amount of the film. We obtained a dataset that contains information about all the top ten Hollywood grossing movies of each year from starting from 1975 till 2018.

2.0 Data Descriptive.

The Top 10 Highest Grossing films data was obtained from The Kaggle website. which consists of blockbuster rated Hollywood movies ranging from 1975 to 2018. The top 10 highest grossed movies every year were included in the data. The data consists of 487 observation where a single observation represents a movie. Furthermore, the data contains 11 variables which are Main genre of the film(**Main_Genre**), First sub-genre of the film(**Genre_2**), Second sub-genre of the film(**Genre_3**), International Movie Database(IMDB) rating of the film(**imdb_rating**), Length of the film in minutes(**length**), Rank of the film in the specific year(**rank in year**), Motion Picture Association of America(MPAA) rating of the film(**rating**), Distributing studio of the film(**studio**), Name of the film(**title**), Total gross of the film in millions(**worldwide_gross**) and Year of release of the film(**year**). In this study, Total gross of the film in millions**(worldwide_gross)** used as response variable while other all variables except **Genre_2**, **Genre_3** and **title** are part of our interest in this study.

The data structure shows that there are different types of data. There are 6 variable which are factor type namely "Main_Genre", "Genre_2", "Genre_3", "rating", "studio" and "title". There are 2 Numeric data types variables such as "worldwide_gross" and "imdb_rating" while the rest of the variables are Integer.

2.1 Descriptive Statistics

Total Gross Of The Film (**worldwide_gross**)	
Mean	390,240,893
Median	334,201,140
Std. Deviation	315,834,361
Range	2,714,391,228
Minimum	34,673,100
Maximum	2,749,064,328

Table 1: Descriptive statistics of dependant variable

The Table 1 shows descriptive statistics of dependent variable(worldwide gross). The Total Gross of the Film has a mean value of 390.24 million dollar while median of 334.20 million dollar, lower than the mean. The movie's gross value deviated from their mean value shown by standard deviation about 315.83 million dollar. The movie's gross value has a minimum value of 34.67 million dollar to maximum of 2.75 billion dollar with a range of 2.71 billion dollar.

	IMDB rating of the film (**imdb_rating**)	Length of the film in minutes (**length**)	Year of release of the film (**year**)	Rank of the Movie in Year (**rank in year**)	MPAA rating of the film (**rating)**
Mean	7.077	119.87	1996	5.52	2.80
Median	7.100	118.00	1997	6.00	3.00
Std. Deviation	.8203	22.744	12.630	2.870	0.865
Range	4.6	174	43	9	3
Minimum	4.4	27	1975	1	1
Maximum	9.0	201	2018	10	4

Table 2: Descriptive statistics of independent variables

Table 2 shows the descriptive statistics of independent variables which included in the model. IMDB rating of the film has a mean rating of 7.077 while median of 7.10 slightly higher than mean. IMDB relatively has lower standard deviation compare to others variable which is 0.8203. Minimum rating of 4.4 while maximum rating of 9.0 with range of 4.6. Next, The mean length of movies is 119.87 minutes or approximately 2 hour while median is 118 minutes slightly lower than the mean. Minimum length of the movies is 27 minutes which is refer to one of the 90's movies while Maximum length of the movies is 201 minutes or equivalent to 3 hour 21 minutes. Year release of the film was from 1975 to 2018 shown by both minimum and maximum values. Both rank in year and rating variables are ordinal variables which have a total of 10 and 4 subcategories respectively for both variables. Rating has a mean value of 2.80

or round-off to 3 which refer to the third sub-category of rating which is “PG-13” indicates that the movie which needs parental guidance especially children aged 13 or below.

2.2 Data Exploration

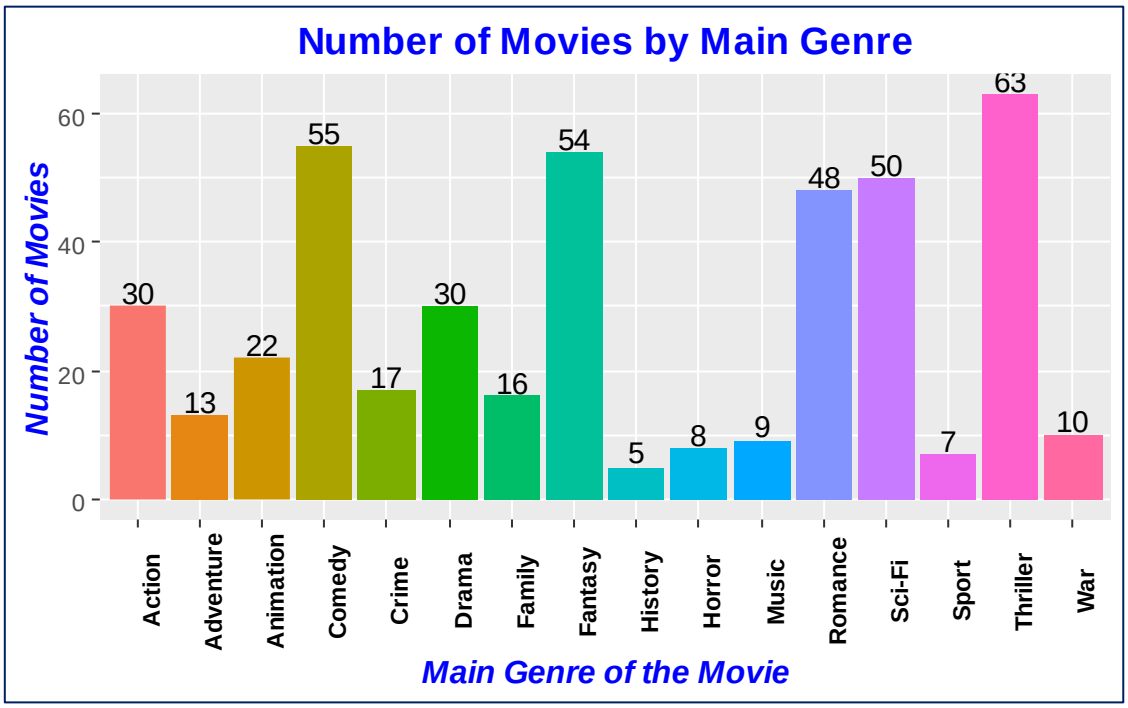


Figure 1: Number Of Movies Produced by Main Genre of the Movies

The Figure 1 above shows number of movies produced according to Main genre. The Main Genre variable consists of 16 genre category. Thriller Genre recorded the highest with 63 or (14.41%) movies produced while History genre was the lowest with only 5 movies produced (1.14%).

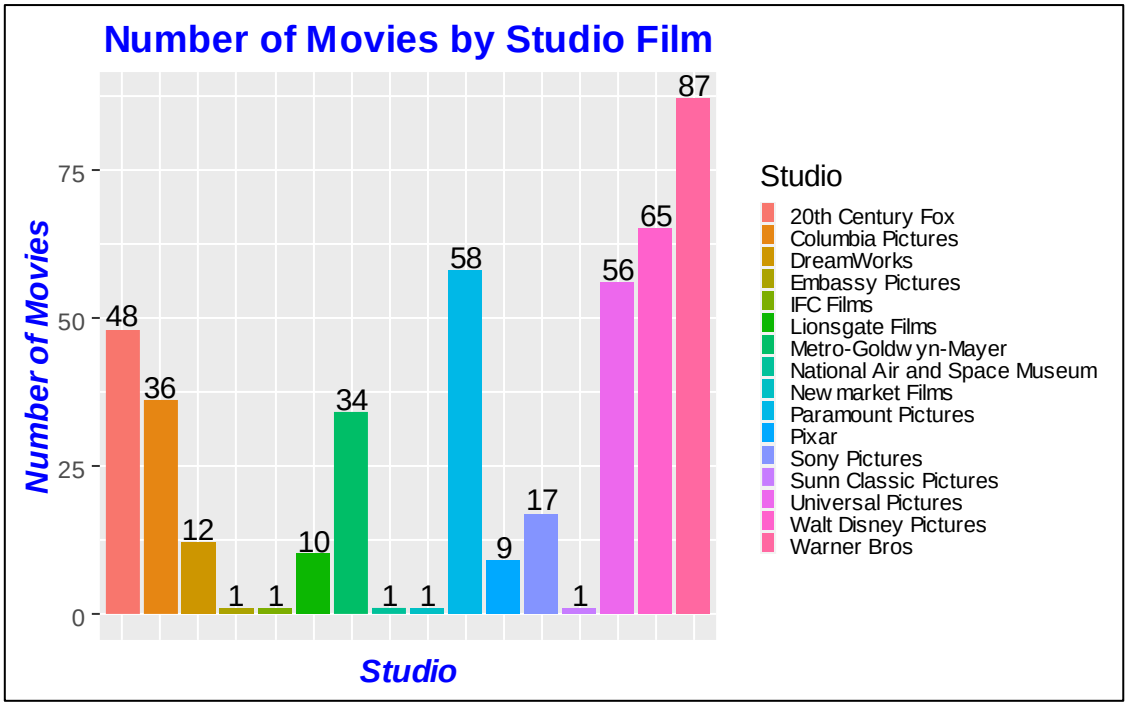


Figure 2: Number Of Movies Produced by Studio Film

Moreover, in Figure 2 above shows the Number of movies produced by the studio film. There are 16 studio companies. Warner Bros was the highest distributing studio company with 87 movies(19.9%) released while lowest distributing movies is 1(0.2%) with combination of 5 studio companies namely Embassy Pictures, IFC Films, National Air And Space Museum, New Market Films And Sunn Classic Pictures.

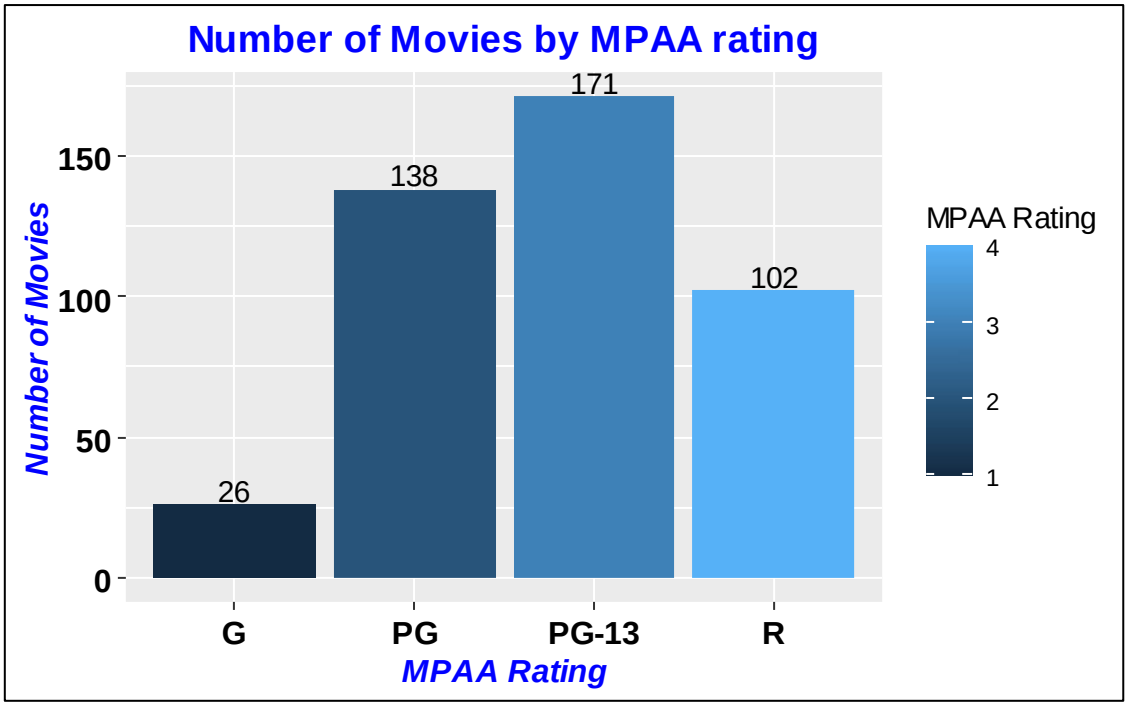


Figure 3: Number Of Movies by MPAA Rating.

Motion Picture Association of America(MPAA) rating of the film(**rating**) consists of 4 ordinal categories which are “G”, “PG”, “PG-13” and “R”. “G” stands for general audience where all ages admitted to watch the movies. “PG” stands for parental guidance suggested where some material may not be suitable for children and needed parental guidance. Moreover, “PG-13” stands for parents strongly cautioned where some material inappropriate for children under 13 and parent should be cautious. Lastly, “R” stands for restricted where under 17 requires accompanying parent or adult guardian and parents are urged to learn more about the film before taking their young children with them. As rating increase from “G” to “R” suitability of everyone to watch the movies decreases. Higher rating such as “R” implies only adult or 17 years old and above only can watch movies. Thus, PG-13 rated movies where produced more compare to others with 171 movies and while G rated movies was the least with only 26 movies.

2.3 Data Cleaning/Manipulation

The raw data was originally taken from crowdflower website. The raw data has been manipulated by third party, The Kaggle. Variables such as URL, Adjusted total gross, rating of audience, rating of score, rating of freshness and audience freshness have been removed by the third party. Furthermore, in raw data the movies only available until year 2014 then the third party website further updated the database up to year 2018. The data we are using from the third party source, The Kaggle.

There are total of 170 missing values found in this data. 29 missing values from Genre_2 and 141 from Genre_3. The missing values were replaced with "NA" string in respective cells and the worldwide_gross variable was originally in factor type then converted into numerical by removing commas and dollar sign in the values, both process were done in Microsoft Excel before importing to Rstudio. Moreover, "rating" variable has been recoded into integer since it's an ordinal data. The subcategories of data "G", "PG", "PG-13" and "R" have been recoded into integer with 1,2,3,4 respectively. Finally, due to missing values, Genre_2 and Genre_3 were left out for this study.

3.0 Methodology

$$\text{Worldwide\_gross} = ß_0 + ß_1(\text{rank in year}) + ß_2(\text{rating}) + ß_3(\text{year}) + ß_4(\text{imdb rating}) + ß_5(\text{length})$$

We have chosen multiple linear regression model to regress all of the variables to show the relationship between worldwide-gross and 5 predictors variables which are rank in year, rating, year, imdb rating and the length. Variables like Main genre, Title, and Studio has been left out of the model since its all categorical variable and we only include numerical and Integer type as our independent variable for simplicity. We are using this model because there are more than 1 predictors variable used and moreover all variables both dependant and independent are in either numerical or integer type has made easy to use multiple linear regression model.

To show that there are the correlation between the worldwide gross and all of the predictors variable, we are looking at their coefficient estimation which are ß1 (coefficient estimation for rank in year) , ß2 (coefficient estimation for rating), ß3 (coefficient estimation for year), ß4 (coefficient estimation for imdb rating), ß5 (coefficient estimation of length) and ß0 is an intercept.

The fitted model:

$$\text{Worldwide gross} = -31{,}374{,}041{,}943 - 38{,}985{,}792(\text{rank in year}) - 31{,}952{,}902(\text{rating}) + 15{,}918{,}419(\text{year}) + 2{,}456{,}167(\text{imdb rating}) + 2{,}237{,}479\ (\text{length})$$

We can see that there are two variable that have negative correlation with worldwide gross which are the rank in year and the rating indicated by negative sign. This means that the increase in rank in year (for example rank 1 to rank 2 which means in term of number its increasing but in term of ranking is decreasing) will decrease the worldwide gross on average by $38,985,792($38.99 million) holding other variable constant and the 1 unit increase in rating(for example from ‘PG’ to ‘R’) will decrease the worldwide gross on average by $31,952,902($31.95 million) holding other variable constant.

Furthermore, the others variables which are year, imdb rating and length have the positive correlation. This means that the increase of 1 year or 1 unit or 1 minute will increase worldwide gross on average by $15,918,419($15.92 million), $2,456,167($2.46 million), and $2,237,479($2.24 million) respectively holding other variable constant.

4.0 Result and Discussion

4. 1 Summary Of The Fitted Model

Independent Variables	Coefficient	Standard Error	t-value	p-value
intercept	-31,374,041,943	1,486,436,562.00	-21.107	0.00
rank in year	-38,985,792	3,404,880.30	-11.45	0.00
rating	-31,952,902	11,544,112.86	-2.768	0.01
year	15,918,419	750,596.77	21.208	0.00
IMDB rating	2,456,167	12,061,507.37	0.204	0.84
length	2,237,478	467,882.34	4.782	0.00
R^2= 0.6328	Adjusted R^2=0.6285	F (5,431) = 148.5	p-value = 0.00	

Table 3: Summary Output of The Fitted Model

Coefficient Interpretation

- 1 unit increase in rank in year (for example, Rank 1 to Rank 2 which means in term of number its increasing but in term of ranking is decreasing) will decrease the worldwide gross by average $38,985,792 ($38.99 million) holding others variable constant.
- 1 unit increase in rating will decrease the worldwide gross by average of $31,952,902 ($31.95 million) holding others variable constant.
- increase in 1 year will increase the worldwide gross by average $15,918,419 ($15.92 million) with holding others variable constant.
- Increase in 1 unit of IMDB rating will increase $2,456,167($2.46 Million) in average worldwide gross, holding others variable constant.
- 1 minute increase in length of the movie will increase average of worldwide gross by $2,237,479($2.24 Million) holding others variable constant.

Based on 5% level of significance, all coefficients are statistically significant where p-value less than 0.05 except β4 or IMDB rating is statistically insignificant where values greater than 0.05(0.84). So we re-run the model again without the variable IMDB rating shows only a slight increase in Adjusted R^2 by 0.0009(0.6294) while with the inclusion of that variable Adjusted R^2 is 0.6285. However, R^2 remain the same for the both scenario. Thus, we decided to proceed with this model. Based on $R^2 = 0.6328$ shows that 63.28% of the variation in worldwide gross is explained by the variation in independent variables (rank in year, rating, year, IMDB_rating and length)

4.2 Testing The Overall Significance Of The Model

H0: $\beta 1 = \beta 2 = \cdots = \beta 5 = 0$

Ha: at least one βj is non-zero.

F stat = 148.5, $F_{\alpha=0.05,5,431}$ (2.21)

Since F-stat(148.5) >F(2.21),

Reject H0, Conclude at least one of the independent variable must be related to the worldwide gross.

4.3 Independent variable: Rank In Year

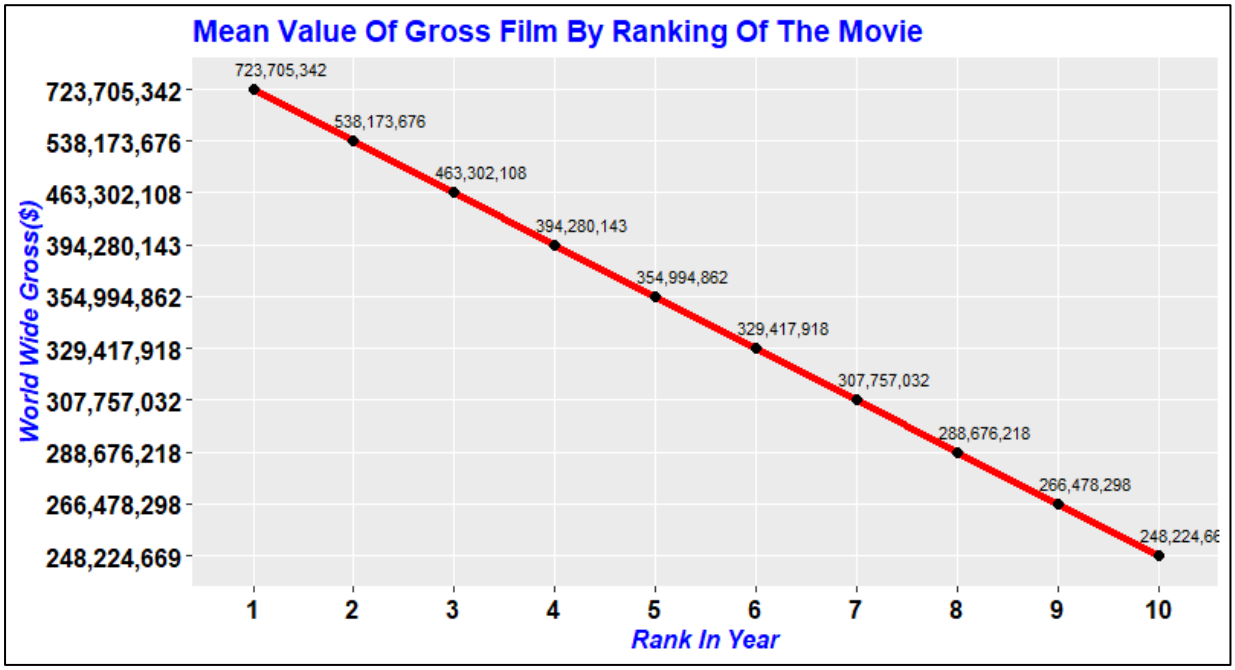


Figure 4: Mean Value Of Gross Film By Ranking Of The Movie

Base on the line graph above we can see downward-sloping straight line shows the negative relationship between rank in year and worldwide gross. As the ranking of the movie increases (example: from rank 1 to rank 10 meaning decreases in term of rank) and causes average worldwide gross to decrease from $723.71 million to $248.22 million. So, higher ranked movie tends to yield higher gross amount compare to lower ranked movies.

4.4 Independent variable: MPAA Rating(rating)

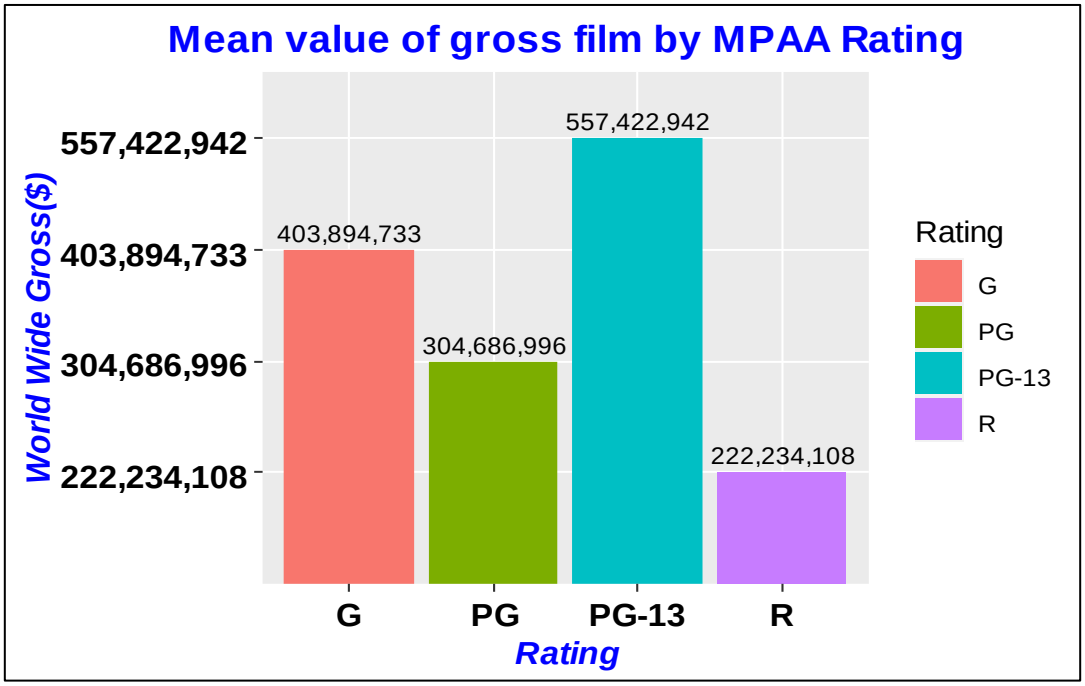


Figure 5: Mean Value Of Gross Film By Rating Of The Movie

The Bar chart above shows the movies by Motion Picture Association of America(MPAA) rating. The movie rated as 'PG-13' had achieved the worldwide gross by average $557,422,942 ($557.42 Million) and the movie rated 'R' is the lowest by average $222,234,108 ($222.23 Million). PG-13 is represent parents strongly cautioned because of some material may be inappropriate for the children under age 13. Rated R stand for restricted which is under age 17 requires accompanying parent or adult guardians. 'PG-13' might be higher because that movies are suitable for kids and adults. So, most of them prefer to that kind of rated movie compare to the others. "R" rated movie category was the lowest worldwide gross because the content is not suitable for children and teenagers aged 17 below and only suitable for adults.

4.5 Independent variable: Year of The Movie Release.

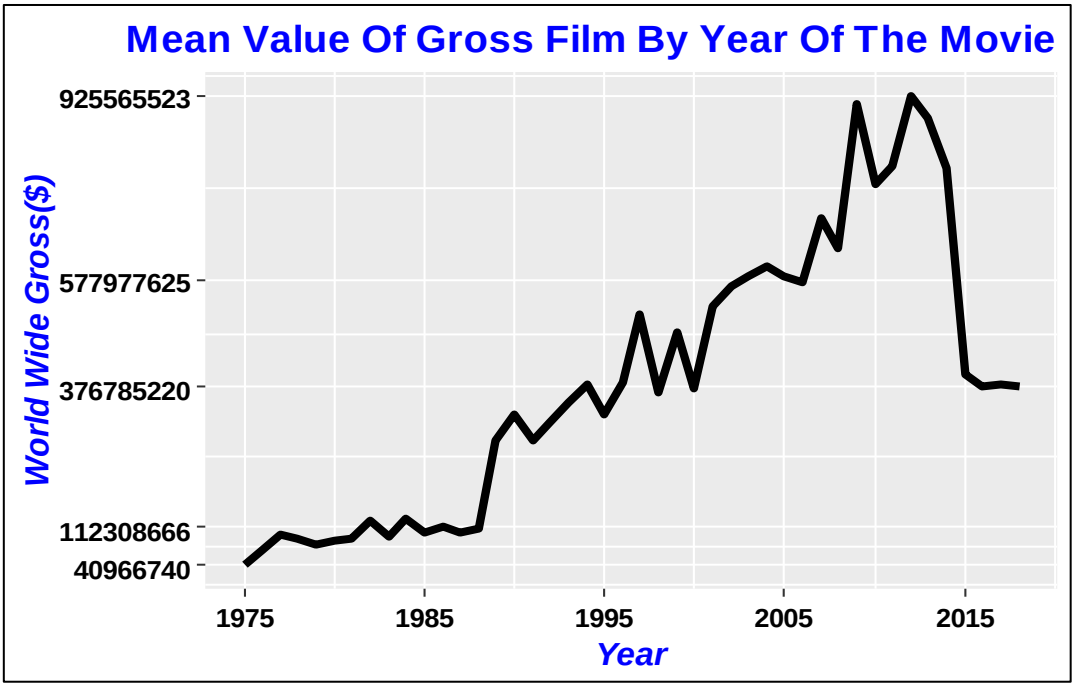


Figure 6: Mean Value Of Gross Film By Year Of The Movie

The Figure 3 above shows that as the year increases the mean value of worldwide gross also increases as well. This is because many advanced technology and facilities are available as the year progresses. Thus, many people have access to movies and it's further increases the quality of movie via advanced technology such as ultra HD 3D technology, autonomous drone and so on.

We can see that, the highest peak of worldwide gross occurs in year 2009 and 2012 at average of $909,438,969($909.44 Million) and $925,565,523 ($925.57 Million) respectively. In 2009, release of the movie "Avatar" has made remarkable history by recording one of the highest gross ever recorded with staggering value at $2,749,064,328($2.75 Billion). Avatar movie was the main reason, behind peak in 2009 beside that movies such as "Harry Potter and the Half Blood Prince", "Ice Age: Dawn of the Dinosaurs", "Transformers: Revenge of the Fallen" contribute as well. During year 2012, films that aired is most interesting and give impact to fans such as "The Avengers", "Skyfall", "The Dark Knight Rises", "The Hobbit: An Unexpected Journey" each of these movies recorded gross value of approximately $1 Billion above and thus in 2012 recorded the highest mean value compare to all of other year. After 2012, the mean gross started to drop because fans have put higher expectation for upcoming movies since the release of such great movies in year 2012. Lastly, Avatar movie record was

later broken by recent release of “The Avengers: Endgame” in 2019 which collected $2.790 Billion.

4.6 Independent variable: IMDB Rating

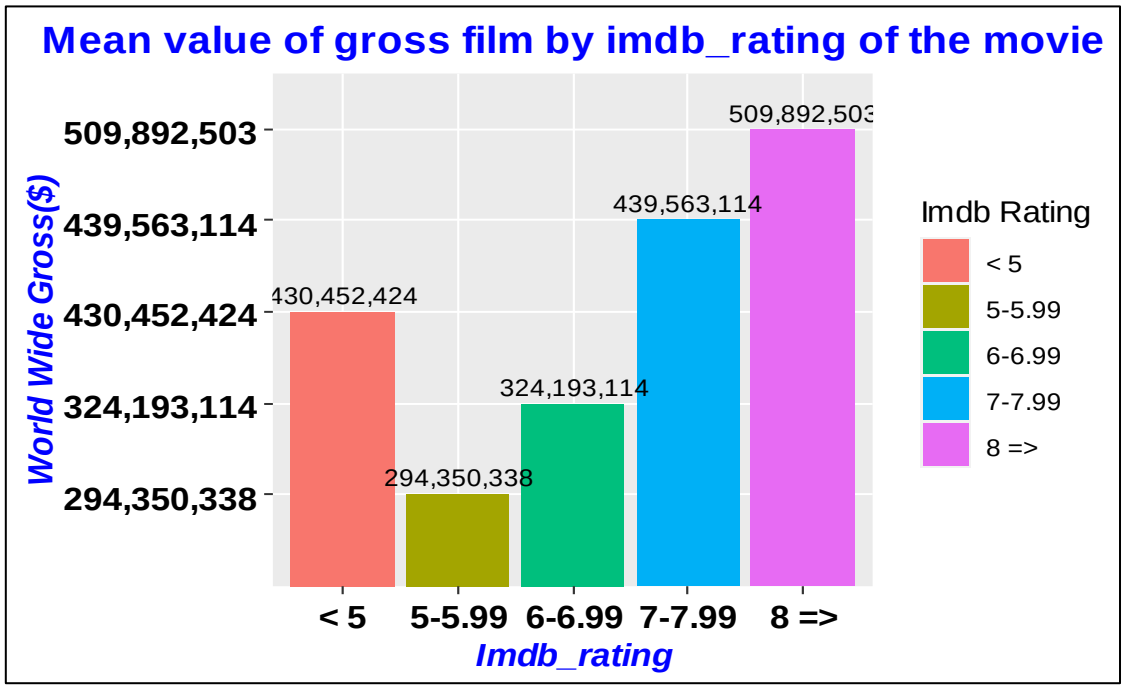


Figure 7: Mean Value of Gross Film by Film by IMDB Rating Of The Movie

International Movie Database(IMDB) is the world's most popular and reliable source for film, TV, and celebrity content which are created to help fans explore the world of movies. IMDB also helps fans to decides what to watch based on review and rating. This bar chart showed the films with IMDB rating equals 8 or above have the highest of mean worldwide gross $509,892,503($509.89 Million) and the lowest is films with a 5-5.99 rating showed mean value of $294,350,338($294.35 Million). Mean gross value increases as the imdb rating increases from 5 to 8 onwards. Rating below 5 somehow have higher mean compare to rating between 5 to 6.99 this is because fans have percepation that underrated movies sometimes will meet their satisfication. Commonly, people willing to watch movies depend on the rate, so worldwide gross is high for 8 or above rate.

4.7 Independent variable: Length of the Movie

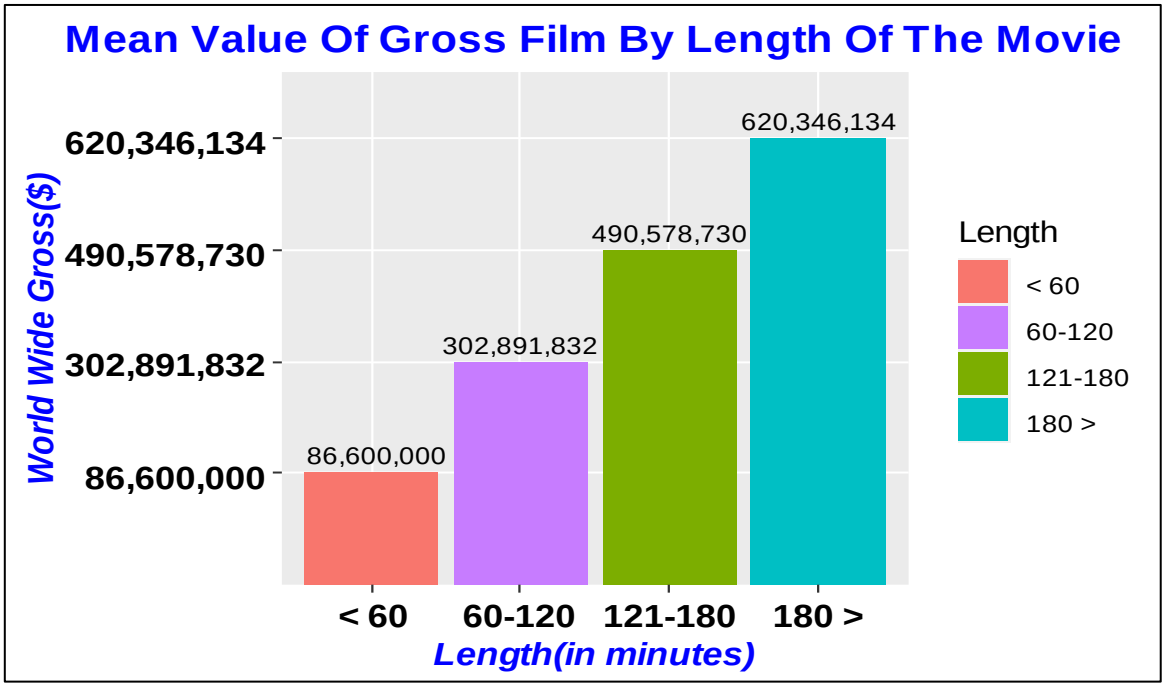


Figure 8: Mean Value of Gross Film by Length of the Movie

From we can see that the mean value of gross film increasing as length of the movie increasing. The mean value of gross movie with length more than 180 minutes or 3 hours was the highest with worldwide gross $620,346,134($620.35 Million) and the lowest is length of the movie is 60 minutes less with only $86,600,000(86.6 Million). This is because of long length movie with bunch of stuff happening make people feel their time and money worth. In long movies have many epic and plot twist can be occurred that give more impact to people who watching. Thus, longer the length of the movie higher the gross value of the film.

5.0 Conclusion:

The results demonstrate that the length of the movie, IMDB rating, MPAA rating, year of film production and rank of the movie play a major role in determining the movie's total gross. The influence of each variable is the different from one to another. The movies with longer length time, recent year of release, higher IMDB rating, higher ranking of the year and the movie either rated as "PG" or "PG-13" tend to produce higher gross value. Taking into account the results of this predictive model, we can say that many independent variables such as genre of the movie, budget of the movie, actors, director must be taken into account in order to explain more on influence of the dependant variable(Total gross of the film).

6.0 Appendix

```
getwd()
```

```
## [1] "C:/Users/Nordin/Desktop/Stat.Comp"
```

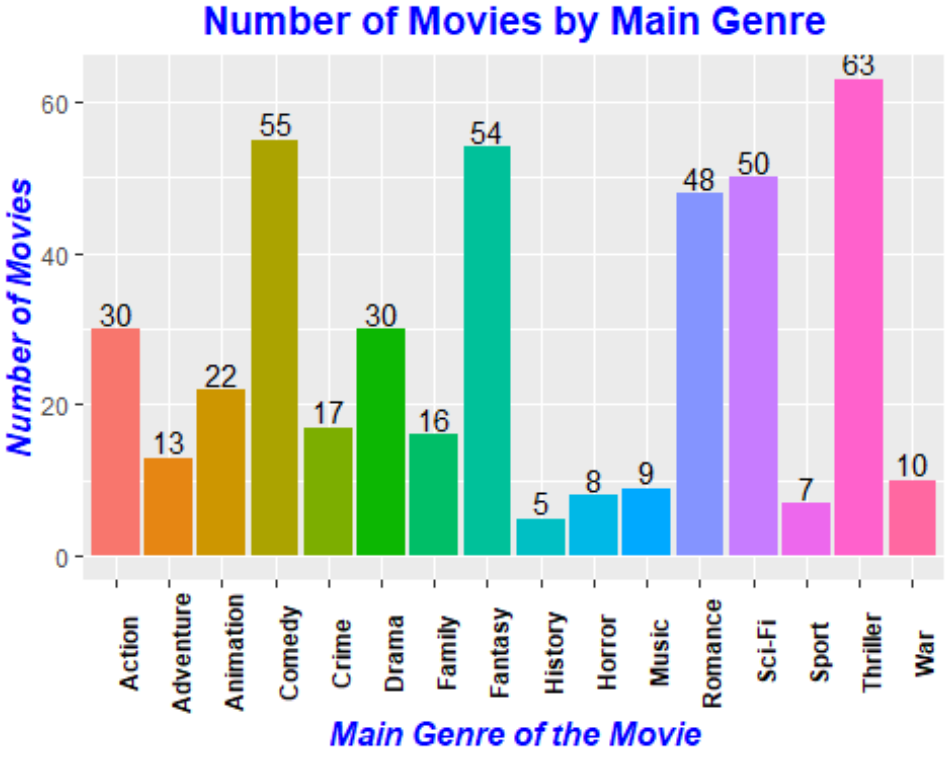
```
setwd("C:\\Users\\Nordin\\Desktop\\Stat.Comp")
```

```
#Importing Data, fileEncoding argument was added to prevent error in readi
ng the data
blockbuster <- read.csv("SCgroupH.csv", fileEncoding = "UTF-8-BOM")
str(blockbuster)
```

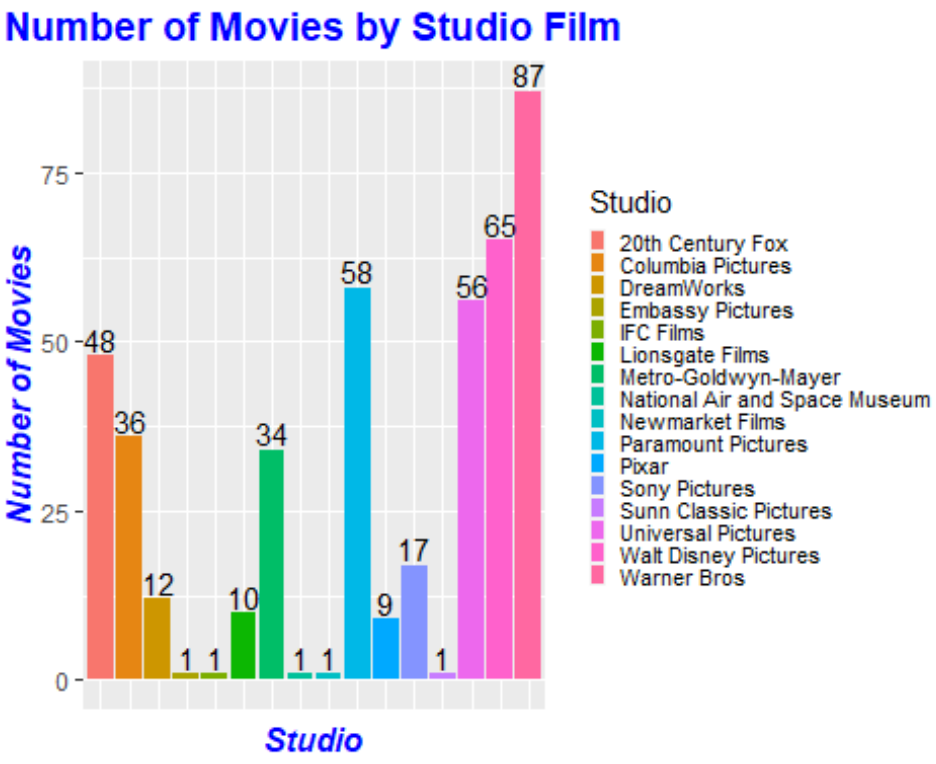
```
## 'data.frame':    437 obs. of  11 variables:
##  $ Main_Genre     : Factor w/ 16 levels "Action","Adventure",..: 1 1 3
1 1 1 1 1 1 11 ...
##  $ Genre_2        : Factor w/ 17 levels "Action","Adventure",..: 2 2 1
2 4 2 4 2 2 7 ...
##  $ Genre_3        : Factor w/ 14 levels "Action","Adventure",..: 7 11 2
7 NA 7 NA 7 7 NA ...
##  $ imdb_rating    : num  7.4 8.5 7.8 6.2 7.8 7.9 7.2 7 6.9 8.1 ...
##  $ length         : int  135 156 118 129 119 147 118 135 112 135 ...
##  $ rank_in_year   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ rating         : int  3 3 2 3 4 3 3 3 3 4 ...
##  $ studio         : Factor w/ 16 levels "20th Century Fox",..: 15 15 11
14 1 10 15 15 2 16 ...
##  $ title          : Factor w/ 436 levels "\"Crocodile\" Dundee",..: 56
38 153 174 84 202 28 277 421 13 ...
##  $ worldwide_gross: num  7.00e+08 6.79e+08 6.09e+08 4.17e+08 3.18e+08 .
..
##  $ year           : int  2018 2018 2018 2018 2018 2018 2018 2018 2018 2
018 ...
```

```
attach(blockbuster)
```

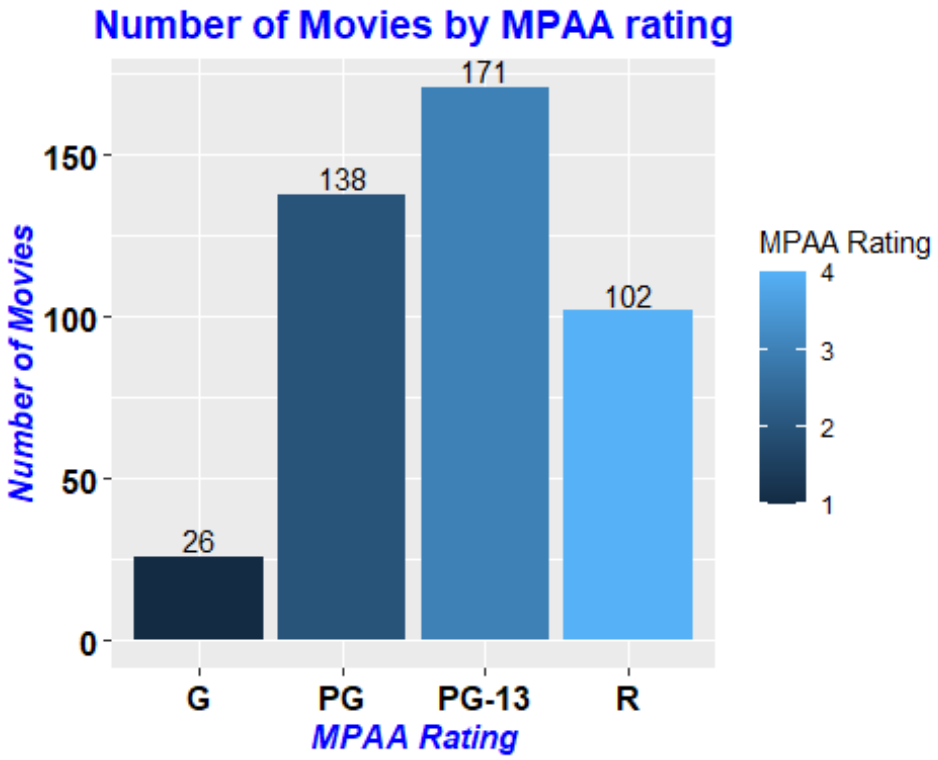
```
#------------------DATA-EXPLORATION-------------------------------------
-
library(ggplot2) #activate ggplot library for ggplot graph

## Warning: package 'ggplot2' was built under R version 3.5.3

# Number Of Movies Produced by Main Genre of the Movies
# use bar graph
ggplot(blockbuster, aes(x=Main_Genre, fill=Main_Genre))+geom_bar()+

  # set the label for x and y axis & title
  labs( x="Main Genre of the Movie",y="Number of Movies",title="Number of
Movies by Main Genre")+

  #show the count label at the top of the bar
  geom_text(stat="count", aes(label=..count..), vjust =-0.2)+

  #Change the colour, size, font of the labels
  theme(
    #change x axis values view, angle at 90 to show label vertically
    axis.text.x =element_text(colour="black", size=9, face="bold",angle =
90),
    #set title colour, font size, bold and horizontal adjustment
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    #set of x and y labels axis colour, font size, bold and italic
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
    #disable legend
    legend.position = "none"
  )
```



```r
#Number Of Movies Produced by Studio Film
# use bar graph
ggplot(blockbuster, aes(x=studio, fill=studio))+geom_bar()+

  # set the label for x and y axis & title
  labs(fill="Studio",x="Studio", y="Number of Movies",title="Number of Mov
ies by Studio Film")+

  #show the count label at the top of the bar
  geom_text(stat="count", aes(label=..count..), vjust =-0.2)+

  #Change the colour, size, font of the labels
  theme(
    #set title colour, font size, bold and horizontal adjustment
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    #set of x and y labels axis colour, font size, bold and italic
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
    #remove x axis labels
    axis.text.x = element_blank(),
    #remove x axis ticks
    axis.ticks.x = element_blank(),
    #set legend text size
    legend.text = element_text(size = 8),
    #set legend box size
    legend.key.size = unit(0.2,"cm")
  )
```



```r
#Number Of Movies by MPAA Rating
# use bar graph
ggplot(blockbuster, aes(x=factor(rating), fill=rating))+geom_bar()+

  # set the label for x and y axis & title
  labs(fill="MPAA Rating", x="MPAA Rating",y="Number of Movies",title="Num
ber of Movies by MPAA rating")+

  #show the count label at the top of the bar
  geom_text(stat="count", aes(label=..count..), vjust =-0.2)+

  #Change the colour, size, font of the labels
  theme(
    #set title colour, font size, bold and horizontal adjustment
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    #set of x and y axis labels colour, font size, bold and italic
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
    #change x and y axis values colour, font size, bold
    axis.text = element_text(colour="black", size=12, face="bold"),
  )+
  #rename x axis values label
  scale_x_discrete(labels=c("G","PG","PG-13","R"))
```



```r
#---------------RESULT AND DISCUSSION-----------------------------------

#Command for Multiple Regression Model
Mlm.fit <- lm(worldwide_gross~rank_in_year+rating+year+imdb_rating+length,
data = blockbuster)
summary(Mlm.fit)
```

```
##
## Call:
## lm(formula = worldwide_gross ~ rank_in_year + rating + year +
##     imdb_rating + length, data = blockbuster)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -547708584  -87765141   18517336   79235949 1895971270
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.137e+10  1.486e+09 -21.107  < 2e-16 ***
## rank_in_year -3.899e+07  3.405e+06 -11.450  < 2e-16 ***
## rating       -3.195e+07  1.154e+07  -2.768  0.00589 **
## year          1.592e+07  7.506e+05  21.208  < 2e-16 ***
## imdb_rating   2.456e+06  1.206e+07   0.204  0.83873
## length        2.237e+06  4.679e+05   4.782 2.38e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 192500000 on 431 degrees of freedom
## Multiple R-squared:  0.6328, Adjusted R-squared:  0.6285
## F-statistic: 148.5 on 5 and 431 DF,  p-value: < 2.2e-16
```

```r
#Plotting graph of independent variables against dependent variable using
GGPLOT
library(ggplot2)

#1st independent variable (Rank_in_year)

#compute mean of value worldwide_gross by ranking of the movie using tappl
y function
mean_ranking=data.frame(value=tapply(worldwide_gross,rank_in_year,mean))
mean_ranking <- mean_ranking[,1]
mean_ranking
```

```
##  [1] 723705342 538173676 463302108 394280143 354994862 329417918 307757
032
##  [8] 288676218 266478298 248224669
```

```r
class(mean_ranking)
```

```
## [1] "array"
```

```r
#when we plot "mean_ranking" variable on y-axis, the values are in scienti
fic notation in the graph
#remove scientific notation and change it to decimal separator for more cl
ear visualization
mean_ranking=format(mean_ranking, big.mark = ",",scientific = F)
mean_ranking
```

```
##  [1] "723,705,342" "538,173,676" "463,302,108" "394,280,143" "354,994,8
62"
##  [6] "329,417,918" "307,757,032" "288,676,218" "266,478,298" "248,224,6
69"
```

```r
class(mean_ranking)
```

```
## [1] "array"
```

```r
#storing the label for x axis(ranking) into new variable
names_ranking<- names(tapply(worldwide_gross,rank_in_year,mean))
names_ranking
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```r
class(names_ranking)
```

```
## [1] "character"
```

```r
#change to integer type since rank is a ordinal variable
names_ranking <- as.integer(names_ranking)
names_ranking
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```r
class(names_ranking)
```

```
## [1] "integer"
```

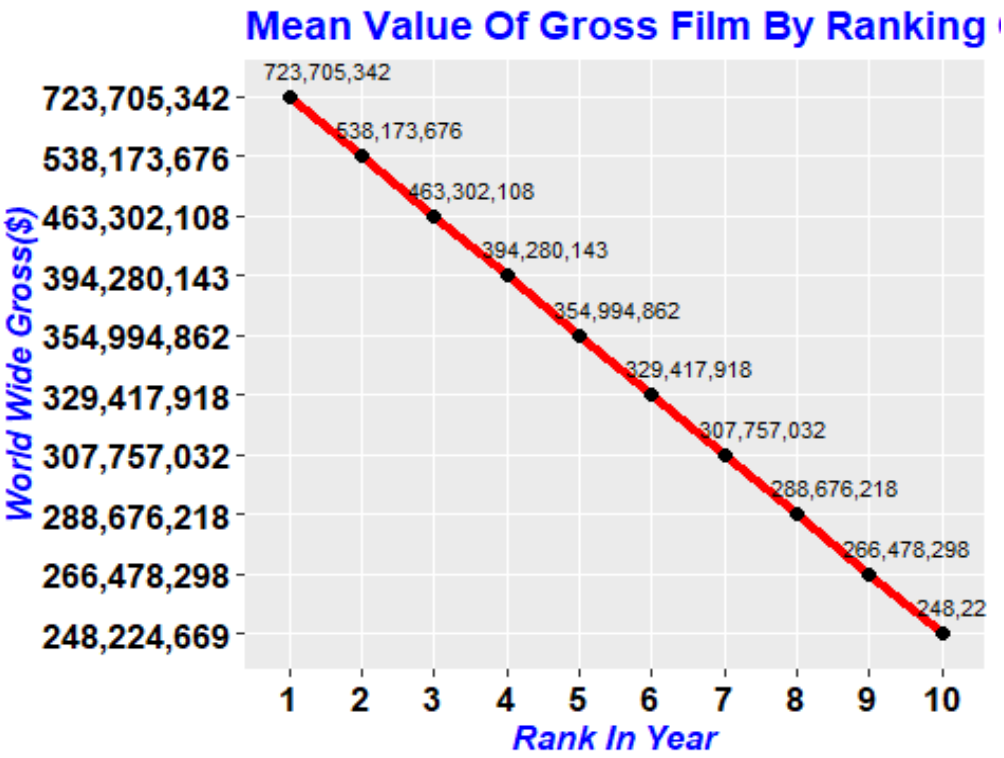
```r
#store "mean_ranking" and "names_ranking" into a new dataframe
df_ranking <- data.frame(names_ranking, mean_ranking)
df_ranking
```

```
##    names_ranking mean_ranking
## 1              1  723,705,342
## 2              2  538,173,676
## 3              3  463,302,108
## 4              4  394,280,143
## 5              5  354,994,862
## 6              6  329,417,918
## 7              7  307,757,032
## 8              8  288,676,218
## 9              9  266,478,298
## 10            10  248,224,669
```

```r
str(df_ranking)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ names_ranking: int  1 2 3 4 5 6 7 8 9 10
##  $ mean_ranking : chr [1:10(1d)] "723,705,342" "538,173,676" "463,302,1
08" "394,280,143" ...
```

```r
#plotting graph
#plotting line graph with size of width equal to "1" and colour of the lin
e set to "red"
#add points on lines

ggplot(df_ranking,aes(x=factor(names_ranking), y=mean_ranking, group=1)) +
geom_line(size=1.5,col="red")+geom_point(size=2)+

  #set the label for x, y axis and title
  labs(x="Rank In Year",y="World Wide Gross($)",title="Mean Value Of Gross
Film By Ranking Of The Movie")+

  #set theme
  theme_gray()+

  #label the values inside the graph
  geom_text(aes(label=mean_ranking),vjust=-1,hjust=0.2,size=3.0)+

  #Change colour, size, font of the labels
  theme(
    #change x and y axis values colour, font size, bold
    axis.text = element_text(colour="black", size=12, face="bold"),
    #set title colour, font size, bold and horizontal adjustment
    plot.title = element_text(color="blue", size=14, face="bold"),
    #set of x and y labels axis colour, font size, bold and italic
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
  )
```




```
#These are the steps we used to plot all the graphs using other independen
t variables. The only changes in commands occurs when the graph ploting di
ffers which is changing the command from (geom_line to geom_bar).
#2nd independent variable (Rating)
mean_rating=data.frame(value=tapply(worldwide_gross,rating,mean))
mean_rating <- mean_rating[,1]
mean_rating

## [1] 403894733 304686996 557422942 222234108

class(mean_rating)

## [1] "array"

mean_rating=format(mean_rating, big.mark = ",",scientific = F)
mean_rating

## [1] "403,894,733" "304,686,996" "557,422,942" "222,234,108"

class(mean_rating)

## [1] "array"

names_rating<- names(tapply(worldwide_gross,rating,mean))
names_rating

## [1] "1" "2" "3" "4"

class(names_rating)

## [1] "character"
```

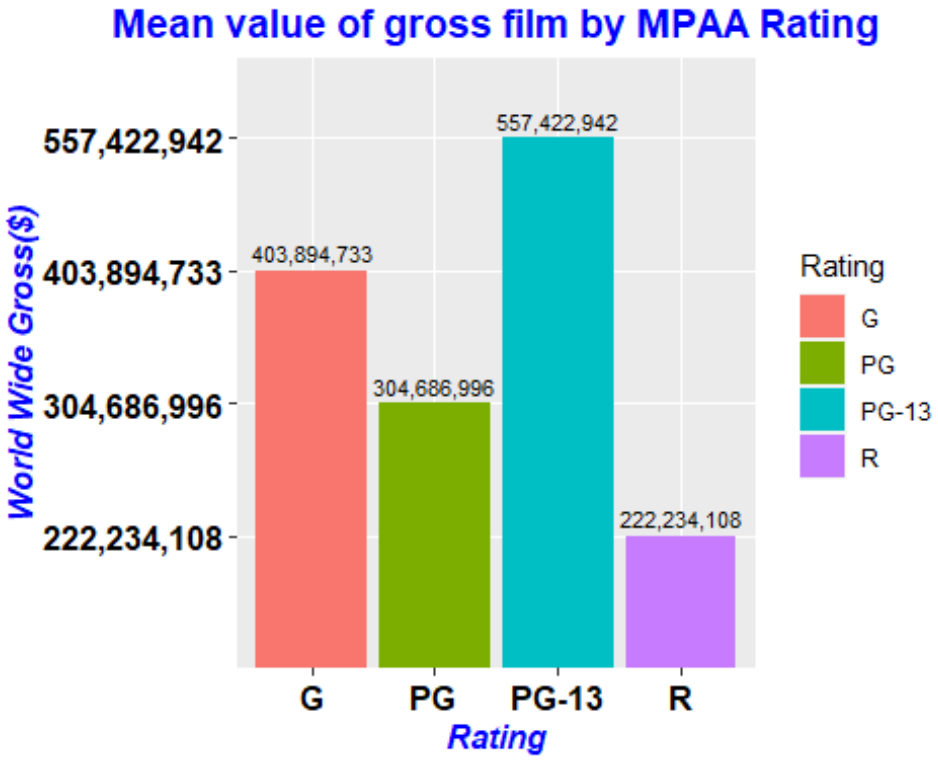
```r
df_rating <- data.frame(names_rating, mean_rating)
df_rating
```

```
##   names_rating mean_rating
## 1            1 403,894,733
## 2            2 304,686,996
## 3            3 557,422,942
## 4            4 222,234,108
```

```r
str(df_rating)
```

```
## 'data.frame':    4 obs. of  2 variables:
##  $ names_rating: Factor w/ 4 levels "1","2","3","4": 1 2 3 4
##  $ mean_rating : chr [1:4(1d)] "403,894,733" "304,686,996" "557,422,942
" "222,234,108"
```

```r
ggplot(df_rating, aes(x=factor(names_rating), y=mean_rating, fill=factor(n
ames_rating))) + geom_bar(stat="identity")+
  geom_text(aes(label=mean_rating),vjust=-0.5,size=3.0)+
  labs(fill="Rating",x="Rating",y="World Wide Gross($)",title="Mean value
of gross film by MPAA Rating")+
  theme(
    axis.text = element_text(colour="black", size=12, face="bold"),
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
  )+
  #rename legend values label with category name instead of integer
  scale_fill_discrete(labels=c("G","PG","PG-13","R"))+
  #rename x axis values label
  scale_x_discrete(labels=c("G","PG","PG-13","R"))
```



```r
#3rd independent variable(Year)

mean_year=data.frame(value=tapply(worldwide_gross,year,mean))
mean_year <- mean_year[,1]
mean_year
```

```
##  [1]  40966740  66420764  98085759  89444160  80860148  88737517  89921
036
##  [8] 125698036  95256174 129173927 102161795 112833598 102056691 110733
872
## [15] 276146112 324112385 277512821 308933826 346776840 380108731 325859
029
## [22] 384996139 513734582 367085213 480067972 374965891 527090730 566844
970
## [29] 586733335 603092517 585861736 575349588 695240613 636573773 909438
969
## [36] 757926658 792010032 925565523 884716153 788573049 400483675 378913
732
## [43] 380147731 378604549
```

```r
class(mean_year)
```

```
## [1] "array"
```

```r
names_year<- names(tapply(worldwide_gross,year,mean))
names_year
```

```
##  [1] "1975" "1976" "1977" "1978" "1979" "1980" "1981" "1982" "1983" "19
84"
## [11] "1985" "1986" "1987" "1988" "1989" "1990" "1991" "1992" "1993" "19
94"
## [21] "1995" "1996" "1997" "1998" "1999" "2000" "2001" "2002" "2003" "20
04"
## [31] "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012" "2013" "20
14"
## [41] "2015" "2016" "2017" "2018"
```

```r
class(names_year)
```

```
## [1] "character"
```

```r
names_year <- as.integer(names_year)
names_year
```

```
##  [1] 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1
988 1989
## [16] 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2
003 2004
## [31] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2
018
```

```r
class(names_year)
```

```
## [1] "integer"
```

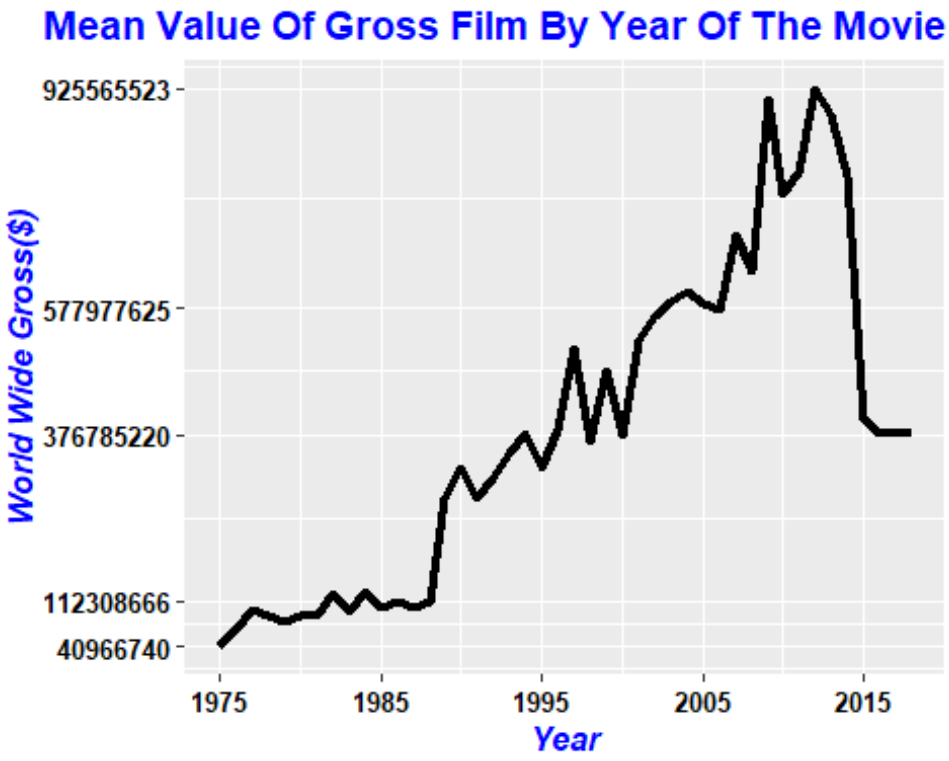
```r
df_year <- data.frame(names_year, mean_year)
df_year
```

```
##    names_year mean_year
## 1        1975  40966740
## 2        1976  66420764
## 3        1977  98085759
## 4        1978  89444160
## 5        1979  80860148
## 6        1980  88737517
## 7        1981  89921036
## 8        1982 125698036
## 9        1983  95256174
## 10       1984 129173927
## 11       1985 102161795
## 12       1986 112833598
## 13       1987 102056691
## 14       1988 110733872
## 15       1989 276146112
## 16       1990 324112385
## 17       1991 277512821
## 18       1992 308933826
## 19       1993 346776840
## 20       1994 380108731
## 21       1995 325859029
## 22       1996 384996139
## 23       1997 513734582
## 24       1998 367085213
## 25       1999 480067972
## 26       2000 374965891
## 27       2001 527090730
## 28       2002 566844970
## 29       2003 586733335
## 30       2004 603092517
## 31       2005 585861736
## 32       2006 575349588
## 33       2007 695240613
## 34       2008 636573773
## 35       2009 909438969
## 36       2010 757926658
## 37       2011 792010032
## 38       2012 925565523
## 39       2013 884716153
## 40       2014 788573049
## 41       2015 400483675
## 42       2016 378913732
## 43       2017 380147731
## 44       2018 378604549
```

```r
str(df_year)
```

```
## 'data.frame':    44 obs. of  2 variables:
##  $ names_year: int  1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 .
..
##  $ mean_year : num [1:44(1d)] 40966740 66420764 98085759 89444160 80860
148 ...
```

```r
ggplot(df_year,aes(x=names_year, y=mean_year, group=1)) +geom_line(size=1.
5)+
  labs(x="Year",y="World Wide Gross($)",title="Mean Value Of Gross Film By
Year Of The Movie")+
  theme_gray()+
  theme(
    axis.text = element_text(colour="black", size=10, face="bold"),
    plot.title = element_text(color="blue", size=14, face="bold", hjust=1)
,
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
  )+
  #set the breaks for x axis
  scale_x_continuous(limits=c(1975, 2018), breaks=c(1975, 1985, 1995, 2005
,2015))+
  #set the breaks for y axis according to the quantile
  scale_y_continuous(limits=c(min(mean_year),max(mean_year)), breaks=c(409
66740,112308666,376785220,577977625,925565523)) #quantiles
```




```r
#for y axis, the breaks set according to the quantile below
quantile(sort(mean_year))
```

```
##        0%       25%       50%       75%      100%
##  40966740 112308666 376785220 577977625 925565523
```

```r
#4th independent variable(Imdb_rating)

mean(imdb_rating)# find mean of imdb rating
```

```
## [1] 7.076659
```

```r
summary(imdb_rating)#find min and max of imdb rating
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.400   6.500   7.100   7.077   7.700   9.000
```

```r
#bining the continuous variable "imdb_rating" into certain values and set
labels
disImdb <- cut(imdb_rating, breaks = c(0,5,6,7,8,Inf), labels = c("< 5","5
-5.99","6-6.99","7-7.99","8 =>"))
disImdb
```

```
##   [1] 7-7.99 8 =>   7-7.99 6-6.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 8
=>
##  [11] 7-7.99 7-7.99 7-7.99 6-6.99 7-7.99 7-7.99 7-7.99 7-7.99 6-6.99 6-
6.99
##  [21] 7-7.99 7-7.99 7-7.99 6-6.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 7-
7.99
##  [31] 7-7.99 6-6.99 7-7.99 8 =>   7-7.99 6-6.99 6-6.99 7-7.99 6-6.99 6-
6.99
##  [41] 5-5.99 7-7.99 8 =>   7-7.99 6-6.99 8 =>   7-7.99 6-6.99 7-7.99 8
=>
##  [51] 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 7-
7.99
##  [61] 8 =>   7-7.99 8 =>   7-7.99 6-6.99 5-5.99 7-7.99 6-6.99 7-7.99 6-
6.99
##  [71] 8 =>   6-6.99 6-6.99 < 5    7-7.99 7-7.99 7-7.99 6-6.99 5-5.99 6-
6.99
##  [81] 8 =>   6-6.99 7-7.99 8 =>   6-6.99 < 5    7-7.99 7-7.99 7-7.99 8
=>
##  [91] 7-7.99 7-7.99 6-6.99 5-5.99 5-5.99 8 =>   < 5    7-7.99 6-6.99 7-
7.99
## [101] 8 =>   6-6.99 7-7.99 6-6.99 6-6.99 6-6.99 6-6.99 7-7.99 8 =>   6-
6.99
## [111] 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 7-7.99 7-7.99 7-7.99 6-6.99 7-
7.99
## [121] 7-7.99 6-6.99 6-6.99 7-7.99 6-6.99 7-7.99 6-6.99 6-6.99 6-6.99 6-
6.99
## [131] 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 6-6.99 6-6.99 6-6.99 8 =>   6-
6.99
## [141] 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 5-5.99 6-
6.99
## [151] 8 =>   8 =>   7-7.99 8 =>   6-6.99 7-7.99 6-6.99 6-6.99 7-7.99 6-
6.99
## [161] 8 =>   7-7.99 7-7.99 6-6.99 5-5.99 6-6.99 6-6.99 7-7.99 6-6.99 7-
7.99
## [171] 7-7.99 8 =>   8 =>   7-7.99 7-7.99 5-5.99 6-6.99 5-5.99 5-5.99 6-
6.99
## [181] 5-5.99 8 =>   7-7.99 6-6.99 6-6.99 5-5.99 6-6.99 6-6.99 7-7.99 6-
6.99
```

```
## [191] 6-6.99 8 =>   7-7.99 8 =>   7-7.99 6-6.99 6-6.99 6-6.99 8 =>   6-
6.99
## [201] 6-6.99 8 =>   5-5.99 7-7.99 7-7.99 6-6.99 7-7.99 5-5.99 7-7.99 6-
6.99
## [211] 7-7.99 6-6.99 7-7.99 6-6.99 6-6.99 7-7.99 6-6.99 6-6.99 7-7.99 7-
7.99
## [221] 6-6.99 6-6.99 7-7.99 7-7.99 6-6.99 5-5.99 6-6.99 5-5.99 7-7.99 5-
5.99
## [231] 7-7.99 8 =>   7-7.99 7-7.99 6-6.99 5-5.99 8 =>   5-5.99 6-6.99 6-
6.99
## [241] 8 =>   8 =>   7-7.99 6-6.99 7-7.99 < 5    7-7.99 7-7.99 7-7.99 6-
6.99
## [251] 7-7.99 6-6.99 7-7.99 8 =>   6-6.99 5-5.99 6-6.99 6-6.99 7-7.99 6-
6.99
## [261] 7-7.99 6-6.99 6-6.99 6-6.99 6-6.99 6-6.99 7-7.99 6-6.99 7-7.99 6-
6.99
## [271] 8 =>   6-6.99 7-7.99 6-6.99 8 =>   7-7.99 6-6.99 7-7.99 6-6.99 6-
6.99
## [281] 6-6.99 7-7.99 6-6.99 7-7.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 5-
5.99
## [291] 8 =>   7-7.99 7-7.99 5-5.99 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 7-
7.99
## [301] 7-7.99 7-7.99 6-6.99 7-7.99 5-5.99 5-5.99 8 =>   7-7.99 5-5.99 7-
7.99
## [311] 5-5.99 6-6.99 6-6.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 6-
6.99
## [321] 6-6.99 6-6.99 8 =>   5-5.99 7-7.99 6-6.99 8 =>   6-6.99 5-5.99 6-
6.99
## [331] 8 =>   6-6.99 6-6.99 7-7.99 7-7.99 6-6.99 5-5.99 7-7.99 7-7.99 6-
6.99
## [341] 7-7.99 7-7.99 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 8 =>   6-6.99 6-
6.99
## [351] 8 =>   7-7.99 6-6.99 7-7.99 7-7.99 6-6.99 6-6.99 < 5    6-6.99 6-
6.99
## [361] 7-7.99 7-7.99 6-6.99 6-6.99 6-6.99 7-7.99 6-6.99 7-7.99 5-5.99 6-
6.99
## [371] 8 =>   7-7.99 6-6.99 6-6.99 6-6.99 6-6.99 7-7.99 6-6.99 7-7.99 6-
6.99
## [381] 8 =>   6-6.99 6-6.99 7-7.99 5-5.99 5-5.99 7-7.99 5-5.99 5-5.99 7-
7.99
## [391] 7-7.99 6-6.99 7-7.99 8 =>   6-6.99 8 =>   5-5.99 7-7.99 6-6.99 7-
7.99
## [401] 7-7.99 7-7.99 7-7.99 6-6.99 6-6.99 6-6.99 5-5.99 7-7.99 6-6.99 8
=>
## [411] 8 =>   6-6.99 7-7.99 7-7.99 6-6.99 6-6.99 7-7.99 6-6.99 7-7.99 8
=>
## [421] 8 =>   6-6.99 6-6.99 7-7.99 7-7.99 < 5    5-5.99 6-6.99 6-6.99 6-
6.99
## [431] 8 =>   6-6.99 7-7.99 6-6.99 6-6.99 5-5.99 6-6.99
## Levels: < 5 5-5.99 6-6.99 7-7.99 8 =>
```

```r
table(disImdb) #frequency table for imdb_rating
```

```
## disImdb
##    < 5 5-5.99 6-6.99 7-7.99   8 =>
##      6     38    169    174     50
```

```r
mean_imdb_rating=data.frame(value=tapply(worldwide_gross,disImdb,mean))
mean_imdb_rating <- mean_imdb_rating[,1]
mean_imdb_rating
```

```
## [1] 430452424 294350338 324193114 439563114 509892503
```

```r
class(mean_imdb_rating)
```

```
## [1] "array"
```

```r
mean_imdb_rating=format(mean_imdb_rating, big.mark = ",",scientific = F)
mean_imdb_rating
```

```
## [1] "430,452,424" "294,350,338" "324,193,114" "439,563,114" "509,892,50
3"
```

```r
class(mean_imdb_rating)
```

```
## [1] "array"
```

```r
names_imdb_rating<- c("< 5","5-5.99","6-6.99","7-7.99","8 =>")
names_imdb_rating
```

```
## [1] "< 5"    "5-5.99" "6-6.99" "7-7.99" "8 =>"
```

```r
class(names_imdb_rating)
```

```
## [1] "character"
```

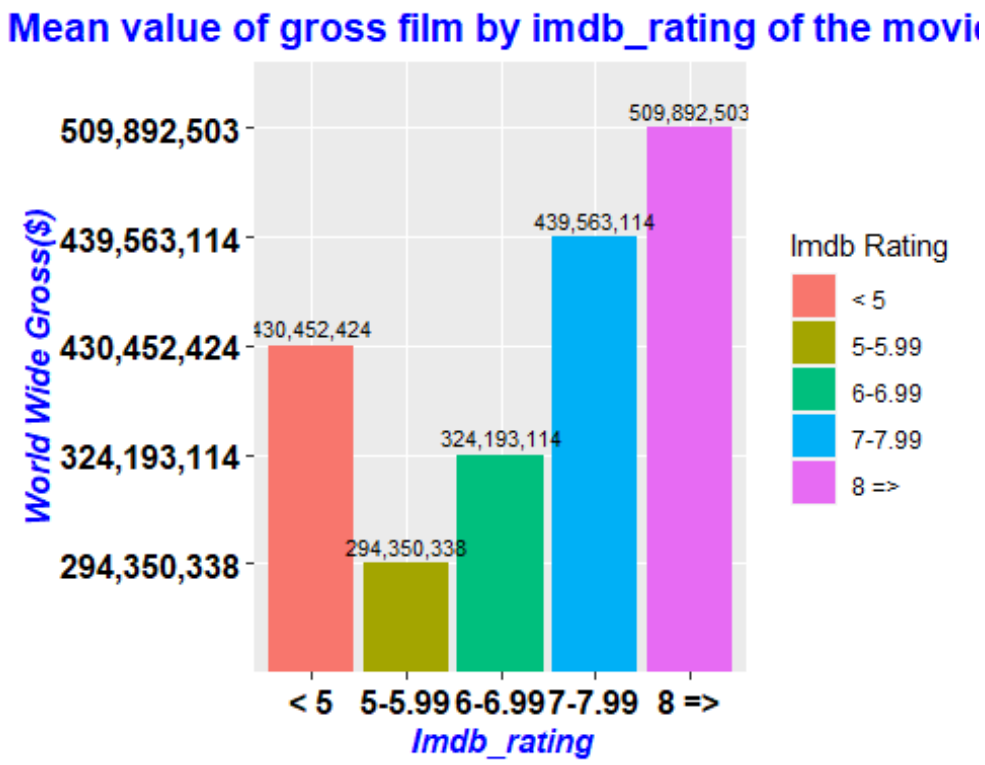
```r
df_imdb_rating <- data.frame(names_imdb_rating, mean_imdb_rating)
df_imdb_rating
```

```
##   names_imdb_rating mean_imdb_rating
## 1               < 5      430,452,424
## 2            5-5.99      294,350,338
## 3            6-6.99      324,193,114
## 4            7-7.99      439,563,114
## 5              8 =>      509,892,503
```

```r
str(df_imdb_rating)
```

```
## 'data.frame':    5 obs. of  2 variables:
##  $ names_imdb_rating: Factor w/ 5 levels "< 5","5-5.99",..: 1 2 3 4 5
##  $ mean_imdb_rating : chr [1:5(1d)] "430,452,424" "294,350,338" "324,19
3,114" "439,563,114" ...
```

```
ggplot(df_imdb_rating, aes(x=factor(names_imdb_rating), y=mean_imdb_rating
, fill=names_imdb_rating)) + geom_bar(stat="identity")+
  geom_text(aes(label=mean_imdb_rating),vjust=-0.5,size=3.0)+
  labs(fill="Imdb Rating",x="Imdb_rating",y="World Wide Gross($)",title="M
ean value of gross film by imdb_rating of the movie")+
  theme(
    axis.text = element_text(colour="black", size=12, face="bold"),
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
  )
```



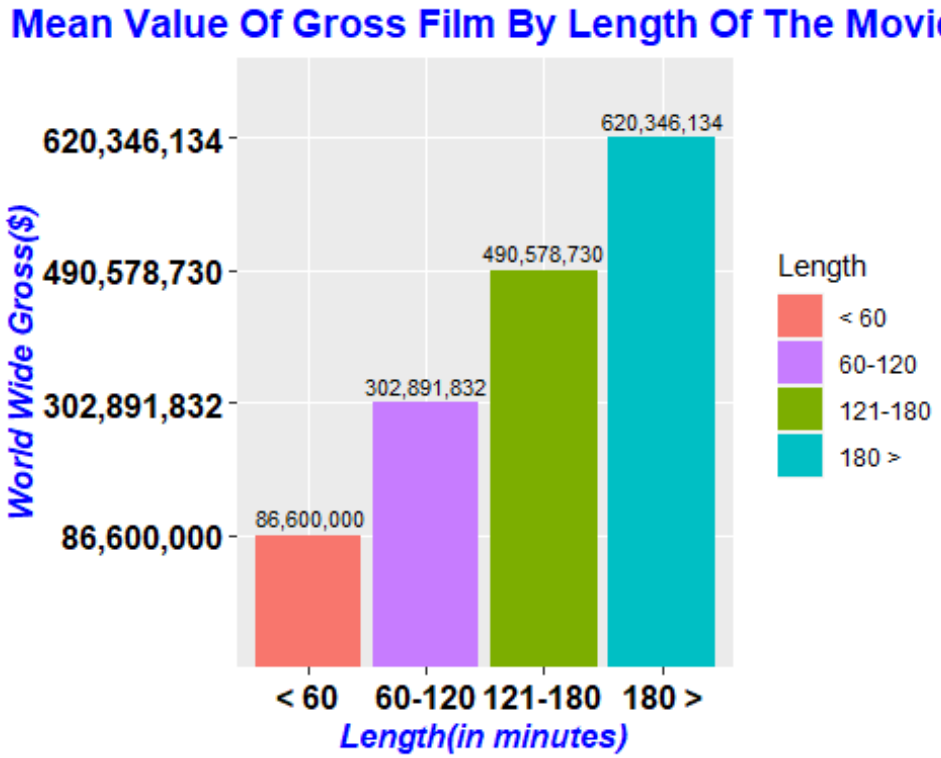

```
#5th independent variable(Length)

mean(length) #find mean length of movies

## [1] 119.8719

summary(length) #find min and max of imdb rating

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    27.0   103.0   118.0   119.9   134.0   201.0

#bin the continuous variable "length" and set labels
dislength <- cut(length, breaks = c(0,60,120,180,Inf), labels = c(("< 60",
"60-120","121-180","180 >"))
dislength

  [1] 121-180 121-180 60-120  121-180 60-120  121-180 60-120

  [8] 121-180 60-120  121-180 121-180 121-180 121-180 60-120
```

```
 [15] 121-180 121-180 121-180 121-180 60-120  60-120  121-180
 [22] 60-120  121-180 60-120  60-120  60-120  60-120  121-180
 [29] 121-180 60-120  121-180 121-180 121-180 60-120  121-180
 [36] 60-120  121-180 121-180 60-120  121-180 121-180 121-180
 [43] 121-180 60-120  121-180 121-180 121-180 121-180 121-180
 [50] 121-180 60-120  121-180 60-120  121-180 121-180 121-180
 [57] 60-120  60-120  121-180 60-120  121-180 121-180 121-180
 [64] 121-180 60-120  60-120  121-180 60-120  121-180 60-120
 [71] 121-180 121-180 121-180 60-120  121-180 60-120  121-180
 [78] 60-120  60-120  60-120  60-120  60-120  121-180 121-180
 [85] 60-120  121-180 121-180 60-120  60-120  60-120  121-180
 [92] 121-180 60-120  121-180 121-180 60-120  121-180 121-180
 [99] 121-180 60-120  121-180 121-180 60-120  60-120  60-120
[106] 60-120  60-120  121-180 60-120  121-180 121-180 121-180
[113] 121-180 60-120  121-180 60-120  60-120  60-120  121-180
[120] 60-120  121-180 121-180 60-120  121-180 60-120  60-120
[127] 60-120  121-180 121-180 60-120  121-180 121-180 121-180
[134] 60-120  180 >   60-120  60-120  60-120  121-180 60-120
[141] 60-120  121-180 121-180 60-120  121-180 121-180 60-120
[148] 121-180 60-120  121-180 180 >   60-120  121-180 121-180
[155] 60-120  121-180 60-120  121-180 121-180 121-180 121-180
[162] 121-180 121-180 121-180 60-120  121-180 60-120  60-120
[169] 60-120  121-180 121-180 121-180 60-120  60-120  60-120
[176] 180 >   121-180 60-120  60-120  121-180 121-180 121-180
[183] 121-180 121-180 60-120  60-120  60-120  121-180 60-120
[190] 121-180 121-180 60-120  60-120  121-180 60-120  121-180
[197] 121-180 121-180 121-180 60-120  121-180 121-180 121-180
[204] 60-120  60-120  60-120  60-120  60-120  121-180 121-180
[211] 180 >   121-180 60-120  60-120  121-180 121-180 60-120
[218] 60-120  121-180 60-120  121-180 60-120  60-120  121-180
```

```
[225] 60-120  60-120  121-180 60-120  121-180 60-120  121-180
[232] 60-120  121-180 121-180 60-120  121-180 121-180 60-120
[239] 121-180 60-120  60-120  121-180 121-180 60-120  60-120
[246] 60-120  60-120  60-120  121-180 121-180 121-180 121-180
[253] 121-180 180 >   121-180 60-120  60-120  60-120  121-180
[260] 121-180 60-120  121-180 60-120  121-180 60-120  121-180
[267] 121-180 60-120  121-180 60-120  121-180 121-180 60-120
[274] 121-180 60-120  180 >   60-120  121-180 60-120  60-120
[281] 121-180 60-120  60-120  180 >   60-120  60-120  121-180
[288] 121-180 60-120  60-120  121-180 121-180 60-120  60-120
[295] 121-180 60-120  60-120  60-120  60-120  121-180 121-180
[302] 60-120  60-120  60-120  60-120  60-120  121-180 60-120
[309] 60-120  60-120  60-120  60-120  60-120  121-180 60-120
[316] 60-120  60-120  60-120  60-120  60-120  60-120  60-120
[323] 60-120  60-120  60-120  60-120  121-180 60-120  60-120
[330] 60-120  60-120  60-120  60-120  121-180 121-180 60-120
[337] 60-120  60-120  60-120  60-120  60-120  60-120  60-120
[344] 60-120  121-180 60-120  60-120  60-120  60-120  60-120
[351] 121-180 121-180 60-120  60-120  60-120  121-180 60-120
[358] 60-120  60-120  60-120  60-120  60-120  121-180 60-120
[365] 60-120  60-120  60-120  60-120  60-120  121-180 60-120
[372] 60-120  121-180 60-120  60-120  60-120  60-120  121-180
[379] 60-120  60-120  121-180 60-120  60-120  60-120  60-120
[386] 60-120  121-180 60-120  60-120  121-180 60-120  60-120
[393] 60-120  121-180 121-180 60-120  121-180 60-120  121-180
[400] 60-120  60-120  121-180 60-120  60-120  60-120  60-120
[407] 60-120  121-180 60-120  180 >   121-180 60-120  121-180
[414] 60-120  60-120  60-120  121-180 121-180 121-180 60-120
[421] 60-120  < 60    121-180 121-180 60-120  60-120  121-180
[428] 60-120  60-120  121-180 121-180 60-120  60-120  121-180
```

```
[435] 60-120  60-120  60-120

Levels: < 60 60-120 121-180 180 >

> table(dislength) #frequency table for length

dislength

   < 60  60-120 121-180   180 >

      1     237     191       8

mean_length=data.frame(value=tapply(worldwide_gross,dislength,mean))
mean_length <- mean_length[,1]
mean_length

## [1]  86600000 302891832 490578730 620346134

class(mean_length)

## [1] "array"

mean_length=format(mean_length, big.mark = ",",scientific = F)
mean_length

## [1] " 86,600,000" "302,891,832" "490,578,730" "620,346,134"

class(mean_length)

## [1] "array"

names_length<- c("< 60","60-120","121-180","180 >")
names_length

## [1] "< 60"    "60-120"  "121-180" "180 >"

class(names_length)

## [1] "character"


df_length <- data.frame(names_length, mean_length)
df_length

##   names_length mean_length
## 1         < 60  86,600,000
## 2       60-120 302,891,832
## 3      121-180 490,578,730
## 4        180 > 620,346,134

str(df_length)

## 'data.frame':    4 obs. of  2 variables:
##  $ names_length: Factor w/ 4 levels "< 60","121-180",..: 1 4 2 3
##  $ mean_length : chr [1:4(1d)] " 86,600,000" "302,891,832" "490,578,730
" "620,346,134"
```

```
ggplot(df_length, aes(x=names_length, y=mean_length, fill=names_length)) +
geom_bar(stat="identity")+
  geom_text(aes(label=mean_length),vjust=-0.5,size=3.0)+
  labs(fill="Length",x="Length(in minutes)",y="World Wide Gross($)",title=
"Mean Value Of Gross Film By Length Of The Movie")+
  theme(
    axis.text = element_text(colour="black", size=12, face="bold"),
    plot.title = element_text(color="blue", size=14, face="bold",hjust = 0
.5),
    axis.title = element_text(color="blue", size=12, face="bold.italic"),
  )+
  #rename x axis values label
  scale_x_discrete(limits=c("< 60","60-120","121-180","180 >"))+
  #rename legend values label with category name
  scale_fill_discrete(breaks=c("< 60","60-120","121-180","180 >"))
```




```
#----------------------------END----------------------------------------
```