Data analyzation and visualization through Imdb movie dataset

ACKNOWLEDGEMENT

- We would like to express my thanks to our Professor **Mr. Awnish Kumar** for giving us a great opportunity to excel in our learning through this project.
- We have achieved a good amount of knowledge through the research and the help that we got from him.

HELLO GUY'S!!

We are here to present this auxillary project to you.

Done by Sagnik(20BCE7169) and Arindam(20BCE7104)

You can find us on instagram @madlybengalee and @arindam

1

Introduction

We all watch movies who doesn't. It is an art form unlike any other. The success of a movie depends on its gross surpassing the budget of the movie. In this project we will be estimating the gross of a movie relating to the various factors of the movie by using appropriate R queries

Data Description

The dataset is from Kaggle. It contains 28 variables for 5043 rows.

There are 2398 unique director names. 4917 unique movie titles

We are trying to predict gross while other attributes are predictors.

Kaggle Link-

https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset

Problem Statement

- Based on the massive movie information, it would be interesting to understand what are the important factors that make a movie more successful than others.
- So, we would like to analyze what kind of movies are grossing more and getting higher profits. We also want to show the results of this analysis in an intuitive way by visualizing outcome using ggplot2 in R.
- In this project, we take gross as response variable and focus on operating predictions by analyzing the rest of variables in the IMDB 5000 movie data.

LIBRARY:

For visualization: library(ggplot2)

For contains extra geoms for ggplot2:

library(ggrepel)

For visualization: library(ggthemes)

For data.frame: library(data.table)

For data manipulation: library(dplyr)

For character manipulation: library(stringr)

For read data: library(readr)

REMOVING DUPLICATES

- #checking duplicate rows
sum(duplicated(IMDB_Movies
))
- # delete duplicate rows
IMDB_Movies <-
IMDB_Movies[!duplicated(
IMDB_Movies),

- In this data, we will check for duplicate rows and delete them

- We have 4998 rows left.

Remove rows containing NA values

- complete.cases(IMDB_Movies)
  which(complete.cases(IMDB_Movies))
- no_NA <-
  which(!complete.cases(IMDB_Movies))
  no_NA
- # removing rows from dataset
  IMDB_Movies <- IMDB_Movies[-no_NA,]
- Now we have 3723 rows left in the dataset

Split genres

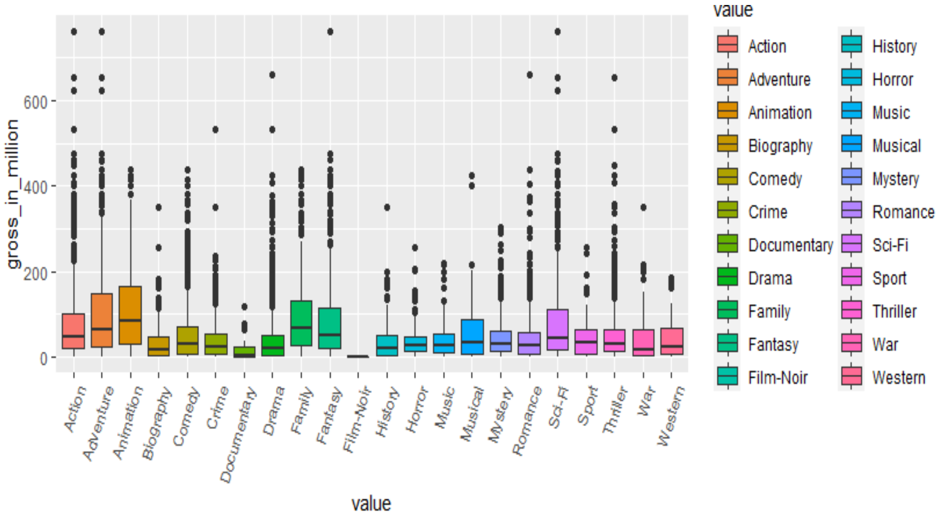
- We want to see relationship between genres and gross as one movie is having multiple genres .
- ```
  genre_type <- IMDB_Movies %>% select(movie_title,genres,containns('name'))
  genre_type <- data.frame(lapply(genre_type, as.character),          stringsAsFactors=FALSE)
  # Separating our Genre variable
  library(reshape)
  break_genre <-
  colsplit(IMDB_Movies$genres,split="\\|",names=c("n1","n2","n3","n4","n5","n6","n7","n8"))
  break_genre <- data.frame(lapply(break_genre, as.character), stringsAsFactors=FALSE)
  ```
  ```

Genre based movie gross from boxplot visualization

Gross vs Value

- ggplot(aes(y = gross_in_million, x = value, fill=value), data=genre_gross) + geom_boxplot() + theme(axis.text.x = element_text(angle=70,hjust=1)) like 1

#As we can see from the plot the Animation and Adventure genre movies are the highest grossing movies whereas Film-Noir and Documentary are the least grossing genre.



Data cleaning

```
> table(IMDB_Movies$aspect_ratio)

1.18 1.33 1.37  1.5 1.66 1.75 1.77 1.78 1.85    2  2.2 2.24 2.35 2.39  2.4 2.55 2.76   16
   1   18   48    1   39    2    1   34 1577    3   10    1 1969   11    3    1    3    1
> mean(IMDB_Movies$gross[IMDB_Movies$aspect_ratio == 1.85])
[1] 44705354
> mean(IMDB_Movies$gross[IMDB_Movies$aspect_ratio == 2.35])
[1] 58769780
> mean(IMDB_Movies$gross[IMDB_Movies$aspect_ratio != 1.85 & IMDB_Movies$aspect_ratio != 2.35])
[1] 51786710
> IMDB_Movies <- subset(IMDB_Movies, select = -c(aspect_ratio))
> #   assign 'PG' rating to 'M'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'M']   <- 'PG'
> #   assign 'PG' rating to 'GP'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'GP']  <- 'PG'
> #   assign 'NC-17' rating to 'X'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'X']   <- 'NC-17'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'Approved']  <- 'R'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'Not Rated'] <- 'R'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'Passed']    <- 'R'
> IMDB_Movies$content_rating[IMDB_Movies$content_rating == 'Unrated']   <- 'R'
> #   convert character to factor
> IMDB_Movies$content_rating <- factor(IMDB_Movies$content_rating)
> table(IMDB_Movies$content_rating)

   G NC-17    PG PG-13     R
  87    16   566  1291  1763
> IMDB_Movies <- IMDB_Movies %>% mutate(profit = gross - budget, return_on_investment_perc = (profit/budget)*100)
```

- From the means of gross for different aspect ratios, we can see there is not much difference.
  # For aspect ratio = 1.85, average gross is 44 Million$.
  # For aspect ratio = 2.35, average gross is 58 Million$.
  # Combining both ratios average is 51 Million$.

- According to the history of naming these different content ratings, we find M = GP = PG, X = NC-17. We want to replace M and GP with PG, replace X with NC-17, because these two are what we use nowadays.

- We want to replace "Approved", "Not Rated", "Passed", "Unrated" with the most common rating "R".

Add Columns

We have gross and budget information. So let's add two colums: profit and percentage return on investment for further analysis.

```
IMDB_Movies <- IMDB_Movies %>% mutate(profit = gross - budget,
return_on_investment_perc = (profit/budget)*100)
```

Remove Columns

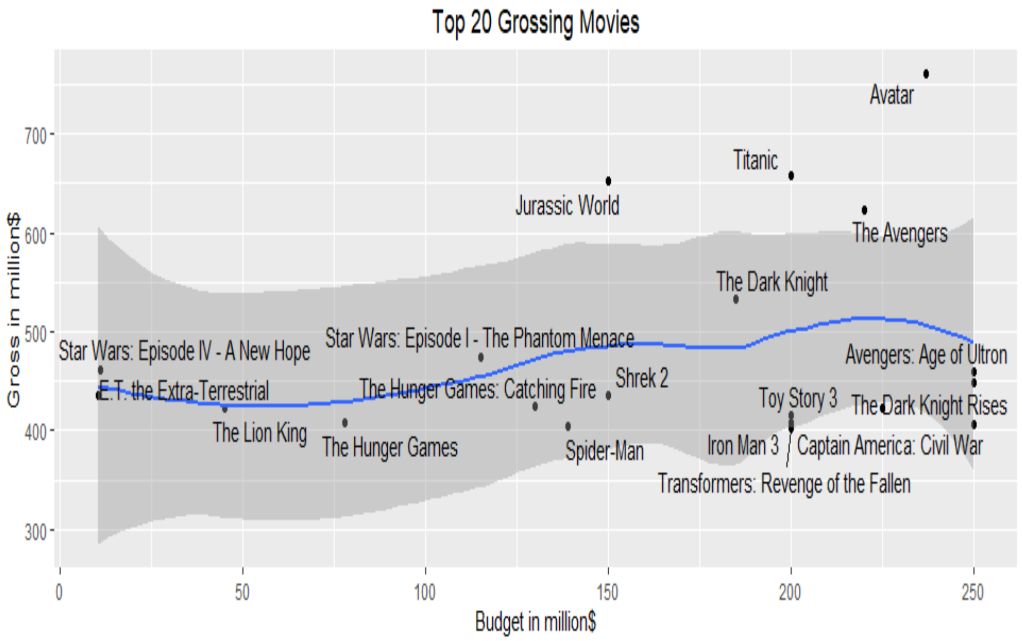
```
table(IMDB_Movies$color)
IMDB_Movies <- subset(IMDB_Movies, select = -c(color))
```

#More than 96% movies are colored, which indicates that this predictor is nearly constant.
Let's remove this predictor.

Data visualization

Top 20 grossing movies

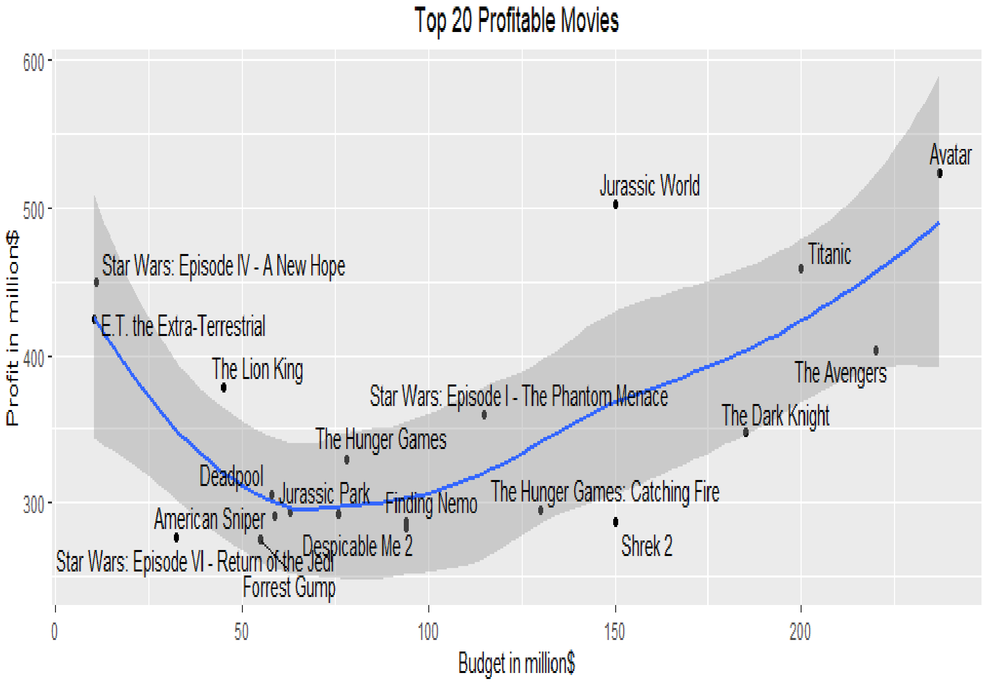
```
IMDB_Movies %>%
arrange(desc(gross_in_million)) %>%
top_n(20, gross) %>%
ggplot(aes(x=budget/1000000,
y=gross_in_million)) + geom_point() +
geom_smooth() +
geom_text_repel(aes(label=movie_title)) +
labs(x = "Budget in million$", y = "Gross in
million$", title = "Top 20 Grossing Movies")
+
theme(plot.title = element_text(hjust = 0.5))
```



Data visualization

- Top 20 profitable movies

IMDB_Movies %>% arrange(desc(profit)) %>% top_n(20, profit) %>% ggplot(aes(x=budget/1000000, y=profit/1000000)) + geom_point() + geom_smooth() + geom_text_repel(aes(label=movie_title)) + labs(x = "Budget in million$", y = "Profit in million$", title = "Top 20 Profitable Movies") + theme(plot.title = element_text(hjust = 0.5))

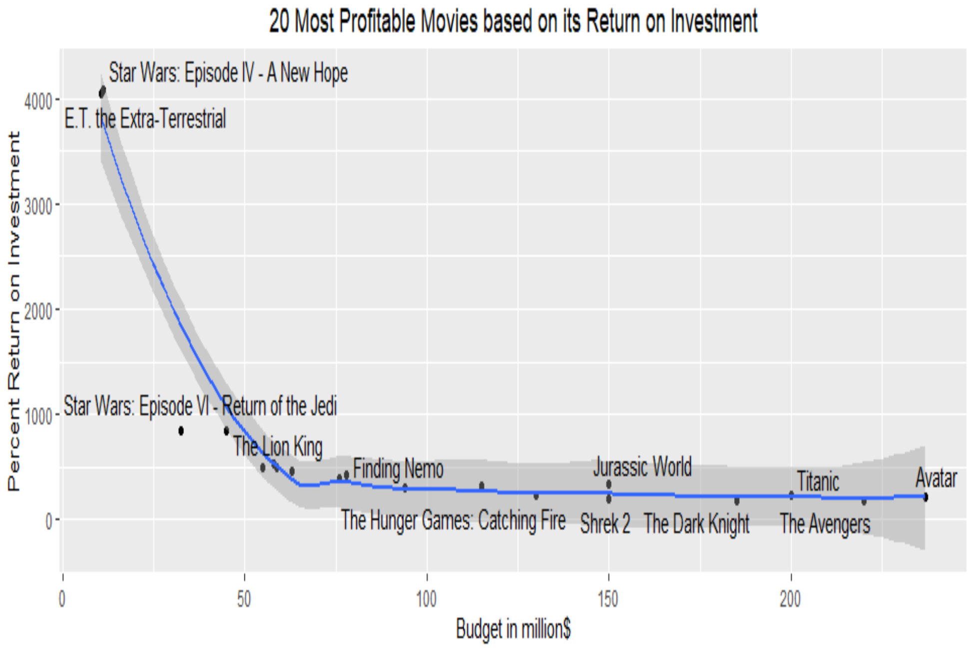


Data visualization

- Top 20 movies on its Return on Investment

```
IMDB_Movies %>% arrange(desc(profit)) %>% top_n(20, profit) %>% ggplot(aes(x=budget/1000000, y = return_on_investment_perc)) + geom_point() + geom_smooth() + geom_text_repel(aes(label = movie_title)) + labs(x = "Budget in million$", y = "Percent Return on Investment", title = "20 Most Profitable Movies based on its Return on Investment") + theme(plot.title = element_text(hjust = 0.5))
```

- # These are top 20 movies based on the percentage return on investment. We conclude from this plot that movies # made on a high budget are low on returns percentage.



Data visualization

Top 20 directors with highest grossing movies

```
library(formattable)
IMDB_Movies %>%
group_by(director_name) %>%
summarise(Average_Gross =
mean(gross_in_million)) %>%
arrange(desc(Average_Gross)) %>%
top_n(20, Average_Gross) %>%
formattable(list(Average_Gross =
color_bar("orange")), align = 'l')
```

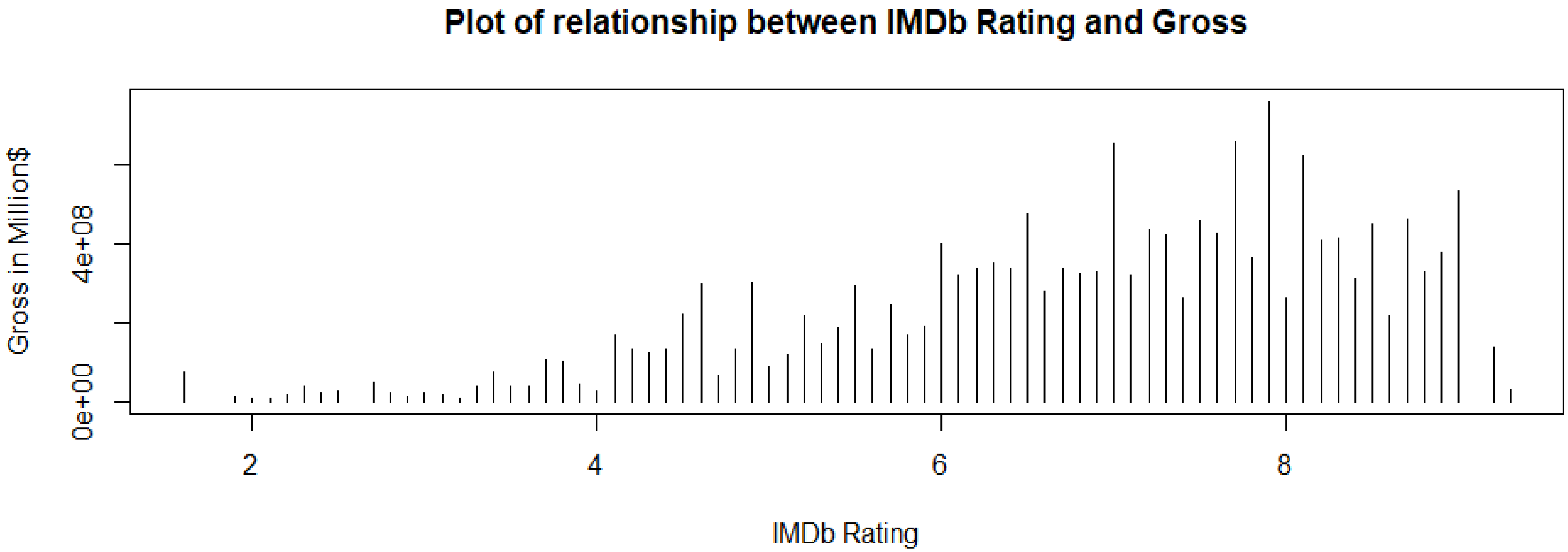
director_name	Average_Gross
Lee Unkrich	414.9845
Chris Buck	400.7366
Joss Whedon	369.2024
Tim Miller	363.0243
George Lucas	348.2837
Kyle Balda	336.0296
Colin Trevorrow	328.0925
Yarrow Cheney	323.5055
Pete Docter	313.1138
Pierre Coffin	309.7756

Effect of imdb_score on gross

```
plot(IMDB_Movies$imdb_score,IMDB_Movies$gross,xlab="IMDb Rating", ylab="Gross in Million$", type =
"h",
main = "Plot of relationship between IMDb Rating and
Gross")
```

This is an analysis on Average Gross earnings by movies for a particular imdb score.
The highest grossing movies are int the range of imdb_score range of 6-9.

VISUALIZATION OF IMDB SCORE VS GROSS



THANK YOU!!!

-By Arindam and Sagnik

20BCE7104 and
20BCE7169