PES University, Bangalore Established under the Karnataka Act No. 16 of 2013

UE21CS342AA2 - Data Analytics

Worksheet 1a : Introduction to R and Exploratory Data Analysis

Richa Shahi - shahiricha2412@gmail.com , Abhay K Iyengar – abzee2002@gmail.com

Exploring Data with R

Prerequisites

This worksheet aims to develop your understanding of summary statistics and basic visualizations through a pragmatic approach. You can download the dataset from here.

Resources

- Check out this beautifully comprehensive resource for everything you need to get started with R.
- This online book provides guided explananations about visualizations in R using the ggplot2 library.

Use the following libraries and read the dataset:

```r
char_preds <- read.csv('movie_dataset.csv')
```

About the Data

To make this worksheet interesting for you all, we have picked this dataset from Kaggle which comprises of the Movies and the metadata associated with it collected using The Movie Database (TMDB). You can download the dataset from here. This dataset is the subset of this Kaggle dataset.

Data Dictionary

```
title - Name or Title of the movie.
budget - The budget of the movie in American Dollar(USD).
genres - The genres for  the entire movie.
id - The identifier for the movie in The Movie Database(TMDB).
original_language - The language associated with the original version of the
                    film.
popularity - Lifetime popularity score of a movie that is impacted by attributes
             like number of votes, number of views, etc.
release_date - The release date of the movie.
revenue - The revenue generated by the movie in American Dollar(USD).
runtime - The duration of the movie in minutes.
```

vote_average - The average of all the votes on the scale of 10.
vote_count - The number of votes for a movie.
director - The director of the movie.

Assignment Submission Format

The following problems are to be completed using the R programming language and should be submitted as a R markdown file (.rmd). Since the dataset is public and many of you students will have the same numerical answers, the grades are allocated on the analysis of the problems and personalized answers within the conclusion section.

Preliminary Guided Exercises

Make sure you have the R programming language installed on your system. It is also recommended to make sure RStudio, the popular IDE for R, is installed. RStudio provides a lot of useful functionality like R markdown, a script editor and GitHub integration. Use RStudio Projects as a great way of keeping each week's assignment work organized.

1. **Data Import**

   To import data from CSV files into a DataFrame:

```
data <- read.csv('movie_dataset.csv', header=TRUE)
```

The header = TRUE argument specifies that the first row of your data contains the variable names. If thi is not the case you can specify header = FALSE (this is the default value so you can omit this argument entirely).

2. **Compact Summary**

   Use the str() function to return a compact and informative summary of the DataFrame.

```
str(data)
```

```
## 'data.frame':    4041 obs. of  12 variables:
##  $ budget          : num  2.37e+08 3.00e+08 2.45e+08 2.50e+08 2.60e+08 2.58e+08 2.60e+08 2.80e+08 2
##  $ genres          : chr  "Action Adventure Fantasy Science-Fiction" "Adventure Fantasy Action" "Ac
##  $ id              : int  19995 285 206647 49026 49529 559 38757 99861 767 209112 ...
##  $ original_language: chr  "en" "en" "en" "en" ...
##  $ popularity      : num  150.4 139.1 107.4 112.3 43.9 ...
##  $ release_date    : chr  "10-12-2009" "19-05-2007" "26-10-2015" "16-07-2012" ...
##  $ revenue         : num  2.79e+09 9.61e+08 8.81e+08 1.08e+09 2.84e+08 ...
##  $ runtime         : num  162 169 148 165 132 139 100 141 153 151 ...
##  $ title           : chr  "Avatar" "Pirates of the Caribbean: At World's End" "Spectre" "The Dark Kn
##  $ vote_average    : num  7.2 6.9 6.3 7.6 6.1 5.9 7.4 7.3 7.4 5.7 ...
##  $ vote_count      : int  11800 4500 4466 9106 2124 3576 3330 6767 5293 7004 ...
##  $ director        : chr  "James Cameron" "Gore Verbinski" "Sam Mendes" "Christopher Nolan" ...
```

Here we see that data is a 'data.frame' object which contains 4041 rows and 12 variables (columns). Each the variables are listed along with their data class and the first 10 values.

3. **Summary Statistics**

   To access the data in any of the variables (columns) in our data frame we can use the $ notation. Indexing in R starts at 1, which means the first element is at index 1. Access the first 10 values of the title column:

```r
data$title[1:10]
```

```
##  [1] "Avatar"
##  [2] "Pirates of the Caribbean: At World’s End"
##  [3] "Spectre"
##  [4] "The Dark Knight Rises"
##  [5] "John Carter"
##  [6] "Spider-Man 3"
##  [7] "Tangled"
##  [8] "Avengers: Age of Ultron"
##  [9] "Harry Potter and the Half-Blood Prince"
## [10] "Batman v Superman: Dawn of Justice"
```

```
We can assign a column to another variable and calculate a mean of a numeric variable or get a summary
a variable using the summary() function.
```

```r
movie_names <- data$title
summary(movie_names)
```

```
##    Length     Class      Mode
##      4041 character character
```

```r
movie_budget <- data$budget
mean(movie_budget)
```

```
## [1] 32853716
```

```r
summary(movie_budget)
```

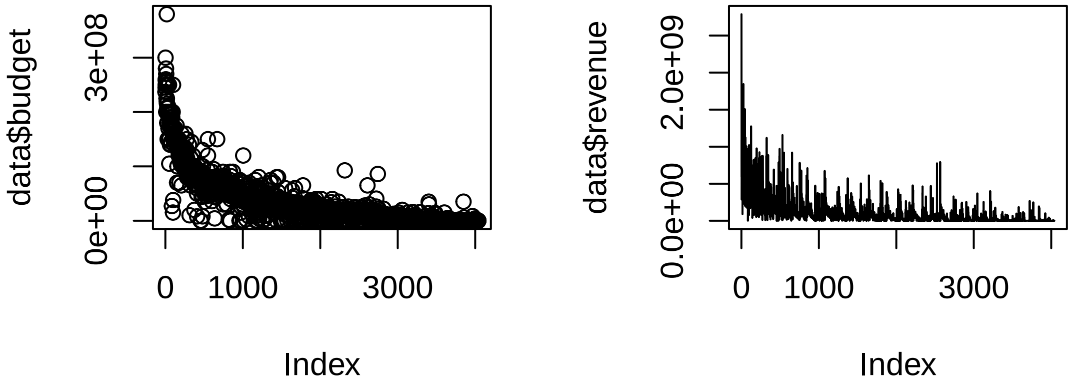
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         0   3000000  18000000  32853716  45000000 380000000
```

Notice how the behavior of the summary function changes with different types of variables. Let’s now try to explore how we can visualize our data.

4. **Scatter Plots and Line Plots**

   The most common high level function used to produce plots in R is the plot function.

```r
par(mfrow = c(1,2)) # To plot different plots in the same row

plot(data$budget, type="p") # scatter plot
plot(data$revenue, type="l") # line plot
```



```
The horizontal axis in these scatter plots represents the index or the row number of data point.
```

5. **Sorting a data frame**

   To sort a dataframe with respect to a column we can use the order() function. Let us sort the dataframe to get the top 10 highest grossing movies.

```
sorted_data <- data[order(data$revenue, decreasing = TRUE), ] # To sort in descending order

# The head function is used to get the first 10 rows
top_10_rows <- head(sorted_data, n = 10)
```

6. **Column Transformation**

   Highest Revenue might not be the right indicator for a successful movie. So lets plot the ROI (Return on Investment for all movies)

   ROI = Net Return/Cost of Investment

```
data$ROI = data$revenue / data$budget

# Print the first 5 rows with their title and ROI
data[1:5, c("title", "ROI")]
```

```
##                                       title       ROI
## 1                                    Avatar 11.763566
## 2 Pirates of the Caribbean: At World's End  3.203333
## 3                                   Spectre  3.594590
## 4                     The Dark Knight Rises  4.339756
## 5                               John Carter  1.092843
```

```
Next, you can sort the data frame with respect to ROI to get movies with highest returns.
```

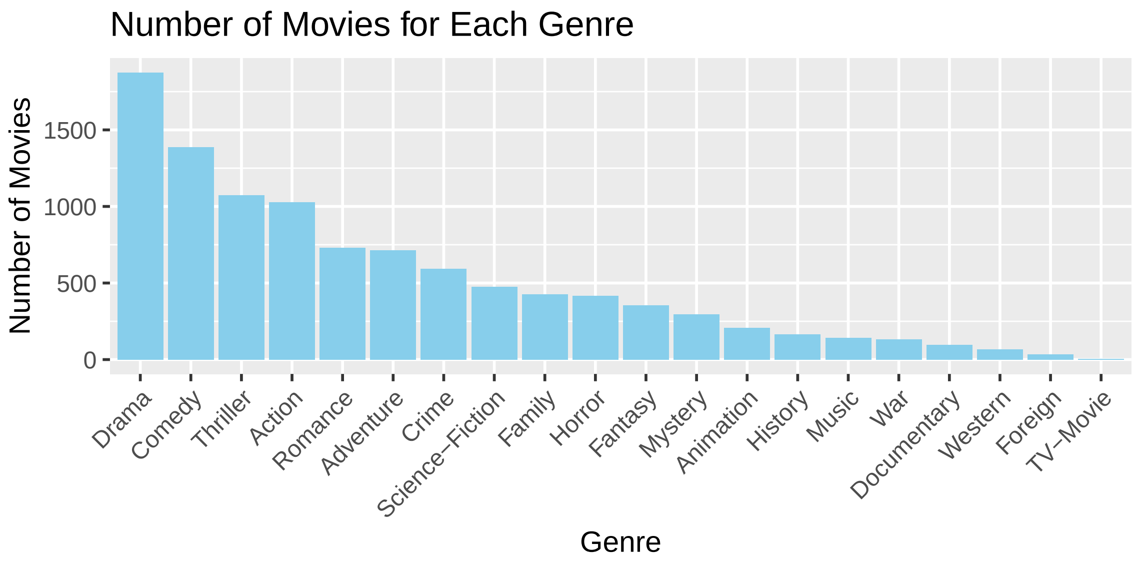
7. **Data Pre-processing**

   A lot of times real-world datasets are not curated and cleaned. Values are not stored in proper formats and hence requires cleaning and appropriate transformation before the data is suitable for analysis. In our case we see that the genre is stored as a string. Lets us split the string to get all genre labels.

```r
# Convert the space-separated string of genres to a list of genres for each movie
data$genres <- strsplit(data$genres, " ")

# Extract the individual genres and count their occurrences
label_counts <- table(unlist(data$genres)) # label_counts will be a data frame with "Var1" and "Freq"

# Sort the counts in descending order
label_counts <- sort(label_counts, decreasing = TRUE)
```

8. **Using the ggplot2 Library**

```r
# Load the ggplot2 library
library(ggplot2)

# Make sure you have label_counts data frame with "Var1" and "Freq" columns

# Convert the table object to a data frame
label_counts_df <- as.data.frame(label_counts)

# Plot the bar chart using ggplot2
ggplot(label_counts_df, aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(x = "Genre", y = "Number of Movies", title = "Number of Movies for Each Genre") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
Nearly half of the movies in the dataset are tagged as Drama. Also note that
one movie could belong to multiple genres.
```

Problems

The problems for this part of the worksheet are:

- *Problem 1*
- *Problem 2*
- *Problem 3*
- *Problem 4*
- *Problem 5*
- *Problem 6*

Problems

Problem 1

Get the summary statistics (mean, median, min, max, 1st quartile, 3rd quartile and standard deviation). Calculate these only for the numerical columns. What can you determine from the summary statistics? What summary statistics can be useful for categorical columns? Classify all the variables/columns into their types of data attributes (nominal, ordinal, interval, ratio).

SOLUTION

Numerical columns are budget, popularity, revenue, runtime and vote_count.

```
data <- read.csv("movie_dataset.csv")
numerical_columns <- c("budget", "popularity", "revenue", "runtime", "vote_count")

# Get summary statistics for the specified columns
summary_statistics <- summary(data[, numerical_columns])

# Print the result
print(summary_statistics)
```

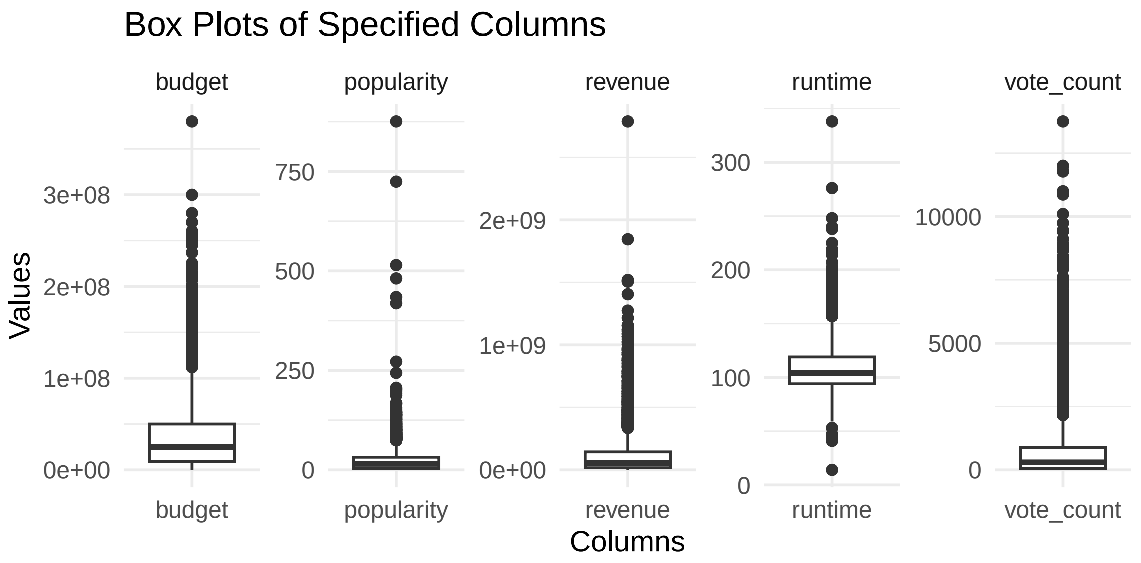
```
##      budget            popularity         revenue              runtime
##  Min.   :        0   Min.   :  0.000   Min.   :0.000e+00   Min.   :  0.0
##  1st Qu.:  3000000   1st Qu.:  3.674   1st Qu.:2.155e+06   1st Qu.: 94.0
##  Median : 18000000   Median : 15.132   Median :3.265e+07   Median :104.0
##  Mean   : 32853716   Mean   : 23.492   Mean   :9.695e+07   Mean   :107.5
##  3rd Qu.: 45000000   3rd Qu.: 31.894   3rd Qu.:1.134e+08   3rd Qu.:119.0
##  Max.   :380000000   Max.   :875.581   Max.   :2.788e+09   Max.   :338.0
##    vote_count
##  Min.   :    0.0
##  1st Qu.:   49.0
##  Median :  299.0
##  Mean   :  784.7
##  3rd Qu.:  890.0
##  Max.   :13752.0
```

We see that the budget, revenue and runtime are zero for movies, which cannot be possible in a real-world scenario. Many times unknown or unavailable values are replaced with default markers (in this case 0).This is a case of unavailability of data. Hence lets subsitute NA values in these column for a zero values.

```
data$budget <- ifelse(data$budget == 0, NA, data$budget)
data$revenue <- ifelse(data$revenue == 0, NA, data$revenue)
data$runtime <- ifelse(data$runtime == 0, NA, data$runtime)
```

```
specified_columns <- c("budget","popularity","revenue", "runtime", "vote_count")
library(ggplot2)
library(tidyr)


selected_data <- data[, specified_columns]

# Reshape the data for ggplot2
melted_data <- stack(selected_data)

# Create the box plots using ggplot2 with facet_wrap
plot <- ggplot(melted_data, aes(x = ind, y = values)) +
  geom_boxplot() +
  labs(x = "Columns", y = "Values", title = "Box Plots of Specified Columns") +
  theme_minimal() +
  facet_wrap(~ind, scales = "free", nrow = 1)

# Display the plot
print(plot)
```

```
## Warning: Removed 1414 rows containing non-finite values (‘stat_boxplot()‘).
```



The columns budget, popularity, revenue and vote_count are left-skewed. We can also see the presence of a lot of outliers in the dataset.

Only mode is useful for categorical columns - original_language and director.

```
# Function to calculate mode of a vector
calculate_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}


mode_lang <- calculate_mode(data$original_language)
print(mode_lang)
```

```
## [1] "en"
```

All the columns should be classified as follows - title - Nominal budget - Ratio genres - Nominal id - Nominal original_language - Nominal popularity - Ratio release_date - Nominal revenue - Ratio runtime - Ratio vote_average - Ordinal vote_count - Ratio director - Nominal

Problem 2

Investigate the data set for missing values. Also classify the missingness as MCAR, MAR or MNAR. Recommend ways to replace missing values in the dataset and apply them for revenue, budget and runtime columns.

Hint: Make sure to capture data from both, missing values in numeric fields and empty strings in descriptive fields. Convert all missing placeholders to type NA. Look at the distribution of the dataset to classify the type of missing values.

SOLUTION:

In the last problem we captured missing values in runtime, budget and revenue columns. Lets print the number of missing values.

```
missing_values_count <- colSums(is.na(data))

# Print the result
print(missing_values_count)
```

```
##            budget            genres                id original_language
##               599                 0                 0                 0
##        popularity      release_date           revenue           runtime
##                 0                 0               778                37
##             title      vote_average        vote_count          director
##                 0                 0                 0                 0
```

Release date can be replaced with empty string.

Runtime can be replaced with median of runtime for movies. Missingness for Runtime can be classified as MCAR. We can replace missing values by median of runtime as there are outliers in runtime.

```
# Calculate the median runtime
runtime_median <- median(data$runtime, na.rm = TRUE)

# Replace the values in the runtime with the calculated median
data$runtime <- ifelse(is.na(data$runtime), runtime_median, data$runtime)
```

For budget and revenue we see that data is missing for movies with lower popularity (popularity<4). This type of missingness can be classified as MAR.

Let us drop the videos where budget and revenue are missing as replacing with any summary statistics might interfere with further analysis. You might choose to deal with missing values by the ones of the ways recommended in the course.

```
# Drop rows with NA values in 'budget'
data <- data[!is.na(data$budget), ]
# Drop rows with NA values in 'revenue'
data <- data[!is.na(data$revenue), ]
```

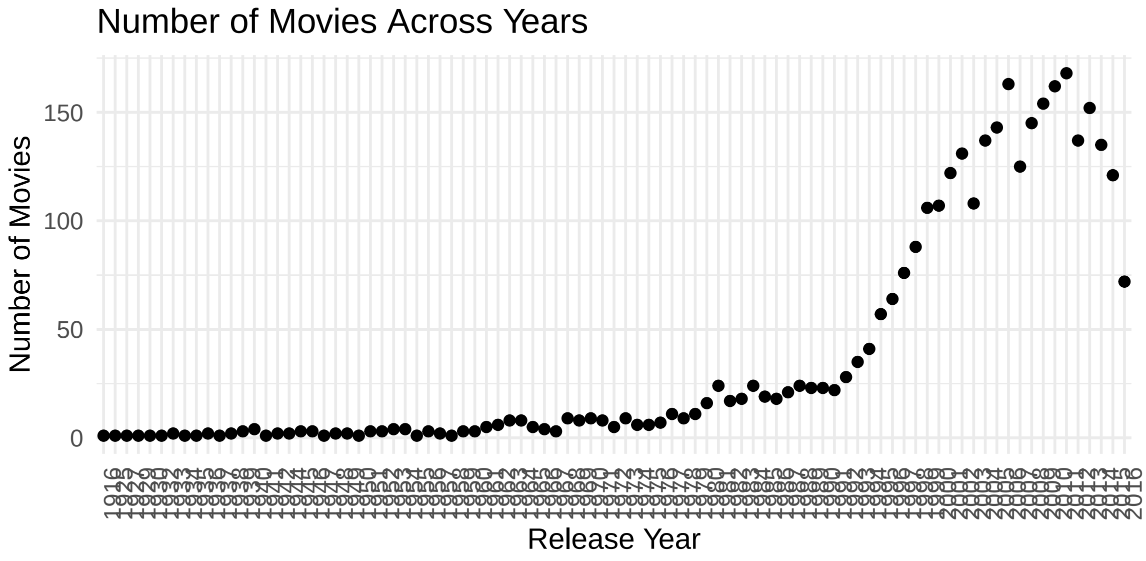
Problem 3

Analyze the spread of the data set along years. How number of movie releases have changed over the years ?

```
# Convert the "release-date" column to a date format
data$release_date <- as.Date(data$release_date, format = "%d-%m-%Y")

# Create the "release_year" column by extracting the year from the "release-date" column
data$release_year <- format(data$release_date, "%Y")

head(data, n = 10)
```

```
##      budget                                   genres     id original_language
## 1  2.37e+08 Action Adventure Fantasy Science-Fiction  19995                en
## 2  3.00e+08                 Adventure Fantasy Action    285                en
## 3  2.45e+08                   Action Adventure Crime 206647                en
## 4  2.50e+08              Action Crime Drama Thriller  49026                en
## 5  2.60e+08         Action Adventure Science-Fiction  49529                en
## 6  2.58e+08                 Fantasy Action Adventure    559                en
## 7  2.60e+08                         Animation Family  38757                en
## 8  2.80e+08         Action Adventure Science-Fiction  99861                en
## 9  2.50e+08                 Adventure Fantasy Family    767                en
## 10 2.50e+08                 Action Adventure Fantasy 209112                en
##    popularity release_date    revenue runtime
## 1   150.43758   2009-12-10 2787965087     162
## 2   139.08262   2007-05-19  961000000     169
## 3   107.37679   2015-10-26  880674609     148
## 4   112.31295   2012-07-16 1084939099     165
## 5    43.92699   2012-03-07  284139100     132
## 6   115.69981   2007-05-01  890871626     139
## 7    48.68197   2010-11-24  591794936     100
## 8   134.27923   2015-04-22 1405403694     141
## 9    98.88564   2009-07-07  933959197     153
## 10  155.79045   2016-03-23  873260194     151
##                                       title vote_average vote_count
## 1                                    Avatar          7.2      11800
## 2  Pirates of the Caribbean: At World’s End          6.9       4500
## 3                                   Spectre          6.3       4466
## 4                     The Dark Knight Rises          7.6       9106
## 5                               John Carter          6.1       2124
## 6                              Spider-Man 3          5.9       3576
## 7                                   Tangled          7.4       3330
## 8                   Avengers: Age of Ultron          7.3       6767
## 9    Harry Potter and the Half-Blood Prince          7.4       5293
## 10       Batman v Superman: Dawn of Justice          5.7       7004
##              director release_year
## 1      James Cameron         2009
## 2     Gore Verbinski         2007
## 3         Sam Mendes         2015
## 4  Christopher Nolan         2012
## 5     Andrew Stanton         2012
## 6          Sam Raimi         2007
## 7       Byron Howard         2010
## 8        Joss Whedon         2015
## 9        David Yates         2009
## 10       Zack Snyder         2016
```

```r
# Count the number of movies released in each year and convert to a data frame
movie_counts <- as.data.frame(table(data$release_year))
names(movie_counts) <- c("year", "count")

# Sort the data frame by the "year" column
movie_counts <- movie_counts[order(movie_counts$year), ]

# Create the scatter plot using ggplot2 with x-axis labels rotated by 90 degrees
library(ggplot2)

plot <- ggplot(movie_counts, aes(x = year, y = count)) +
  geom_point() +
  labs(x = "Release Year", y = "Number of Movies", title = "Number of Movies Across Years") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# Display the plot
print(plot)
```



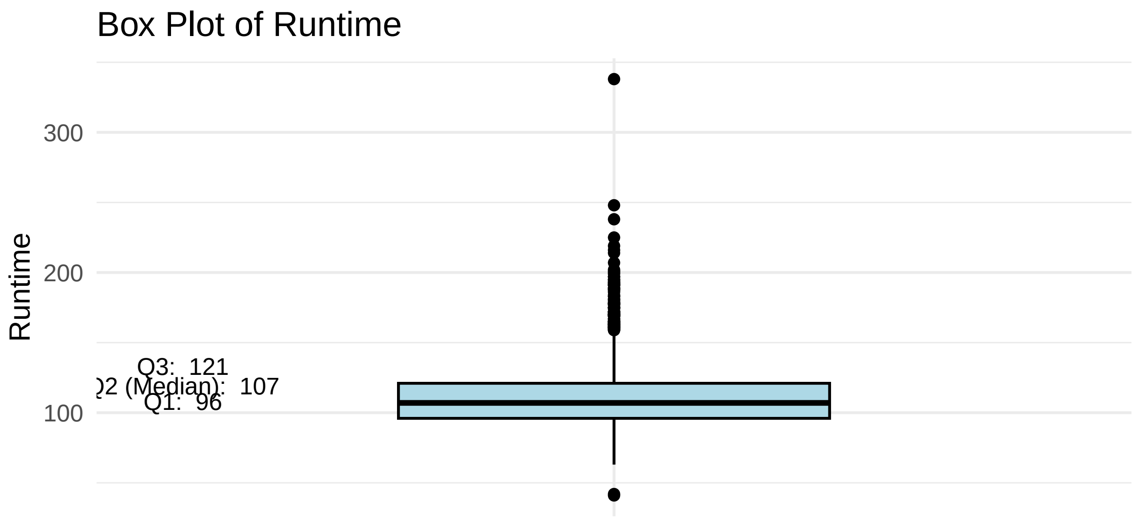
We see that after 1990 the number of movies getting released every year starts increasing drastically.

Problem 4

Create a horizontal box plot using the column "runtime". What inferences can you make from this box and whisker plot? Comment on the skew of the runtime column (visual inspection is enough).

SOLUTION:

```r
library(ggplot2)

# Calculate quartile and IQR values
quartiles <- quantile(data$runtime, probs = c(0.25, 0.5, 0.75), na.rm = TRUE)
iqr_value <- IQR(data$runtime, na.rm = TRUE)

# Create a data frame for labels
labels_df <- data.frame(
  x = rep(0.5, length(quartiles)),
  y = quartiles,
  label = c(paste("Q1: ", quartiles[1]),
            paste("Q2 (Median): ", quartiles[2]),
            paste("Q3: ", quartiles[3]))
)

# Create the horizontal box plot with quartile and IQR labels
plot <- ggplot(data, aes(x = "", y = runtime)) +
  geom_boxplot(width = 0.5, fill = "lightblue", color = "black") +
  labs(x = "", y = "Runtime", title = "Box Plot of Runtime") +
  theme_minimal() +
  theme(axis.text.x = element_blank(),  # Remove x-axis labels
        axis.ticks.x = element_blank()) +  # Remove x-axis ticks
  geom_text(data = labels_df, aes(x = x, y = y, label = label),
            vjust = -0.5, size = 3)

# Display the plot
print(plot)
```

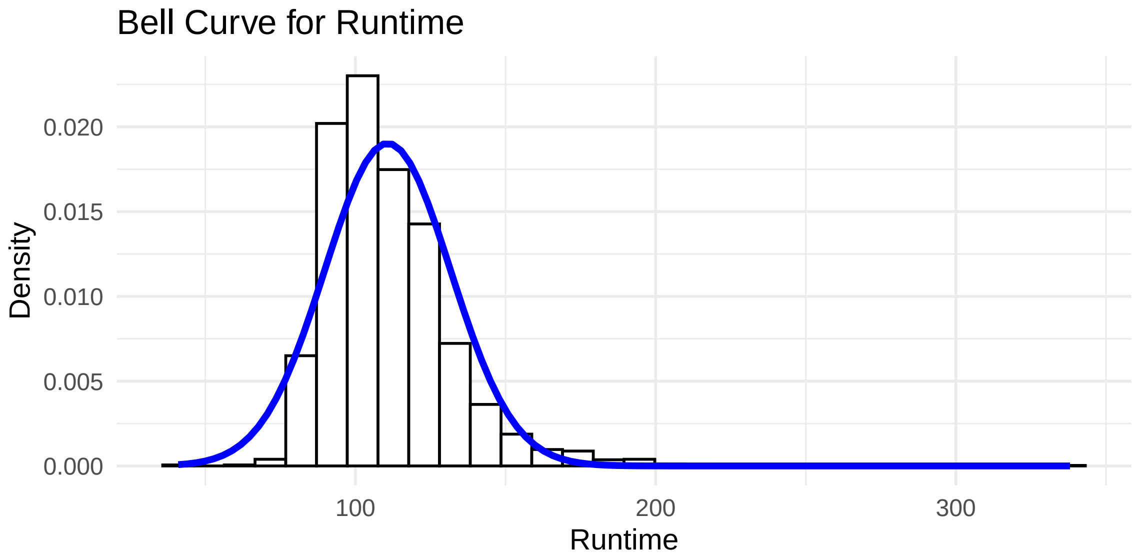


```r
# Calculate mean and standard deviation of the "popularity" column
runtime_mean <- mean(data$runtime)
runtime_sd <- sd(data$runtime)

# Create the plot using ggplot2
plot <- ggplot(data, aes(x = runtime)) +
  geom_histogram(aes(y = ..density..), bins = 30, color = "black", fill = "white") +
  stat_function(fun = dnorm, args = list(mean = runtime_mean, sd = runtime_sd), color = "blue", size =
  labs(x = "Runtime", y = "Density", title = "Bell Curve for Runtime") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
# Display the plot
print(plot)
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



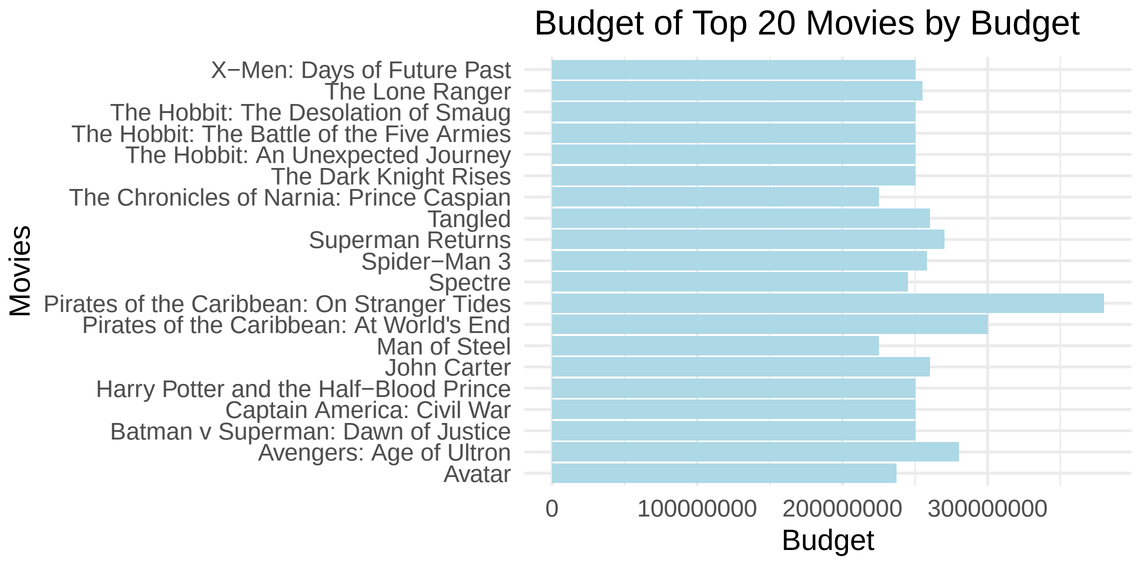
Bell Curve for Runtime

The Runtime column is not skewed (atleast not as skewed as the other fields) but we can see the presence of outliers in the box plot. We see the median runtime of movies to be 107 minutes which is the close to the runtime of films we see. Also another infernce we know from the box plot is the presence of outliers after the third quartile. There are no outliers before Q1 (Movies are generally not less than an hour)

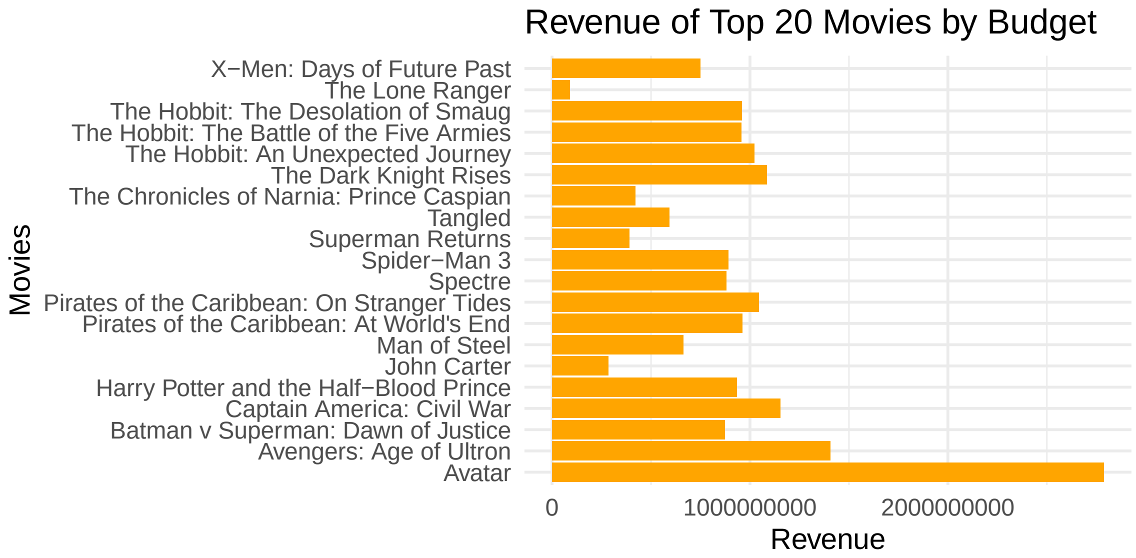
Problem 5

Analyze the top 20 titles with highest budget, revenue and ROI. Plot a horizontal bar graph for all three metrics in each case. What analysis can you make by looking at these graphs? What kind of movies attracts the highest investments and do they promise a better ROI ?

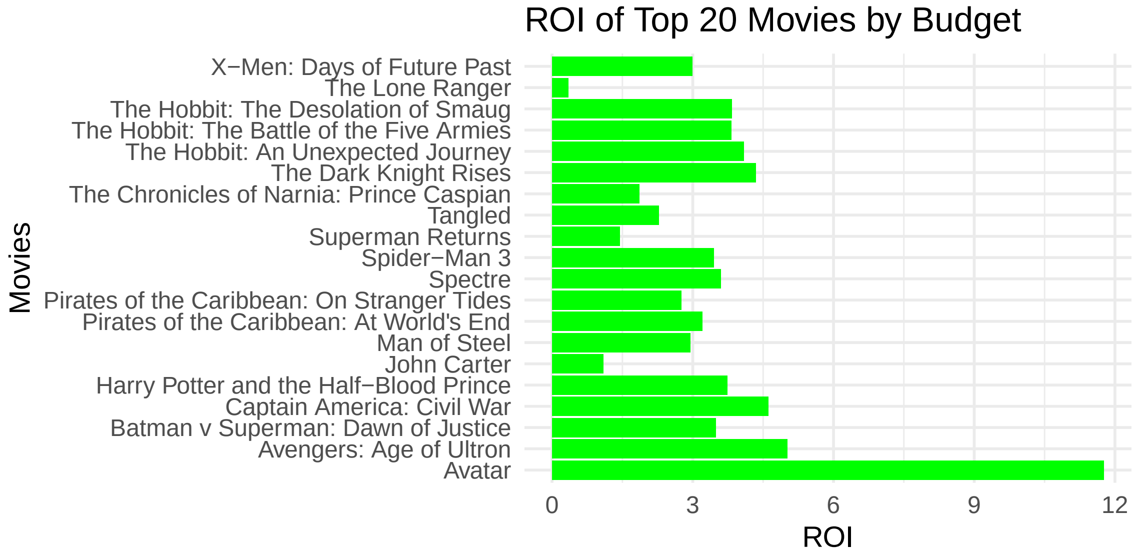
```
options(scipen = 999)

# Load the library
library(ggplot2)

data$ROI <- data$revenue/data$budget
# Sort the DataFrame by budgets in descending order
data <- data[order(data$budget, decreasing = TRUE), ]

# Select the top 20 rows with the highest budgets
top_20_data <- head(data, 20)

# Create the horizontal bar plot
plot <- ggplot(top_20_data, aes(x = title, y = budget)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(x = "Movies", y = "Budget", title = " Budget of Top 20 Movies by Budget") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



```
# Create the horizontal bar plot for revenue
plot <- ggplot(top_20_data, aes(x = title, y = revenue)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(x = "Movies", y = "Revenue", title = "Revenue of Top 20 Movies by Budget") +
  theme_minimal() +
  coord_flip()

print(plot)
```



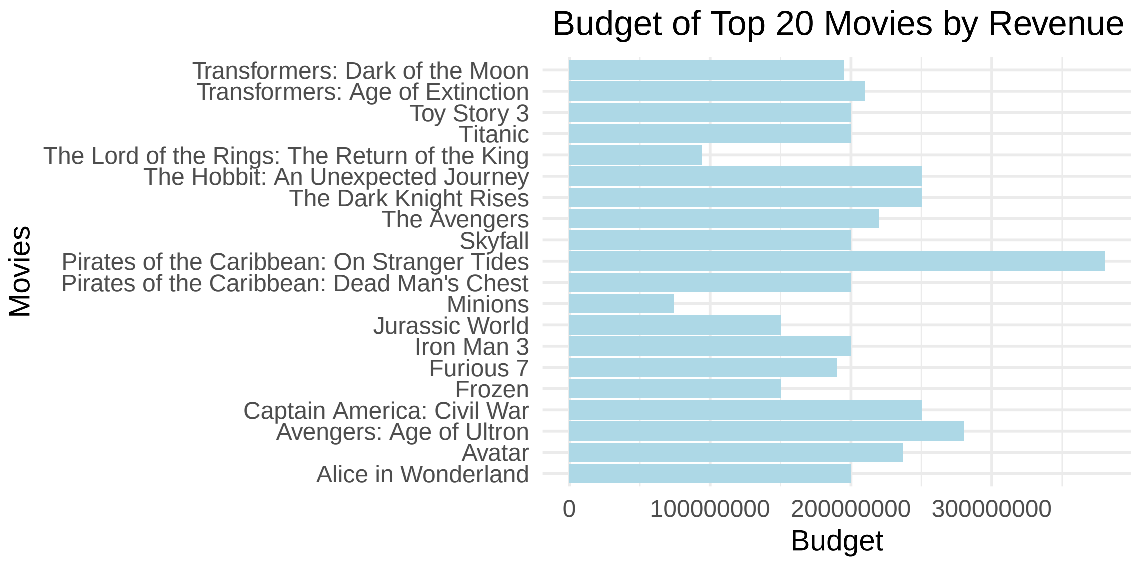
```r
# Create the horizontal bar plot for ROI
plot <- ggplot(top_20_data, aes(x = title, y = ROI)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(x = "Movies", y = "ROI", title = "ROI of Top 20 Movies by Budget") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



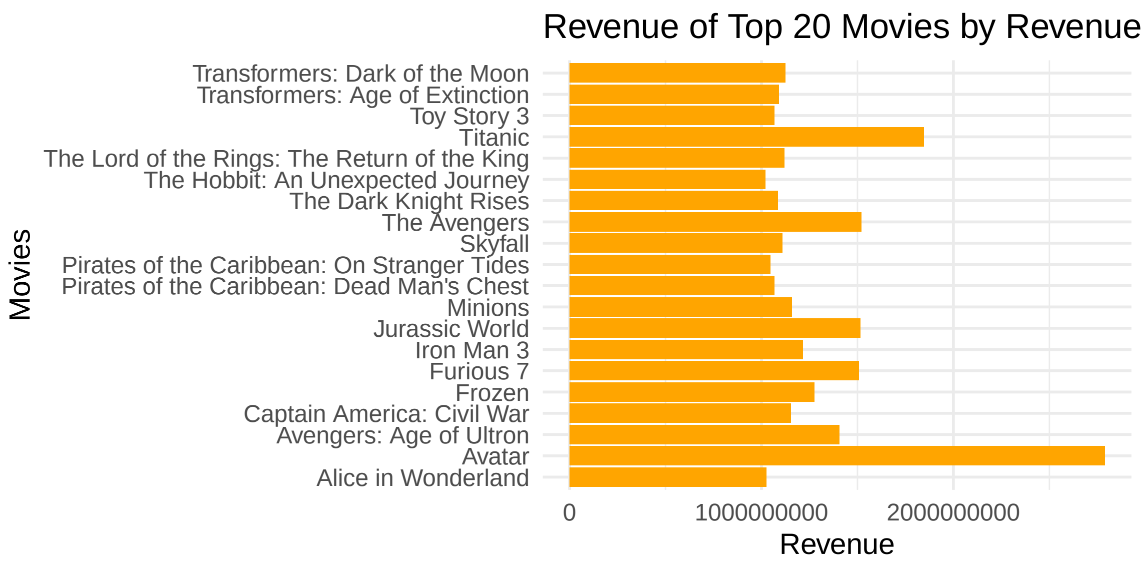
As you might expect, this is basically a list of blockbusters. These are the ones that the studios are happy to throw buckets of cash at in the hopes that it pays off. It mostly does: with the exception of The Lone Ranger and John Carter, all of these films made at least twice their budget in revenue.

Next. let's run the same chart, but sorted by the revenue column:

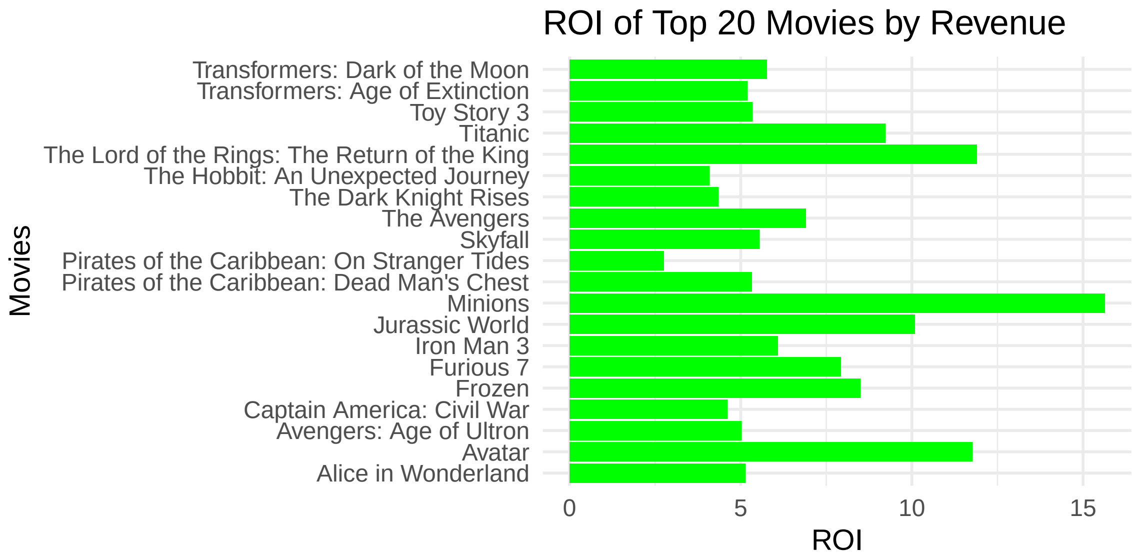
```r
# Load the library
library(ggplot2)

# Sort the DataFrame by budgets in descending order
data <- data[order(data$revenue, decreasing = TRUE), ]

# Select the top 10 rows with the highest budgets
top_20_data <- head(data, 20)

# Create the horizontal bar plot
plot <- ggplot(top_20_data, aes(x = title, y = budget)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(x = "Movies", y = "Budget", title = " Budget of Top 20 Movies by Revenue") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



```r
# Create the horizontal bar plot for revenue
plot <- ggplot(top_20_data, aes(x = title, y = revenue)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(x = "Movies", y = "Revenue", title = "Revenue of Top 20 Movies by Revenue") +
  theme_minimal() +
  coord_flip()

print(plot)
```



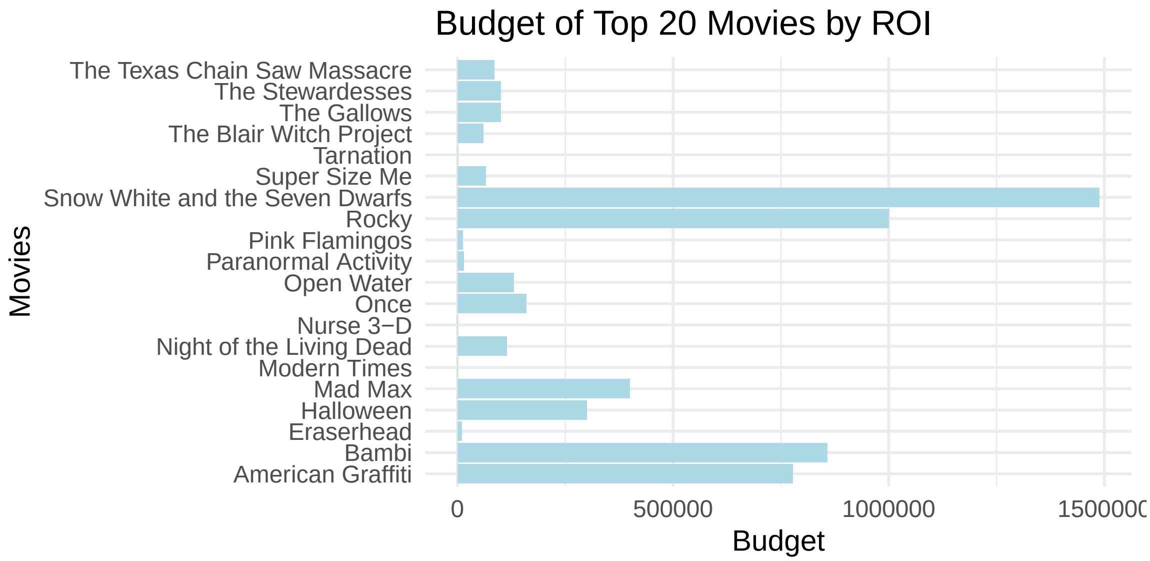
```r
# Create the horizontal bar plot for ROI
plot <- ggplot(top_20_data, aes(x = title, y = ROI)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(x = "Movies", y = "ROI", title = "ROI of Top 20 Movies by Revenue") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



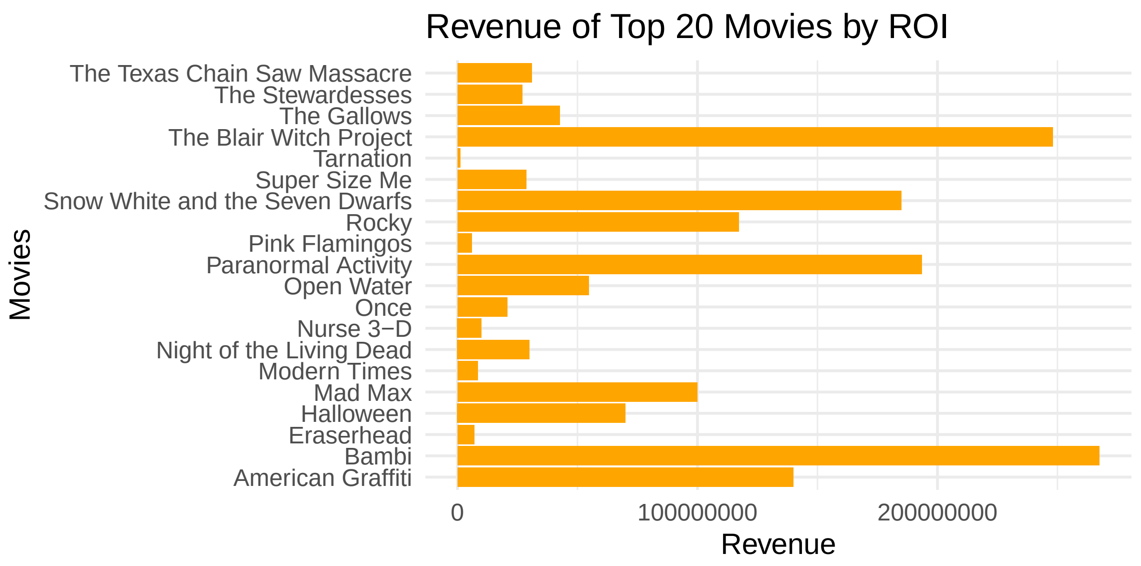
Still blockbusters, but it's interesting that it is a very different list of blockbusters. In fact, only a small number of rows in the data are in both the 20 highest budgets and revenues.

Finally for this section, we'll run that chart again but sort on the ROI column.

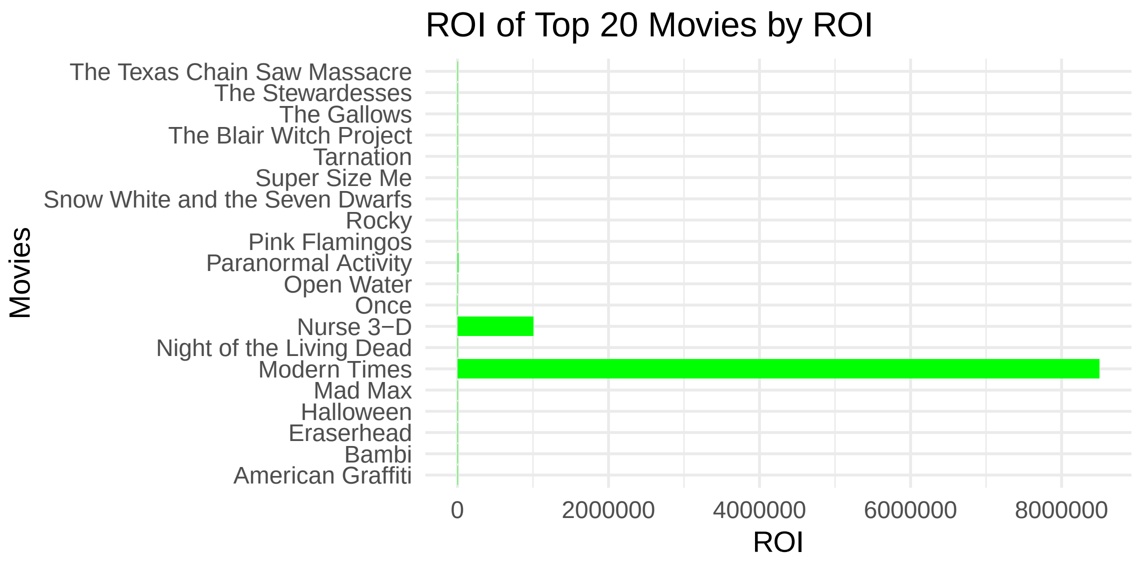
```r
# Load the library
library(ggplot2)

# Sort the DataFrame by budgets in descending order
data <- data[order(data$ROI, decreasing = TRUE), ]

# Select the top 10 rows with the highest budgets
top_20_data <- head(data, 20)

# Create the horizontal bar plot
plot <- ggplot(top_20_data, aes(x = title, y = budget)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(x = "Movies", y = "Budget", title = " Budget of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



```r
# Create the horizontal bar plot for revenue
plot <- ggplot(top_20_data, aes(x = title, y = revenue)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(x = "Movies", y = "Revenue", title = "Revenue of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

print(plot)
```



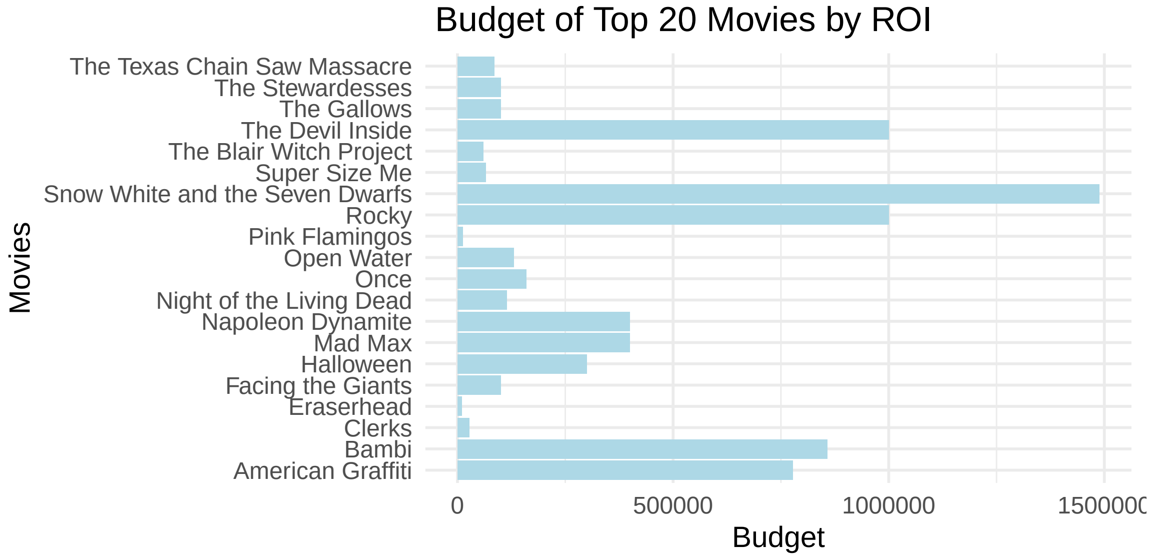
```r
# Create the horizontal bar plot for ROI
plot <- ggplot(top_20_data, aes(x = title, y = ROI)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(x = "Movies", y = "ROI", title = "ROI of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



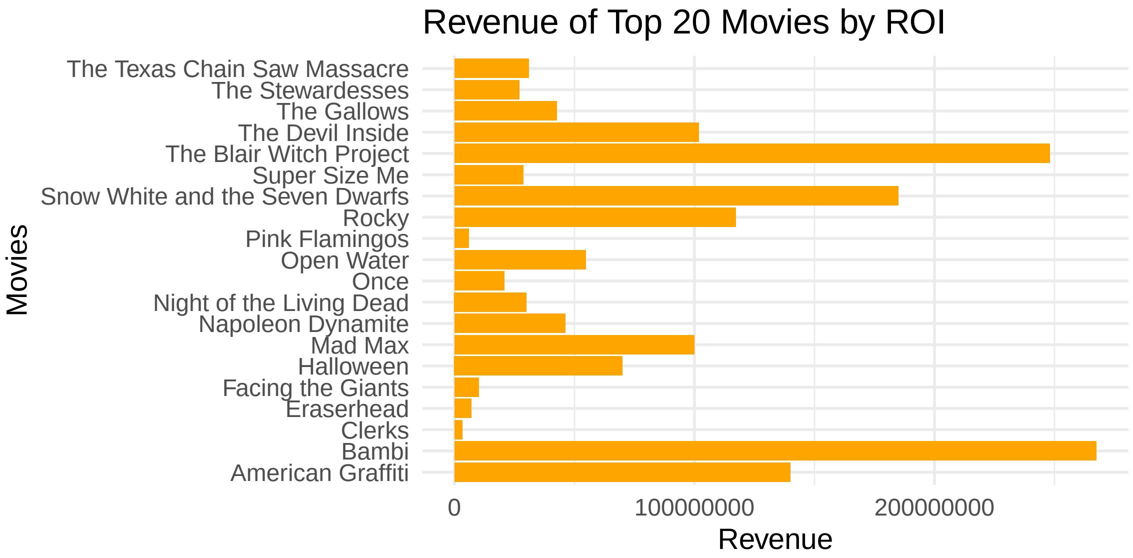
The scale of the graph is inappropriate to visualize the bar plot because of the presence of outliers. Lets remove outliers i.e. movies having ROI > 5000

```r
# Load the library
library(ggplot2)
```

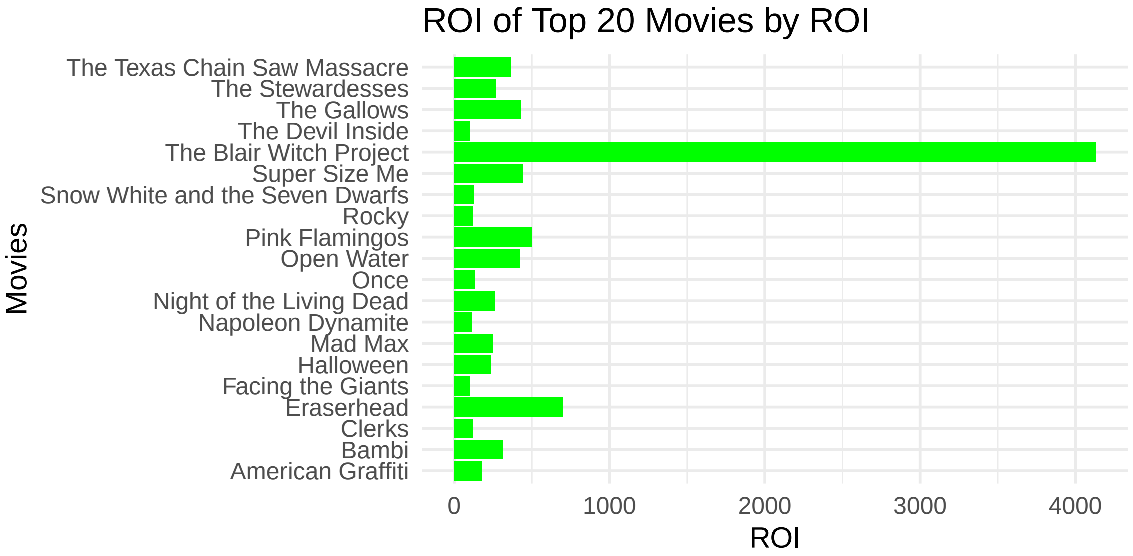
```r
# Sort the DataFrame by ROI in descending order
data <- data[order(data$ROI, decreasing = TRUE), ]
filtered_data <- data[data$ROI <= 5000, ]
# Select the top 10 rows with the highest budgets
top_20_data <- head(filtered_data, 20)

# Create the horizontal bar plot
plot <- ggplot(top_20_data, aes(x = title, y = budget)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(x = "Movies", y = "Budget", title = " Budget of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



```r
# Create the horizontal bar plot for revenue
plot <- ggplot(top_20_data, aes(x = title, y = revenue)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(x = "Movies", y = "Revenue", title = "Revenue of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

print(plot)
```



```
# Create the horizontal bar plot for ROI
plot <- ggplot(top_20_data, aes(x = title, y = ROI)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(x = "Movies", y = "ROI", title = "ROI of Top 20 Movies by ROI") +
  theme_minimal() +
  coord_flip()

# Display the plot
print(plot)
```



Obviously, the major outlier here is The Blair Witch Project.It's worth noting that a number of other films in this list are also horror movies; it seems they're cheap to make and sometimes become cult classics.

In general, the films that make the most revenue are the ones with a significant budget, but generally not the most investment. Yet outliers such as The Blair Witch Project buck the trend and demonstrate that even lower-budget films can be smash hits in the right circumstances, while The Lone Ranger(in the graph sorted by budget) shows that higher-budget films can still flop.

Problem 6

Put yourself in the shoes of a production house. You want to produce the next big blockbuster. Plot the ROI, revenue and budget across genres to finalize the genre of your upcoming movie as you did in the previous problem. Elaborate your answers with proper explanation. Since one movie can fall in multiple genre categories, you are free to choose a combination. You can also understand how the popularity of different genres have changed along the years. Do provide a nice name to your movie and your dream cast ;)

To be attempted by the student