5/12/2023

Josephine Quagraine

GMCA 5

Data analysis and visualization using R and Power BI

Table of Contents

[1 Initial Exploratory Analysis 3](#_Toc134968853)

[1.1 Import/ load data into R 3](#_Toc134968854)

[1.2 View/ take a look at the data 3](#_Toc134968855)

[1.3 Load library 4](#_Toc134968856)

[1.4 Check data types 4](#_Toc134968857)

[2 Clean data and remove outliers 4](#_Toc134968858)

[2.1 Clean data 4](#_Toc134968859)

[2.1.1 Check for missing values 4](#_Toc134968860)

[2.1.2 Drop missing values 5](#_Toc134968861)

[2.1.3 To verify that rows have been omitted/removed 5](#_Toc134968862)

[2.1.4 Check for duplicates 5](#_Toc134968863)

[2.1.5 Round off numeric values to 2 places 6](#_Toc134968864)

[2.1.6 View data 6](#_Toc134968865)

[2.2 Remove outlier 7](#_Toc134968866)

[2.2.1 Boxplot 7](#_Toc134968867)

[2.2.2 Remove outliers in Profitability 7](#_Toc134968868)

[2.2.3 Remove outliers in Worldwide.Gross 8](#_Toc134968869)

[3 Exploratory Data Analysis 8](#_Toc134968870)

[3.1 Univariate analysis 8](#_Toc134968871)

[3.2 Bivariate analysis 9](#_Toc134968872)

[3.2.1 Scatterplot 9](#_Toc134968873)

[3.2.2 Bar chart 10](#_Toc134968874)

[4 Export clean data 10](#_Toc134968875)

[5 Power BI visualizations 0](#_Toc134968876)

**List of figures**

l

[Figure 1 Figure showing a section of the imported data frame into R 3](#_Toc134967962)

[Figure 2 Figure showing the output console with the correct and wrong code for viewing the data in the data frame 3](#_Toc134967963)

[Figure 3 Figure showing the successful loading of the tidyverse 4](#_Toc134967964)

[Figure 4 Figure showing the 3 data types within the data frame 4](#_Toc134967965)

[Figure 5 Output after running the code colSums(is.na(HW\_df)) 5](#_Toc134967966)

[Figure 6 Output of code HW\_df <-na.omit(HW\_df) in console indicating that it run successfully 5](#_Toc134967967)

[Figure 7 Console showing results that verify that missing values had been omitted successfully 5](#_Toc134967968)

[Figure 8 Console output in R showing there are zero duplicates in the data frame 5](#_Toc134967969)

[Figure 9 Output showing successful run of both HW\_df$Profitability<- round (HW\_df$Profitability, digit =2) and HW\_df$Worldwide.Gross<-round (HW\_df$Worldwide.Gross, digit=2) 6](#_Toc134967970)

[Figure 10 Figure showing the partially cleaned dataset with numeric values rounded up to 2 decimal places 6](#_Toc134967971)

[Figure 11 Boxplot showing the outliers (coloured purple) in the data frame 7](#_Toc134967972)

[Figure 12 Output of univariate analysis using code summary(HW\_df1) 8](#_Toc134967973)

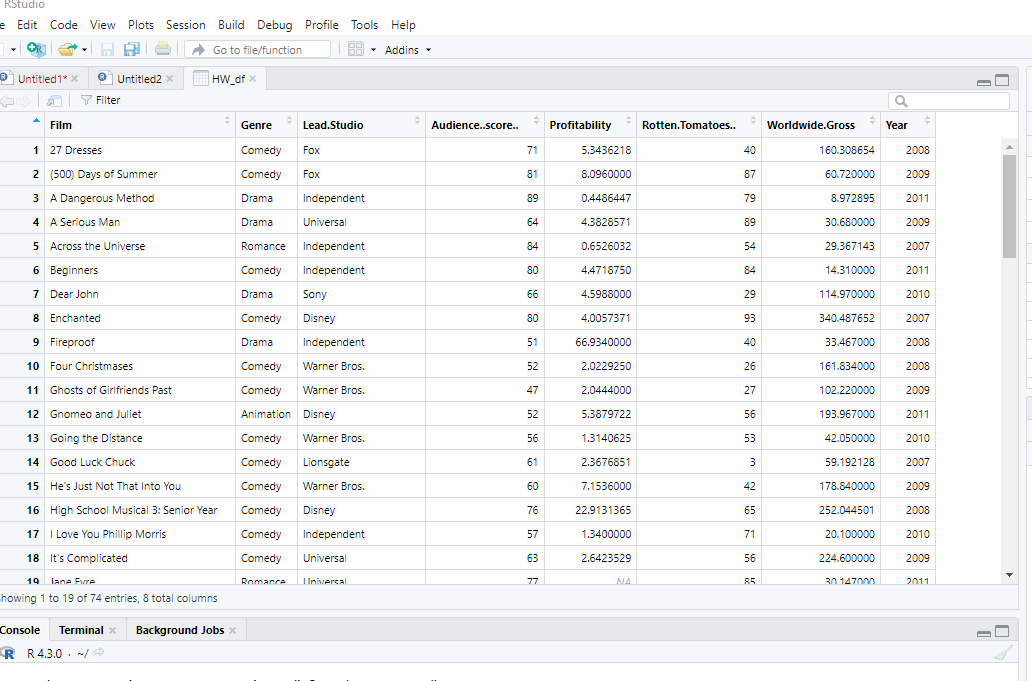
[Figure 13 Scatterplot showing the relationship between rotten tomatoes ratings and lead studios 9](#_Toc134967974)

[Figure 14 Bar chart showing the count of movies according to the year they were produced 10](#_Toc134967975)

# Initial Exploratory Analysis

## Import/ load data into R

Downloaded link as shown in PowerPoint and imported into R studio as a data frame (df). Data frames are data displayed in a tabular format and could have different types of data in different columns. To access data within dfs, single or double brackets ([ ], [[ ]]) or $ can be used. For this analysis, the file name, **HollywoodsMostProfitableStories** was changed to **HW\_df** while importing into R for easier manipulation.



**Figure 1** Figure showing a section of the imported data frame into R

## View/ take a look at the data

To view the data in R, used the code **View (HW\_df).** As R is case sensitive, it only accepted the capital **V** and not the lower case **v** as indicated in Figure 2 below.

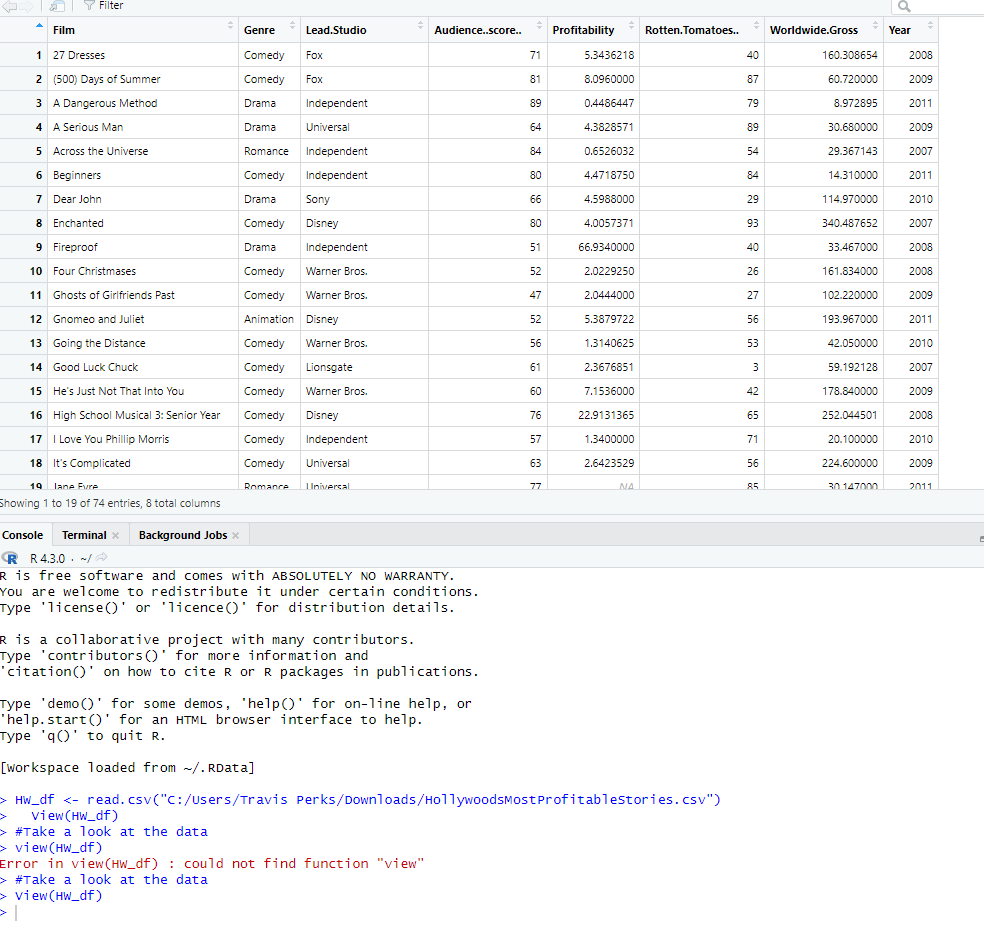
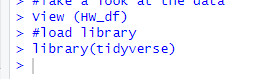


Figure 2 Figure showing the output console with the correct and wrong code for viewing the data in the data frame

## Load library

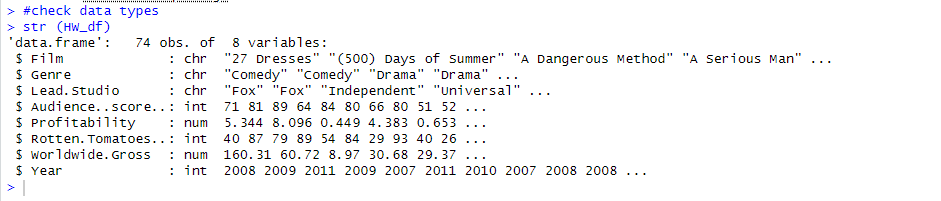
To be able to carry out data cleaning, needed to load tidyverse, a collection of R packages designed for working with data. It is a tool that provides methods to diagnose and clean datasets in R. Already had Tidyverse library installed so skipped the install tidyverse package and loaded it using the code: **library(tidyverse)** which when executed successfully, turned blue in the console as indicated in the figure below.



**Figure 3** Figure showing the successful loading of the tidyverse

## Check data types

To check data types, used the command code **str (HW\_df)**. This brought up a quick overview of the first 5 rows for all the columns or variables in the dataset. As indicated in the output console in the figure below, there are 3 data types within the data frame. These are character (chr) or string, integer (int) and numeric (num). There are also 74 observations or rows and 8 variables or columns in the data frame.



**Figure 4** Figure showing the 3 data types within the data frame

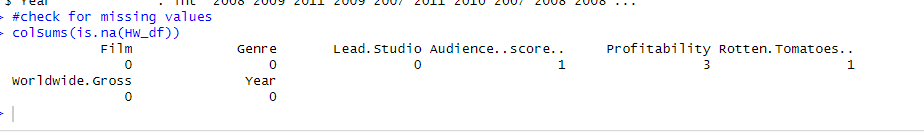
# Clean data and remove outliers

## Clean data

To be able to analyse data and create visuals, the data has to be consistent for statistical analysis. As such missing values, errors such as duplicates and outliers (abnormality in the data) need to be removed or corrected.

### Check for missing values

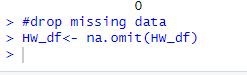
A missing value is represented by NA in R. When present, it indicates that though the data type is known, its value is not and thus cannot be used to perform statistical analysis. To verify if there were any null data within the data frame, used the code **colSums(is.na(HW\_df))** which basically gives a summation of all the columns (colSums) as well as that of any null values (is.na). After running the code, most of the columns had complete datasets aside from 3. The output console indicated that there were 3 nulls in the column profitability, 1 in audience score and 1 in rotten tomatoes respectively as indicated in figure 5.



**Figure 5** Output after running the code colSums(is.na(HW\_df))

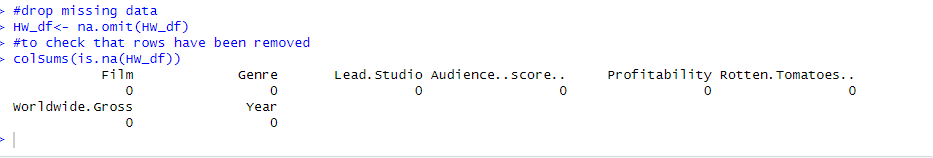
### Drop missing values

Before any analysis can be done, need to deal with missing values. For this, used the command or code **HW\_df <-na.omit(HW\_df**) which omits elements from a dataset containing missing values. After running the code, showed up in the output console in blue which indicated that it had run successfully.



**Figure 6** Output of code HW\_df <-na.omit(HW\_df) in console indicating that it run successfully

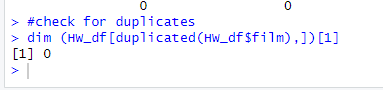
### To verify that rows have been omitted/removed

Next was to verify that missing values had been dropped by checking to ensure that the affected rows had been successfully removed. To do this task, entered the code, **colSums (is.na(HW\_df))** again. This shows that there are zero missing data (figure 7)as compared to the initial check that yielded some missing data shown in figure 5 above and thus verified that the previously affected rows had been removed.

**Figure 7** Console showing results that verify that missing values had been omitted successfully

### Check for duplicates

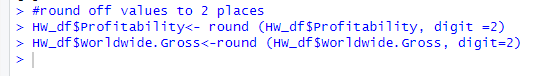
To check for duplicates, utilized the code, **dim (HW\_df[duplicated(HW\_df$film),])[1].** The duplicated() function in R checks which elements of the data frame are duplicates and returns a logical vector suggesting which elements or rows are duplicates. Run gave a successful prompt in the console of R as indicated below.



**Figure 8** Console output in R showing there are zero duplicates in the data frame

### Round off numeric values to 2 places

From figure 1, the are 2 columns (Profitability and Worldwide.Gross) that have numeric values. These numeric values have more than 2 decimal places which could be problematic during analysis. As such, 2 codes were used to round off the values to 2 decimal places. These are **HW\_df$Profitability<- round (HW\_df$Profitability, digit =2)** and **HW\_df$Worldwide.Gross<-round (HW\_df$Worldwide.Gross, digit=2)**. After running the 2 codes, generated the output in the figure 9 below.



**Figure 9** Output showing successful run of both HW\_df$Profitability<- round (HW\_df$Profitability, digit =2) and HW\_df$Worldwide.Gross<-round (HW\_df$Worldwide.Gross, digit=2)

### View data

To verify that the data within those two columns have been rounded off, run the code **View (HW\_df)** again. After running the code, displayed the amended data which shows the numeric values within the columns Profitability and Worldwide.Gross, have been rounded to 2 place values. To check on the number of rows and columns available within the amended data, used the code **dim (HW\_df)**. This showed there were still 8 columns in the data frame. However, there were now 70 rows left from the initial 75 rows.

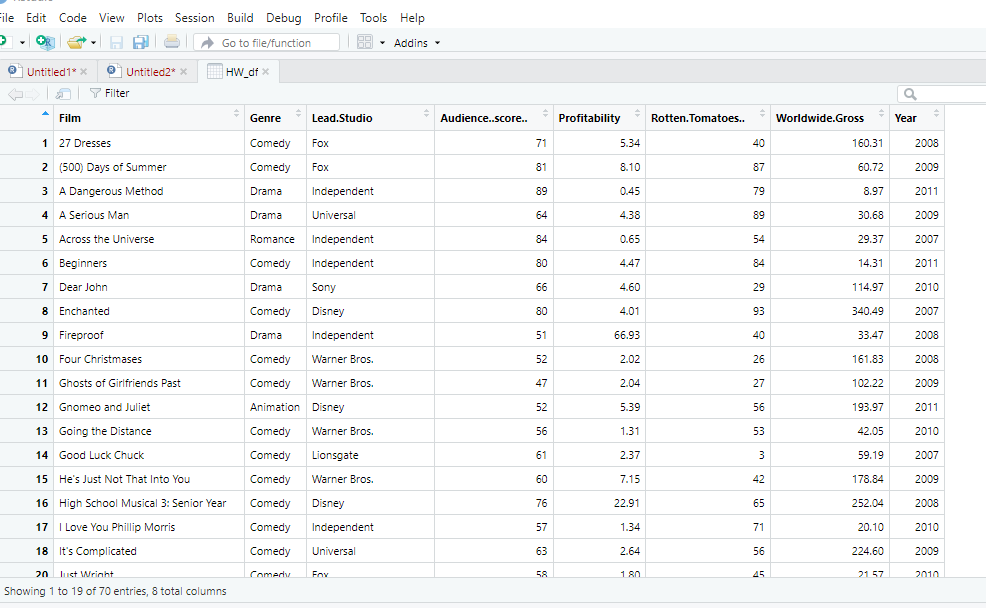


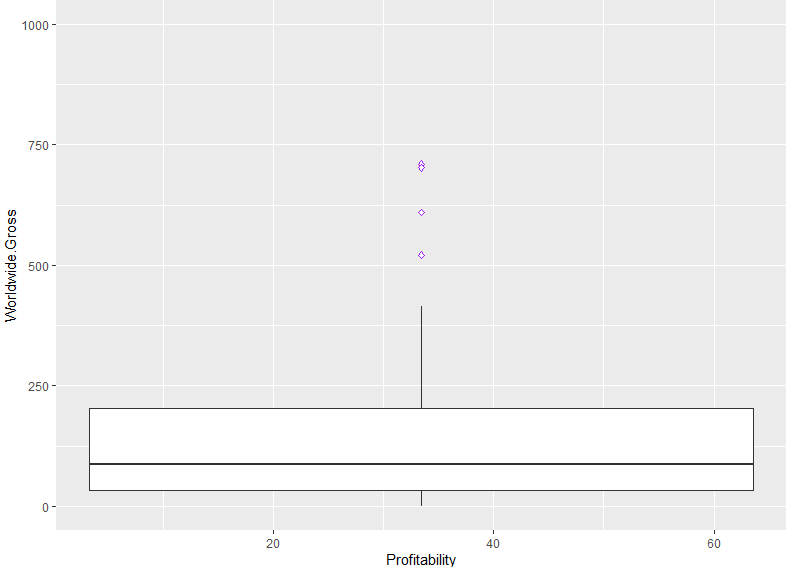
Figure 10Figure showing the partially cleaned dataset with numeric values rounded up to 2 decimal places

## Remove outlier

An observation is an outlier or an abnormally when it is larger or smaller than the ``whiskers'' of the set of observations in a boxplot. The upper whisker is computed by adding 1.5 times the interquartile range to the third quartile and rounding to the nearest lower observation. The lower whisker is computed likewise. This is can be visualized using the box (and whisker) plot, where the box indicates the interquartile range and the median. The whiskers are represented at the ends of the box and outliers are indicated as separate points above or below the whiskers

### Boxplot

To create a boxplot in R, used the library(ggplot2) which is an inbuilt package in R. Used code **ggplot (HW\_df,aes(x=Profitability,y=Worldwide.Gross)) + geom\_boxplot (outlier.colour = "red" ,outlier.shape= 1) + scale\_x\_continuous (labels = scales::comma) + coord\_cartesian(ylim= c(0,1000)).** Changed the shapes and colours as indicated in figures 11 from **circle (1)** to **diamond (23)** and red to purple. From the graph, the outliers were approximately above 400 and lower than 750.



**Figure 11** Boxplot showing the outliers (coloured purple) in the data frame

### Remove outliers in Profitability

After using the boxplot to determine the outliers in profitability, used the codes below to remove the outliers from the data set. After running the codes, **dim(no\_outliers)** showed the number of observations (rows) reduced to 65. Code used to remove outliers in 'Profitability' was **Q1 <- quantile(HW\_df$Profitability, .25), Q3 <- quantile(HW\_df$Profitability, .75), IQR <- IQR(HW\_df$Profitability) and no\_outliers <- subset(HW\_df, HW\_df$Profitability> (Q1 - 1.5\*IQR) & HW\_df$Profitability< (Q3 + 1.5\*IQR))**

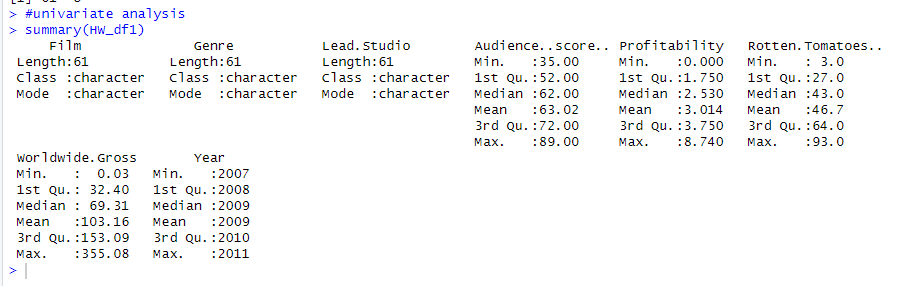
### Remove outliers in Worldwide.Gross

To remove outliers in **'Worldwide.Gross', used Q1 <- quantile(no\_outliers$Worldwide.Gross, .25), Q3 <- quantile(no\_outliers$Worldwide.Gross, .75), IQR <- IQR(no\_outliers$Worldwide.Gross) and HW\_df1 <- subset(no\_outliers, no\_outliers$Worldwide.Gross> (Q1 - 1.5\*IQR) &** **no\_outliers$Worldwide.Gross< (Q3 + 1.5\*IQR)).** After running **dim(HW\_df1)** the number of observations (rows) reduced to 61.

# Exploratory Data Analysis

## Univariate analysis

This is also known as summary statistics and gives 6 statistical output for each column/variable as indicated below. From the analysis, it gives information on min, max, 1st and 3rd quartiles as well as the mean and median. For example, the minimum year for a film to have been included is 2007 while the maximum is 2011. There are now also 61 observations. This step is done using the code, **summary(HW\_df1).**

****

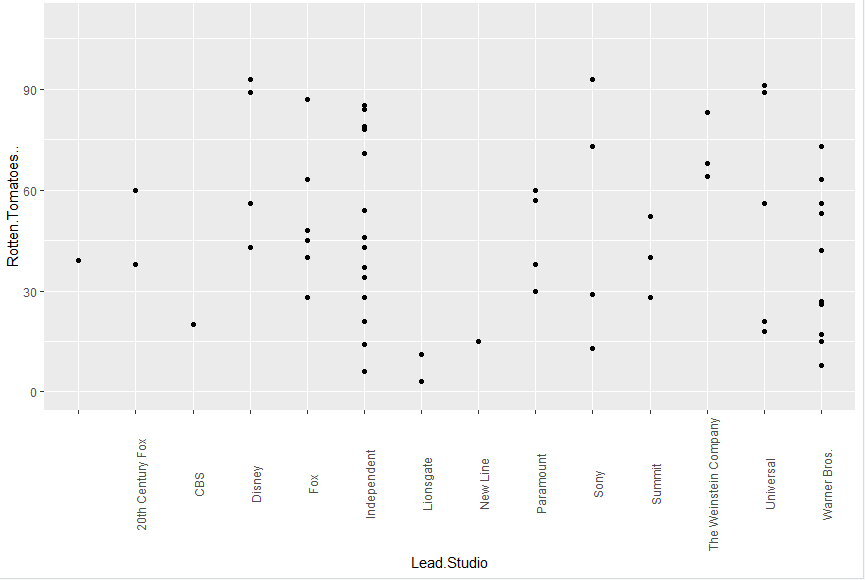
**Figure 12** Output ofunivariate analysis using code summary(HW\_df1)

## Bivariate analysis

The bivariate analysis is to find the correlation or relationship between two variables or columns. For this task, the scatterplot and bar chart were used.

### Scatterplot

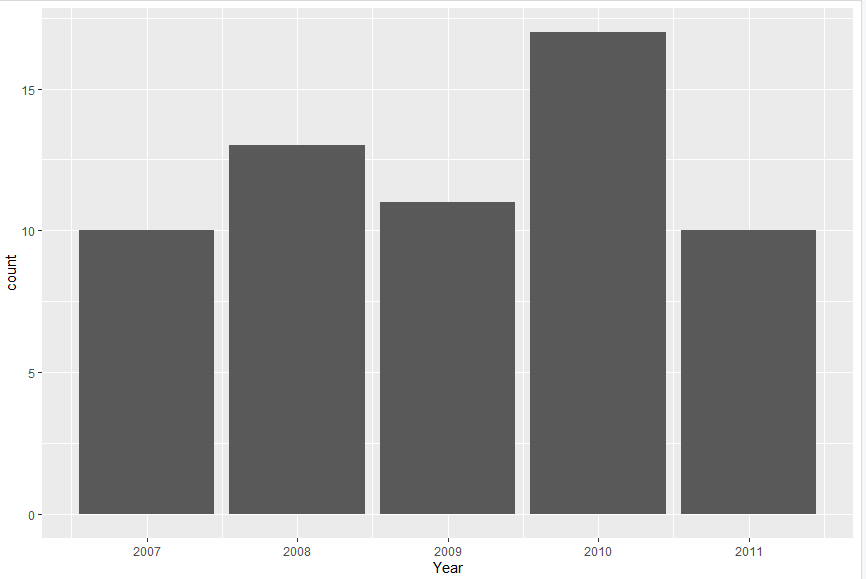
For this graph, utilized the code: **ggplot(HW\_df1, aes(x=Lead.Studio, y=Rotten.Tomatoes..)) + geom\_point()+ scale\_y\_continuous(labels = scales::comma)+coord\_cartesian(ylim = c(0, 110))+theme(axis.text.x = element\_text(angle = 90))** which generated the graph in figure 13. Graph shows a high correlation between the 2 columns. More movies from Independent studio was rated followed by Warner Brothers. Could be inferred that they produced more movies within those years. However, the highest rated movies were from Disney and Sony followed closely by Universal and Independent.



**Figure 13** Scatterplot showing the relationship between rotten tomatoes ratings and lead studios

### Bar chart

Used code **ggplot(HW\_df1, aes(x=Year)) + geom\_bar()** to generated the graph in figure 14 below. From this chart, can infer that more films were produced in 2010 followed by 2008 while 2007 and 2011 had less with 10 being produced.



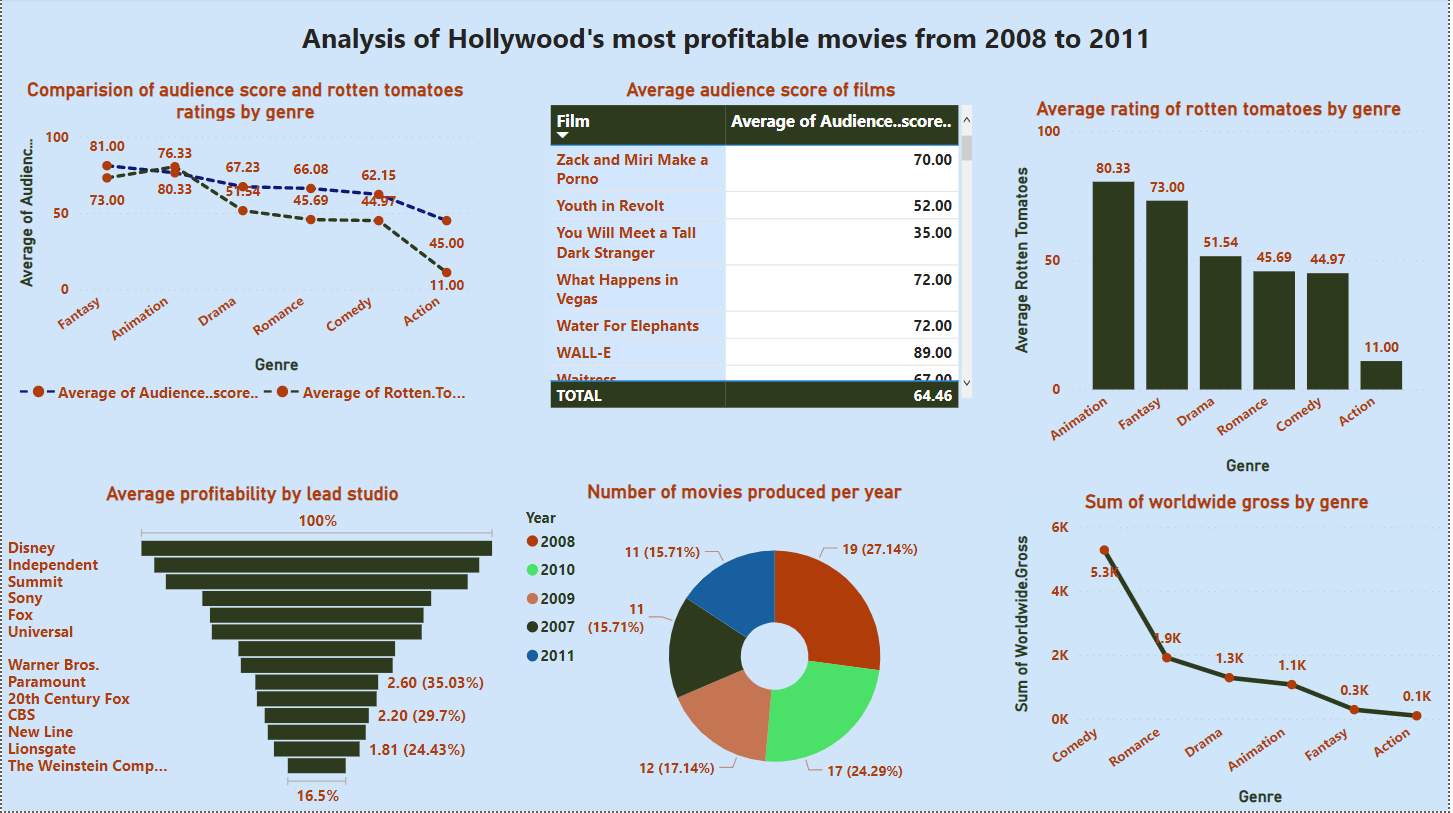
**Figure 14** Bar chart showing the count of movies according to the year they were produced

# Export clean data

After obtaining satisfactory results from the exploratory data analysis, the cleaned consistent data is now ready to be used in downstream data analysis in Power BI. As such, the cleaned data, renamed as HW\_df1 has to be exported. To do this, used the code, **write.csv(HW\_df1,”clean\_HW\_df.csv”)** to export it. However, as I could not find the CSV file, wrote a file path to easily located the file. Code used was **write.csv** **(HW\_df1,"C:\\Users\\Travis Perks\\Downloads\\clean\_HW\_df.csv").** The CSV file was then imported into Power BI.

# Power BI visualizations

For the dashboard, instructions were to use company brand colors of blue, green and brown. Visualizations had to include average rotten tomatoes ratings of each genre, the number of movies produced per year, the audience score for each film, the profitability per studio and the worldwide gross per genre. Added comparison of audience score and rotten tomatoes to make 6 charts in total as seen in the figure below. Link to published work is <https://app.powerbi.com/links/MWd37YbOKE?ctid=6efd0f20-57c8-4447-b53f-00d4992ca50b&pbi_source=linkShare>



**Figure 15** Dashboard showing the analysis of Hollywood’s most profitable movies from 2008 to 2011