AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)

INTRODUCTION TO DATA SCIENCE

Section: [C]

Spring 2022-2023

Project Title:

Interactive Dashboard using Shiny based on Web Scraping Data

Supervised By: Dr. Akinul Islam Jony

Submitted By

Name	ID
A.S.M. Fazle Rabbi	19-39714-1
Farhan Sadik Ferdous	20-42072-1
Tapu Biswas	20-42073-1
Salahuddin Elias Khan	20-44139-2

Date of Submission: 30th April 2023

Project Overview:

For this project, we have been assigned to scrap data from webpages, perform preprocessing techniques on them, describe them in the light of descriptive statistics and visualize them using R language.

In our project firstly, we chose movie revenue data from the box office. To build an interactive dashboard that displays information about movie rankings based on their worldwide box office revenue, domestic box office revenue, and international box office revenue. The data from 1977 to the present day will be collected through web scraping from "**The Numbers"** website. After that, we did many comparisons on data like why **"Avatar"** was the highest ranked movie and analyzed the dataset. Real-world data is frequently incomplete, noisy, and inconsistent, meaning it needs to be cleaned up before it can be put to the intended use. Data pre-processing is a common term for this. Data preprocessing is a data mining technique used to turn raw data into a practical and effective format. The most important tasks involved in data pre-processing are Data Cleaning, Data Integration, Data Transformation, Data Reduction, and Data Discretization. We did data preprocessing where it was needed. In Descriptive analysis, we described our data with the help of descriptive methods. In the descriptive analysis, we describe our data in some manner and present it in a meaningful way so that it can be easily understood. To describe a comparison between different things we did the Mean, Median, Mode, Range, Variance, Quartile & Percentile. Lastly, we did data visualization to see and understand as visualizations can more effectively allow the reader to gather information. Graphics can allow users to deliver insights in a much easier fashion than describing through text and can also have a greater impact. Here we tried to visualize almost every aspect of comparison & relation.

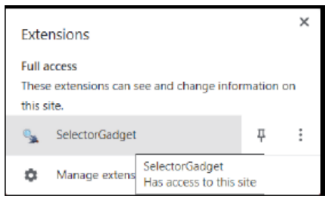
Project Solution Design:

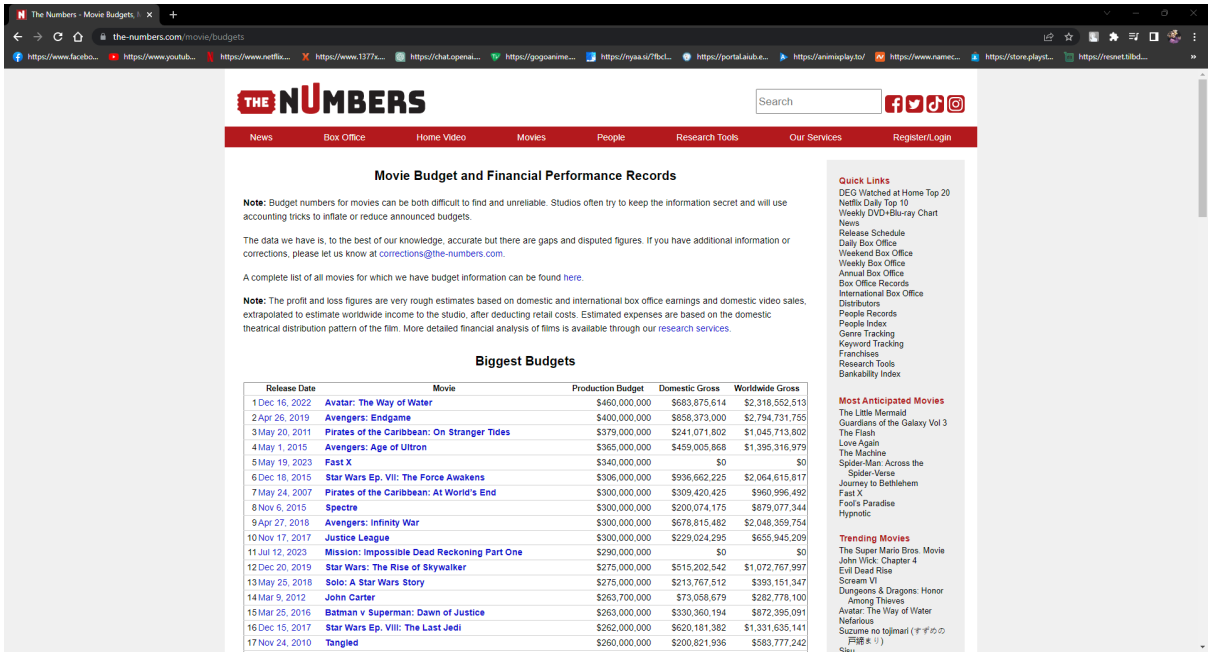
We initially gathered our movie lists and box office income from "**The Numbers"** website in order to prepare the dataset for data analysis. We then recorded the information in a CSV file. The data pre-processing is then done. Data cleaning is the process of inspecting a raw dataset to find and eliminate errors, duplication, and superfluous data. The table had some missing data, which we removed. Then we tried to manage every item of noisy data that was in the dataset. After performing data cleaning, measures for data integration, data transformation, data reduction, and data discretization were taken to further clean the data set. We concentrated on using descriptive statistics to rationally simplify our enormous volumes of data after completing the data preprocessing. Moreover, to sum up, the dataset's approximate data. In our data collection, we used the following metrics: Mean, Median, Mode, Range, Variance, Standard Deviation, Quartiles, Percentiles, and Interquartile Ranges. We used data visualization to present facts and data graphically after finishing the descriptive statistics.

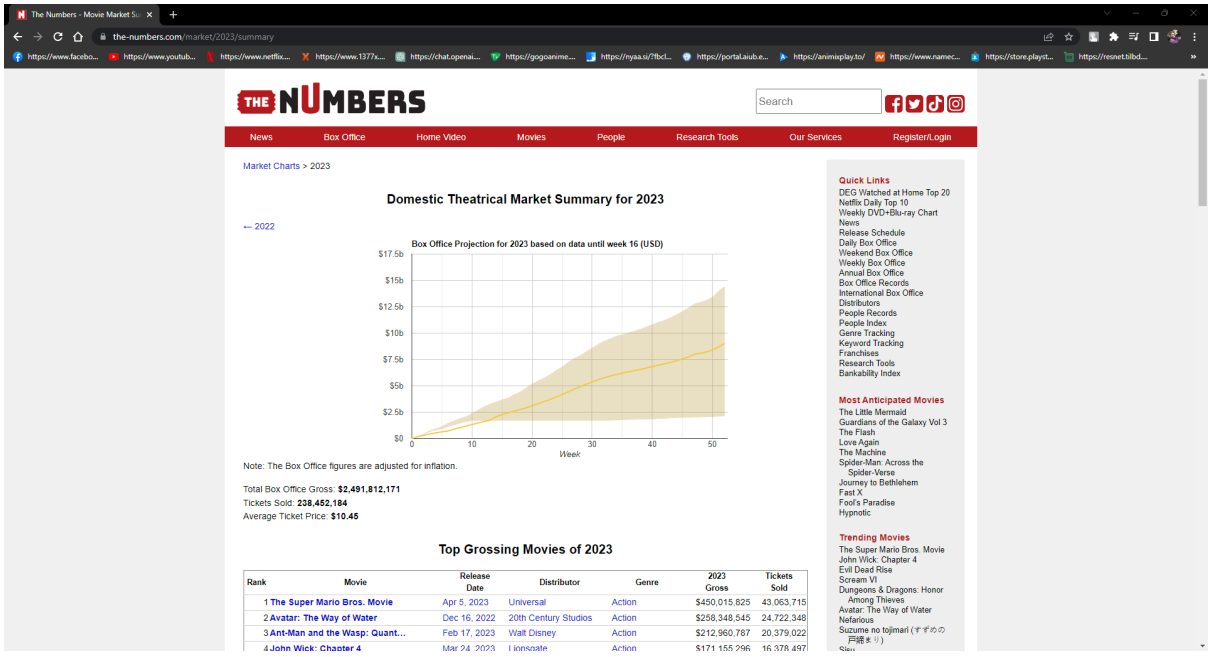
Data Collection:

For this project, we start to scrap the data from the website. First, we start to scrap the data from "**The Numbers**" website. In this process, we use a selector gadget to simply select data on a website and it will determine its HTML/CSS tags, ids and classes.

Data Source: The Numbers (https://www.the-numbers.com/movie/budgets/all)







Source code:

```
#Loading Library
install.packages("rvest","dplyr")
library(rvest)
library(dplyr)

#Creating data frame
Movies = data.frame()

for (page_result in seq(from = 1, to = 2000, by = 100)) {
  #URL link
  link = paste0("https://www.the-numbers.com/box-office-records/worldwide/all-
Movies/cumulative/all-time/",page_result)

  #Read the HTML content of the page
  page = read_html(link)

  #Fetching data from webpage
  rank = page %>% html_nodes(".data:nth-child(1)") %>% html_text()
  year = page %>% html_nodes(".data a") %>% html_text()
  name = page %>% html_nodes("#page_filling_chart b a") %>% html_text()
  worldwide_box_office = page %>% html_nodes("td:nth-child(4)") %>% html_text()
  domestic_box_office = page %>% html_nodes("td:nth-child(5)") %>% html_text()
  international_box_office = page %>% html_nodes("td:nth-child(6)") %>% html_text()

  #Adding data into Data Frame
  Movies = rbind(Movies,
data.frame(rank,name,year,worldwide_box_office,domestic_box_office,
                             international_box_office, stringsAsFactors = FALSE))

}

#Storing the data frame into csv file for future use
write.csv(Movies, "Movies_data.csv", append = TRUE)
#read.csv("Movies_data.csv")
print(Movies)
```

Output: First 50 entries

	X	rank	name	year	worldwide_box_office	domestic_box_office	international_box_office
1	1	1	Avatar	2009	$2,923,706,026	$785,221,649	$2,138,484,377
2	2	2	Avengers: Endgame	2019	$2,794,731,755	$858,373,000	$1,936,358,755
3	3	3	Avatar: The Way of Water	2022	$2,318,552,513	$683,875,614	$1,634,676,899
4	4	4	Titanic	1997	$2,222,985,568	$674,396,795	$1,548,588,773
5	5	5	Star Wars Ep. VII: The Force Awakens	2015	$2,064,615,817	$936,662,225	$1,127,953,592
6	6	6	Avengers: Infinity War	2018	$2,048,359,754	$678,815,482	$1,369,544,272
7	7	7	Spider-Man: No Way Home	2021	$1,910,048,245	$814,115,070	$1,095,933,175
8	8	8	Jurassic World	2015	$1,669,963,641	$652,306,625	$1,017,657,016
9	9	9	The Lion King	2019	$1,647,733,638	$543,638,043	$1,104,095,595
10	10	10	The Avengers	2012	$1,515,100,211	$623,357,910	$891,742,301
11	11	11	Furious 7	2015	$1,514,553,486	$353,007,020	$1,161,546,466
12	12	12	Top Gun: Maverick	2022	$1,481,369,482	$718,732,821	$762,636,661
13	13	13	Frozen II	2019	$1,437,862,795	$477,373,578	$960,489,217
14	14	14	Avengers: Age of Ultron	2015	$1,395,316,979	$459,005,868	$936,311,111
15	15	15	Black Panther	2018	$1,336,494,320	$700,059,566	$636,434,754
16	16	16	Star Wars Ep. VIII: The Last Jedi	2017	$1,331,635,141	$620,181,382	$711,453,759
17	17	17	Harry Potter and the Deathly Hallows:…	2011	$1,316,278,261	$381,193,157	$935,085,104
18	18	18	Jurassic World: Fallen Kingdom	2018	$1,308,323,302	$417,719,760	$890,603,542
19	19	19	Beauty and the Beast	2017	$1,268,697,483	$504,014,165	$764,683,318
20	20	20	Frozen	2013	$1,256,887,580	$400,953,009	$855,934,571
21	21	21	Incredibles 2	2018	$1,242,805,359	$608,581,744	$634,223,615
22	22	22	The Fate of the Furious	2017	$1,236,703,796	$225,764,765	$1,010,939,031
23	23	23	Iron Man 3	2013	$1,215,392,272	$408,992,272	$806,400,000
24	24	24	Minions	2015	$1,157,271,759	$336,045,770	$821,225,989
25	25	25	Captain America: Civil War	2016	$1,151,899,586	$408,084,349	$743,815,237
26	26	26	Aquaman	2018	$1,143,758,700	$335,061,807	$808,696,893
27	27	27	Spider-Man: Far From Home	2019	$1,132,107,522	$390,532,085	$741,575,437
28	28	28	Captain Marvel	2019	$1,129,576,094	$426,829,839	$702,746,255
29	29	29	Transformers: Dark of the Moon	2011	$1,123,794,079	$352,390,543	$771,403,536
30	30	30	The Lord of the Rings: The Return of …	2003	$1,121,386,981	$379,021,990	$742,364,991
31	31	31	Skyfall	2012	$1,110,526,981	$304,360,277	$806,166,704
32	32	32	Transformers: Age of Extinction	2014	$1,104,054,072	$245,439,076	$858,614,996
33	33	33	The Dark Knight Rises	2012	$1,082,228,107	$448,139,099	$634,089,008
34	34	34	Toy Story 4	2019	$1,073,064,540	$434,038,008	$639,026,532
35	35	35	Star Wars: The Rise of Skywalker	2019	$1,072,767,997	$515,202,542	$557,565,455
36	36	36	Joker	2019	$1,069,121,583	$335,451,311	$733,670,272
37	37	37	Toy Story 3	2010	$1,068,879,522	$415,004,880	$653,874,642
38	38	38	Pirates of the Caribbean: Dead Man's …	2006	$1,066,179,725	$423,315,812	$642,863,913
39	39	39	Rogue One: A Star Wars Story	2016	$1,055,083,596	$533,539,991	$521,543,605
40	40	40	Aladdin	2019	$1,046,587,513	$355,559,216	$691,028,297
41	41	41	Pirates of the Caribbean: On Stranger…	2011	$1,045,713,802	$241,071,802	$804,642,000
42	42	42	Jurassic Park	1993	$1,045,573,035	$402,523,348	$643,049,687
43	43	43	Despicable Me 3	2017	$1,032,809,657	$264,624,300	$768,185,357
44	44	44	Star Wars Ep. I: The Phantom Menace	1999	$1,027,044,677	$474,544,677	$552,500,000
45	45	45	Alice in Wonderland	2010	$1,025,491,110	$334,191,110	$691,300,000
46	46	46	Finding Dory	2016	$1,025,006,125	$486,295,561	$538,710,564
47	47	47	The Hobbit: An Unexpected Journey	2012	$1,014,938,545	$303,003,568	$711,934,977
48	48	48	The Dark Knight	2008	$1,006,234,167	$534,987,076	$471,247,091
49	49	49	Jurassic World: Dominion	2022	$1,003,775,632	$376,851,080	$626,924,552
50	50	50	Zootopia	2016	$1,002,462,578	$341,268,248	$661,194,330

Data Pre-processing:

Now the most important phase of the data analysis starts which is data pre-processing. We are going to use pre-processing techniques on this dataset to prepare a complete dataset for analysis and visualization.

1) Data Cleaning:

1) ***Handling Missing Data:*** To handle missing data we first need to search the data set for any value that is not assigned. To do so we write a code that will show us the row which contains the missing value.

Source code:

```
#Dropping the X column
Movies_DF <- Movies_DF[c(-1)]

#Handling Missing Data
#Replacing "missing value" with NA
Movies_DF[Movies_DF == ""] <- NA
print(Movies_DF)
```

```
#Dropping rows with NA
Movies_DF<- na.omit(Movies_DF)

#Removing $ and , from the dataset
Movies_DF$rank <- gsub("\\$|,", "", as.character(Movies_DF$rank))
Movies_DF$worldwide_box_office <- gsub("\\$|,", "",
as.character(Movies_DF$worldwide_box_office))
Movies_DF$domestic_box_office<-gsub("\\$|,", "",
as.character(Movies_DF$domestic_box_office))
Movies_DF$international_box_office<-gsub("\\$|,", "",
as.character(Movies_DF$international_box_office))
```

Output:

```
#Handling Missing Data
#Replacing "missing value" with NA
Movies_DF[Movies_DF == ""] <- NA

#Dropping rows with NA
Movies_DF<- na.omit(Movies_DF)

#Removing $ and , from the dataset
Movies_DF$rank <- gsub("\\$|,", "", as.character(Movies_DF$rank))
Movies_DF$worldwide_box_office <- gsub("\\$|,", "", as.character(Movies_DF$worldwide_box_office))
Movies_DF$domestic_box_office<-gsub("\\$|,", "", as.character(Movies_DF$domestic_box_office))
Movies_DF$international_box_office<-gsub("\\$|,", "", as.character(Movies_DF$international_box_office))
```

2) ***Smooth Noisy Data:*** In the dataset, there were many outliers so first we identified them using several methods and removed them.

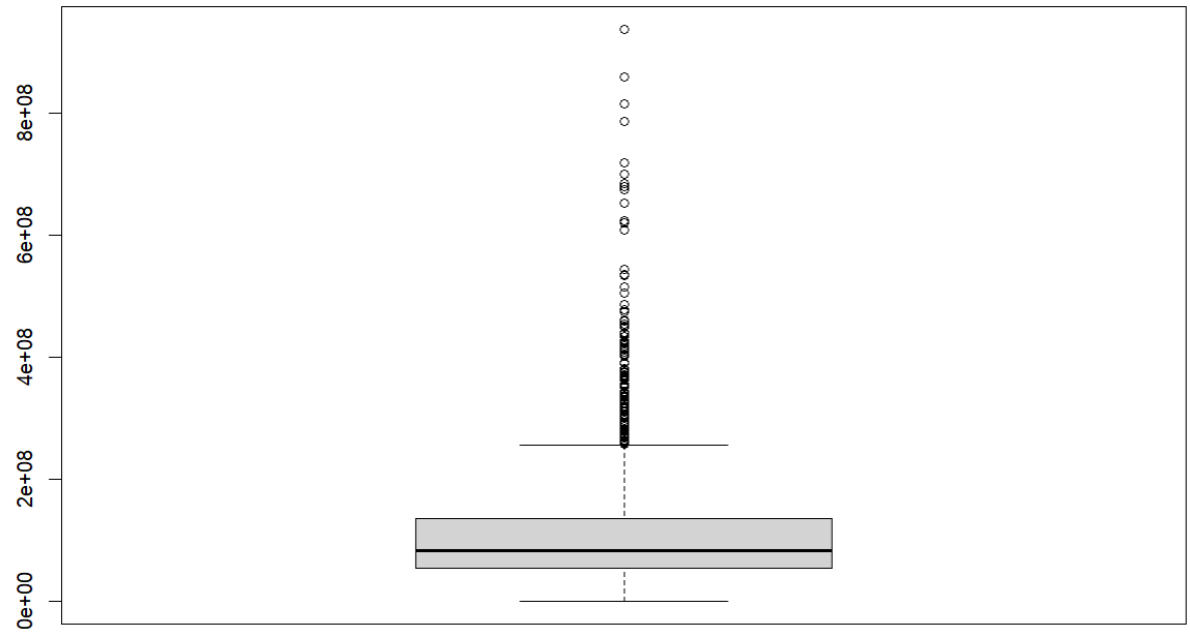
Source Code:

```
#Smoothing Noisy Data
#Finding Outliers
max(Movies_DF$worldwide_box_office)
min(Movies_DF$worldwide_box_office)
boxplot(Movies_DF$worldwide_box_office)
worldwide_box_office_sorted <-
Movies_DF[order(Movies_DF$worldwide_box_office),"worldwide_box_office"]
worldwide_box_office_sorted

max(Movies_DF$international_box_office)
min(Movies_DF$international_box_office)
boxplot(Movies_DF$international_box_office)
international_box_office_sorted <-
Movies_DF[order(Movies_DF$international_box_office),"international_box_office"]
international_box_office_sorted

max(Movies_DF$domestic_box_office)
min(Movies_DF$domestic_box_office)
boxplot(Movies_DF$domestic_box_office)
```

```
domestic_box_office_sorted <-
Movies_DF[order(Movies_DF$domestic_box_office),"domestic_box_office"]
domestic_box_office_sorted

#Removing Outlier
Movies_DF <- subset(Movies_DF, international_box_office >= 100000)
Movies_DF <- subset(Movies_DF, domestic_box_office >= 100000)
```

Output:

```
> #Smoothing Noisy Data
> #Finding Outliers
> max(Movies_DF$worldwide_box_office)
[1] 2923706026
> min(Movies_DF$worldwide_box_office)
[1] 85527867
> boxplot(Movies_DF$worldwide_box_office)
> worldwide_box_office_sorted <- Movies_DF[order(Movies_DF$worldwide_box_office),"worldwide_box_office"]
> worldwide_box_office_sorted

> max(Movies_DF$international_box_office)
[1] 2138484377
> min(Movies_DF$international_box_office)
[1] 81
> boxplot(Movies_DF$international_box_office)
> international_box_office_sorted <- Movies_DF[order(Movies_DF$international_box_office),"international_box_office"]
> international_box_office_sorted
```



1) Data Munging:

The dataset does not require munging because all the data are within the same range.

2) Data Integration:

The dataset does not require data integration as all the data are taken from the same dataset.

3) Data Transformation:

In this phase, we need to transform some variables for better analysis of the dataset. We need to transform the variables such as rank, year & the box office revenues to numeric.

Source code:

```
#Data Transformation
#Converting data into numeric
Movies_DF <- transform(Movies_DF,
                       rank = as.numeric(rank),
                       year = as.numeric(year),
                       worldwide_box_office = as.numeric(worldwide_box_office),
                       domestic_box_office = as.numeric(domestic_box_office),
                       international_box_office = as.numeric(international_box_office))
```

Output:

```
> #Data Transformation
> #Converting data into numeric
> Movies_DF <- transform(Movies_DF,
+                        rank = as.numeric(rank),
+                        year = as.numeric(year),
+                        worldwide_box_office = as.numeric(worldwide_box_office),
+                        domestic_box_office = as.numeric(domestic_box_office),
+                        international_box_office = as.numeric(international_box_office))
> |
```

4) Data Reduction:

In the data reduction part, we converted the income in millions and added them in new column and changed their name.

Source code:

```
#Data Reduction
# Create a new column called "worldwide_box_office_million" by dividing
"worldwide_box_office" by 1 million
Movies_DF$worldwide_box_office_million <- Movies_DF$worldwide_box_office /
1000000

# Create a new column called "domestic_box_office_million" by dividing
"domestic_box_office" by 1 million
Movies_DF$domestic_box_office_million <- Movies_DF$domestic_box_office /
1000000
```

```
# Create a new column called "international_box_office_million" by dividing
"international_box_office" by 1 million
Movies_DF$international_box_office_million <- Movies_DF$international_box_office
/ 1000000

#Dropping the column
Movies_DF <- Movies_DF[c(-4,-5,-6)]

#Renaming the column
library(dplyr)
Movies_DF <- rename(Movies_DF, worldwide_box_office =
"worldwide_box_office_million",
          domestic_box_office= "domestic_box_office_million",
          international_box_office = "international_box_office_million")
```

Output:

```
> #Data Reduction
> # Create a new column called "worldwide_box_office_million" by dividing "worldwide_box_office" by 1 million
> Movies_DF$worldwide_box_office_million <- Movies_DF$worldwide_box_office / 1000000
>
> # Create a new column called "domestic_box_office_million" by dividing "domestic_box_office" by 1 million
> Movies_DF$domestic_box_office_million <- Movies_DF$domestic_box_office / 1000000
>
> # Create a new column called "international_box_office_million" by dividing "international_box_office" by 1 million
> Movies_DF$international_box_office_million <- Movies_DF$international_box_office / 1000000
>
> #Dropping the column
> Movies_DF <- Movies_DF[c(-4,-5,-6)]
```

5) Data Discretization:

No discretization is needed for this dataset as it is already in better shape. So, we skip this process and move on to descriptive statistics.

Descriptive Statistics:

Now, we are going to compute various descriptive statistics parameters for our dataset. Firstly, let's try to inspect the central tendency for the various variables of our dataset.

1) Mean:

Mean of worldwide box office, domestic box office & international box office.

Source code:

```
#Mean
mean_worldwide_box_office <- mean(Movies_DF$worldwide_box_office)
paste("Mean of worldwide box office :", mean_worldwide_box_office)

mean_domestic_box_office <- mean(Movies_DF$domestic_box_office)
paste("Mean of domestic box office :", mean_domestic_box_office)

mean_international_box_office <- mean(Movies_DF$international_box_office)
paste("Mean of international box office :", mean_international_box_office)
```

Output:

```
> #Mean
> mean_worldwide_box_office <- mean(Movies_DF$worldwide_box_office)
> paste("Mean of worldwide box office :", mean_worldwide_box_office)
[1] "Mean of worldwide box office : 276847394.236045"
> mean_domestic_box_office <- mean(Movies_DF$domestic_box_office)
> paste("Mean of domestic box office :", mean_domestic_box_office)
[1] "Mean of domestic box office : 111679151.810739"
>
> mean_international_box_office <- mean(Movies_DF$international_box_office)
> paste("Mean of international box office :", mean_international_box_office)
[1] "Mean of international box office : 165168242.425306"
>
```

2) Median:

Now we calculate the median for the amount of worldwide box office, domestic box office & international box office.

Source code:

```
#Median
median_worldwide_box_office <- median(Movies_DF$worldwide_box_office)
paste("Median of worldwide box office :", median_worldwide_box_office)

median_domestic_box_office <- median(Movies_DF$domestic_box_office)
paste("Median of domestic box office :", median_domestic_box_office)

median_international_box_office <- median(Movies_DF$international_box_office)
paste("Median of international box office :", median_international_box_office)
```

Output:

```
> #Median
> median_worldwide_box_office <- median(Movies_DF$worldwide_box_office)
> paste("Median of worldwide box office :", median_worldwide_box_office)
[1] "Median of worldwide box office : 183291893"
>
> median_domestic_box_office <- median(Movies_DF$domestic_box_office)
> paste("Median of domestic box office :", median_domestic_box_office)
[1] "Median of domestic box office : 83586447"
>
> median_international_box_office <- median(Movies_DF$international_box_office)
> paste("Median of international box office :", median_international_box_office)
[1] "Median of international box office : 103980200"
```

3) Mode:

As the mode doesn't have a built-in function, we first implement the function.

Source code:

```
#Mode
mode <- function(x){
   unique_values <- unique(x)
    table <- tabulate(match(x, unique_values))
    unique_values[table == max(table)]
}
paste("Mode of year :",mode(Movies_DF$year))
```

Output:

```
> #Mode
> mode <- function(x){
+   unique_values <- unique(x)
+   table <- tabulate(match(x, unique_values))
+   unique_values[table == max(table)]
+ }
>
> paste("Mode of year :",mode(Movies_DF$year))
[1] "Mode of year : 2016"
```

4) Range:

Now we calculate the range of variables.

Source code:

```
#Range
range_worldwide_box_office <- max(Movies_DF$worldwide_box_office) -
min(Movies_DF$worldwide_box_office)
paste("Range of worldwide box office :", range_worldwide_box_office)

range_domestic_box_office <- max(Movies_DF$domestic_box_office) -
min(Movies_DF$domestic_box_office)
paste("Range of domestic box office :", range_domestic_box_office)

range_international_box_office <- max(Movies_DF$international_box_office) -
min(Movies_DF$international_box_office)
paste("Range of international box office :", range_international_box_office)
```

Output:

```
> #Range
> range_worldwide_box_office <- max(Movies_DF$worldwide_box_office) - min(Movies_DF$worldwide_box_office)
> paste("Range of worldwide box office :", range_worldwide_box_office)
[1] "Range of worldwide box office : 2838178159"
>
> range_domestic_box_office <- max(Movies_DF$domestic_box_office) - min(Movies_DF$domestic_box_office)
> paste("Range of domestic box office :", range_domestic_box_office)
[1] "Range of domestic box office : 936660433"
>
> range_international_box_office <- max(Movies_DF$international_box_office) - min(Movies_DF$international_box_office)
> paste("Range of international box office :", range_international_box_office)
[1] "Range of international box office : 2138484296"
```

5) Variance:

Source code:

```
#variance
variance_worldwide_box_office <- var(Movies_DF$worldwide_box_office)
paste("Variance of worldwide box office :", variance_worldwide_box_office)

variance_domestic_box_office <- var(Movies_DF$domestic_box_office)
paste("Variance of domestic box office :", variance_domestic_box_office)

variance_international_box_office <- var(Movies_DF$international_box_office)
paste("Variance of international box office :", variance_international_box_office)
```

Output:

```
> #variance
> variance_worldwide_box_office <- var(Movies_DF$worldwide_box_office)
> paste("Variance of worldwide box office :", variance_worldwide_box_office)
[1] "Variance of worldwide box office : 69170097960558320"
>
> variance_domestic_box_office <- var(Movies_DF$domestic_box_office)
> paste("Variance of domestic box office :", variance_domestic_box_office)
[1] "Variance of domestic box office : 9743125685743832"
>
> variance_international_box_office <- var(Movies_DF$international_box_office)
> paste("Variance of international box office :", variance_international_box_office)
[1] "Variance of international box office : 32717288595471256"
```

6) Standard Deviation:

Source code:

```
#Standard Deviation
standard_deviation_worldwide_box_office <- sd(Movies_DF$worldwide_box_office)
paste("Standard Deviation of worldwide box office :",
standard_deviation_worldwide_box_office)

standard_deviation_domestic_box_office <- sd(Movies_DF$domestic_box_office)
paste("Standard Deviation of domestic box office :",
standard_deviation_domestic_box_office)

standard_deviation_international_box_office <-
sd(Movies_DF$international_box_office)
paste("Standard Deviation of international box office :",
standard_deviation_international_box_office)
```

Output:

```
> #Standard Deviation
> standard_deviation_worldwide_box_office <- sd(Movies_DF$worldwide_box_office)
> paste("Standard Deviation of worldwide box office :", standard_deviation_worldwide_box_office)
[1] "Standard Deviation of worldwide box office : 263002087.369204"
>
> standard_deviation_domestic_box_office <- sd(Movies_DF$domestic_box_office)
> paste("Standard Deviation of domestic box office :", standard_deviation_domestic_box_office)
[1] "Standard Deviation of domestic box office : 98707272.7094809"
>
> standard_deviation_international_box_office <- sd(Movies_DF$international_box_office)
> paste("Standard Deviation of international box office :", standard_deviation_international_box_office)
[1] "Standard Deviation of international box office : 180879209.959219"
```

7) Quantile:

Source code:

```
#Quantile
quantile(Movies_DF$worldwide_box_office)

quantile(Movies_DF$domestic_box_office)

quantile(Movies_DF$international_box_office)
```

Output:

```
> #Quantile
> quantile(Movies_DF$worldwide_box_office)
       0%        25%        50%        75%       100%
 85527867  122519874  183291893  321887208 2923706026
>
> quantile(Movies_DF$domestic_box_office)
      0%       25%       50%       75%      100%
    1792  54979992  83586447 135560942 936662225
>
> quantile(Movies_DF$international_box_office)
       0%        25%        50%        75%       100%
       81   61972530  103980200  196027687 2138484377
```

8) Percentiles:

Source code:

```
#Percentiles
percentiles_worldwide_box_office <- IQR(Movies_DF$worldwide_box_office)
paste("Percentiles of worldwide box office :", percentiles_worldwide_box_office)
percentiles_domestic_box_office <- IQR(Movies_DF$domestic_box_office)
paste("Percentiles of domestic box office :", percentiles_domestic_box_office)

percentiles_international_box_office <- IQR(Movies_DF$international_box_office)
paste("Percentiles of international box office :", percentiles_international_box_office)
```

Output:

```
> #Percentiles
> percentiles_worldwide_box_office <- IQR(Movies_DF$worldwide_box_office)
> paste("Percentiles of worldwide box office :", percentiles_worldwide_box_office)
[1] "Percentiles of worldwide box office : 199367334"
>
> percentiles_domestic_box_office <- IQR(Movies_DF$domestic_box_office)
> paste("Percentiles of domestic box office :", percentiles_domestic_box_office)
[1] "Percentiles of domestic box office : 80580950"
>
> percentiles_international_box_office <- IQR(Movies_DF$international_box_office)
> paste("Percentiles of international box office :", percentiles_international_box_office)
[1] "Percentiles of international box office : 134055157"
```

Data Visualization:

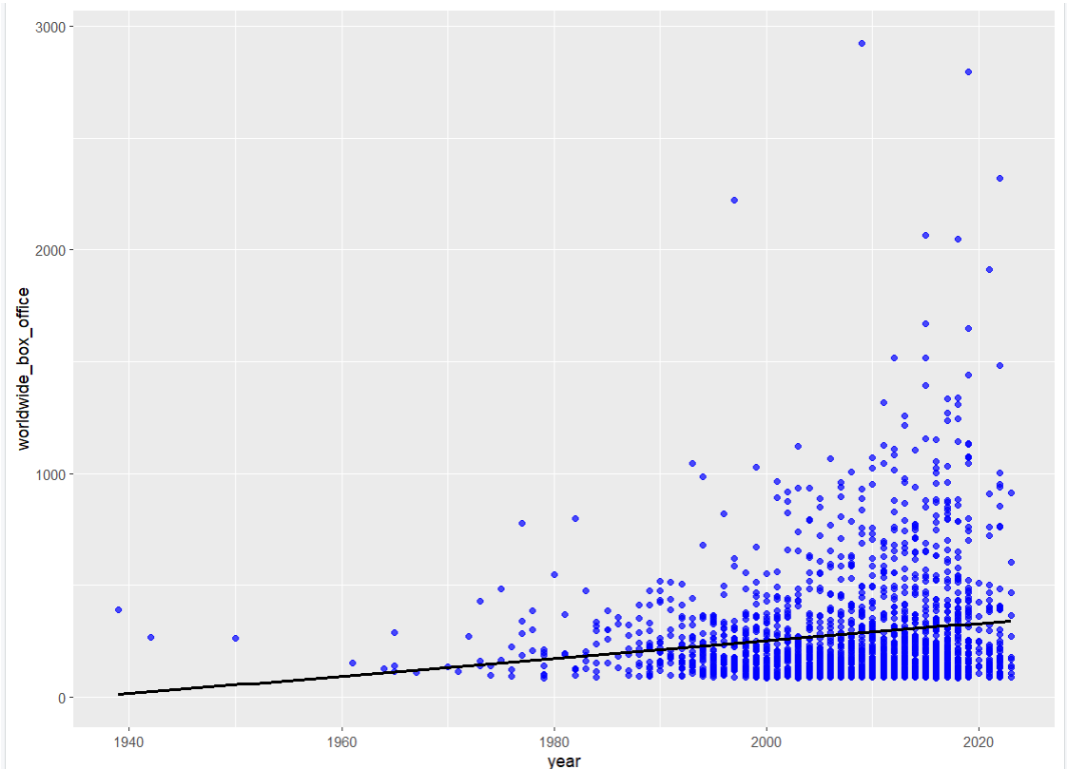
Now we plot point, pie chart, bar chart & density to represent the data.

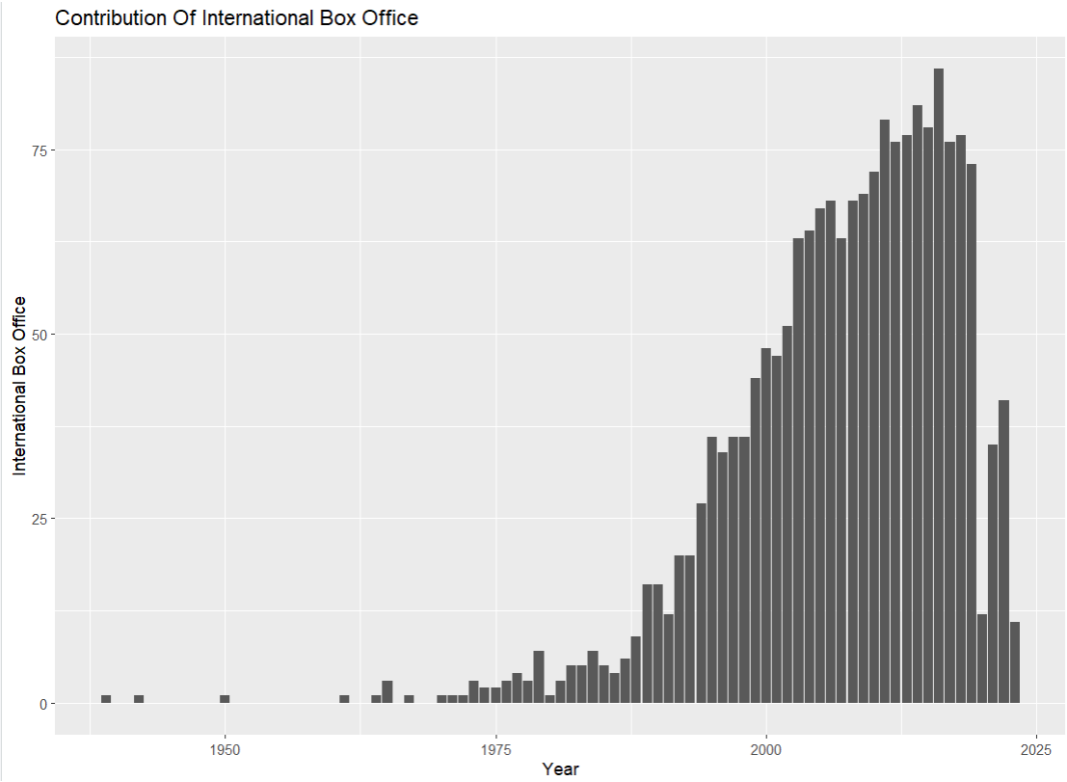
Source code:

```
#Data Visualization
#Geom-point
library(ggplot2)
ggplot(data = Movies_DF, mapping = aes(x = year, y = worldwide_box_office)) +
geom_point(color='blue', alpha = .7, size = 1.5)+geom_smooth(color="black",method =lm,
se= FALSE)
```

```
#piechart
library(ggpie)
Movies_DF %>% ggpie(group_key = "year",count_type = "full", label_type = "circle",
           label_info = "ratio", label_pos = "out",label_size = 3,nudge_x = 20)


#Geom-bar
ggplot(Movies_DF,aes(x=year, fill=international_box_office))+
  geom_bar()+
  labs(title = "Contribution Of International Box Office", x ="Year", y="International Box
Office")


Movies_DF %>% ggplot(aes(x= year,y= worldwide_box_office, fill=year))+
  geom_bar(stat = "identity")+
  labs(x="Year",y="Worldwide_box_office", title = "Year By Worldwide Box Office")


#Density
Movies_DF%>% ggplot(aes(x= year, y= domestic_box_office))+
  geom_density(stat = "identity", fill="red", bw= 1)+
  labs(x="Year",y="Domestic Box Office", title = "Year Vs Domestic Box Office")

#New Data Frame

# Create a new dataframe from a subset of the old dataframe "Movies_DF"
New_Movies_DF <- Movies_DF[, c("name", "year",
"domestic_box_office","international_box_office",
                        "worldwide_box_office")]

# Change the column names of the new dataframe
colnames(New_Movies_DF) <- c("Name", "Year", "Domestic Box Office ($ Million)",
"International Box Office ($ Million)",
                      "Worldwide Box Office ($ Million)")
```

Output:


3000
2000
1000
0
worldwide_box_office
1940
1960
1980
2000
2020
year


Contribution Of International Box Office
75
50
25
0
International Box Office
1950
1975
2000
2025
Year

Shiny Dashboard Implementation:

For the shiny dashboard implementation, we tried to create a reactive app based on our topic. We tried to show a reactive line plot, scatter plot and a bar plot.

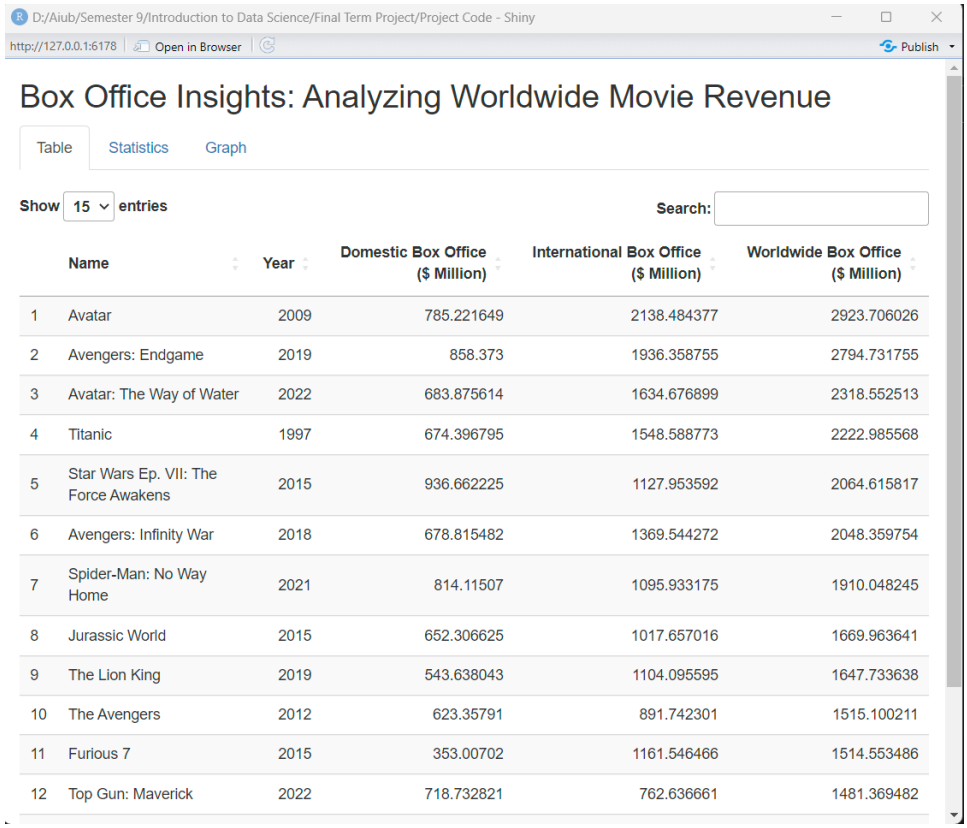
Source code:

```
#Interactive dashboard
library(shiny)
library (DT)
library(ggplot2)

ui = fluidPage(
 titlePanel("Box Office Insights: Analyzing Worldwide Movie Revenue"),
 tabsetPanel(
  tabPanel("Table", div(dataTableOutput("table"), style="margin-top: 20px")),
  tabPanel("Statistics",
        verbatimTextOutput("stats1"),
        verbatimTextOutput("stats2"),
        verbatimTextOutput("stats3"),
        verbatimTextOutput("stats4")),

  tabPanel("Graph",
        fluidRow(
         column(3,
              wellPanel(
               selectInput("x_var", "X Variable",
                        choices = c("domestic_box_office",
"international_box_office","worldwide_box_office","year"),
                        selected = "year"
               )
              )
         ),
         column(3,
              wellPanel(
               selectInput("y_var", "Y Variable",
                        choices = c("domestic_box_office",
"international_box_office","worldwide_box_office","year"),
                        selected = "domestic_box_office"
               )
              )
         ),
         column(3,
              wellPanel(
               selectInput("plot_type", "Select Plot Type",
                        choices = c("Point", "Bar", "Density"),
                        selected = "Point"
```

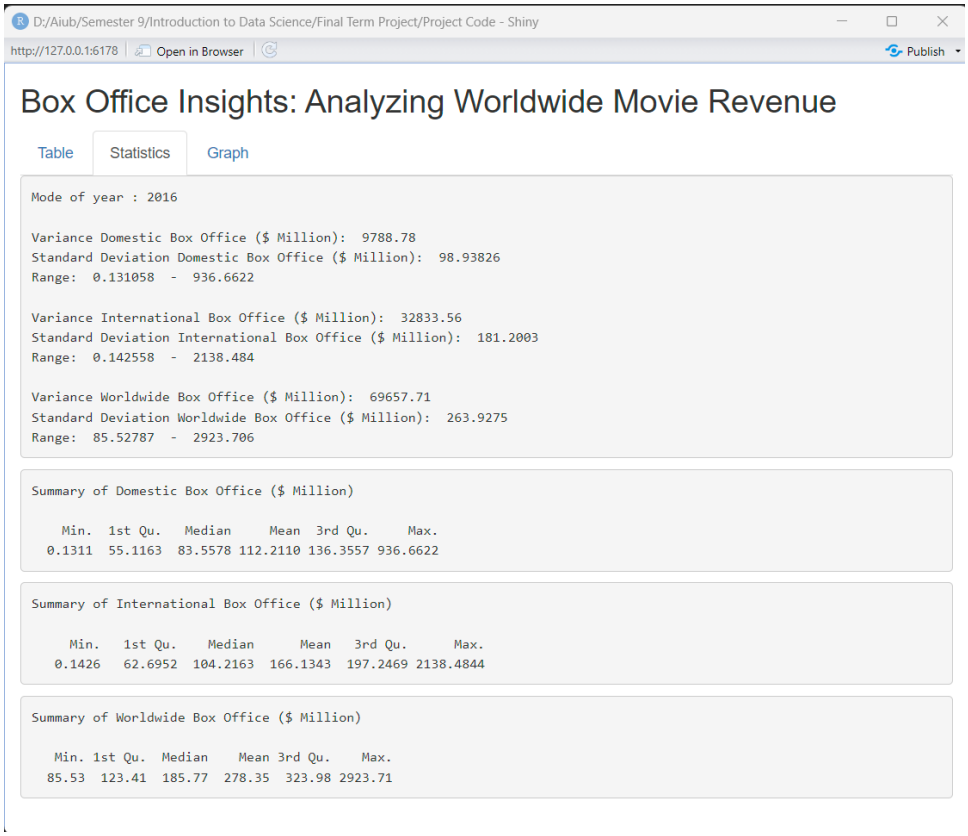
```
                  )
                )
             )
           ),
           plotOutput("plot"))
  )
)


server = function(input, output) {
  output$table <- renderDataTable({
    datatable(New_Movies_DF,
              options = list(pageLength = 15,
                             lengthMenu = c(5, 10, 15, 20),
                             searching = TRUE))
  })

  output$stats1 <- renderPrint({
    # Calculate the variance, standard deviation and range
    variance_domestic <- var(Movies_DF$domestic_box_office)
    sd_domestic <- sd(Movies_DF$domestic_box_office)
    range_domestic <- range(Movies_DF$domestic_box_office)

    variance_international <- var(Movies_DF$international_box_offic)
    sd_international <- sd(Movies_DF$international_box_offic)
    range_international <- range(Movies_DF$international_box_office)

    variance_worldwide <- var(Movies_DF$worldwide_box_office)
    sd_worldwide <- sd(Movies_DF$worldwide_box_office)
    range_worldwide <- range(Movies_DF$worldwide_box_office)

    # Print the results
    cat("Mode of year :",mode(Movies_DF$year), "\n\n")
    cat("Variance Domestic Box Office ($ Million): ", variance_domestic, "\n")
    cat("Standard Deviation Domestic Box Office ($ Million): ", sd_domestic, "\n")
    cat("Range: ", range_domestic[1], " - ", range_domestic[2], "\n")
    cat("\n")
    cat("Variance International Box Office ($ Million): ", variance_international, "\n")
    cat("Standard Deviation International Box Office ($ Million): ", sd_international, "\n")
    cat("Range: ", range_international[1], " - ", range_international[2], "\n")
    cat("\n")
    cat("Variance Worldwide Box Office ($ Million): ", variance_worldwide, "\n")
    cat("Standard Deviation Worldwide Box Office ($ Million): ", sd_worldwide, "\n")
    cat("Range: ", range_worldwide[1], " - ", range_worldwide[2], "\n")

  })
```

```
  output$stats2 <- renderPrint({
    cat("Summary of Domestic Box Office ($ Million)", "\n\n")
    summary(Movies_DF$domestic_box_office)
  })

  output$stats3 <- renderPrint({
    cat("Summary of International Box Office ($ Million)", "\n\n")
    summary(Movies_DF$international_box_office)
  })

  output$stats4 <- renderPrint({
    cat("Summary of Worldwide Box Office ($ Million)", "\n\n")
    summary(Movies_DF$worldwide_box_office)
  })

  output$plot <- renderPlot({

    # get user inputs
    x_col <- input$x_var
    y_col <- input$y_var
    plot_type <- input$plot_type

    # check if the input columns exist in the data frame
    if(!all(c(x_col, y_col) %in% colnames(Movies_DF))) {
      return(NULL)
    }

    # create plot based on selected plot type
    if(plot_type == "Point") {
      ggplot(Movies_DF, aes_string(x = x_col, y = y_col)) +
        geom_point() +
        labs(x = x_col, y = y_col)
    } else if (plot_type == "Bar") {
      ggplot(Movies_DF, aes_string(x = x_col, y = y_col, fill = factor(Movies_DF$year))) +
        geom_bar(stat = "identity") +
        labs(x = x_col, y = y_col, fill = "Year")
    } else if (plot_type == "Density") {
      ggplot(Movies_DF, aes_string(x = y_col)) +
        geom_density() +
        labs(x = y_col, y = "Density")
    }
  })
}

shinyApp(ui, server)
```

Output:



	Name	Year	Domestic Box Office ($ Million)	International Box Office ($ Million)	Worldwide Box Office ($ Million)
1	Avatar	2009	785.221649	2138.484377	2923.706026
2	Avengers: Endgame	2019	858.373	1936.358755	2794.731755
3	Avatar: The Way of Water	2022	683.875614	1634.676899	2318.552513
4	Titanic	1997	674.396795	1548.588773	2222.985568
5	Star Wars Ep. VII: The Force Awakens	2015	936.662225	1127.953592	2064.615817
6	Avengers: Infinity War	2018	678.815482	1369.544272	2048.359754
7	Spider-Man: No Way Home	2021	814.11507	1095.933175	1910.048245
8	Jurassic World	2015	652.306625	1017.657016	1669.963641
9	The Lion King	2019	543.638043	1104.095595	1647.733638
10	The Avengers	2012	623.35791	891.742301	1515.100211
11	Furious 7	2015	353.00702	1161.546466	1514.553486
12	Top Gun: Maverick	2022	718.732821	762.636661	1481.369482

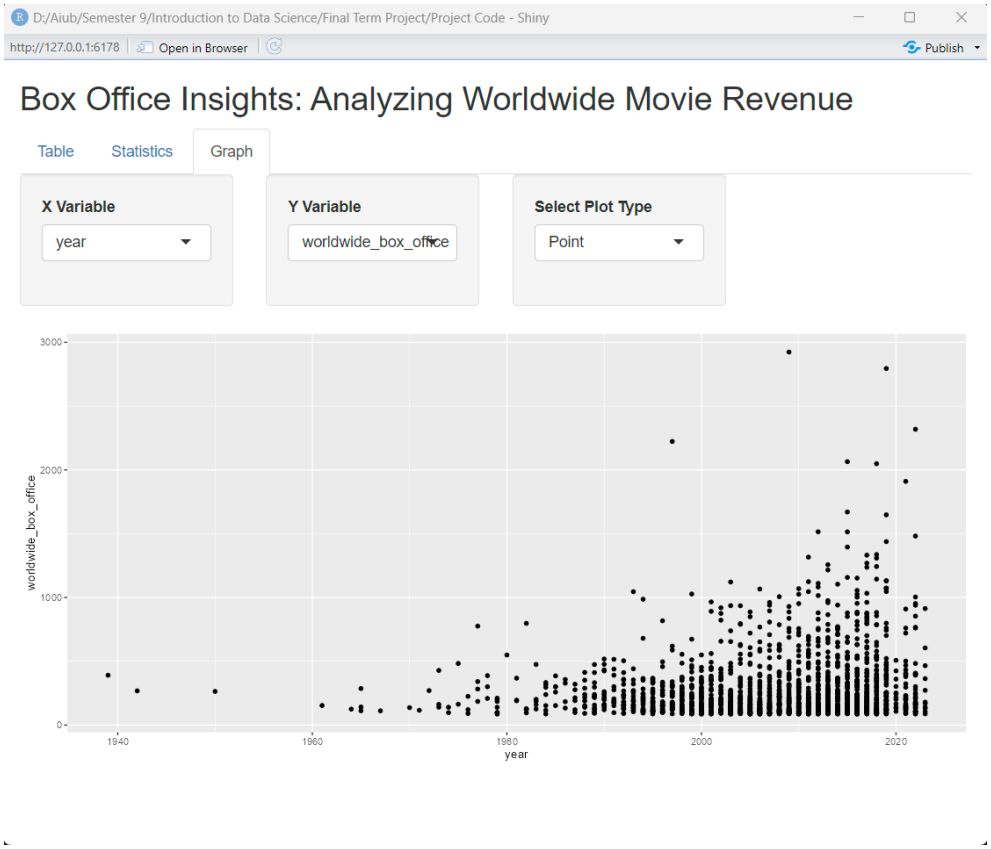


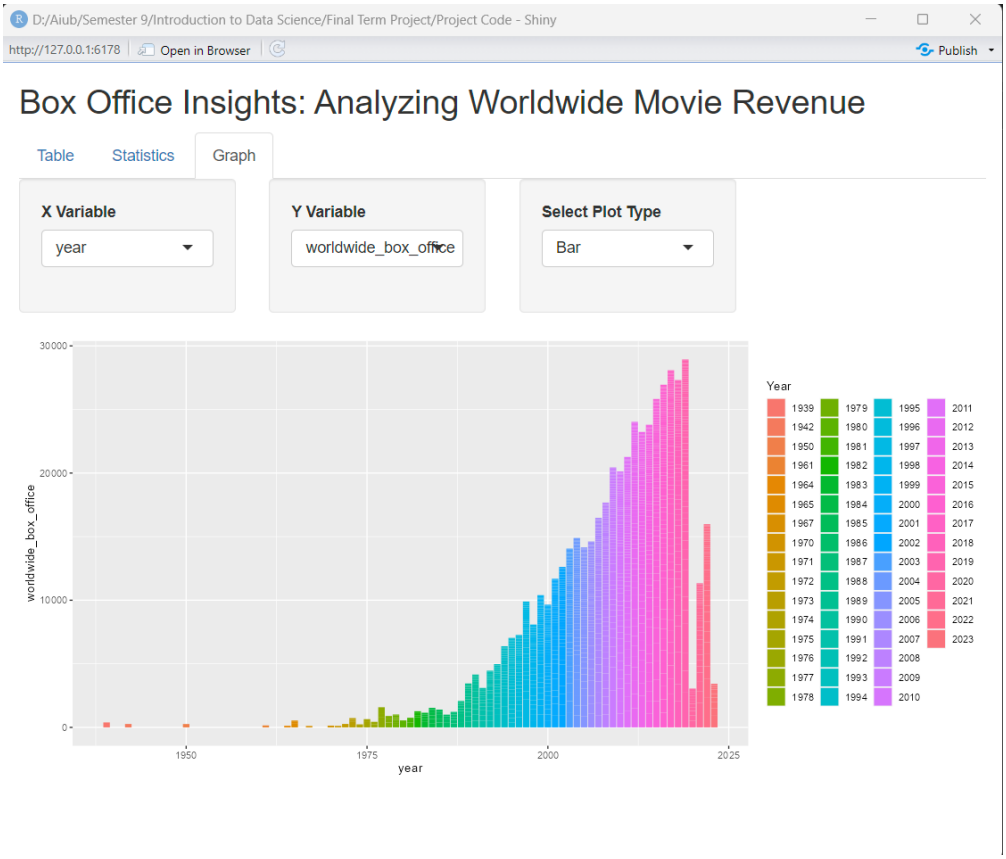

```
Mode of year : 2016

Variance Domestic Box Office ($ Million):  9788.78
Standard Deviation Domestic Box Office ($ Million):  98.93826
Range:  0.131058  -  936.6622

Variance International Box Office ($ Million):  32833.56
Standard Deviation International Box Office ($ Million):  181.2003
Range:  0.142558  -  2138.484

Variance Worldwide Box Office ($ Million):  69657.71
Standard Deviation Worldwide Box Office ($ Million):  263.9275
Range:  85.52787  -  2923.706
```

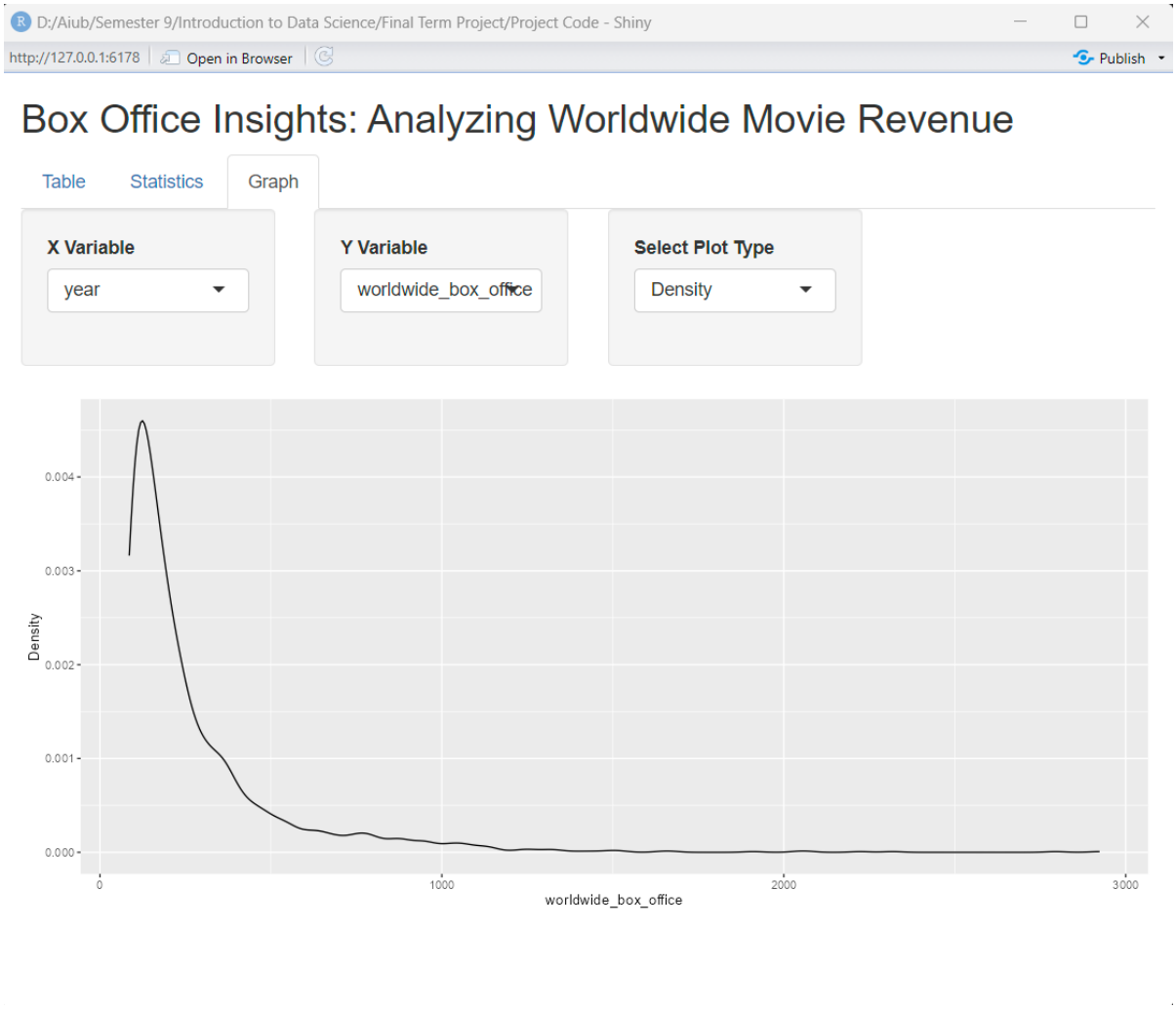
```
Summary of Domestic Box Office ($ Million)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.1311  55.1163  83.5578 112.2110 136.3557 936.6622
```

```
Summary of International Box Office ($ Million)

     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
   0.1426   62.6952  104.2163  166.1343  197.2469 2138.4844
```

```
Summary of Worldwide Box Office ($ Million)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  85.53  123.41  185.77  278.35  323.98 2923.71
```


D:/Aiub/Semester 9/Introduction to Data Science/Final Term Project/Project Code - Shiny
http://127.0.0.1:6178
Open in Browser
Publish
Box Office Insights: Analyzing Worldwide Movie Revenue
Table
Statistics
Graph
X Variable
year
Y Variable
worldwide_box_office
Select Plot Type
Point
worldwide_box_office
3000
2000
1000
0
1940
1960
1980
2000
2020
year


D:/Aiub/Semester 9/Introduction to Data Science/Final Term Project/Project Code - Shiny
http://127.0.0.1:6178
Open in Browser
Publish
Box Office Insights: Analyzing Worldwide Movie Revenue
Table
Statistics
Graph
X Variable
year
Y Variable
worldwide_box_office
Select Plot Type
Bar
worldwide_box_office
30000
20000
10000
0
1950
1975
2000
2025
year
Year
1939 1942 1950 1961 1964 1965 1967 1970 1971 1972 1973 1974 1975 1976 1977 1978
1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994
1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023



Discussion:

The project aimed to build an interactive dashboard using Shiny framework based on web scraping data from "The Numbers" website. The dashboard allows users to filter the movie rankings data by revenue type & year to get a more personalized view of the movie rankings.

Web scraping was used to collect the data from "The Numbers" website. The data collected included movie titles, release year, worldwide box office revenue, domestic box office revenue, and international box office revenue. Rvest package in R was used to scrap the data.

Shiny framework was used to build the interactive dashboard. The dashboard consists of three tabs, each displaying the table, statistical summary and graph. The dashboard was interactive for user to check and search from the table as well as generate various graphs.

The project has demonstrated the usefulness and potential of web scraping and interactive dashboard development using Shiny framework. The dashboard provides a valuable tool for anyone interested in analyzing and exploring movie rankings data. The interactive components of the dashboard make it user-friendly and responsive.

Conclusion:

In conclusion, the project has successfully developed an interactive dashboard using Shiny framework that displays information about movie rankings based on their worldwide box office revenue, domestic box office revenue, and international box office revenue. The project demonstrated the usefulness of web scraping in collecting data and the tidy verse package in R for data processing.

The interactive dashboard provides users with the ability to filter and explore the data in a personalized way. The dashboard is user-friendly and responsive, making it easy to use and navigate. The project has also demonstrated the potential of deploying the dashboard on a web server to make it accessible to the public.

Overall, this project has provided valuable insights into web scraping and interactive dashboard development using Shiny framework. It provides a valuable tool for anyone interested in analyzing and exploring movie rankings data. The project has shown that with the right tools and techniques, it is possible to collect and process large amounts of data and create interactive dashboards that are user-friendly and responsive.