# 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.

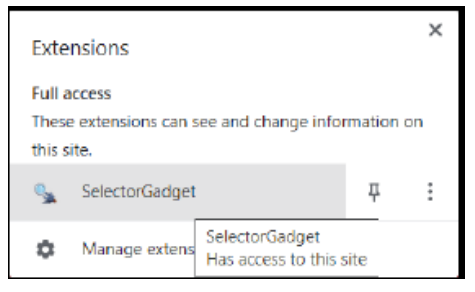
**Project Solution Design:**

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)



Graphical user interface, text

Description automatically generated

Chart

Description automatically generated

**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**

Table

Description automatically generated

**Data Pre-processing:**

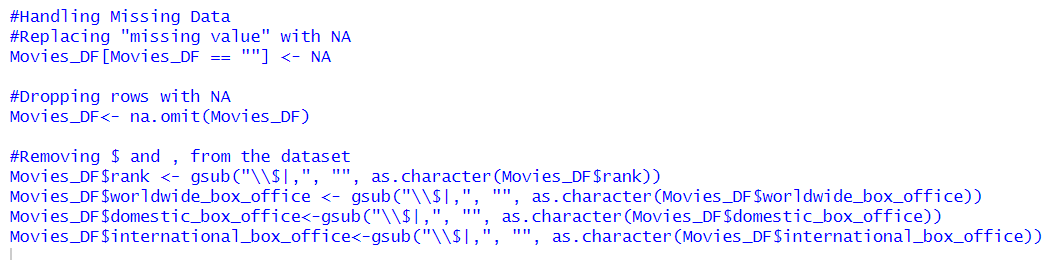
Now the most important phase of the data analysis starts which is data pre-processing. We are going to use pre-processing techniques on this dataset to prepare a complete dataset for analysis and visualization.

1. **Data Cleaning:**
2. ***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:**

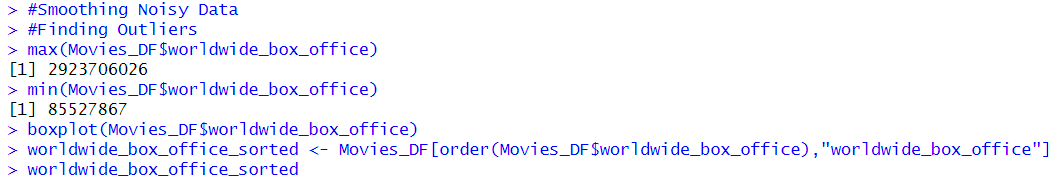
****

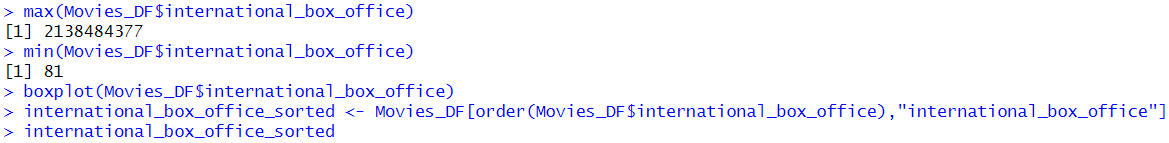
1. ***Smooth Noisy Data:*** In the dataset, there were many outliers so first we identified them using several methods and removed them.

**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:**

****

****

**Chart, box and whisker chart

Description automatically generated**

1. **Data Munging:**

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

1. **Data Integration:**

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

1. **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:**

**Text

Description automatically generated**

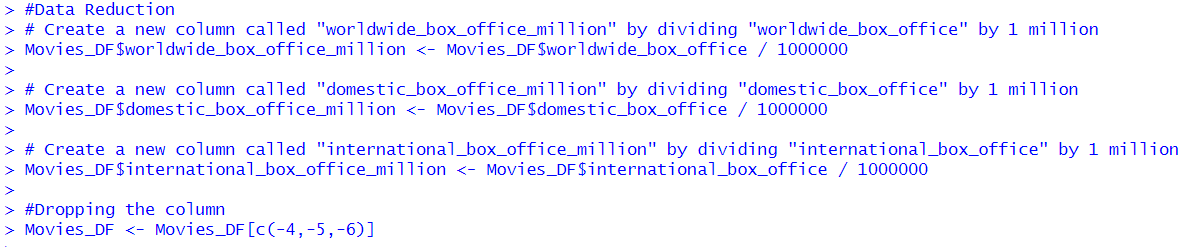
1. **Data Reduction:**

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:**

****

1. **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.

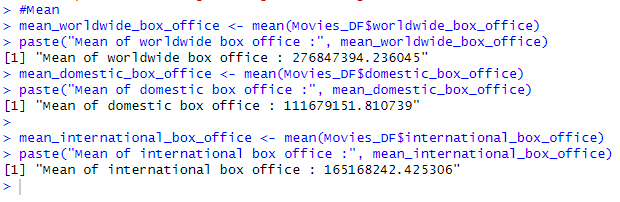
1. **Mean:**

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:**



1. **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:**

**Text, letter

Description automatically generated**

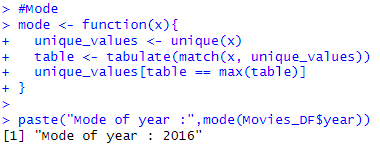
1. **Mode:**

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:**

****

1. **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:**

**Text

Description automatically generated**

1. **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:**

**Text

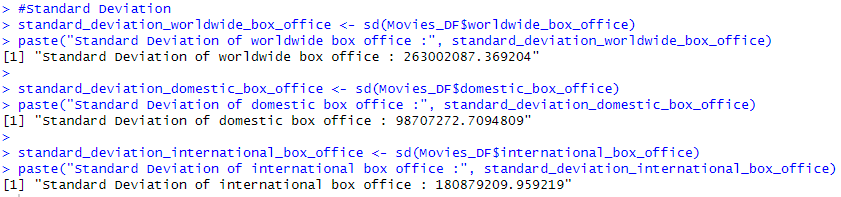
Description automatically generated**

1. **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:**



1. **Quantile:**

**Source code:**

|  |
| --- |
| #Quantile  quantile(Movies\_DF$worldwide\_box\_office)  quantile(Movies\_DF$domestic\_box\_office)  quantile(Movies\_DF$international\_box\_office) |

**Output:**

**Text

Description automatically generated**

1. **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:**

**Text, letter

Description automatically generated**

**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:**

**Chart, scatter chart

Description automatically generated**

**Chart, histogram

Description automatically generated**

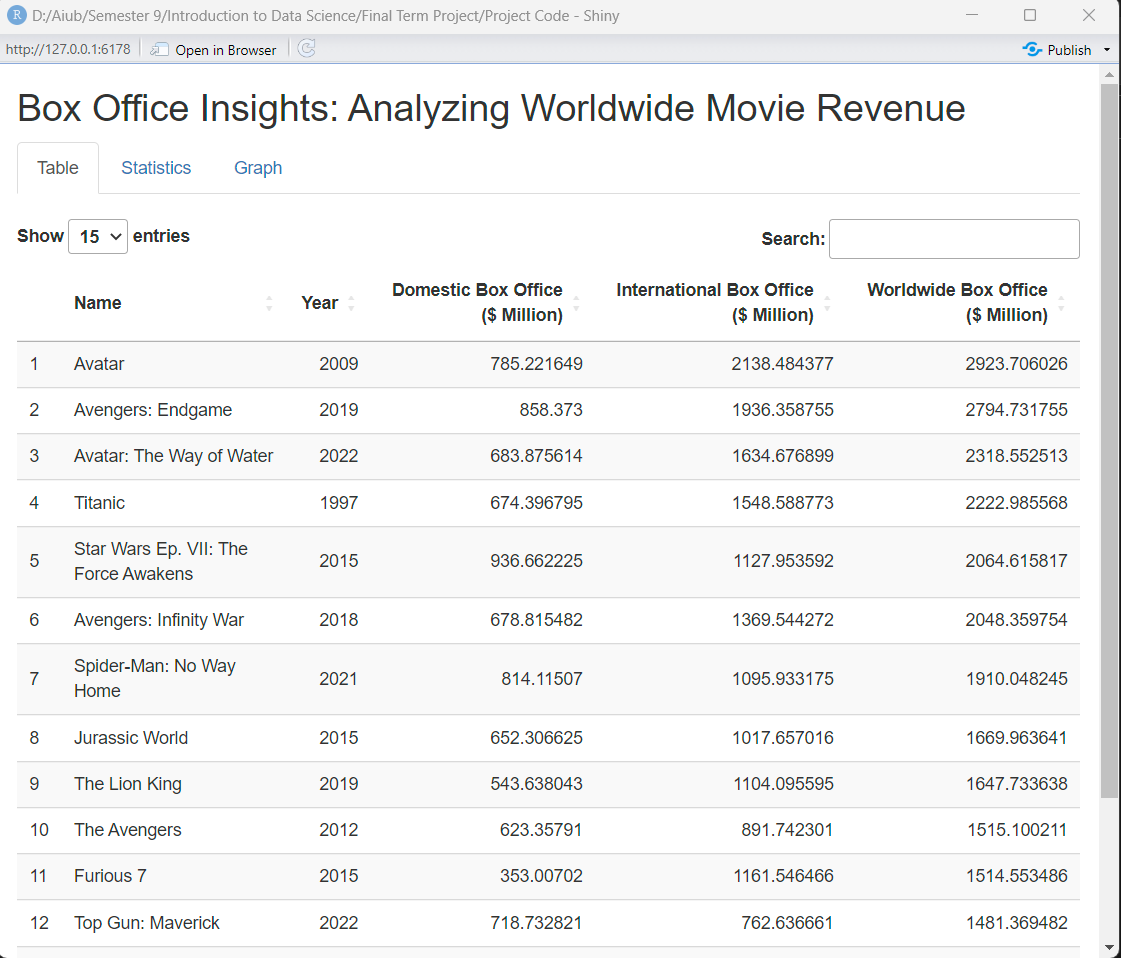
**Shiny Dashboard Implementation:**

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:**



Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, chart

Description automatically generated

Chart, histogram

Description automatically generated

Graphical user interface

Description automatically generated

**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.