Dale Randall

06-04-2023

R & Power BI Project

Table of Contents

1. Initial exploratory analysis  
2. Clean Data  
3. Outlier Removal  
4. Exploratory Data Analysis  
5. Export data  
6. Power BI Dashboard creation

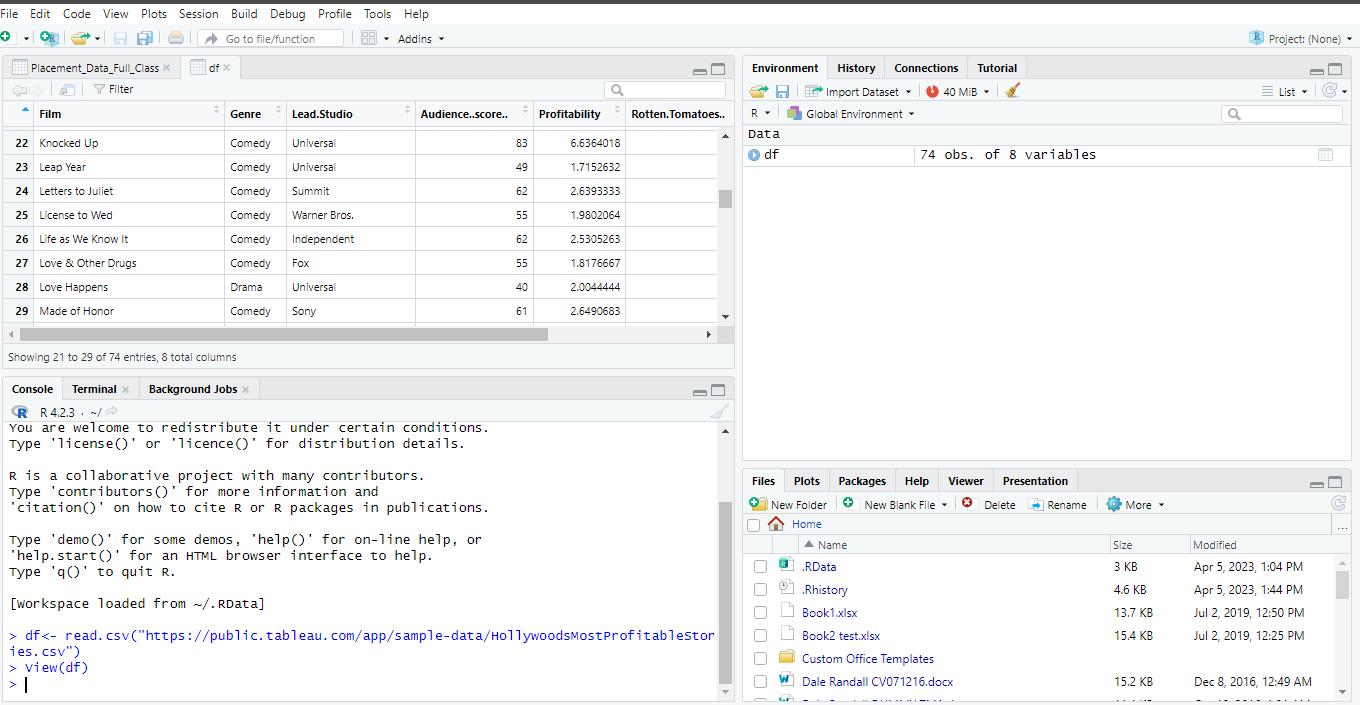
1. Initial exploratory analysis

Load data  
df<- read.csv(“https://public.tableau.com/app/sample data/HollywoodsMostProfitableStories.csv") 

df <- read.csv is a command in R language that reads a Comma-Separated Values (CSV) file into a data frame.  
The read.csv() function is used to read a CSV file and create a data frame from its contents. The df in this command is the name of the data frame where the data will be stored after reading the CSV file.  
In other words, this code creates a data frame in R and loads data from a CSV file into it. The data can then be manipulated and analysed using various R functions and packages.  
  
View data

View(df) is an R command that launches a data viewer window in RStudio or R console to display the contents of a data frame named df.

The View() function provides an interactive way to explore and examine the data in a data frame. It allows you to easily navigate through the rows and columns of the data, sort and filter the data, and perform other operations on the data using the GUI interface provided by the data viewer.  
In other words, the View() function provides a graphical representation of the data frame in a separate window, which allows you to easily visualize and explore the data frame.



Load library/ Install library  
install.packages("tidyverse") is an R command that installs the tidyverse package.  
  
The tidyverse is a collection of several R packages (including dplyr, ggplot2, tidyr, readr, etc.) that are designed to work together and provide a consistent and cohesive set of tools for data manipulation, visualization, and analysis.  
  
To use the functions and capabilities provided by the tidyverse package, it needs to be installed on the system where R is being used. The install.packages() function is used to install packages in R, and the argument passed to it is the name of the package to be installed.  
  
In other words, the install.packages("tidyverse") command downloads and installs the tidyverse package and any other packages it depends on, making the functions and capabilities of the tidyverse package available for use in R.  
  
Check data types  
str(df) is an R command that displays the structure of a data frame named df.  
  
The str() function is a powerful tool for exploring the structure of an R object. When applied to a data frame, str() displays a compact summary of the object that includes the number of rows and columns, the names and types of the columns, and a preview of the first few rows of data.  
In other words, the str(df) command provides a quick way to get an overview of the structure of a data frame in R, which can be useful for understanding the data, identifying any issues or inconsistencies in the data, and planning the next steps in data manipulation or analysis.

A screenshot of a computer

Description automatically generated with medium confidence

1. Clean Data

A screenshot of a computer

Description automatically generated with medium confidenceCheck for missing values  
colSums(is.na(df)) is an R command that returns a vector of the number of missing values in each column of a data frame df.  
  
The is.na(df) part of the command creates a logical matrix of the same dimensions as df, with TRUE values wherever there is a missing value in df and FALSE values wherever there is not a missing value in df. The colSums() function then calculates the column sums of this logical matrix, treating TRUE values as 1 and FALSE values as 0. The resulting vector displays the total number of missing values in each column of the data frame.  
For example, if you have a data frame df and you want to check how many missing values there are in each column, you can use the following command  
  
This will return a vector where each element represents the number of missing values in the corresponding column of df. This can be useful in identifying which columns have the most missing values and how much missing data there is in the data frame overall.

A screenshot of a computer

Description automatically generated with medium confidenceDrop/ remove missing values   
df <- na.omit(df) is an R command that removes rows with missing values (NA) from a data frame df and assigns the resulting data frame to df.  
In the command df <- na.omit(df), the original data frame df is replaced with a new data frame that contains only the rows that do not have any missing values. This can be useful when working with data that has missing values because it allows you to remove incomplete cases and focus only on complete cases.  
  
It is important to note that removing missing values can result in a loss of information and can potentially bias the analysis if the missing values are not missing completely at random.  
  
Check for missing values have been removed  
colSums(is.na(df)) can be used to check if any missing values (NA) remain in the columns of a data frame df after removing rows with missing values using the na.omit(df) function.

Check for duplicates  
dim(df[duplicated(df$Film),])[1] is an R command that counts the number of duplicate values in a specific column of a data frame df.  
  
In this command, duplicated(df$Film) creates a logical vector of the same length as the column df$Film, where TRUE values indicate that the corresponding element is a duplicate of a previous element in the column. The expression df[duplicated(df$Film),] subsets the data frame to only include rows where the df$Film column contains a duplicate value. Finally, dim(df[duplicated(df$Film),])[1] returns the number of rows in this subsetted data frame.  
So, the overall effect of the command dim(df[duplicated(df$Film),])[1] is to count the number of rows in the data frame where the df$Film column contains a duplicate value.  
  
Round off values by 2 places  
df$Profitability <- round(df$Profitability, digit = 2) is an R command that rounds the values in the Profitability column of a data frame df to two decimal places.  
  
In this command, round() is an R function used to round off the specified values to a given number of decimal places. The first argument of the round() function is df$Profitability, which is the column to be rounded, and the second argument digit = 2 specifies the number of decimal places to round the values to.  
The assignment operator <- is used to assign the rounded values back to the Profitability column of the data frame df  
  
df$Worldwide.Gross <- round(df$Worldwide.Gross ,digit=2) is an R command that rounds the values in the Worldwide.Gross column of a data frame df to two decimal places  
dim(df) is an R command that returns the dimensions of a data frame df as a vector with A screenshot of a computer

Description automatically generated with medium confidencetwo elements: the number of rows and the number of columns.

1. Outlier Removal

Check for outliers using a boxplot  
library(ggplot2) is an R command that loads the ggplot2 package, which is a popular data visualization package used for creating high-quality graphics and charts.  
  
By running library(ggplot2), the ggplot2 package is loaded into the R session, and its functions and other objects become available for use. Once the package is loaded, you can start creating visualizations using the ggplot2 syntax and functions.  
  
How to create a boxplot that highlights outliers  
ggplot(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)) creates a box plot of the Profitability and Worldwide.Gross columns of a data frame df.  
Overall, this code will produce a box plot of the distribution of Worldwide.Gross values for each level of Profitability. Outliers will be marked with red dots, and the y-axis will be limited to the range 0 to 1000.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generatedRemove outliers in Profitability.  
Q1 <- quantile(df$Profitability, .25) is an R command that calculates the first quartile (Q1) of the Profitability column of a data frame df.  
By calculating the first quartile of the Profitability column, we can get an idea of the distribution of the data and how it is spread out. If the first quartile is significantly smaller than the median or third quartile, for example, it might suggest that the data is skewed towards lower values.  
  
The code Q3 <- quantile(df$Profitability, .75) calculates the third quartile, also known as the 75th percentile, of the "Profitability" variable in a dataframe named "df"  
  
The code IQR <- IQR(df$Profitability) calculates the interquartile range (IQR) of the "Profitability" variable in a dataframe named "df".

The code no\_outliers <- subset(df, df$Profitability> (Q1 - 1.5\*IQR) & df$Profitability< (Q3 + 1.5\*IQR)) creates a new subset of the original dataframe "df" by removing any outliers in the "Profitability" variable.

The subset() function is used to create the new subset based on the condition defined by the logical expression df$Profitability> (Q1 - 1.5\*IQR) & df$Profitability< (Q3 + 1.5\*IQR), which returns TRUE for observations that are within the range and FALSE for observations that are outside the range.

**You may repeat these steps using the 'Worldwide.Gross' metric in place of 'Profitability'.**

1. Exploratory Data Analysis

Summary statistics  
The summary(df1) is a command that provides a statistical summary of the data contained in the data frame "df1". The summary function generates a brief statistical summary of all variables in the data frame, including the minimum and maximum values, quartiles, mean, median, and count of non-missing values (NAs).  
The output of the "summary(df1)" command will display the summary statistics of each variable present in the data frame, making it a quick way to assess the overall distribution and characteristics of the data. This can be useful for identifying any missing values, outliers, or patterns in the data.

A screenshot of a computer

Description automatically generated with medium confidence

Scatterplot  
The code ggplot(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)) is written in R programming language and uses the ggplot2 package to create a scatter plot from the data contained in the data frame "df1".  
In summary, this code generates a scatter plot that displays the relationship between the Lead Studio and Rotten Tomatoes ratings for movies in the dataset, with each point representing a movie. The y-axis is formatted with comma-separated values and the x-axis labels are rotated for better readability.

Bar chart

The code ggplot(df1, aes(x=Year)) + geom\_bar() uses the ggplot2 package to create a bar plot from the data contained in the data frame "df1".  
The "ggplot" function initializes the plot and specifies the data frame and the aesthetic mappings, where "Year" is mapped to the x-axis. The "geom\_bar" function adds a layer of bars to the plot, with the height of each bar representing the count of movies in each year.  
In summary, this code generates a bar plot that displays the distribution of movies across different years in the dataset. Each bar represents the count of movies in a specific year. This plot can be useful for visualizing trends and patterns in the distribution of movies over time.

Graphical user interface, application

Description automatically generated

1. Export data

Export clean data  
The code "write.csv(df1, “clean\_df.csv")" is written in R programming language and is used to export the contents of the data frame "df1" to a CSV (Comma-Separated Values) file named "clean\_df.csv".  
  
The "write.csv" function is a built-in function in R that allows the user to write data to a CSV file. The first argument of the function specifies the data frame to be written, while the second argument specifies the filename to be used for the exported file.  
In summary, this code exports the data contained in the data frame "df1" to a CSV file named "clean\_df.csv", which can be used for further analysis or sharing with other users.

Graphical user interface, application

Description automatically generated

6. Power BI Dashboard creation

Chart

Description automatically generated<https://app.powerbi.com/links/SgHrazxoDb?ctid=6efd0f20-57c8-4447-b53f-00d4992ca50b&pbi_source=linkShare>