28/05/2024

Elizabeth Muir

Hollywood Movies

R and PowerBi

Contents

[Introduction 2](#_Toc167813343)

[RStudio 3](#_Toc167813344)

[PowerBi 11](#_Toc167813345)

[Conclusion 15](#_Toc167813346)

# Introduction

In this assignment I will be analysing the performance of Hollywood movies from data dated between 2007 – 2012. I will be using RStudio and PowerBi to complete this.

I will be using RStudio to be looking at the data and exploring if it has the right data types and to check for missing data. I will then use RStudio to do a scatterplot to display lead studios and their ratings on rotten tomatoes and a bar chart to show the number of films produced each year.

I will then export the data file using a csv format ready to insert into PowerBi. I will be making a dashboard for the client presenting different charts, I will cover:

* The average Rotten Tomatoes rating of each genre
* The number of movies produced per year
* The audience scores for each film
* The profitability per studio
* The worldwide gross per genre

I will also include any additional charts or information that I feel will fit nicely into the dashboard to give more insights into Hollywood.

# RStudio

I will take you through a step-by-step guide in using RStudio with our dataset for Hollywood Movies. I first want to download our data onto our PC so that we are ready to insert into RStudio, I was given a link on where I had to get my data from.

A close up of a logo

Description automatically generated

This is the [link](https://public.tableau.com/app/sample-data/HollywoodsMostProfitableStories.csv) to where I will be getting my data from. This will instantly download the file to my downloads folder ready to open in RStudio but for now we are going to directly use the link to bring the data into RStudio. The data we are using is from the actual Tableau website, where they have loads of datasets made for users to create dashboards and other analytics depending on clients’ criteria.

A logo on a black surface

Description automatically generated

Now we are going to go to RStudio on our PC and load it up.

A screenshot of a computer

Description automatically generated

Once loaded our screen should look a bit like this or similar, we have a console on the left, environment, and files on the right side.

A screenshot of a computer menu

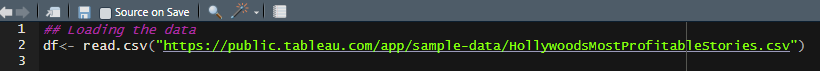
Description automatically generated

We now want to come up to the top left corner of the screen and under file, select the paper with the plus sign on it to select a R Script. This is a file that allows us to enter code to run on many lines rather than using the console where you are limited to one line. Using a R Script allows us to keep an eye on our code to ensure that we have no typos or syntax errors.

A screenshot of a computer

Description automatically generated

You will now see on the left side of the screen that another panel has been added to the screen. This is where our script will be written. The console below will display the results when we run our lines of code.



The first line of code that we need to execute is reading our csv file we are going to use the website for now to add our file. We are assigning our file to a variable called ‘df’. This makes it easier to execute code without having to refer to the file name all the time. The next function is read.csv, this is telling RStudio that we want to look at a file that has a csv file extension. In brackets and in speech marks we paste our link of our dataset, this tells RStudio where to grab that data from.

A screen shot of a computer

Description automatically generated

When we look at the right side of our screen under environment, we can see that it has changed. This displays our variable df that we have set, 74 obs. Which is the number of entries that is in the file and 8 variables which are how many columns/headers in the file.

A black background with purple and blue text

Description automatically generated

Now to display our file to see that everything has been pulled in correctly we have to run a line of code called View(df). View is telling RStudio we want to display a spreadsheet style view of our data. In brackets we include our variable that we have assigned our data to so RStudio knows what dataset we want to look at.

A screenshot of a computer

Description automatically generated

Once that line has been executed this screen will appear. You can see it looks like an Excel spreadsheet with all our data inputted. This allows us to see everything has appeared correctly and if we need to do some cleaning to our data.

A black screen with yellow and white text

Description automatically generated

Before we can go ahead and clean our data, we need to install some packages that will assist us in doing so. We run a line of code called install.packages(“tidyverse”). This is going to load a package called tidyverse. This package has various tools that will assist us in cleaning our data ready for analysis.

A black screen with white lines

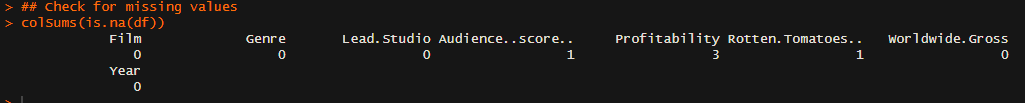
Description automatically generated

Now we are ready to import all the tools that tidyverse has for us to clean our data. We run the code library(tidyverse). You can see here in the output each name of the tools that have come with tidyverse.

A screen shot of a computer

Description automatically generated

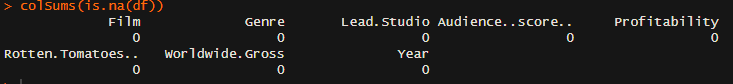
Now that is all prepared and ready to use, we want to first check the datatypes of our entries, this is to make sure that the data is represented according to their datatype and not another one that may cause confusion when creating visualisations. We run the code str(df), this is a function in R that displays the internal structure of an object, in this case our dataset. It breaks it down into columns, with each one and their data type and examples of the data that is entered.



Once we have gone over each data type and we are happy we now check over the data to see if there are any missing values. We do this to ensure that when we go to create visualisations, they will not display gaps or a misrepresentation of what the data is trying to tell us through patterns and relationships. We run the code colSums(is.na(df)), this counts how many null values are true and displays to us where they are within our data.



We are missing 5 entries in our data, so we are going to drop them from our table. We do this by running a line of code df <-na.omit(df), this is going to remove missing values (NAs) from the data, this then returns the same columns but now with the removed rows.



To ensure that the code has worked correctly we run the code colSums(is.na(df)) again and now you will see that each column has 0. This means that the previous missing entries have now been removed and we are ready to go to the next step for visualisation.

A screenshot of a computer

Description automatically generated

We now want to create a summary of our data. The summary() function gives us statistics of each column within our data. You can see that some of our columns display different statistics, as an example Film is a Character datatype, this will show you the length of the longest name under that column which is 70. Class is telling us that it is a Character and always have text. Mode shows Character also which means it indicates in R how it will be stored and handled when used. Now we can compare this to audience.score, this displays the min number, middle, max, average and 1st and 3rd quarters of the scores that are in this column as this data type is an integer.



We are now going to create a scatterplot from data that we are going to select from our data frame. This is where our tidyverse library comes in handy to help us create some visualisations from our data. For this first visualisation I will be using a scatterplot, we are using this graph to show us the two pieces of information and how they are related to each other to see any patterns.

We use this code: ggplot(df,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)).

Ggplot: is a function from ggplot2 package which helps us create visualisations.

Df: is our dataset

Aes(x=Lead.Studio, y=Rotten.Tomatoes..) : This is where we want to select our columns to map out leading to our visualisations.

Geom\_point(): adds dots to the plot that are represented from our data.

Scale\_y\_continuous(labels=scales::comma) : modifies the y axis

Coord\_cartesian(ylim = c(0,110)) : this limits the plot area and ensures that the y-axis values are between 0 and 110

Theme(axis.text.x = element\_text(angle = 90)) : This is the customisation of the plot and ensures that the labels on the x-axis is rotates 90 degrees so that they are readable on the scatterplot.

A screenshot of a computer

Description automatically generated

When we run our code, we can see this is what happens with our data. Each lead studio has many dots, but you can tell which ones have more in terms of rotten tomatoes scores. We get an indication in what studios have more films that have been rated compared to others.



We are now going to create a bar chart graph. This code will display a bar chart. We are going to use the Year column to display how many films have been produced each year. The code broken down:

Ggplot: package to create a visualisation

Df: our data

Aes(x=year): how the data map is visualised for the x-axis and the column needed.

Geom\_bar(): this adds bars to the plot using the year columns data.

A graph on a computer screen

Description automatically generated

Once the code is run this is what appears. This is our bar chart displays how many films have been produced each year. We can see that 2008 was a popular year.



A screenshot of a computer

Description automatically generated



Now that we have cleaned our data and produced a couple of visualisations in RStudio we now want to save our data so that we can now import it into PowerBi to create our dashboard. We have to run code write.csv(df, “clean\_df.csv”), this is a function that saves our file as a csv file as we have stated when we written the name of the file. We need to include the file extension to let RStudio know that is what we want, we can do this for other extensions also. As default RStudio will save our file in our documents so that it is easy to find. So, we head to documents folder and we can see that our file is saved correctly and can be viewed without a problem.



I am now going to save my script that has all my code in, this is in case I need it in the future to refer to or if I would like to add to it.

A screenshot of a computer

Description automatically generated

All I must do is put the file name that I want and select save and that will be ready for when I need it next.

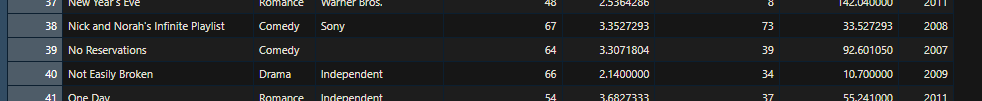
# PowerBi

For this part of my assignment, I will be importing the clean\_df.csv file into PowerBi to create a dashboard for a company with some requirements. Firstly, the company would like the dashboard to be in their brand colours which is blue, green and brown. They don’t mind if the shades are in light or dark variations as long as they are those colours, they also do not mind if blue and green are only used also.

The main charts that they would like to see on the dashboard are as follows:

* The average Rotten Tomatoes ratings for each genre
* The number of movies produced per year
* The audience scores for each film
* The profitability per studio
* The worldwide gross per genre

I did have a slight issue when I started to create my charts, I had noticed that I still had a missing value on my data which could affect the way my data is displayed, and it would make it unreliable, so I had to head back over to RStudio to fix straight away.

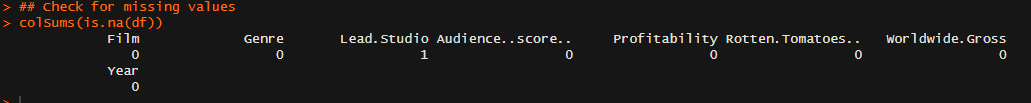


I ran the code view(df) to see exactly where the missing value was. We can see at line 39 that the film ‘No Reservations’ was missing a studio. I had to find a way of correcting this.

A computer screen with text

Description automatically generated

To fix this was to write some code that will search for blank cells that had no value in and then make it display as NA. This will allow me to then drop it afterwards so that all my data is clean. For some reason when I first cleaned the data the code, I ran wouldn’t pick up fully empty cells and the only way I could work around it is using this code.

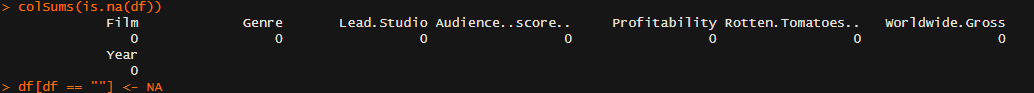


Once I ran the code I quickly checked if that changed the values of missing data which we can see under lead studio we have 1 missing value now. This shows me that the code worked, and we are ready to drop the row so we can continue with our visualisations.

A computer screen with text

Description automatically generated

Again, I run this code so that it will drop the row ready to export into PowerBi.



To confirm the row was dropped I checked the colSums again and now the data has been cleaned fully. Now I can resave my file and input it into PowerBi to create my dashboard.

These are the charts that was requested by the client:

A screenshot of a music book

Description automatically generated

A graph with green bars

Description automatically generated with medium confidenceA graph with a circle and numbers

Description automatically generated with medium confidenceA green pie chart with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

When I created the dashboard, I had some more ideas on what to publish onto it showing different aspects of Hollywood and making the dashboard more interactive for the user. Along with these charts I have included two filter slicers, one for year and one for genre. This gives the user interaction with the dashboard to show what information they would like to see for a particular year with or without a specific genre. When a user selects a year or years or genre and genres each of the charts displayed will update with that information.

A screenshot of a computer screen

Description automatically generated A screenshot of a computer screen

Description automatically generated

I also included 4 cards at the top of the dashboard, this allows the user to see various bits of information as a whole ranging from 2007 – 2012, until a filter is in place and the values change with them:

* An average audience scores
* An average Rotten Tomatoes score
* Overall Profit made
* Worldwide Gross

A close-up of a white square

Description automatically generatedA close up of a sign

Description automatically generated

I have also made a tree map display showing the average rotten tomatoes score per studio, this will allow the user to select individual studios and see the average. Lastly the extra chart that I have added to the dashboard is a bar chart showing how many films were created for each year. I took this idea from when I was cleaning the data in RStudio and thought it would be a great insight to see how each year was for production of films.

A screenshot of a computer screen

Description automatically generated A graph of blue rectangular bars

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

This is my dashboard that I have created. All my charts and cards are displayed together with the filter buttons ready for users to look and play with the data to get information that they would like to see. I have stuck with the brand colours that are listed in the specification

**Link to my dashboard:** <https://app.powerbi.com/links/myrLQMZ0T_?ctid=6efd0f20-57c8-4447-b53f-00d4992ca50b&pbi_source=linkShare>

# Conclusion

Overall, I have really enjoyed this assignment. Being able to learn how to import a file into RStudio and to clean and visualise the data using a scatterplot allows me to see the relationship between data and understand how it can be used for analysis and reporting.

I thoroughly enjoyed taking the cleaned data into PowerBi and to now make a great dashboard ready to show to users so that they can visualised different bits of data for years or genres or even certain films and get to see a thorough analysis of them all.

I am glad I got to see that even though I cleaned the data, not all the data was cleaned effectively so I was able to learn and overcome that situation to change the data so that when I was able to put it into PowerBi again I know I was able to create visualisations that are reliable and not be misunderstood.

In the future I would like to work on bigger datasets and see what other visualisations I can make in RStudio because of this assignment I only saw scatterplots and bar charts and I would like to see more and learn how to maybe customise them.