5/11/2023

Jonathan Theaker

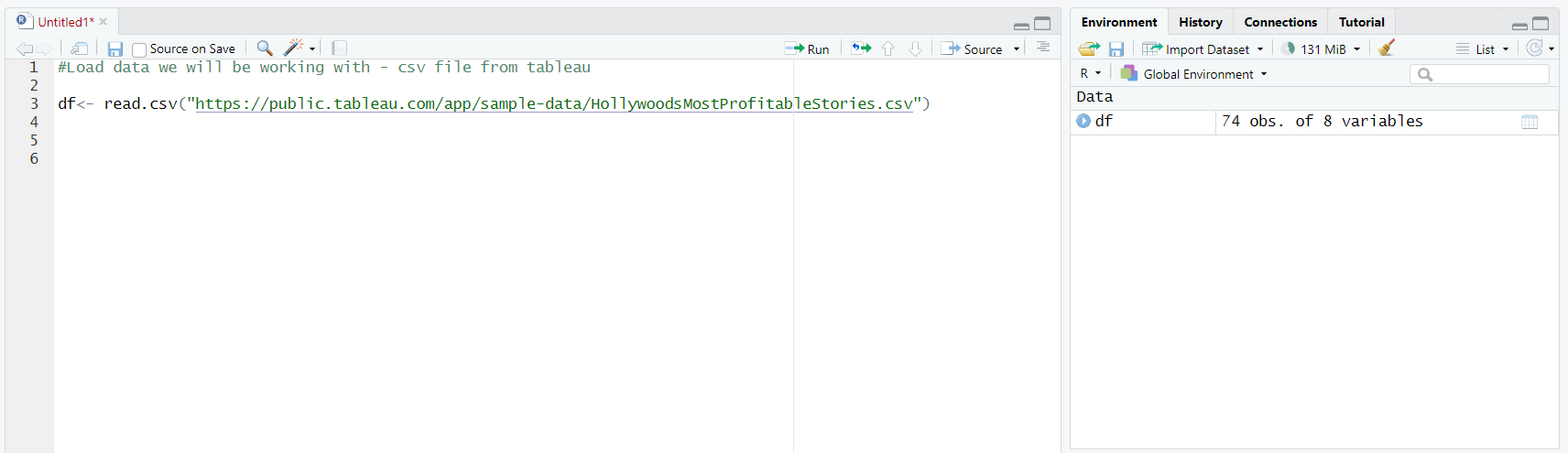
Performance of Hollywood Movies

Using R

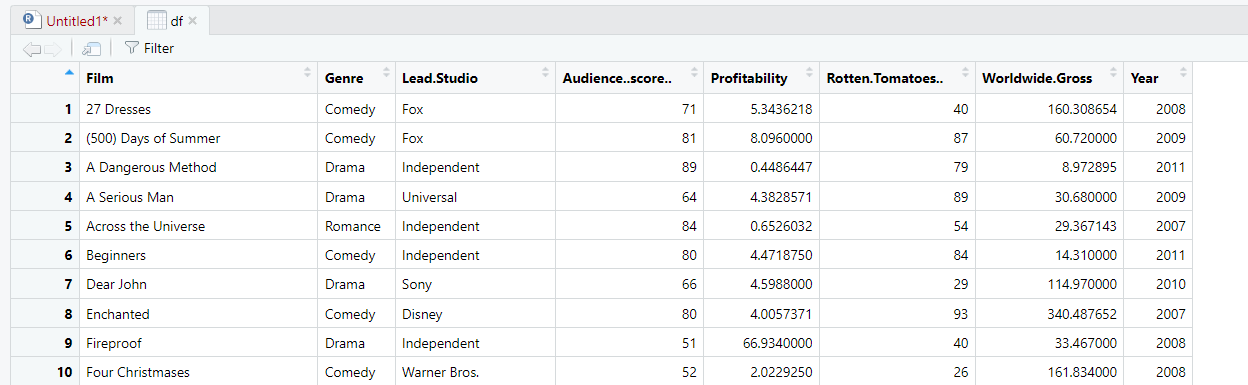
We will be looking at the performance of Hollywood Movies. Lines written in R starting with # are not part of my commands and will be used to annotate briefly what each piece of code does above each command. Blue text is the output shown in the console.

**Setup + Import Data/Tools**

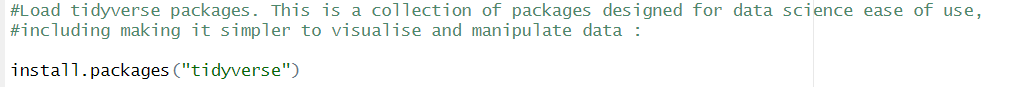
#First, we need to import the data we will be working with into R so that we can work with it



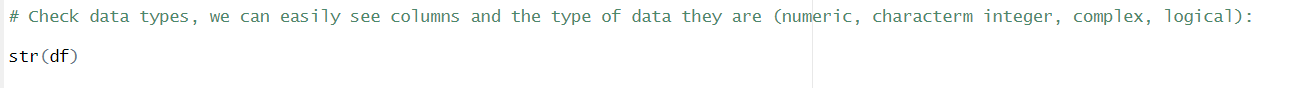
#Once the data is loaded after running the command, we can see the data on the right and select it to view the table, this will add it as a tab we can select at the top, next to our ‘Untitled1’ that we are currently using to write our code.



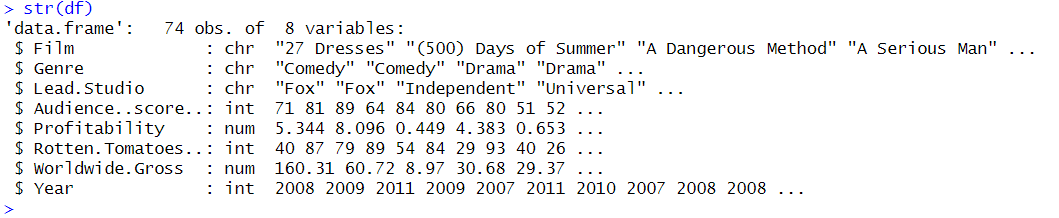
#Tidyverse is a popular R package we have to load into R before we can use it.



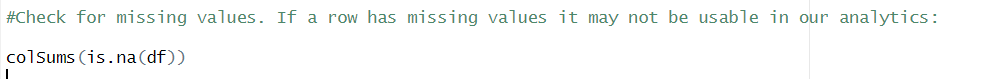
**Data Cleaning**



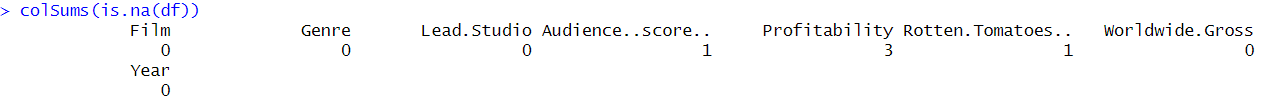
#Below is what is shown in the console when we run str(df)



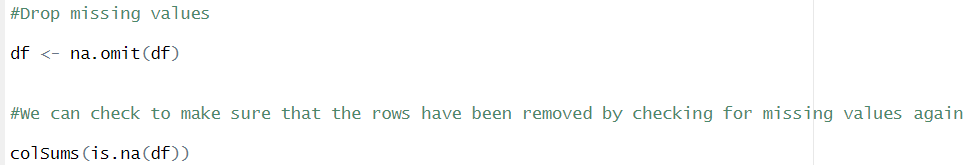
#We can see our data types consist of characters, integers and numeric.



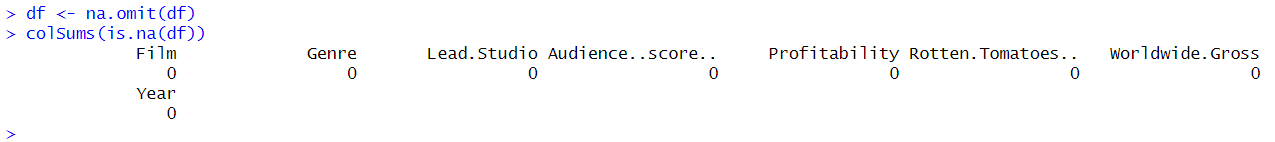
The code above returns the below missing values in the console.



#Now we know we have a missing value in Score and Rotten Tomatoes alongside 3 missing values in Profitability; we can remove them from our dataset as they will not be useable.



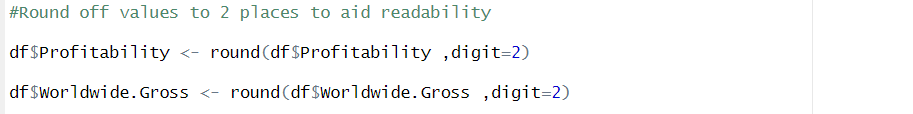
#We can see in the below console that there are no longer any missing values

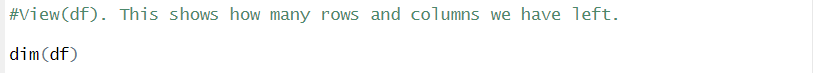




#We do not have any duplicates which is good, we do not want to log the same movie twice.

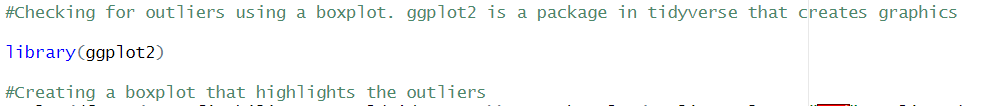






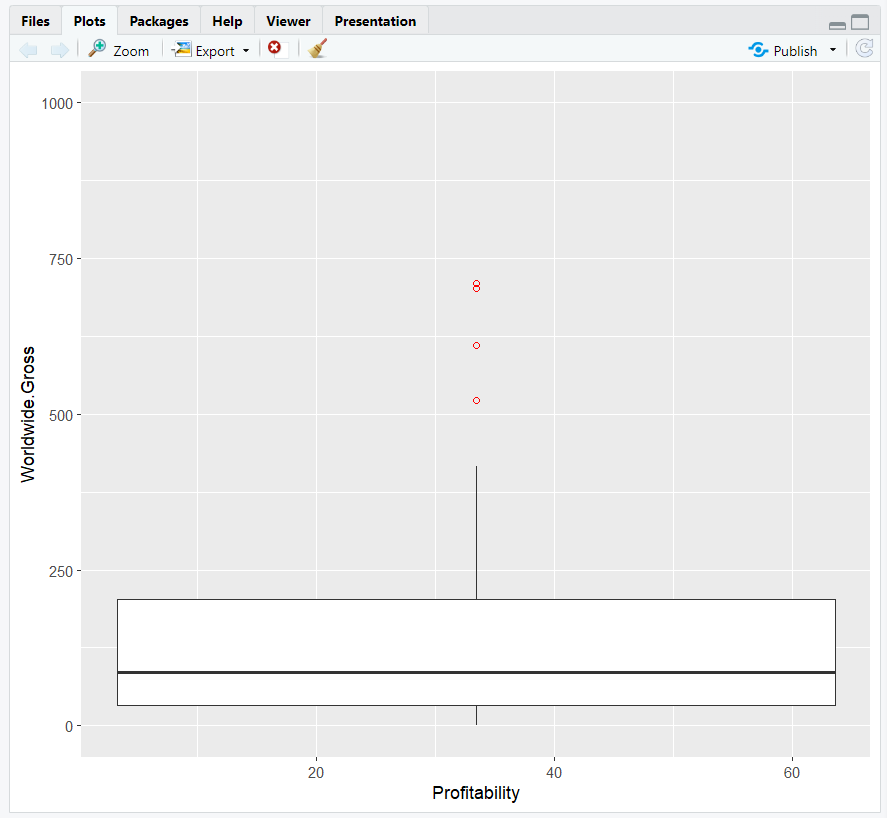


#70 Rows, 8 Columns

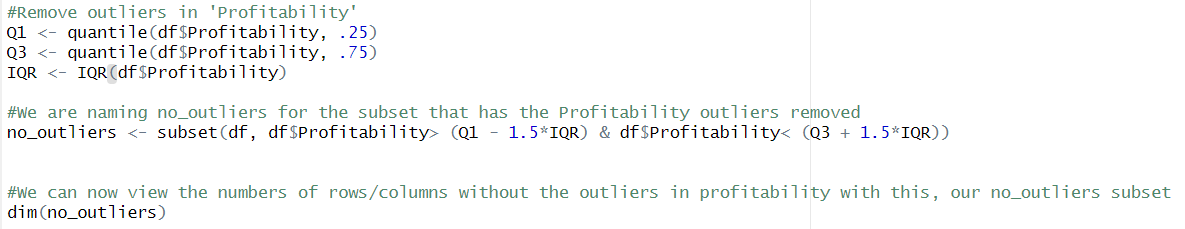


ggplot(df,aes(x=Profitability, y=Worldwide.Gross))+geom\_boxplot(outlier.colour= "red",outlier.shape= 1)+scale\_x\_continuous(labels = scales::comma)+coord\_cartesian(ylim= c(0,1000))

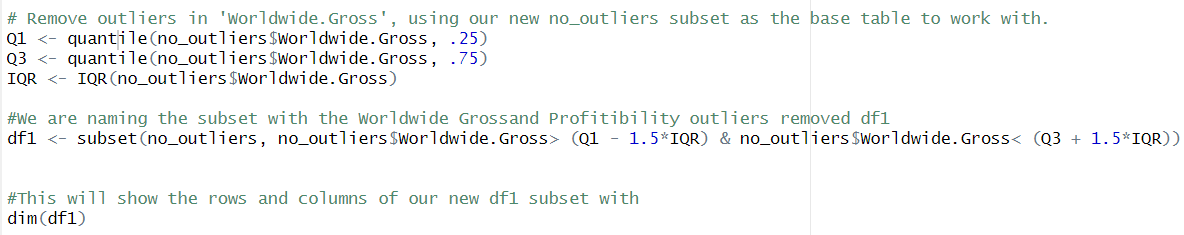
#The above code creates the boxplot below, with outliers shown in red circles



#We now will remove the outliers in our data that we do not trust to use.

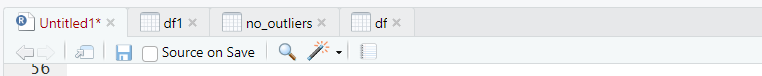






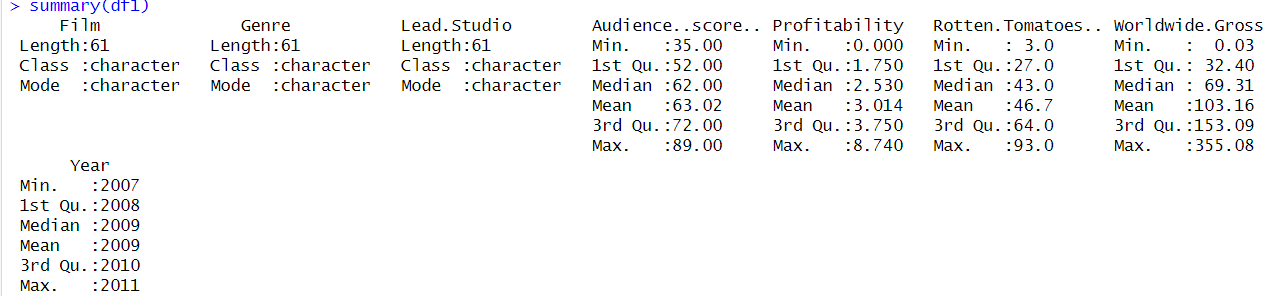


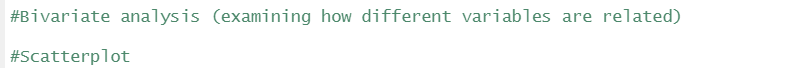
#We can also now see our newly named subset tables in our tab.



**Exploring our Clean Data**

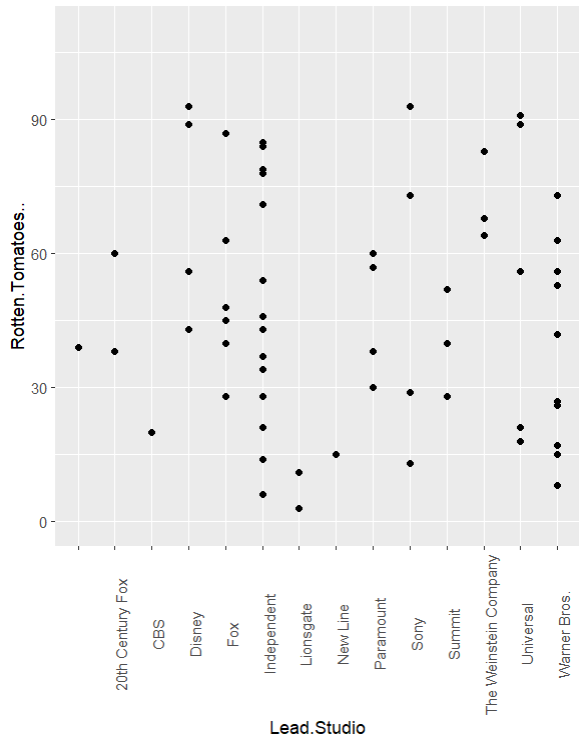
****

****

****

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))

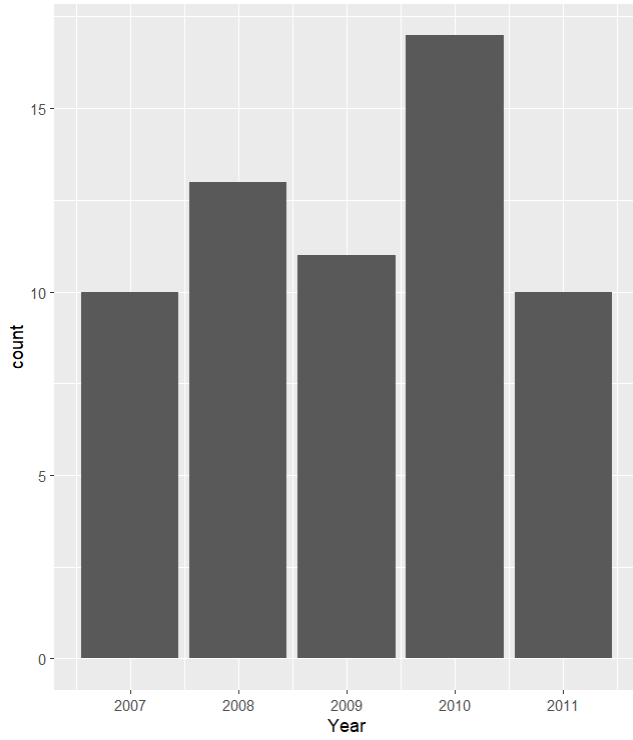
#This shows what score Rotten tomatoes gave to each film by studio. Each dot is an individual film.



#Bar Chart

ggplot(df1, aes(x=Year)) + geom\_bar())

#This shows how many films were reviewed each year within our data set



**Exporting our Clean Data**

#We can create a file with our cleaned data. We will turn it into a csv file named ‘clean\_df’. This will be automatically be saved in our project folder.



#We can not import this file into another program like Power BI or Tableau to create visualisations/dashboards.

**Power BI Visualisations**

**Specifications**

* For the dashboard, the company would like me to use their brand colors which are blue, green and brown.
* See the average Rotten Tomatoes rating for each genre
* Number of movies produced each year
* Audience score for each film
* Profitability per studio
* Worldwide gross per genre

Average ratings and count of films per year are very clear in a simple bar chart. However, Audience score by film does not suit a bar chart as there are way too many individual films to try and squash onto one. A Treemap suited it much better as you can easily see each individual film name and its score. I made the score for each individual film small enough to fit on the dashboard, if you want the full film name you will have to hover over each one unfortunately, but it is hard to represent each full name on a single screen.

