

# ASSIGNMENT 3 – R STUDIO AND POWER BI

Marnie Blades

13/12/24 Data Skills

## Contents

Step 1: Initial Exploratory Analysis .....	2
Load the data:.....	2
Take a look at the data: .....	2
Load library:.....	3
Import library:.....	3
Check data types:.....	4
Step 2: Clean data.....	4
Check for missing values:.....	4
Drop missing values: .....	5
Check to see if the rows have been removed: .....	5
Step 3: Exploratory Data Analysis.....	6
Summary statistics:.....	6
Scatterplot: .....	6
Bar chart: .....	7
Step 4: Export Data .....	8
Export clean data:.....	8
Import clean data into power BI.....	8
Import clean_df in Power BI: .....	8
Creating visulisations and reports: .....	8
Breif: .....	8
The average rotten tomatoes rating for each genre: .....	9
The number of movies produced each year:.....	9
The audience score for each film:.....	9
.....	9
The profitability per studio:.....	10
The worldwide gross per genre: .....	10

# R-studio

## Step 1: Initial Exploratory Analysis

Load the data:

The screenshot shows the RStudio interface with the 'HollywoodsMostProfitableStories' dataset loaded. The console displays the following code and output:

```
R - R 4.4.2 - ~/R
> HollywoodsMostProfitableStories <- read_csv("HollywoodsMostProfitableStories.csv")
Rows: 74 Columns: 8
Column specification
Delimiter: "
chr (3): Film, Genre, Lead Studio
dbl (5): Audience score %, Profitability, Rotten Tomatoes %, Worldwide Gr...
i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(HollywoodsMostProfitableStories)
> |
```

The Environment pane shows the dataset 'HollywoodsMostPr...' with 74 observations and 8 variables. The User Library pane lists various R packages installed.

Take a look at the data:

The screenshot shows the RStudio interface with the 'HollywoodsMostProfitableStories' dataset loaded. The console displays the following code and output:

```
R - R 4.4.2 - ~/R
> view(df)
1
2
```

The Environment pane shows the dataset 'HollywoodsMostPr...' with 74 observations and 8 variables. The User Library pane lists various R packages installed.

## Load library:

RStudio interface showing the process of loading the tidyverse library. The console output is as follows:

```
R - R 4.4.2 ~ ./
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/marni/AppData/Local/R/win-library/4.4'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.4/tidyverse_2.0.0.zip'
Content type 'application/zip' length 431716 bytes (421 KB)
downloaded 421 KB

package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:/Users/marni/AppData/Local/Temp/RtmpkJUKL2/downloaded_packages
>
```

The Environment pane shows the loaded data objects: df (74 obs. of 8 variables) and HollywoodsMostPr... (74 obs. of 8 variables). The Packages pane shows the installed packages, including askpass, backports, base64enc, bit, bit64, blob, broom, bslib, cachem, callr, and cellranger.

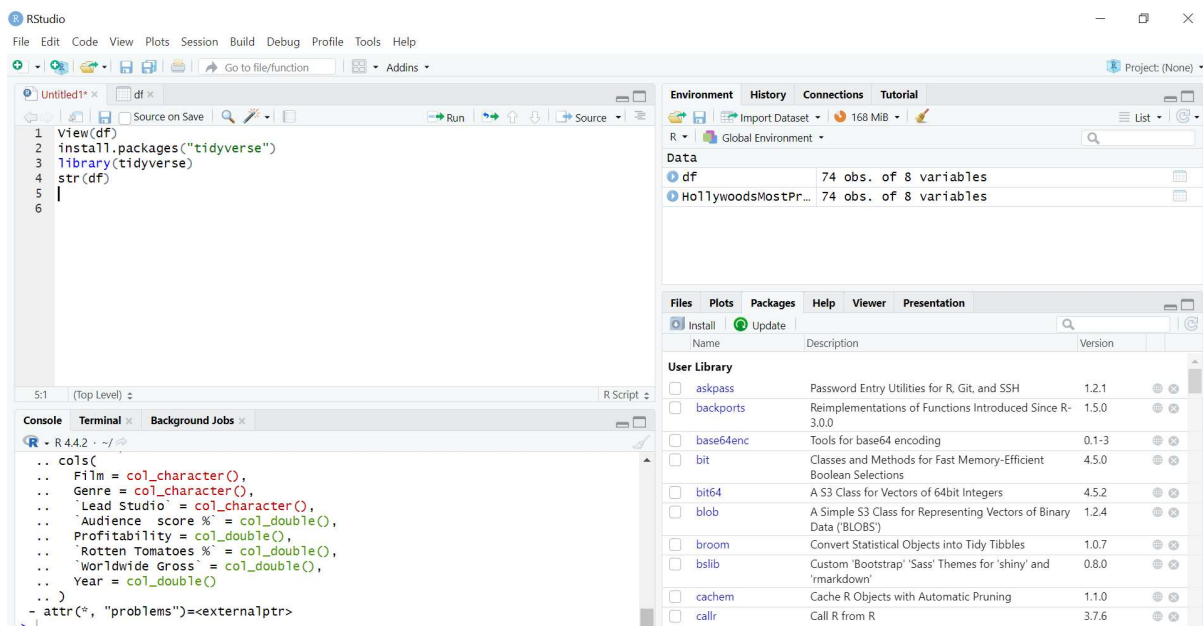
## Import library:

RStudio interface showing the process of importing the tidyverse library. The console output is as follows:

```
R - R 4.4.2 ~ ./
> library(tidyverse)
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr 1.1.4 ✓ readr 2.1.5
✓ forcats 1.0.0 ✓ stringr 1.5.1
✓ ggplot2 3.5.1 ✓ tibble 3.2.1
✓ lubridate 1.9.3 ✓ tidyr 1.3.1
✓ purrr 1.0.2
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag() masks stats::lag()
! use the conflicted package to force all conflicts to become errors
>
```

The Environment pane shows the loaded data objects: df (74 obs. of 8 variables) and HollywoodsMostPr... (74 obs. of 8 variables). The Packages pane shows the installed packages, including askpass, backports, base64enc, bit, bit64, blob, broom, bslib, cachem, callr, and cellranger.

## Check data types:



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 view(df)
2 install.packages("tidyverse")
3 library(tidyverse)
4 str(df)
5
6
```

The console on the bottom left shows the output of `str(df)`:

```
.. cols(
..   Film = col_character(),
..   Genre = col_character(),
..   'Lead Studio' = col_character(),
..   'Audience score %' = col_double(),
..   Profitability = col_double(),
..   'Rotten Tomatoes %' = col_double(),
..   'worldwide Gross' = col_double(),
..   Year = col_double()
.. )
- attr(*, "problems")=<externalptr>
```

The Environment pane on the right shows two data objects:

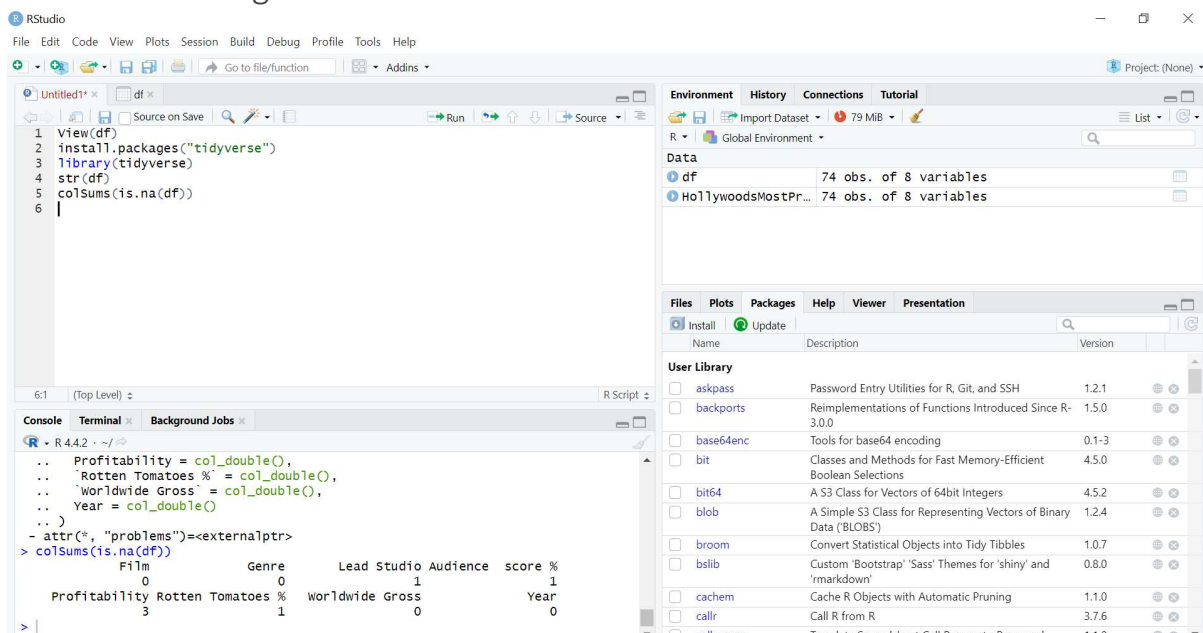
- `df`: 74 obs. of 8 variables
- `HollywoodsMostPr...`: 74 obs. of 8 variables

The Packages pane shows the installed user library:

Name	Description	Version
askpass	Password Entry Utilities for R, Git, and SSH	1.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.5.0
base64enc	Tools for base64 encoding	0.1-3
bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.5.0
bit64	A S3 Class for Vectors of 64bit Integers	4.5.2
blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.4
broom	Convert Statistical Objects into Tidy Tibbles	1.0.7
bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'markdown'	0.8.0
cachem	Cache R Objects with Automatic Pruning	1.1.0
callr	Call R from R	3.7.6

## Step 2: Clean data

### Check for missing values:



The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 view(df)
2 install.packages("tidyverse")
3 library(tidyverse)
4 str(df)
5 colSums(is.na(df))
6
```

The console on the bottom left shows the output of `colSums(is.na(df))`:

```
.. Profitability = col_double(),
.. 'Rotten Tomatoes %' = col_double(),
.. 'worldwide Gross' = col_double(),
.. Year = col_double()
.. )
- attr(*, "problems")=<externalptr>
> colSums(is.na(df))
      Film      Genre      Lead Studio Audience score %
      0         0         1         1
Profitability Rotten Tomatoes % worldwide Gross      Year
      3         1         0         0
```

The Environment pane on the right shows two data objects:

- `df`: 74 obs. of 8 variables
- `HollywoodsMostPr...`: 74 obs. of 8 variables

The Packages pane shows the installed user library:

Name	Description	Version
askpass	Password Entry Utilities for R, Git, and SSH	1.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.5.0
base64enc	Tools for base64 encoding	0.1-3
bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.5.0
bit64	A S3 Class for Vectors of 64bit Integers	4.5.2
blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.4
broom	Convert Statistical Objects into Tidy Tibbles	1.0.7
bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'markdown'	0.8.0
cachem	Cache R Objects with Automatic Pruning	1.1.0
callr	Call R from R	3.7.6
callranner	Translate Spreadsheet Call Ranner to R and	1.1.0

## Drop missing values:

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 View(df)
2 install.packages("tidyverse")
3 library(tidyverse)
4 str(df)
5 colSums(is.na(df))
6 df <- na.omit(df)
7 df <- df %>% drop_na()
8
```

The console shows the execution of the code, with an error message for the `na.omit` function:

```
- attr(*, "problems")=<externalptr>
> colSums(is.na(df))
      Film      Genre      Lead Studio Audience  score %
Profitability Rotten Tomatoes % worldwide Gross      Year
           3             1             0
> df <- na.omit(df) or df <- df %>% drop_na()
Error: unexpected symbol in "df <- na.omit(df) or"
> df <- na.omit(df)
> df <- df %>% drop_na()
>
```

The Environment pane shows two data frames: `df` (69 obs. of 8 variables) and `HollywoodsMostPr...` (74 obs. of 8 variables).

## Check to see if the rows have been removed:

The screenshot shows the RStudio interface after running the code. The script editor contains the following code:

```
1 View(df)
2 install.packages("tidyverse")
3 library(tidyverse)
4 str(df)
5 colSums(is.na(df))
6 df <- na.omit(df)
7 df <- df %>% drop_na()
8 colSums(is.na(df))
9
```

The console shows the execution of the code, with an error message for the `na.omit` function:

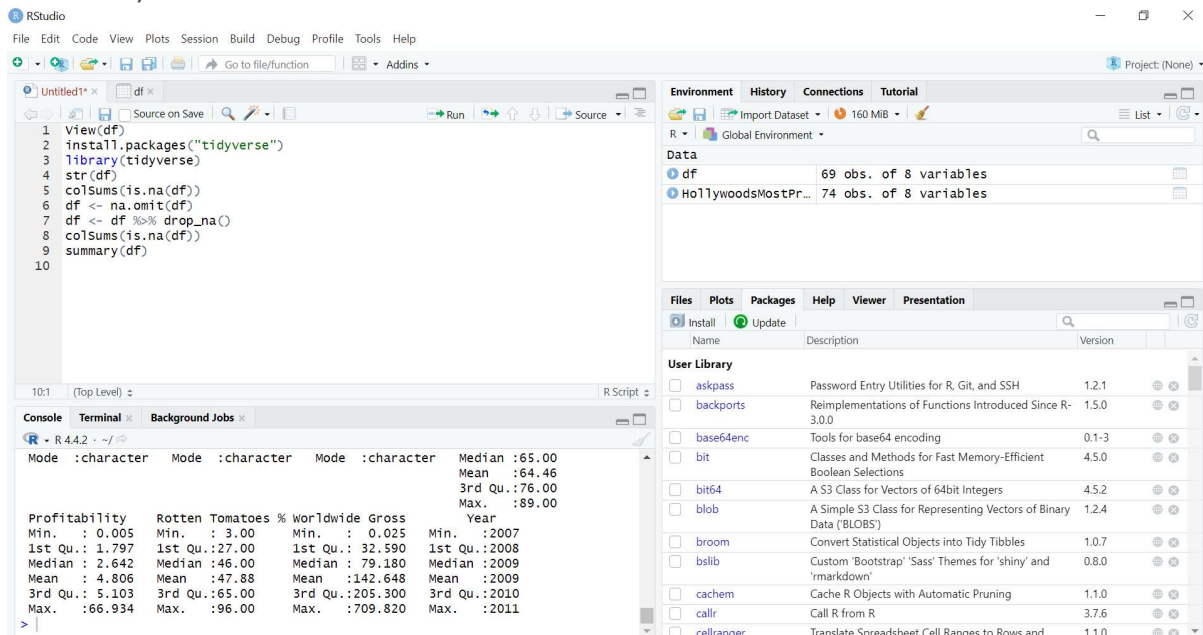
```
Profitability Rotten Tomatoes % worldwide Gross      Year
           3             1             0
> df <- na.omit(df) or df <- df %>% drop_na()
Error: unexpected symbol in "df <- na.omit(df) or"
> df <- na.omit(df)
> df <- df %>% drop_na()
> colSums(is.na(df))
      Film      Genre      Lead Studio Audience  score %
Profitability Rotten Tomatoes % worldwide Gross      Year
           0             0             0
>
```

The Environment pane shows two data frames: `df` (69 obs. of 8 variables) and `HollywoodsMostPr...` (74 obs. of 8 variables).

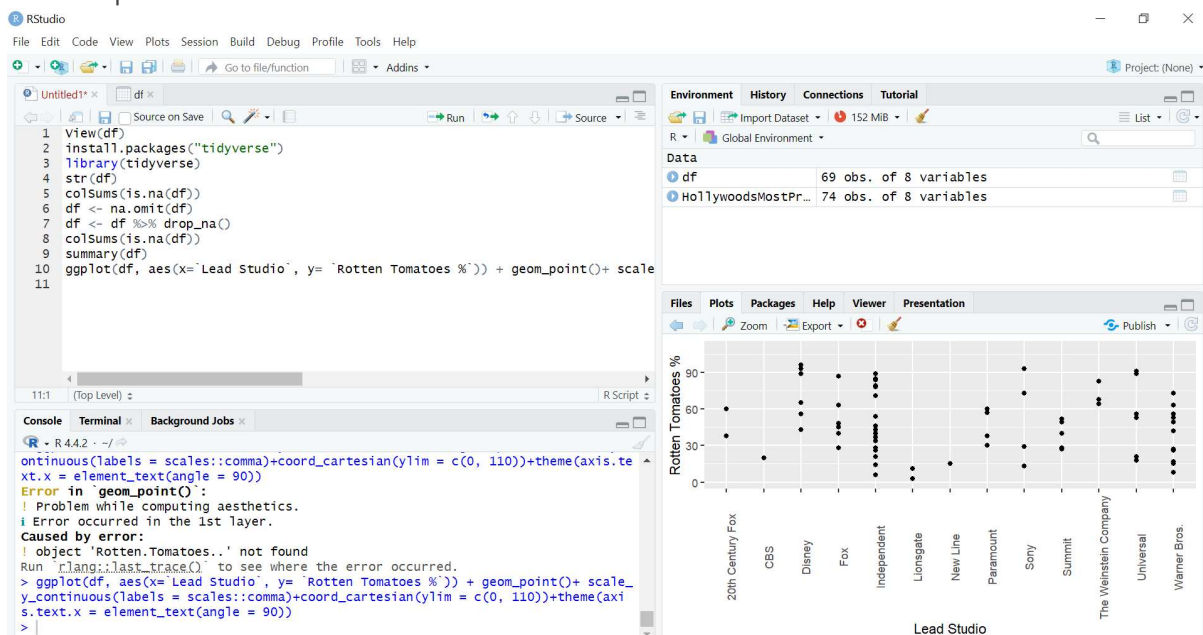


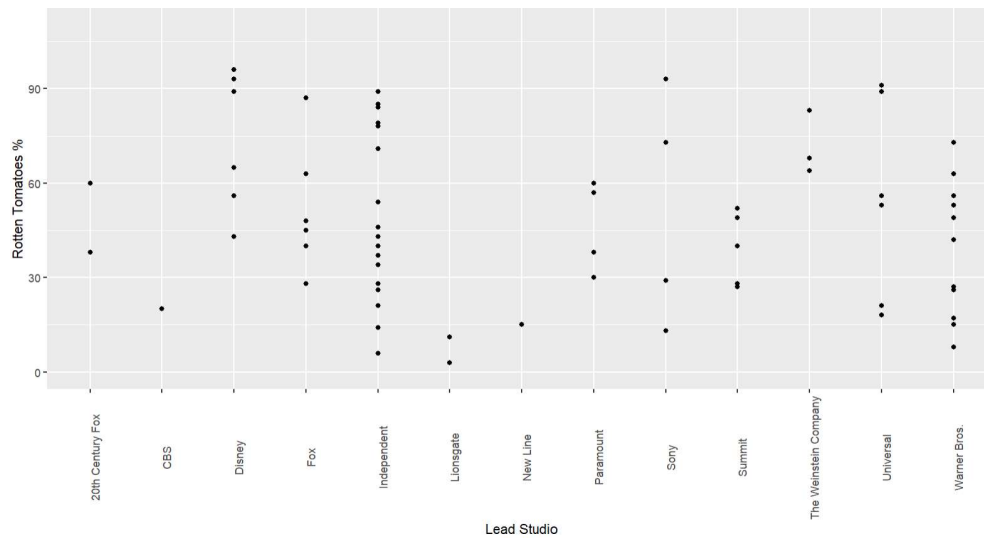
## Step 3: Exploratory Data Analysis

### Summary statistics:

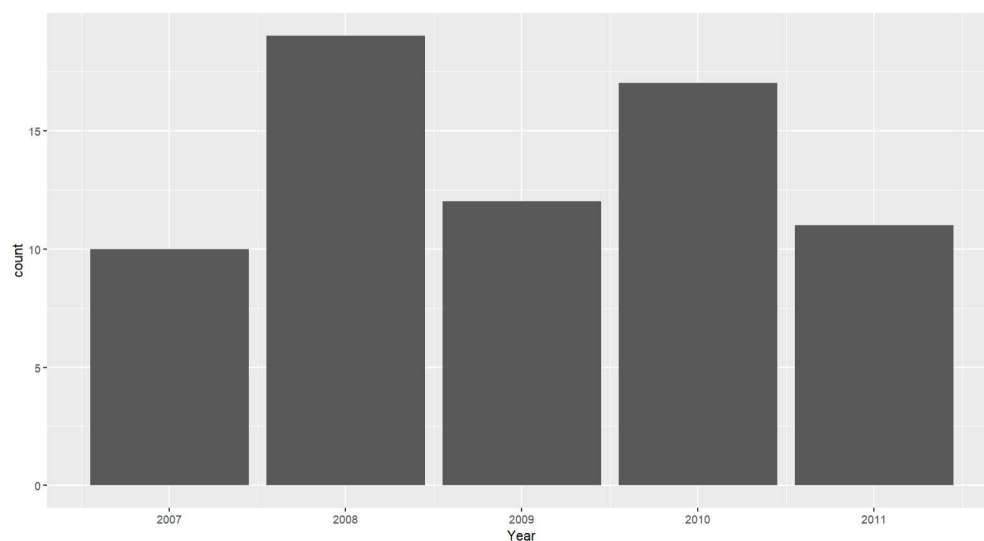
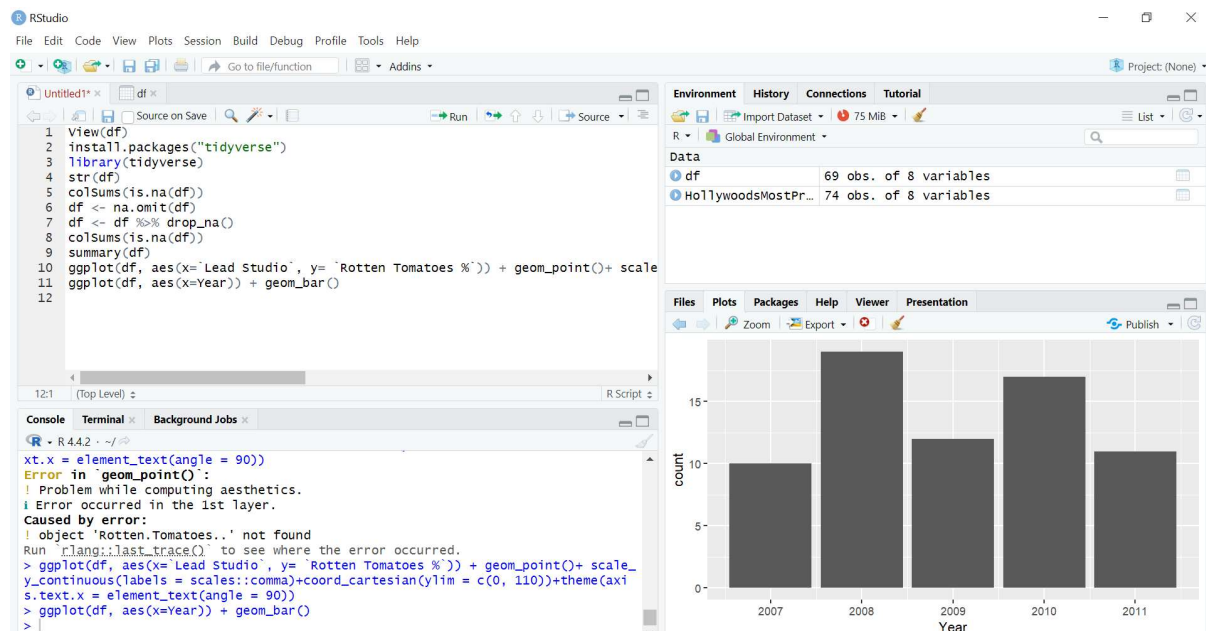


### Scatterplot:





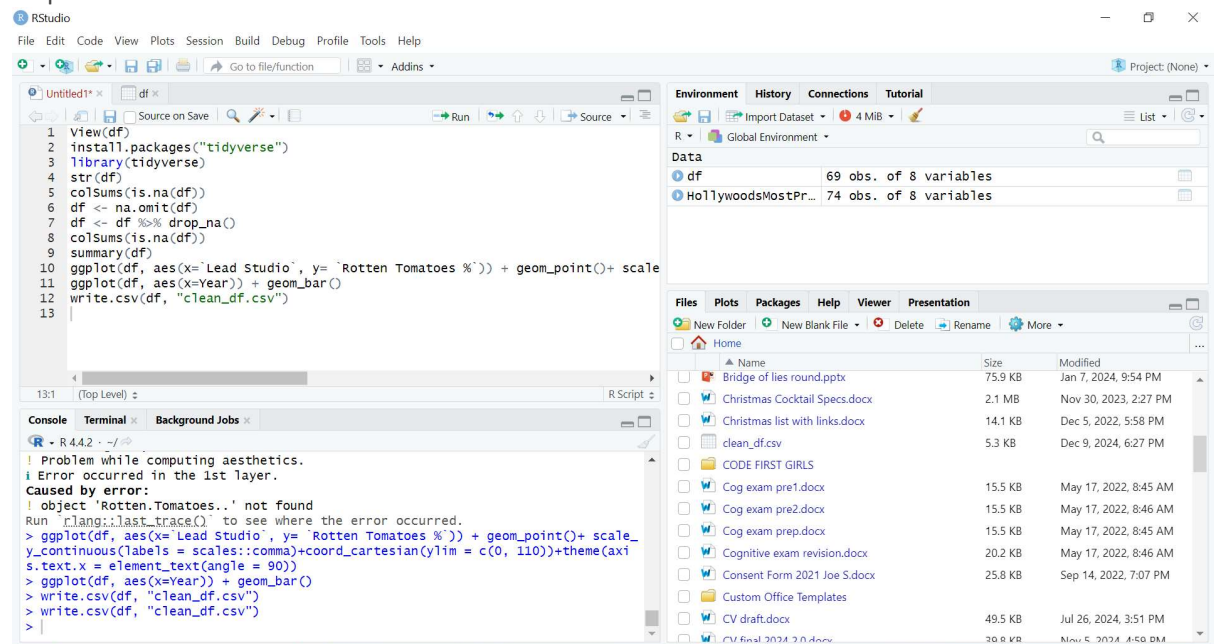
Bar chart:





## Step 4: Export Data

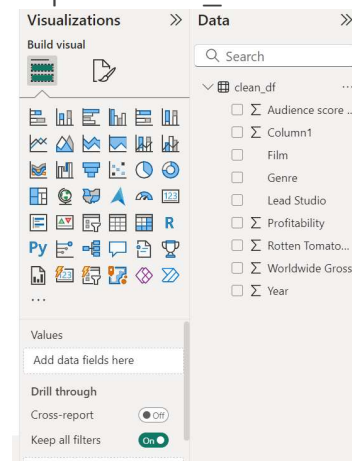
Export clean data:



## Power BI

Import clean data into power BI

Import clean\_df in Power BI:



Creating visulisations and reports:

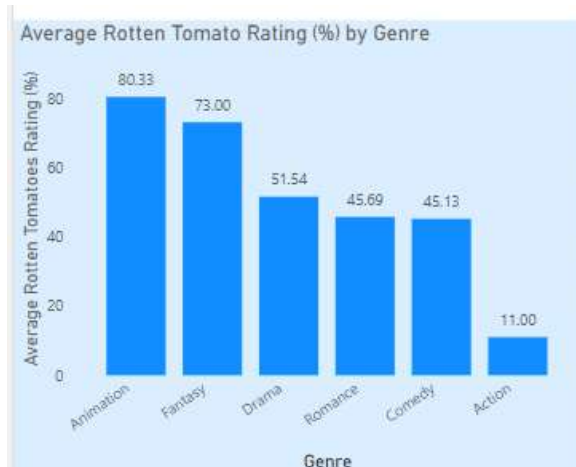
Breif:

Feel free to add more charts apart from the ones mentioned in the next slide.

For the dashboard, the company would like you to use their brand colours which are blue, green and brown. You can use light or dark shades of each colour. For example, light blue and dark blue are acceptable.

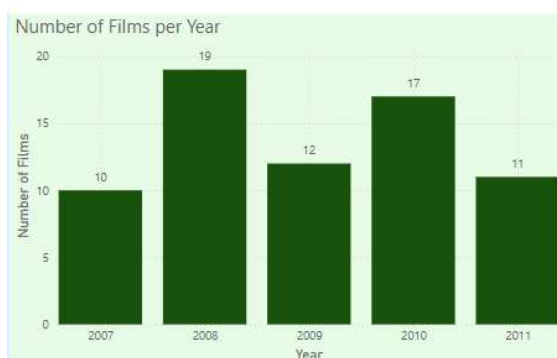
You can combine these colours any way that you like. For example, you can use only blue and green if you want to.

The average rotten tomatoes rating for each genre:



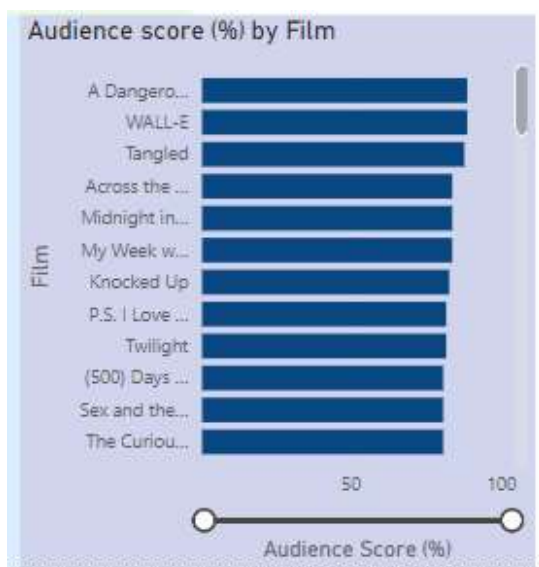
Animation films have the highest Rotten Tomato rating with an average rating of 80.33% scoring a “fresh tomato”. This suggests that Animation films are the most positively reviewed genre in this dataset and are the most enjoyed films by Rotten Tomatoes users. On the other hand, Action films have the lowest average rotten tomatoes rating at 11% scoring a “rotten tomato”. This suggests that Action films are the least well-received genre in this dataset and are the least enjoyed films by the Rotten Tomatoes users.

The number of movies produced each year:



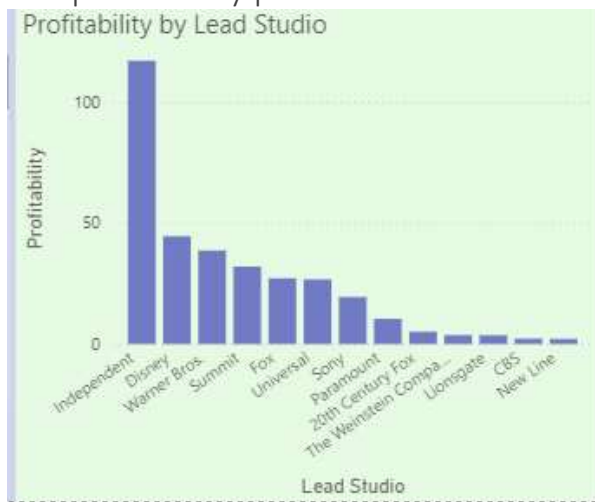
This chart presents trends in film production from 2007 to 2011. This chart shows that the number of films produced peaked in both 2008 at 19 films and 2010 at 17 films. However, these peaks were followed by declines in the following years, with only 12 films produced in 2009 and 11 films in 2011.

The audience score for each film:



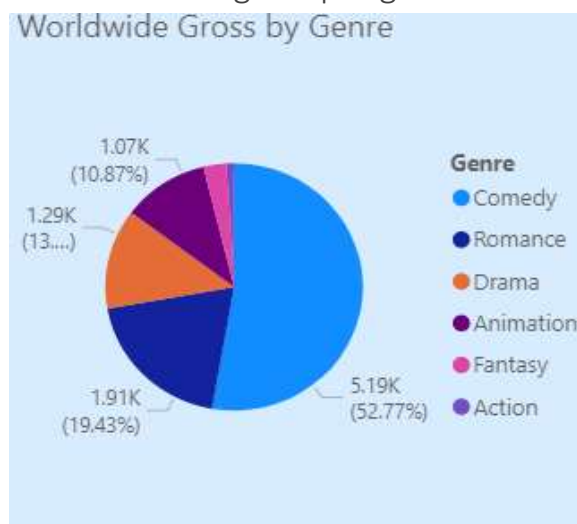
This chart shows the two top rated films by audience score were ‘A Dangerous Method’ and ‘WALL-E’ with both scoring 89% suggesting strong approval from viewers. The two films with the lowest rated films are ‘You Will Meet a Tall Dark Stranger’ scoring 35% and ‘Love Happens’ scoring 40% suggesting low enjoyment from viewers.

The profitability per studio:



This chart compares the profitability of different movie studios, including both major studios (like Disney, Warner Bros., Fox, and Universal) and independent studios. Independent studios are shown to have the highest profitability, miles ahead of the other studios on the chart. They followed by major studios Disney and Warner Bros. The least profitable studios shown on the chart include Lionsgate, CBS, and New Line.

The worldwide gross per genre:



This graph shows that the genre comedy has the highest Worldwide Gross with 52.77% miles ahead of any other genre. Romance genre follows next with 19.43%. Action genre has the lowest Worldwide Gross with 0.95%. This suggests that comedy movies make the most money.

Final Dashboard with extra visualisations:

