



JUST IT, London

DATA TECHNICIAN BOOT CAMP

PROJECT 3

Programming Language R and Power BI

- Registering with Studio R
- Downloading data
- Basic manipulations
- Cleaning Data
- Visualisation

STUDENT: Alexei Kosyhin

Date: 20 Feb 2022

Content

Pages

- About Programming language R and R Studio	3
- Registering and opening Studio R	3
- Uploading data and creating data frame.....	4 -5
- Step1: Initial Exploratory Analysis.....	6
- Cleaning data/ dropping missing values.....	7
- Rounding Values	8
- Step 2.1: Outlier removal	9
- Viewing a cleaned data frame	10
- Step 3: Exploratory Data Analysis	11
- Step 4: Export data.....	12
- Additional: plots, calculating colorations.....	13
- Manipulation and Visualisation in Power BI.....	15 – 17
- Dashboard in Power BI.....	18

About Programming Language R, Registration, Lanching

From Wikipedia: R is a [programming language](#) for [statistical computing](#) and graphics supported by the R Core Team and the R Foundation for Statistical Computing. Created by statisticians [Ross Ihaka](#) and [Robert Gentleman](#), R is used among [data miners](#), [bioinformaticians](#) and [statisticians](#) for [data analysis](#) and developing [statistical software](#).^[7] Users have created packages to augment the functions of the R language.

RStudio is an [integrated development environment](#) for [R](#), a [programming language](#) for [statistical computing](#) and graphics. It is available in two formats: R Studio Desktop is a regular [desktop application](#) while RStudio Server runs on a remote server and allows accessing R Studio using a [web browser](#).

<https://login.posit.cloud> is the link for using an online R Studio version

Login

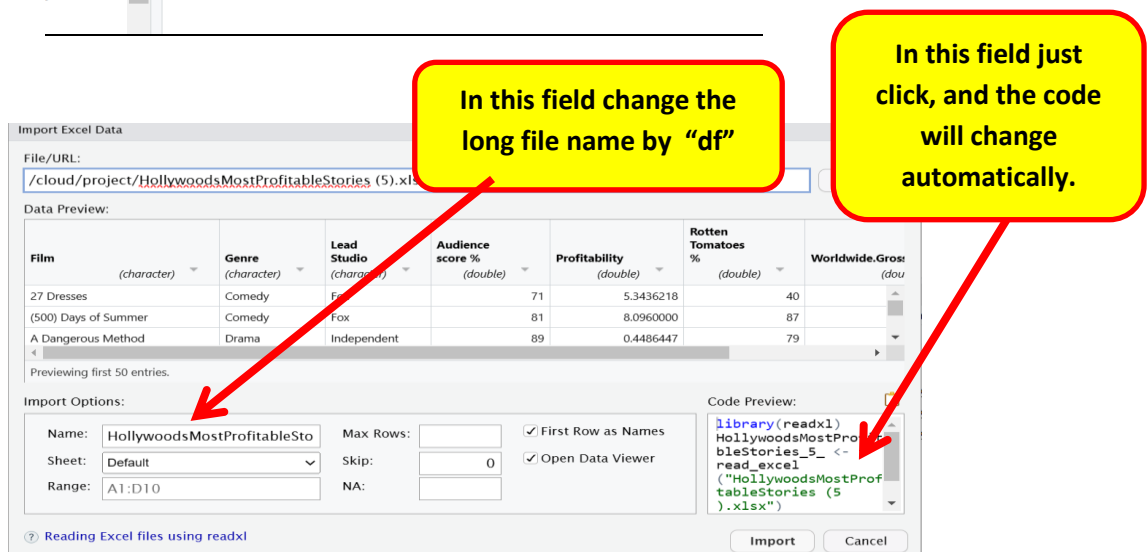
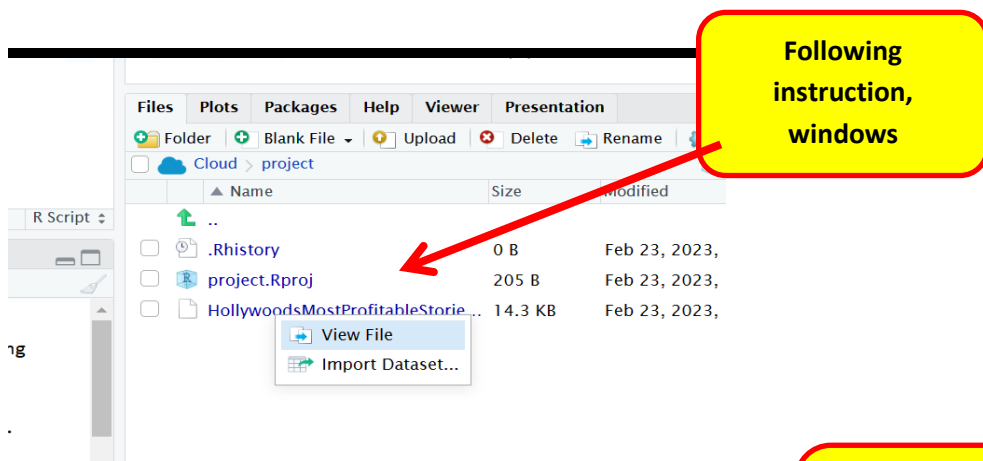
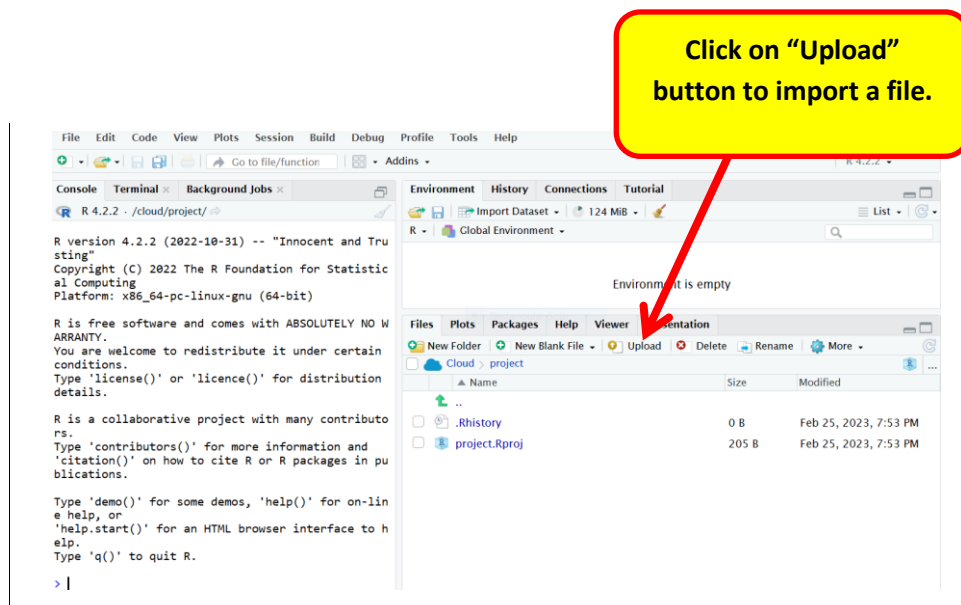
Signing up

Click on here

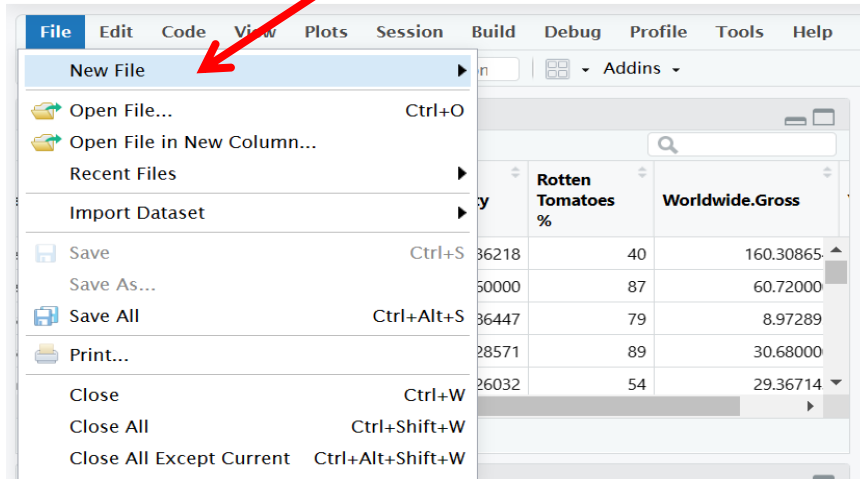
Click on the first project, if first time

This is how R Studio looks like inside

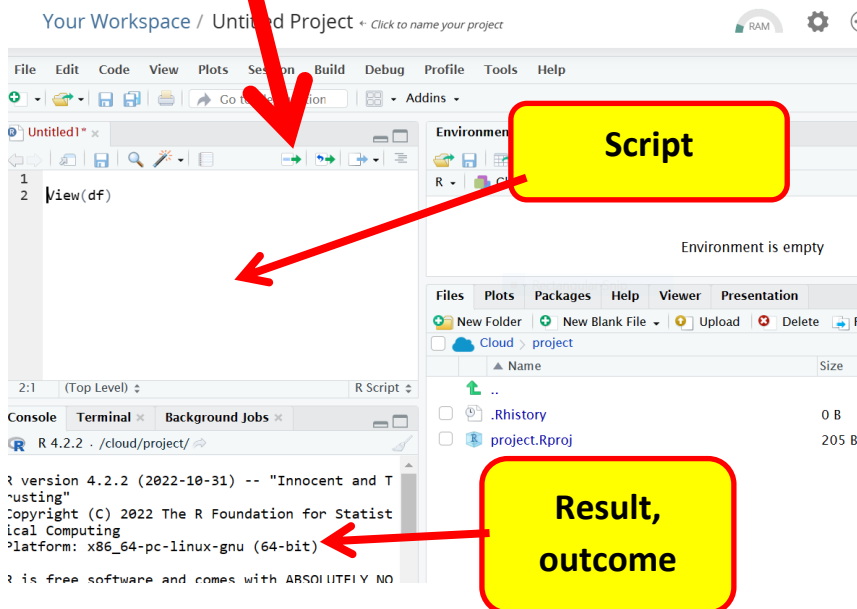
Uploading data and creating data frame



Click a new file,
Then R script, to order to
have a space for scripting



**RUN
COMMAND !!!**



Step1: Initial Exploratory Analysis

Basic manipulation, installing packages and importing libraries.

Viewing the data frame "df" using a command View(df)

Importing a library "tidyverse"

Viewing dataframe dimension - dim(df)

Commands View() and Dim()

```

1 # Viewing date, the table
2 View(df)
3
4 #Checking the table dimension
5 dim(df)
6
7

```

```

> library(readxl)
> df <- read_excel("HollywoodsMostProfitableStories (5).xlsx")
> View(df)
> View(df)
> View(df)
>
> # Viewing date, the table
> View(df)
> #Checking the table dimension
> dim(df)
[1] 74 8

```

The view of the dataframe "df".

Film	Genre	Lead Studio	Audience score %	Profitat
1 27 Dresses	Comedy	Fox	71	
2 (500) Days of Summer	Comedy	Fox	81	
3 A Dangerous Method	Drama	Independent	89	
4 A Serious Man	Drama	Universal	64	
5 Across the Universe	Romance	Independent	84	
6 Beginners	Comedy	Independent	80	

Loading the software packages install.packages("tidyverse")

```

1
2 # Viewing date, the table
3 View(df)
4
5 #Checking the table dimension
6 dim(df)
7
8
9 #Load library: "tidyverse"
10
11 install.packages("tidyverse")
12
13

```

```

R 4.2.2 . /cloud/project/
> install.packages("tidyverse")
* installing *binary* package 'modelr' ...
* DONE (modelr)
* installing *binary* package 'reprex' ...
* DONE (reprex)
* installing *binary* package 'googlesheets4' ...
* DONE (googlesheets4)
* installing *binary* package 'tidyverse' ...
* DONE (tidyverse)
The downloaded source packages are in
'/tmp/Rtmp1PyOKA/downloaded_packages'

```

Importing the library "tidyverse".

```

8
9 #Load library: "tidyverse"
10
11 install.packages("tidyverse")
12
13 #Import library
14 library(tidyverse)
15
16
17
18
19

```

```

R 4.2.2 . /cloud/project/
> library(tidyverse)
The downloaded source packages are in
'/tmp/Rtmp1PyOKA/downloaded_packages'
> library(tidyverse)
Attaching packages:
  tidyr 1.3.0, tibble 3.1.8, dplyr 1.1.0, readr 2.1.4, forcats 1.0.0, tidypurrr 1.0.1, tidypurrr 1.0.1, tidypurrr 1.0.1
Conflicts:
  dplyr::filter() masks stats::filter()
  dplyr::lag() masks stats::lag()

```

Exploring the dataframe structure, Using command "str(df)"

```

12
13 #Import library
14 library(tidyverse)
15
16 # Checking data types:
17 str(df)
18
19
20
21

```

```

R 4.2.2 . /cloud/project/
> dplyr::filter() masks stats::filter()
> dplyr::lag() masks stats::lag()
> str(df)
tibble [74 x 8] (S3: tbl_df/tbl/data.frame)
 $ film_d : chr [1:74] "27 Dresses" "(500) Days of Summer" "A Dangerous Metho
 $ Genre : chr [1:74] "Comedy" "Comedy" "Drama" "Drama" ...
 $ Lead Studio : chr [1:74] "Fox" "Fox" "Independent" "Universal" ...
 $ Audience score % : num [1:74] 71 81 89 64 84 80 66 88 51 52 ...
 $ Profitability : num [1:74] 5.344 8.096 0.449 4.383 0.65 ...
 $ Rotten Tomatoes : num [1:74] 48 87 79 89 54 84 29 93 46 ...
 $ Worldwide.Gross : num [1:74] 160.31 60.72 8.97 30.68 29.37 ...
 $ Year : num [1:74] 2008 2009 2011 2009 2007 ...

```

Step 2: Cleaning Data

Checking missing values. Visually it looks "N/A"

```
# Checking for missing values:
colSums(is.na(df))
```

Film	Genre	Lead Studio	Audience	score %
0	0	1	1	1
Profitability	Rotten Tomatoes %	Worldwide.Gross	Year	0

Using the command `df <- na.omit(df)`

```
#Dropping missing values
#And assigning back to the dataframe
df <- na.omit(df)
```

Checking dimension after "cleaning" command. Dimension was reduced from 74 rows to 69.

```
#checking dataframe dimension
dim(df)
```

```
[1] 69 8
```

Checking data by a different command. All data is cleaned.

```
colSums(is.na(df))
```

Film	Genre	Lead Studio	Audience	score %
0	0	0	0	0
Profitability	Rotten Tomatoes %	Worldwide.Gross	Year	0

Checking data duplicates. No duplicates.

```
#Check for duplicates
dim(df[duplicated(df$Film),])[1]
```

```
[1] 0
```

Rounding values

Your Workspace / Project_3_R+BI

```
35 colSums(is.na(df))
36
37
38
39
40
41
42 #round off values to 2 places
43
44 df$Profitability <- round(df$Profitability ,digit=2)
45
46
47
48
49 (Top Level) :
```

R 4.2.2 . /cloud/project/

Console Terminal Background Jobs

Rounding values to 2 decimal digits

Your Workspace / Project

```
51
52
53 #Check for duplicates
54 dim(df[duplicated(df$Film),])[1]
55 df$Worldwide.Gross <- round(df$Worldwide.Gross ,digit=2)
56
57
58
59
60 (Top Level) :
```

R 4.2.2 . /cloud/project/

Console Terminal Background Jobs

Rounding

Your Workspace / Project_3_R+BI

RAM ⚙️ ⋮ AK Ale

File Edit Code View Plots Session Build De

Filter

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide.Gross	Year
1	27 Dresses	Comedy	Fox	71	5.34	40	160.31	2008
2	(500) Days of Summer	Comedy	Fox	81	8.10	87	60.72	2009
3	A Dangerous Method	Drama	Independent	89	0.45	79	8.97	2011
4	A Serious Man	Drama	Universal	64	4.38	89	30.68	2009
5	Across the Universe	Romance	Independent	84	0.65	54	29.37	2007
6	Beginners	Comedy	Independent	80	4.47	84	14.31	2011

Showing 1 to 7 of 69 entries, 8 total columns

Checking date frame after the commands.

The date was rounded

Removing Outliers

Removing
ouliers/ tails
And checking
dtaframe
dimension

```
Your Workspace / Project_3_R+BI

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1* df
48 dim(df[duplicated(df$Film),])[1]
49 df$Worldwide.Gross <- round(df$Worldwide.Gross, digits=2)
50
51 #Remove outliers in 'Profitability'
52 Q1 <- quantile(df$Profitability, .25)
53 Q3 <- quantile(df$Profitability, .75)
54 IQR <- IQR(df$Profitability)
55
56 no_outliers <- subset(df, df$Profitability > (Q1 - 1.5*IQR) & df$Profitability < (Q3 + 1.5*IQR))
57 dim(no_outliers)
58
59
60

59:1 (Top Level) R Script

Console Terminal Background Jobs
R 4.2.2 . /cloud/project/
> dim(df[duplicated(df$Film),])[1]
[1] 0
>
> Q3 <- quantile(df$Profitability, .75)
> IQR <- IQR(df$Profitability)
> no_outliers <- subset(df, df$Profitability > (Q1 - 1.5*IQR) & df$Profitability < (Q3 + 1.5*IQR))
> dim(no_outliers)
[1] 64 8
>
```

```
Your Workspace / Project_3_R+BI

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1* df
64
65
66 # Remove outliers in 'Worldwide.Gross'
67
68 Q1 <- quantile(no_outliers$Worldwide.Gross, .25)
69
70 Q3 <- quantile(no_outliers$Worldwide.Gross, .75)
71
72 IQR <- IQR(no_outliers$Worldwide.Gross)
73
74
75
76
77

74:1 (Top Level) R Script

Console Terminal Background Jobs
R 4.2.2 . /cloud/project/
> Q3 <- quantile(df$Profitability, .75)
> IQR <- IQR(df$Profitability)
> no_outliers <- subset(df, df$Profitability > (Q1 - 1.5*IQR) & df$Profitability < (Q3 + 1.5*IQR))
> dim(no_outliers)
[1] 64 8
> Q1 <- quantile(no_outliers$Worldwide.Gross, .25)
> Q3 <- quantile(no_outliers$Worldwide.Gross, .75)
> IQR <- IQR(no_outliers$Worldwide.Gross)
>
```

Checking a new cleaned data frame

Your Workspace / Project_3_R+BI

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1* x df x

```
70 Q3 <- quantile(no_outliers$Worldwide.Gross, .75)
71
72 IQR <- IQR(no_outliers$Worldwide.Gross)
73
74
75
76
77
78 df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))
79
80 dim(df1)
81 |
82
```

81:1 (Top Level) R Script

Console Terminal Background Jobs

```
R 4.2.2 . /cloud/project/
>
> df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))
Error: unexpected input in "df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))"
> df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))
> dim(df1)
[1] 60 8
>
```

Checking a new data frame dimension. Dimension was reduced to 60 rows.

Your Workspace / Project_3_R+BI

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1* x df x

Filter

	Film	Genre	Lead Studio	Audience score %	Profitability	Rotten Tomatoes %	Worldwide.Gross	Year
1	27 Dresses	Comedy	Fox	71	5.34	40	160.31	2008
2	(500) Days of Summer	Comedy	Fox	81	8.10	87	60.72	2009
3	A Dangerous Method	Drama	Independent	89	0.45	79	8.97	2011
4	A Serious Man	Drama	Universal	64	4.38	89	30.68	2009
5	Across the Universe	Romance	Independent	84	0.65	54	29.37	2007
6	Beginners	Comedy	Independent	80	4.47	84	14.31	2011

Showing 1 to 7 of 60 entries, 8 total columns

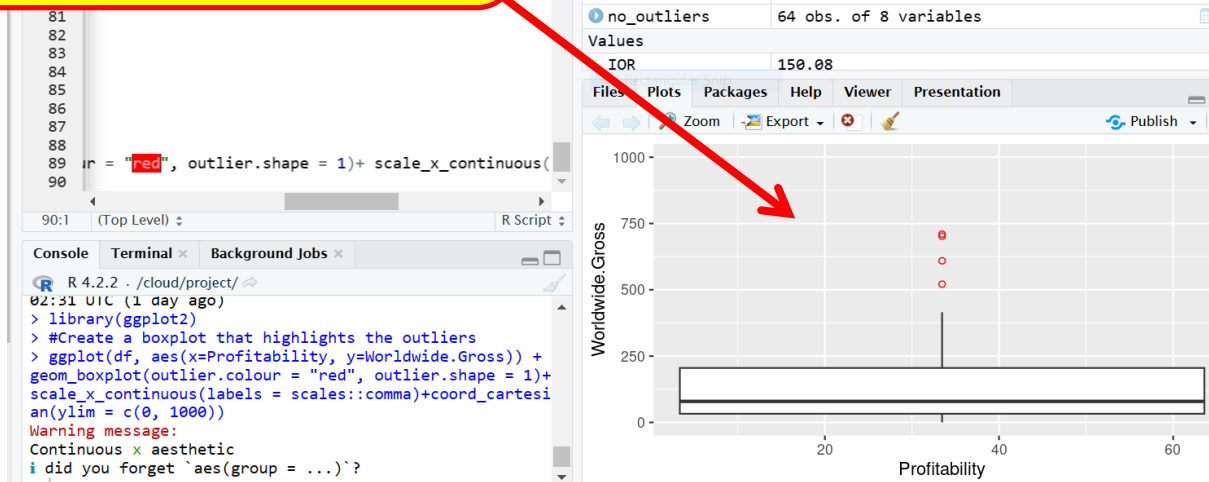
Console Terminal Background Jobs

```
R 4.2.2 . /cloud/project/
> df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))
Error: unexpected input in "df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))"
> df1 <- subset(no_outliers, no_outliers$Worldwide.Gross > (Q1 - 1.5*IQR) & no_outliers$Worldwide.Gross < (Q3 + 1.5*IQR))
> dim(df1)
[1] 60 8
> View(df1)
>
```

Viewing a new cleaned data frame "df1"

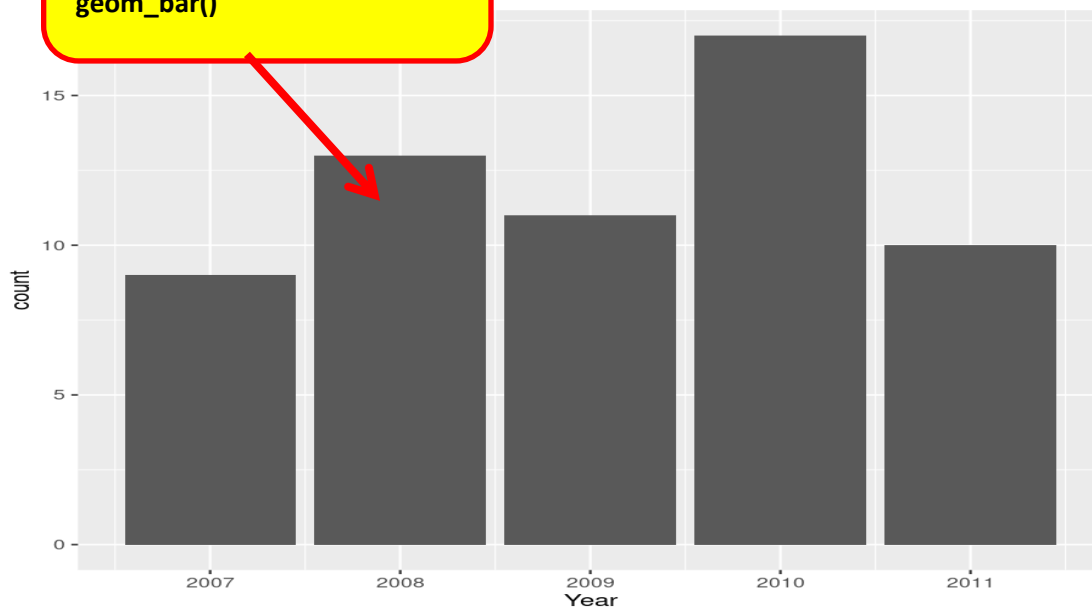
Step 3: Exploratory Data Analysis

```
#scatterplot  
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))
```



#bar chart

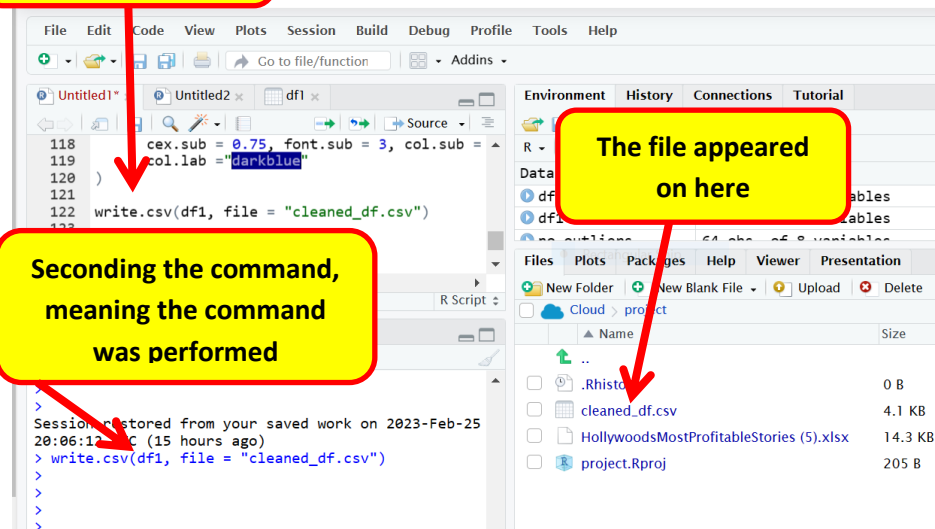
```
ggplot(df1, aes(x=Year)) +  
geom_bar()
```



Step 4: Export data

It's vitally important to save the cleaned data and working scripting files to a local disk !!!
Because it can be used for next analysis, proof of the work done, repeating scripting again.

A command to
copy a data frame.

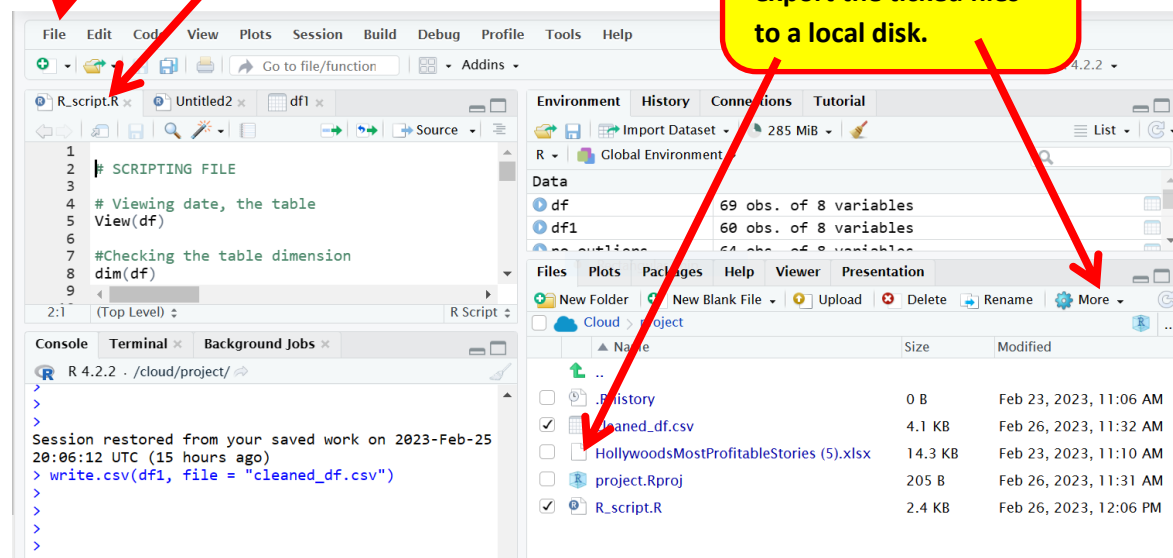


The file appeared
on here

Seconding the command,
meaning the command
was performed

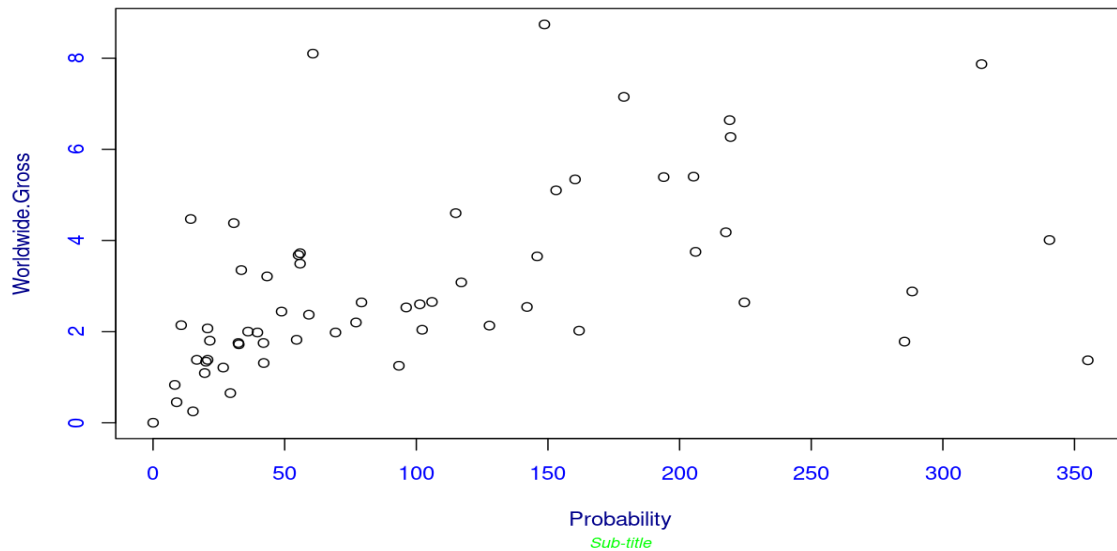
Click on the script file,
click on "File" to save
the script file.

Tick the files,
Click the icon "gear" to
export the ticked files
to a local disk.



Additional plots and analysis

Bivariable Analysis



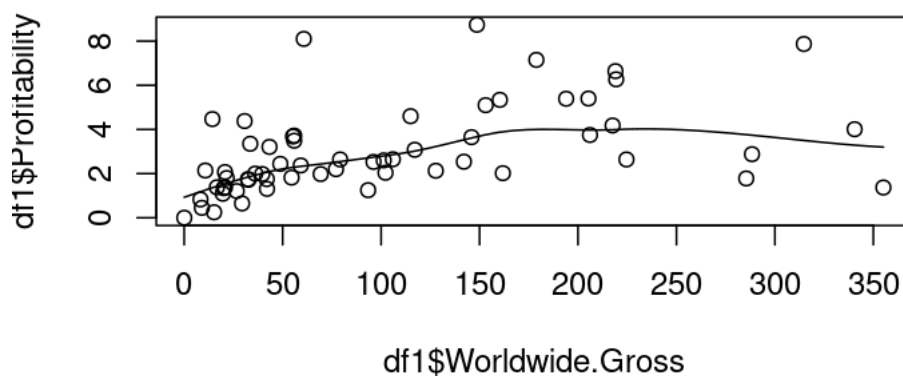
Using this R script:

```
plot(df1$Worldwide.Gross, df1$Profitability, main = "Probability vs World.Gross")
```

Plotting scatter using R

```
scatter.smooth(df1$Worldwide.Gross, df1$Profitability, main='Project 3, gross vs profitability')
```

Project 3, gross vs profitability



Calculating correlations between variables:

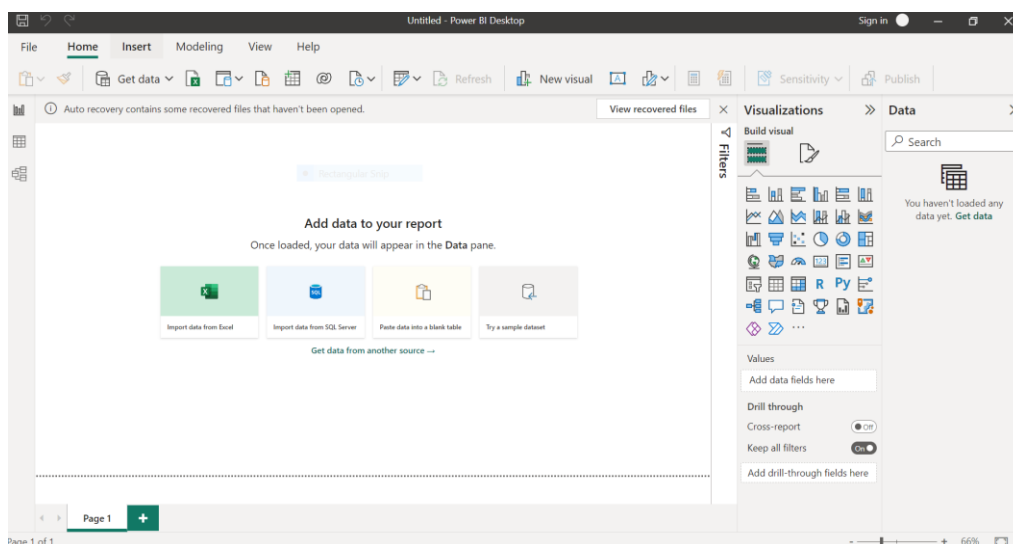
```
78 # CALCULATING COROLATIONS:|
79
80 #Calculating correlations for the variables
81 cor(df1$Profitability, df1$Worldwide.Gross)
82 #0.47 which moderate.
83
84 cor(df1$Profitability, df1$`Rotten Tomatoes %`)
85 #0.146 which very weak
86
87 cor(df1$Worldwide.Gross, df1$`Rotten Tomatoes %`)
88 # 0.035 no correlation
89
90 cor(df1$`Audience score %`, df1$`Rotten Tomatoes %`)
91 # 0.60 is a strong correlation
92
```

Manipulation and Visualisation in Power BI

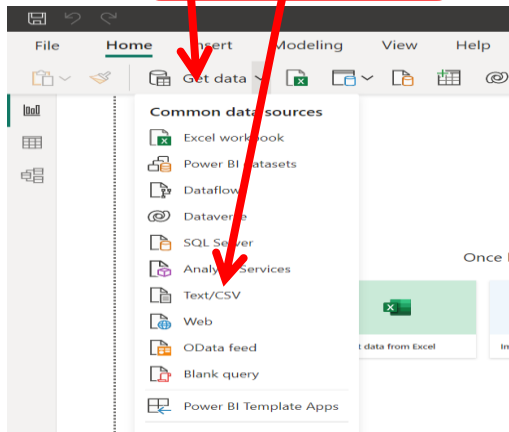
From Wikipedia Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence.[1] It is part of the Microsoft Power Platform. Power BI is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data may be input by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

Microsoft Power BI	
	
Developer(s)	Microsoft
Initial release	11 July 2011; 11 years ago
Stable release	March 2021 Update (2.91.383.0) / March 2021; 2 years ago
Operating system	Microsoft Windows
Type	Data visualization Business intelligence
License	Proprietary
Website	powerbi.microsoft.com

When we open Power BI, the software look like this:



Importing data from a local disk to Power BI as a text file.

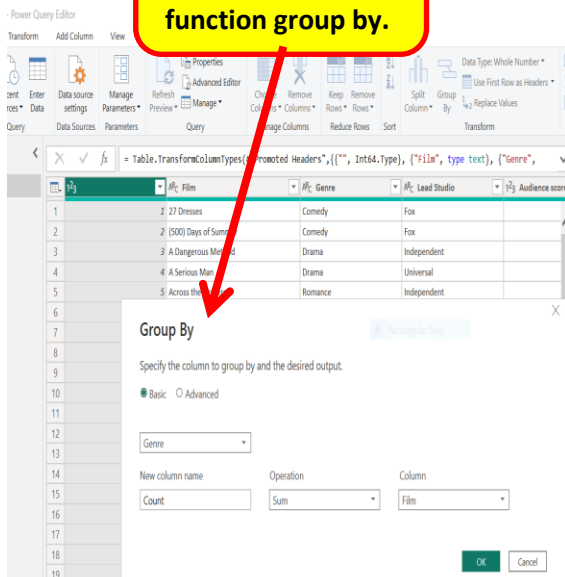


We can view this data as a table in Power BI.

The screenshot shows the 'Table tool' in Power BI Desktop. A table named 'cleaned_df' is displayed with columns: Film, Genre, Lead Studio, Audience score %, and Profitability. The table contains 28 rows of movie data.

	Film	Genre	Lead Studio	Audience score %	Profitability
1	27 Dresses	Comedy	Fox	71	5.34
2	(500) Days of Summer	Comedy	Fox	81	8.1
3	A Dangerous Method	Drama	Independent	89	0.45
4	A Serious Man	Drama	Universal	64	4.38
5	Across the Universe	Romance	Independent	84	0.65
6	Beginners	Comedy	Independent	80	4.47
7	Dear John	Drama	Sony	66	4.6
8	Enchanted	Comedy	Disney	80	4.01
9	Four Christmases	Comedy	Warner Bros.	52	2.02
10	Ghosts of Girlfriends Past	Comedy	Warner Bros.	47	2.04
11	Gnomeo and Juliet	Animation	Disney	52	5.89
12	Going the Distance	Comedy	Warner Bros.	56	1.81
13	Good Luck Chuck	Comedy	Lionsgate	61	2.87
14	He's Just Not That Into You	Comedy	Warner Bros.	60	7.15
15	I Love You Phillip Morris	Comedy	Independent	57	1.94
16	It's Complicated	Comedy	Universal	63	2.64
17	Just Wright	Comedy	Fox	58	1.8
18	Killers	Action	Lionsgate	45	1.25

To group by data we use Power BI function group by.

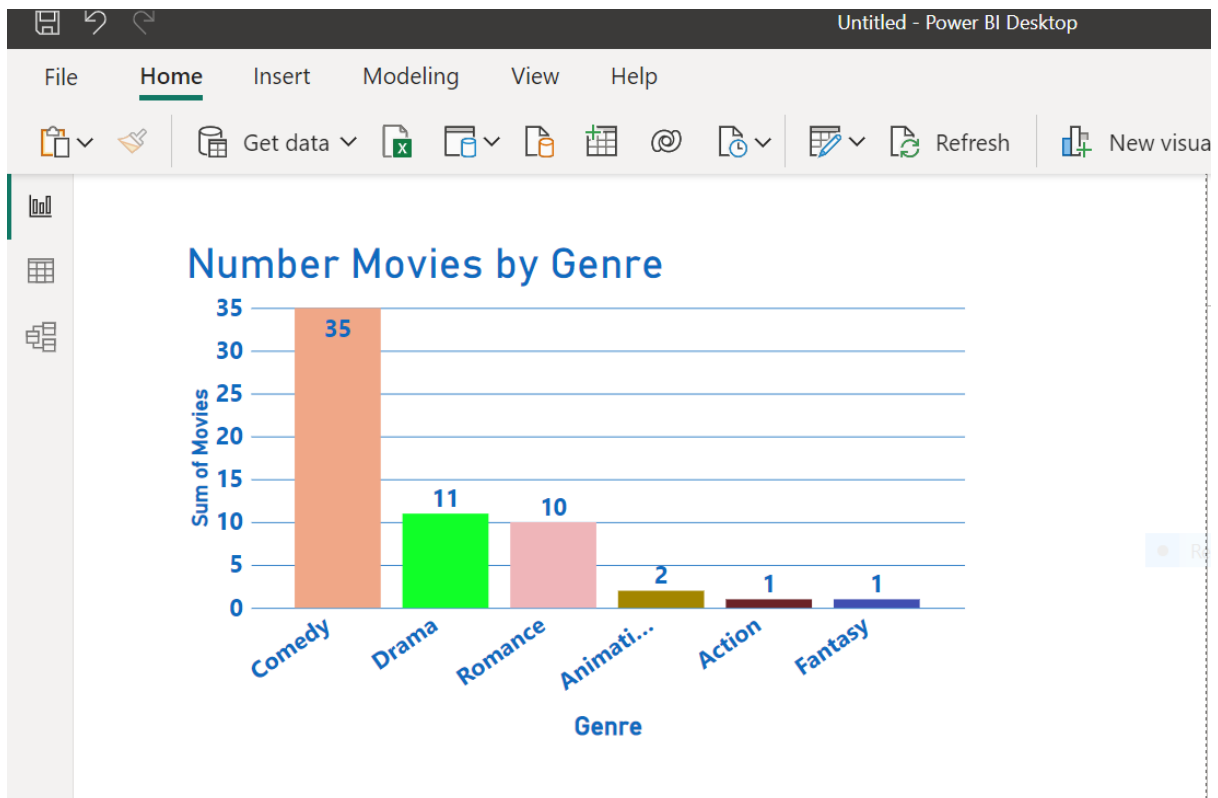
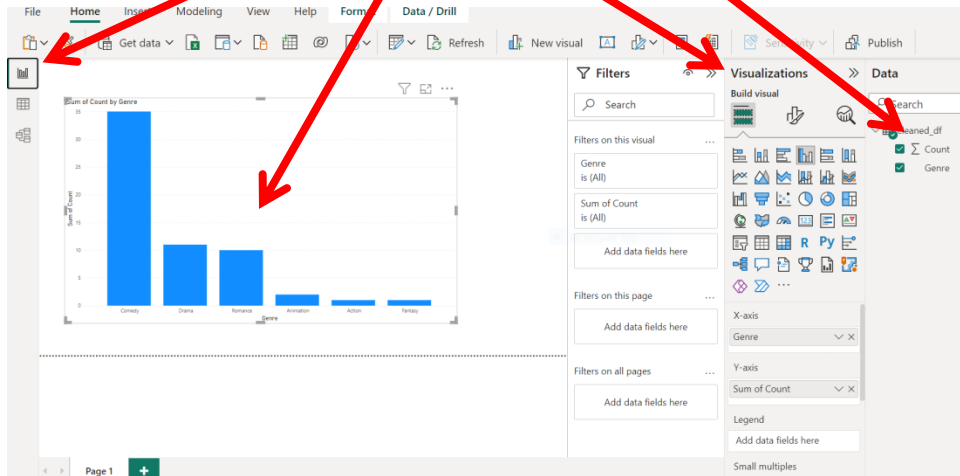


Grouped by data.

The screenshot shows the Power Query Editor with the grouped data table. The formula bar shows: `= Table.Group("#'Changed Type", {"Genre"}, {{"Count", Sum, "Film"}})`. The table has columns: Genre and Count.

Genre	Count
1 Comedy	35
2 Drama	11
3 Romance	10
4 Animation	2
5 Action	1
6 Fantasy	1

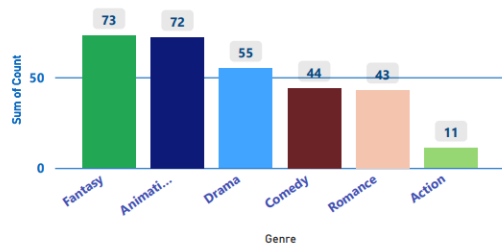
Drawing a chart using Power BI visualisation tools



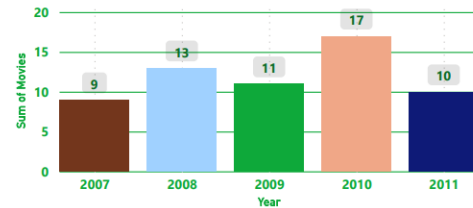
Beautiful dashboard

Rectangular Snip

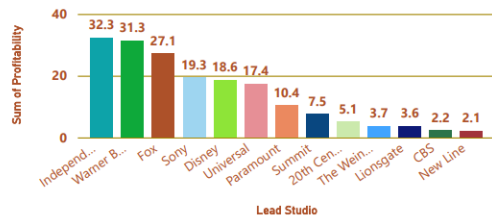
Number Movies by Genre



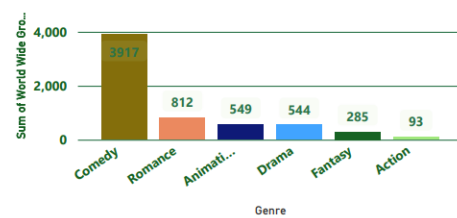
Number Movies by Year



Profitability by Lead Studio



Worldwide Gross by Genre



<https://app.powerbi.com/groups/me/reports/9db75f58-2430-4a1f-84b8-df477a15d763/ReportSectionbc6680698ec7a709c444>

[Beautiful Dashboard](#)