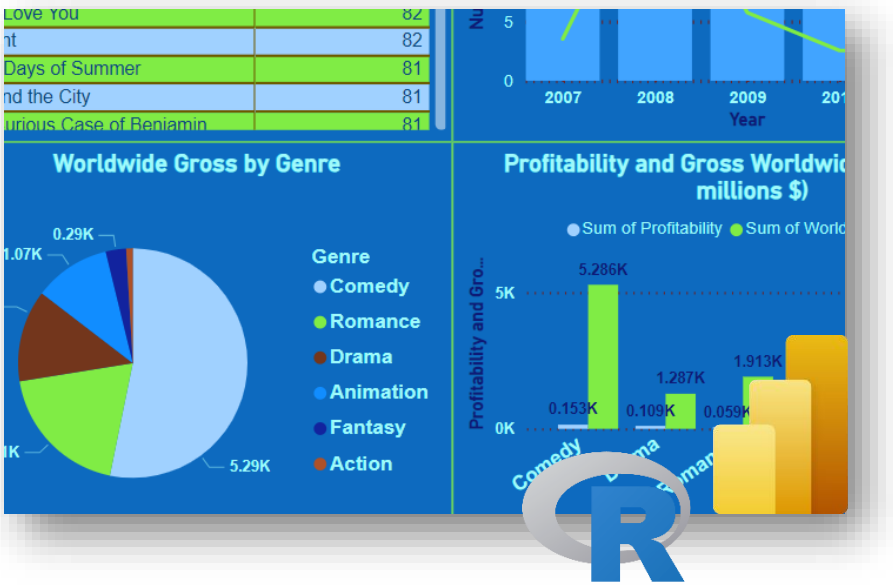
CLEANING AND VISUALISATION OF DATA

(R AND POWER BI)



Jelena Cekmeniova

Cohort: GLA 16

Submission Date: 05.10.2023

Contents

Figures

1. Overview

R programming language tends to be popular tool to analyse data, particular in academic environment. Ross Ihaka and Robert Gentleman stand at the foundation of R in mid 1990s (Tuimala and Kallio, 2013) and until now available as a free version for interested to learn it, under terms of Free Software Foundation's GNU General Public License (The R Foundation, 2023).
R is relevantly easy to use and include wide range of statistical functions (statistical summaries, modelling, graphical representation), enabling statisticians and analysts produce informative reports (The R Foundation, 2023).

This project is based on using RStudio platform to import, perform initial analysis and clean "Hollywood's Most Profitable Stories" data set, acquired from Tableau Public webpage with further visualisation of results in Power BI.

2. Initial Analysis and Cleaning Data with R

This section focuses on importing dataset into RStudio environment as well as installing "Tidyverse" library for data analysis, using R programming. Fig. 1, below, illustrates how the data-frame "df" created from data-set "Hollywood Most Profitable Stories", that was downloaded from Tableau Public website. "view(df)" function opens data table in the script window and allows initial exploration/ observation of rows and columns' content.

Fig 1. – Importing Data Set to RStudio Environment and Installing "Tidyverse" Library

```
# Initial Exploratory Analysis

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

view(df)

install.packages("tidyverse")

view(df)

library(tidyverse)

str(df)


```

Initial data-set / dataframe contains 74 observations across 8 variables. However, some data is unavailable. As per rrequirement, null values were omitted to preserve reliability and consistency of further data analysis to ensure correct business decisions (Fig.2). In real setting, "Data quality is an important parameter in determining the relevant use cases for a data source, as is being able to rely on data for further calculations/ inclusions with other data sets" (Eryurek et al., 2021).

Methods for data cleaning/ normalisation or dealing with missing values/ outliers may include: deleting missing values (entire row / rows or column/ columns); finding and filling-in these values; filling with arbitrary values/ averages/ median/ mode values; replacing nulls with most frequent values; using programming models to predict potentaial missing values (Petrella, 2023).

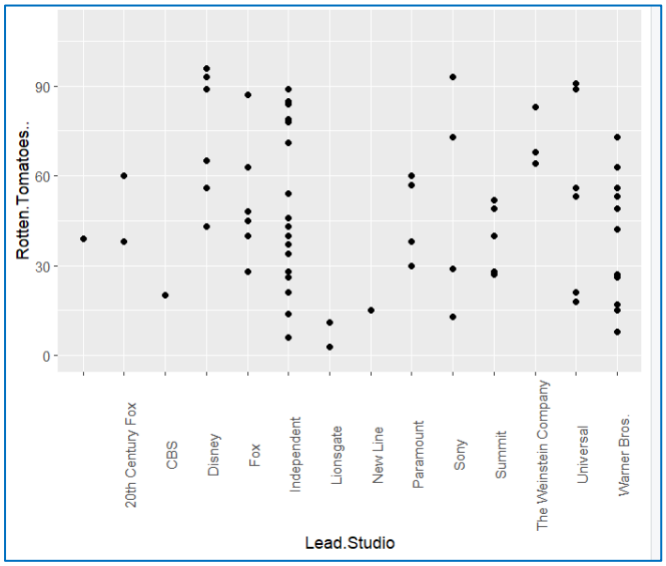
After dropping rows with the null values, new dataframe “df1” was created, containing just 70 observations of 8 variables.

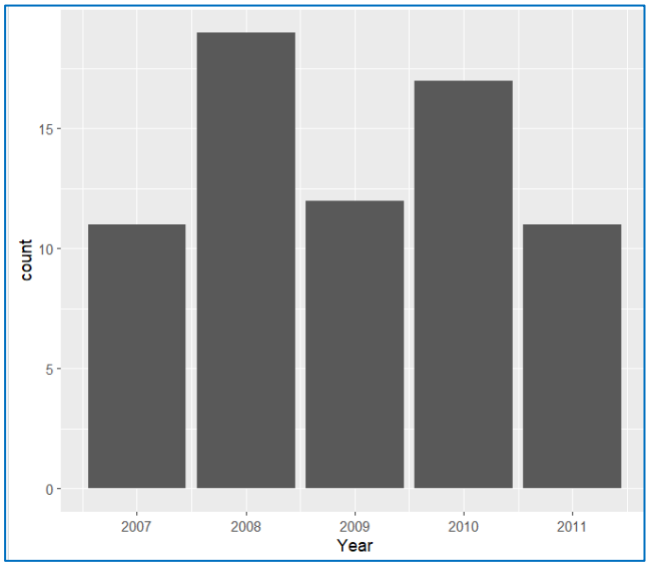
Fig. 2 – Dropping Rows with Null Values and Exporting Cleaned Data

```
# Cleaning Data

colSums(is.na(df))

df1 <-df%>% drop_na()

colSums(is.na(df1))


# Exploratory Data Analysis

colSums(is.na(df1))

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))

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


# Export Cleaned Data

write.csv(df1, "~/bootcamp IT tech/R_assignment/clean_df.csv")
```

Cleaned data was also visualised, using “ggplot” package function for R programming as scatter plot (Rotten Tomatoes Rating / Lead Studio) and bar chart (number of movies released yearly) (Fig.3).

Fig 3. – Scatter Plot and Bar Chart, created using “ggplot” functionality.





Finally cleaned dataset was exported as csv file.

3. Summarising Data

Fig.4 shows additional statistical analysis for entire data-frame and then just columns, containing numeric data. For example, “summary()” function renders results for Min., 1st quartile, Median, Mean, 3rd quartile and Max. values. Statistics may be also calculated using separate functions like “mean()”, “median()”, “sd()” (standard deviation), “cov()” and “cor()” (covariance and correlation).

Fig. 4 – Essential Summary Statistics Analysis

```
40
41  #Additional practice
42
43  write.csv(df1, "~/bootcamp IT tech/R_assignment/clean_df2.csv")
44
45  df2 <- df1
46
47  summary(df2)
48  summary(df2$Profitability)
49  summary(df2$Worldwide.Gross)
50
51  mean(df2$Profitability)
52  median(df2$Profitability)
53  sd(df2$Profitability)
54  cov(df2$Profitability, df2$Worldwide.Gross)
55  cor(df2$Profitability, df2$Worldwide.Gross)
56
57  class(df2)
58  class(df2$Genre)
59  class(df2$Profitability)
60  class(df2$Audience..score..)
61
62  typeof(df2$Lead.Studio)
63  typeof(df2$Rotten.Tomatoes..)
64  typeof(df2$Worldwide.Gross)
65
66  table(df2$Film, df2$Genre)
67  df2[c("Film", "Genre")]
68
```

Further, author checked datatype of some variables using “class()” and “typeof()” functions and tested selection of specific columns (“Film” and “Genre”) with “table()” and data-frame viewing functionality. While first one generates temporary table with, showing the logic how count is performed by categories in R, second function just returns two requested columns (Fig.5).

Fig. 5 – “table()” Function Results

	Action	Animation	Comedy	Drama	Fantasy	Romance
(500) Days of Summer	0	0	1	0	0	0
27 Dresses	0	0	1	0	0	0
A Dangerous Method	0	0	0	1	0	0
A Serious Man	0	0	0	1	0	0
Across the Universe	0	0	0	0	0	1
Beginners	0	0	1	0	0	0

In order to understand how sorting functionality works in R programming in comparison to MySQL and Python, author also tested “group by()” and “ifelse()” statements.

For instance, Fig. 6 demonstrates creation of new data frame “df3” with only three columns (“Genre”, “mean_profit” and “mean_gross”). Aggregated results for original “Profitability” and “Worldwide.Gross” variables are grouped by genre.

Fig. 6 – Grouping by Average Profitability by Genre

```
68
69  #group by
70
71  df3 <- df2 %>% group_by(Genre) %>%
72    summarise(mean_profit=mean(Profitability),
73              mean_gross=mean(Worldwide.Gross),
74              .groups = 'drop') %>%
75    as.data.frame()
76  df3
77
```

	Genre	mean_profit	mean_gross
1	Action	1.245333	93.40000
2	Animation	3.216561	356.77681
3	Comedy	3.935434	135.54348
4	Drama	8.407218	99.01137
5	Fantasy	1.783944	285.43100
6	Romance	4.575390	147.13857

On another side, nested “ifelse()” shows how new column can be added to existing data-frame, while values (“Low Score”, “High Score”, “Mid Score”) in this column are assigned depending on condition (“less than”, “greater than or equa”)l, refering to values in another column “Audience..score..” (Fig.7).

Fig. 7 – Using ”if else()” to Sort Data and Place in New Column

```
77
78  #if else
79
80  df4 <- df2
81  df4$Rating = ifelse(df4$Audience..score..<50, "Low Score",
82               ifelse(df4$Audience..score..>=80, "High Score",
83                      "Mid Score"))
84  df4
```

ilm	Genre	Lead.Studio	Audience..score..	Profitability	Rotten.Tomatoes..	Worldwide.Gross	Year	Rating
7 Dresses	Comedy	Fox	71	5.3436218	40	160.308654	2008	Mid Score
500) Days of Summer	Comedy	Fox	81	8.0960000	87	60.720000	2009	High Score
A Dangerous Method	Drama	Independent	89	0.4486447	79	8.972895	2011	High Score
A Serious Man	Drama	Universal	64	4.3828571	89	30.680000	2009	Mid Score
Across the Universe	Romance	Independent	84	0.6526032	54	29.367143	2007	High Score
Beginners	Comedy	Independent	80	4.4718750	84	14.310000	2011	High Score
Dear John	Drama	Sony	66	4.5988000	29	114.970000	2010	Mid Score
Enchanted	Comedy	Disney	80	4.0057371	93	340.487652	2007	High Score
Fireproof	Drama	Independent	51	66.9340000	40	33.467000	2008	Mid Score
Four Christmases	Comedy	Warner Bros.	52	2.0229250	26	161.834000	2008	Mid Score
Ghosts of Girlfriends Past	Comedy	Warner Bros.	47	2.0444000	27	102.220000	2009	Low Score
Gnomeo and Juliet	Animation	Disney	52	5.3879722	56	193.967000	2011	Mid Score
Going the Distance	Comedy	Warner Bros.	56	1.3140625	53	42.050000	2010	Mid Score
Good Luck Chuck	Comedy	Lionsgate	61	2.3676851	3	59.192128	2007	Mid Score

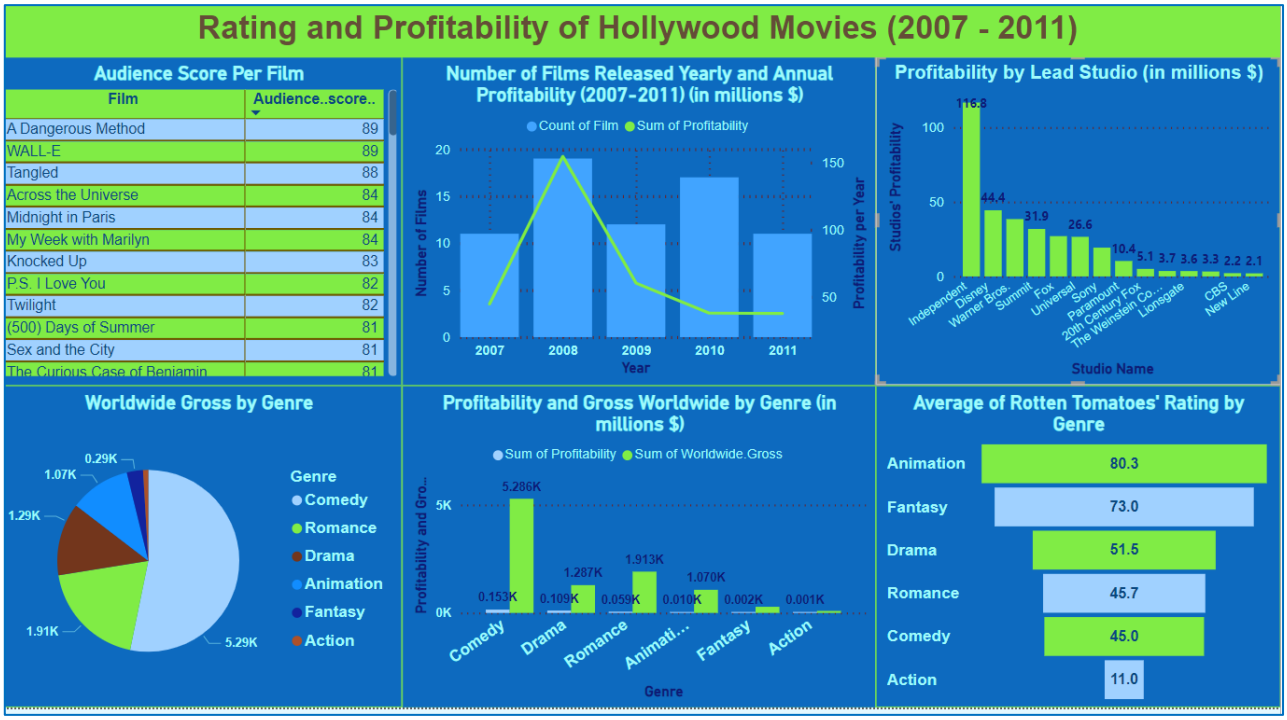
4. Data Visualisation with Power BI

After data cleansing with R, resulted csv file was uploaded to Power BI for creating collection of various interactive charts, as per project requirements. Dashboard contains following visuals: Audience Score per Film; Number of Films Released Yearly and Annual Profitability (industry-wise); Profitability by Lead Studio; Worldwide Gross Revenue by Genre; Profitability and Gross Revenue Worldwide by Genre; Average of "Rotten Tomatoes" Rating by Genre.

Aggregated figures are represented in thousands of millions, and representing differences by categories ("Genre", "Lead Studio") and changes over time.

Dashboard itself was created using Brown/ Blue/ Green colour scheme, as requested by client.

Fig. 8 – Power BI Dashboard "Rating and Profitability of Hollywood Movies (2007–2011)



5. Self-reflection on Project

Summarising experience working with R programming language and then creating visual in Power BI, it can be described as semi-challenging.

On one hand, R programming utilises similar, to some extent, function names as Python and MySQL, however, syntax differs, and for not very experienced user like me, can be difficult to find, understand and debug issues within code.

Though, on other hand, creating dashboards in Power BI seems to be easier than performing same activity in Tableau, due to greater exposure working with various Microsoft Office tools over the years.

Moving forward, I definitely looking to acquire more practice by working with various datasets from publicly available depositories, that would help me to develop more confidence as aspiring data analyst.

6. References

Eryurek, E., Uri, G., Lakshmanan, V., Kibunguchy-Grant, A. and Ashdown, J. (2021) *Data Governance. The definitive Guide.* Sebastopol: O'reilly.

Petrella, A. (2023) *Fundamentals of Data Observability*. O'reilly ed. Sebastopol.

The R Foundation (2023) *What is R?* The R Foundation. Available from: https://www.r-project.org/about.html [Accessed 05 October 2023].

Tuimala, J. and Kallio, A. (2013) R, Programming Language. *In: Encyclopedia of Systems Biology*.: Springer Science+Business Media, 1809–1811.