Introduction to Data Science

Team 15:
Thijs van der Knaap (s2752077)
Hatim Alsayahani (s3183696)
Sebastian Wehkamp (s2589907)
Dimitri Laskaratos

September 10, 2017

1 Overview

1. the group decided to solve the assignment by using two different technology R and Python, this report include the solutions of all Exercises ,Python code and R code also The Images of analysis .

2 Exercise 1.1

1. Brightness as measured by a light meter: continuous, quantitative, ratio
2. Brightness as measured by people's judgements: continuous, qualitative, ordinal: People don't have a linear scale of perceiving light. People however can indicate if some brightness is higher than another brightness.
3. Time in terms of AM or PM: discrete, quantitative, ratio
4. Coat check number: binary, qualitative, nominal

3 Exercise 1.2

We implemented the merge of data from the given CSV file and the OMDB API with both Python an R. Both implementations can be found below and in the Github repository. For every entry in the CSV file we querry the API using the movie title and if given the release date. If the API gives a match we retrieve the extra data and store it in the result table. If the API doesn't give a match we assume that the movie doesn't exist and we don't add it.

3.1 Python

:

```
import pandas as pd
import urllib
import json
import csv
from dateutil.parser import parse


#Needed to check if date is valid
def is_date(string):
    try:
        parse(string)
        return True
    except ValueError:
        return False

reader=pd.read_csv('movievalue.csv', encoding='latin1').dropna(subset=['Title']
, how='all')
i=0

with open('new.csv', 'w', newline='') as new:
    writer=csv.writer(new)
    writer.writerow(('Title', 'ReleaseDate', 'Popularity', 'Budget', 'Revenue'
    , 'Director', 'Genre', 'IMDBRating', 'IMDBVotes', 'BoxOfice', 'Production'))

    for title in reader.Title:

        link = "http://www.omdbapi.com/?apikey=863c5282&t=" + title.replace(" ", "+")

        if is_date(reader.ReleaseDate[i]):
            link += "&y=" + str(parse(reader.ReleaseDate[i]).year)

        try:
          with urllib.request.urlopen(link, timeout=10) as response:
              data=response.read().decode()
              result=json.loads(data)
              writer.writerow((reader.Title[i], reader.ReleaseDate[i]
              , reader.Popularity[i], reader.Budget[i], reader.Revenue[i]
              , result['Director'], result['Genre'], result['imdbRating']
              , result['imdbVotes'], result['BoxOffice'], result['Production']))
              i+=1
        except Exception:
            i+=1
        if i==10000:
            break
```

3.2 R

```
install.packages(c("openssl","httr", "jsonlite", "lubridate"))
library("httr")
library(jsonlite)
library(lubridate)
imdbCSV <- read.csv(file.choose(new = FALSE))

resultFrame <- data.frame( Title=character(),
                   ReleaseDate=as.Date(character()),
                   Popularity=double(),
                   Budget=double(),
                   Revenue=double(),
                   Genre=character(),
                   ImdbRating=double(),
                   ImdbVotes=integer(),
                   Director=character(),
                   Country=character()
                   )

for (i in 1:length(imdbCSV$Title)) {
  #check if the name doesn't contain special characters
  if (grepl('[^[:alnum:]]', imdbCSV$Title[i])) {
    title <- imdbCSV$Title[i]

    #releaseDate
    if (imdbCSV$ReleaseDate[i] != "'NaT'") {
      #it is a properdate
      releaseDate <- as.Date(imdbCSV$ReleaseDate[i], "'%d/%m/%Y'")
    } else {
      releaseDate <- NA
    }

    popularity <- imdbCSV$Popularity[i]

    if (imdbCSV$Budget[i] != 0) {
      budget <- imdbCSV$Budget[i]
    } else {
      budget <- NA
    }

    if (imdbCSV$Revenue[i] != 0) {
      revenue <- imdbCSV$Revenue[i]
    } else {
      revenue <- NA
    }
```

```
    url <- paste0("http://www.omdbapi.com/?apikey=863c5282&t=",
        gsub(' ', "%20", imdbCSV$Title[i]))
    if (!is.na(releaseDate)) {
      url <- paste0(url,"&y=",format(releaseDate, "%Y"))
    }

    #retrieve the movie
    rawData <- GET(url = url)
    if (rawData$status_code == 200) {
      movieData <- fromJSON(rawToChar(rawData$content))
      if (movieData$Response == "True") {
        #the movie exists
        print(paste0("It Exists: ",i))
        print(movieData$Title)
        resultFrame <- rbind(resultFrame, data.frame( Title=title,
                                                      ReleaseDate=releaseDate,
                                                      Popularity=popularity,
                                                      Budget=budget,
                                                      Revenue=revenue,
                                                      Genre=movieData$Genre,
                                                      ImdbRating=movieData$imdbRating,
                                                      ImdbVotes=movieData$imdbVotes,
                                                      Director=movieData$Director,
                                                      Country=movieData$Country
                                                      )
        )
      }
    }
  }
}
```

4 Exercise 1.3

- **Title**: Qualitative, Nominal because it is labeling variables
- **Year**: quantitative, discrete, because there is no fractions
- **Released**: quantitative, discrete, because there is no fractions
- **RunTime**: quantitative, continuous because it could be in fraction like 103.22.15 min
- **Genre**: Qualitative, Nominal because it is labeling variables
- **Director**: Qualitative, Nominal because it is labeling variables
- **Writer**: Qualitative, Nominal because it is labeling variables
- **Actors**: Qualitative, Nominal because it is labeling variables

- **Plot**: Qualitative, Nominal because it is labeling variables
- **Language**: Qualitative, Nominal because it is labeling variables
- **Country**: Qualitative, Nominal because it is labeling variables
- **Awards**: quantitative, discrete, because there is no fractions, there is no half award
- **Rating**: quantitative, continuous, because it could be 7.7 out of 10
- **IMDB votes**: quantitative, discrete, because there is no fractions, there is no 320,566.5 votes
- **Box-office**: quantitative, continuous, because it could be 10000000.554

5 Exercise 1.4

All graphs below are created without the zero values in the respective categories.

5.1 Does a movie get a higher IMDBRating if the movie has a higher budget

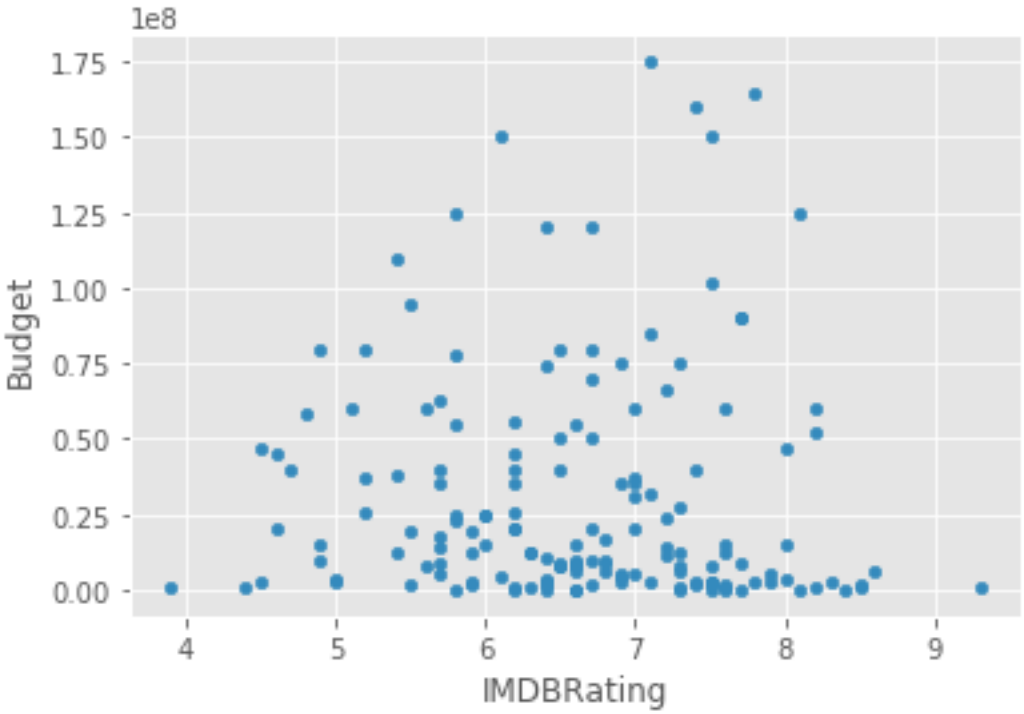

Figure 1: Scatter plot of the IMDB Rating versus the Budget

There does not seem to be much correlation between the IMDBRating and the budget. There are movies which have an IMDB rating of 6 with a very high budget and there are movies with an IMDB rating of 9 with a very low budget.

5.2 Does the revenue increase if the budget increases?

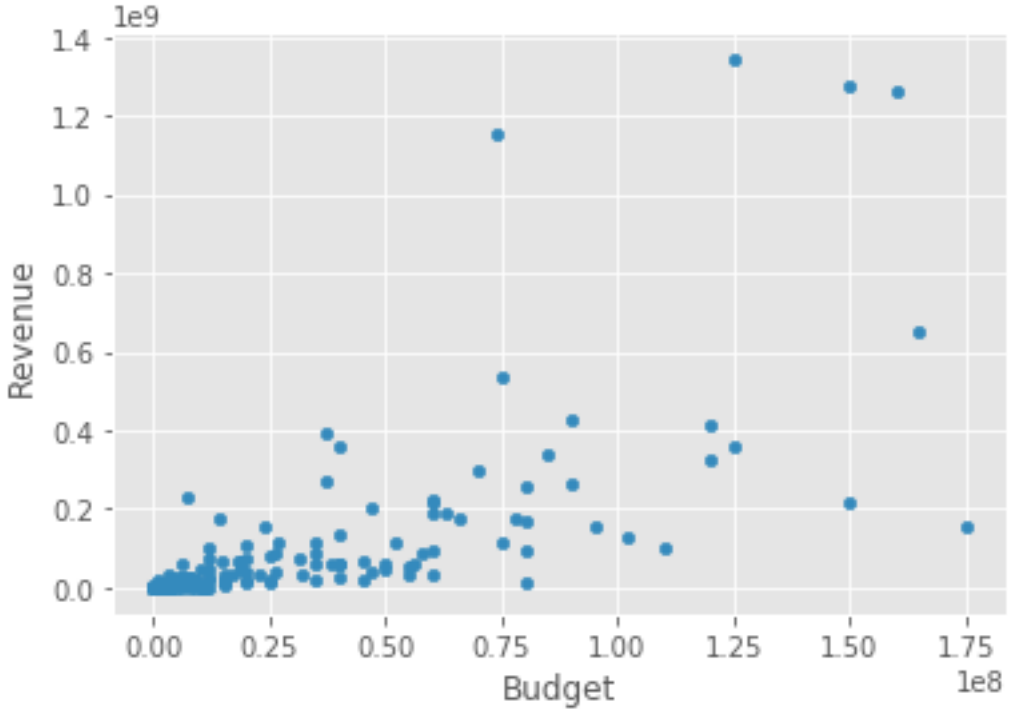

Figure 2: Scatter plot of the Budget versus the Revenue

There is a slight increase in revenue if the budget increases.

5.3 Does the average revenue increase over the years?

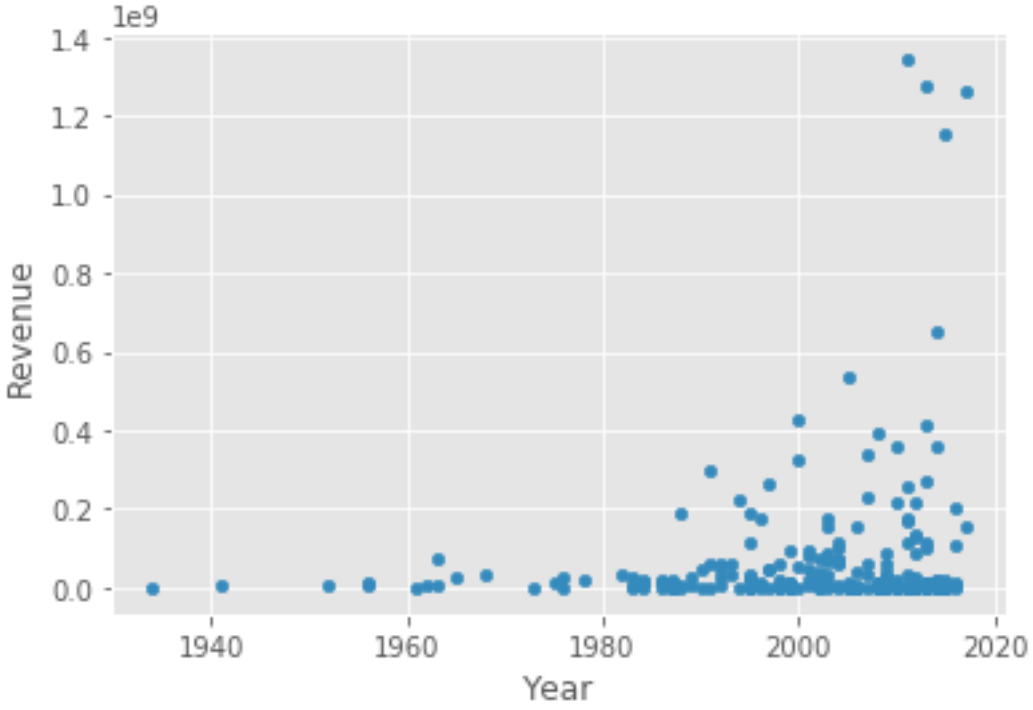

Figure 3: Scatter plot of the Release year versus the Revenue

In plot Figure 3 you can see the revenue plotted with respect to the release year. In general you can say that the revenue started increasing from 1980s onward.

5.4 Does the amount of released movies increase?

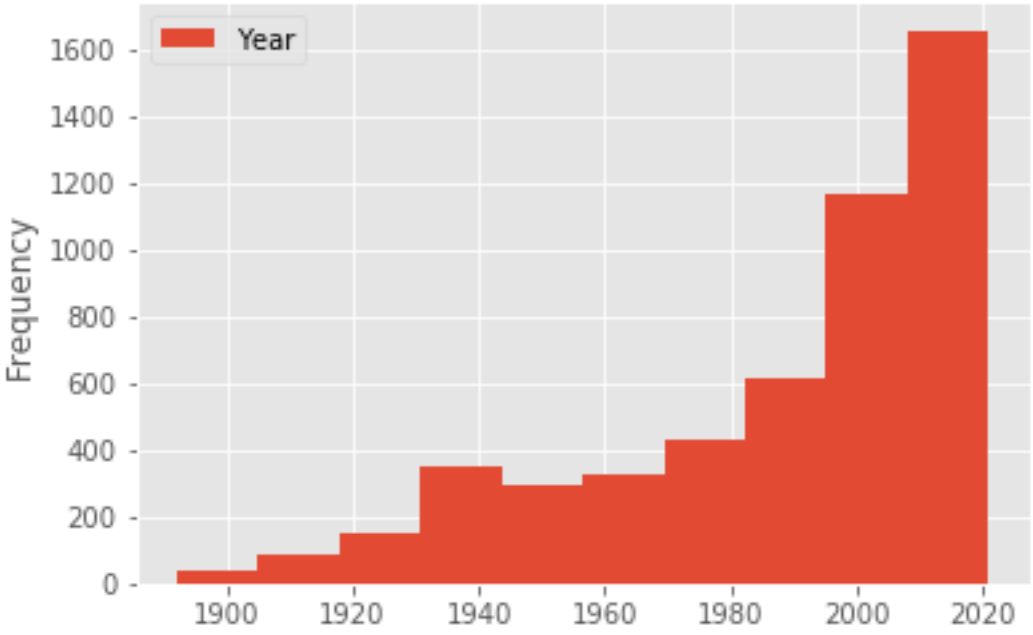

Figure 4: Histogram of the Release years

In the histogram Figure 6 you can clearly see that the amount of movies released each year increases. Especially the last 25 years there has been a huge increase in the number of movies released.

5.5 Does the budget increase as time passes

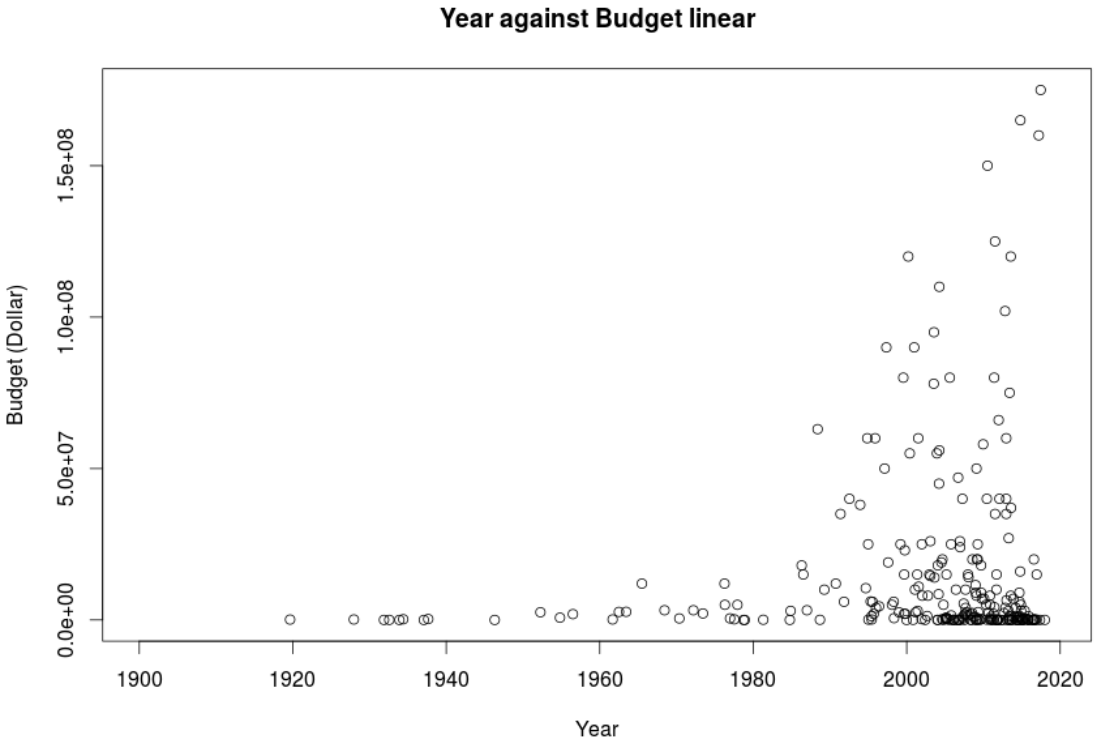

Figure 5: Linear scale

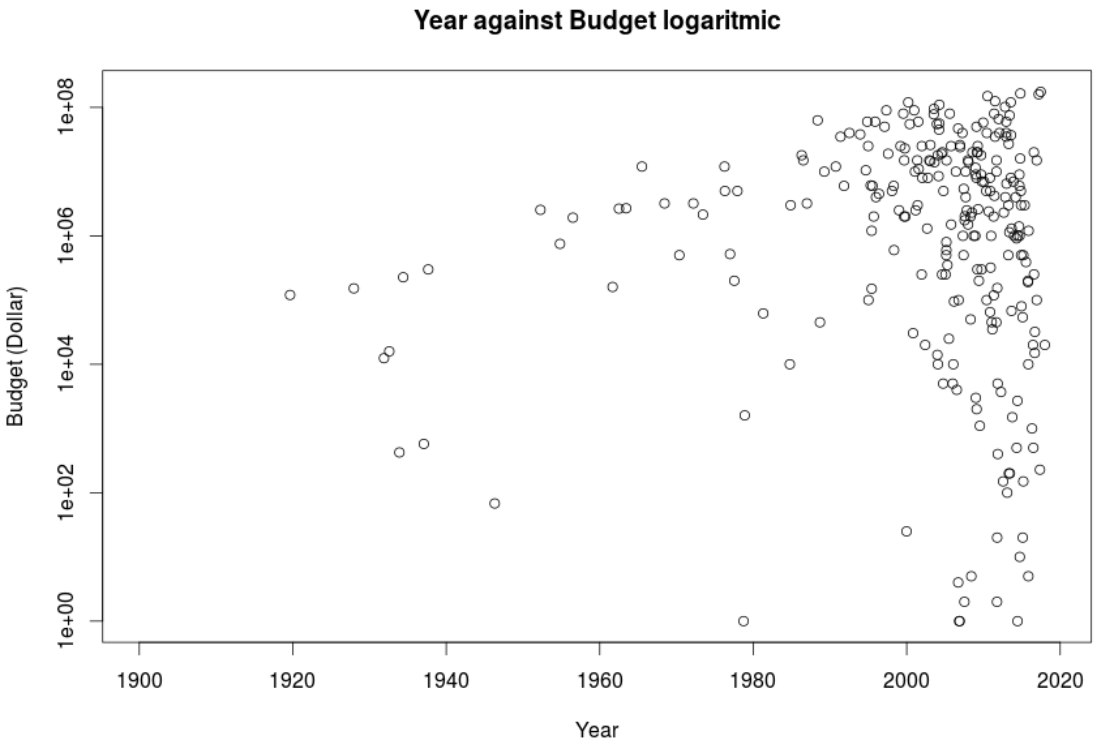

Figure 6: logarithmic scale

The plot shows that earlier movies are more expensive than later 'cheap' movies, but that the majority follows an increasing trend.

5.6 Most populair movies

The five most populair movies ordered using Popularity are:

1. **Beauty and the Beast** by Bill Condon
2. **Big Hero 6** by Don Hall, Chris Williams
3. **Harry Potter and the Deathly Hallows: Part 2** by David Yates
4. **The Fifth Element** by Luc Besson
5. **13 Year** by Emma Bloom, Esther Julis, Rachel Kaye, Olivia Knight