Information of Datasets (CSV files)

This file uses two CSV files downloaded from kaggle;
About the first file;

1. Page Title: Netflix Movies and TV Shows
2. File Name: netflix_titles.csv
3. Link for the file (https://www.kaggle.com/shivamb/netflix-showsv)

About the second file;

1. Page Title: Movies on Netflix, Prime Video, Hulu and Disney+
2. File Name: MoviesOnStreamingPlatforms_updated.csv
3. Link for the file (https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney)

Overview of Dataset

The dataset, Netflix Movies and TV Shows,has the data of shows on Netflix as of 2019. It includes the data listed below;

- show_id
- type
- title
- director
- cast
- country
- date_added
- release_year
- rating
- duration
- listed_in
- description

Because date_added has 11 blank entities, I removed the data from the dataset. The number of data in this dataset is 6223. release_year is converted from string to integer and date_added is converted from string to datetime.

The dataset, Movies on Netflix, Prime Video, Hulu and Disney+,has the data of ratings from IMDb and Rotten Tomatoes. The number of data in this dataset is 16744. This is used to obtain ratings of the movies on Netflix. Values of IMDb and Rotten Tomatoes are converetd from string to float and integer respectively.

The total number of movies having ratings of IMDb and Rotten Tomatoes is 1165 as a result.

Creating Datasets and Cleaning

In [1]:

```
#Import the csv file: Netflix Movies and TV Shows
import csv
file_path = "netflix_titles.csv/netflix_titles.csv"
file = open(file_path, encoding="utf-8")
all_lines = csv.reader(file, delimiter = ",")

#Set header for a dataset
dataset = []
header = next(all_lines)

import datetime

for line in all_lines:
    d = dict(zip(header, line))
    if d['release_year'] != '':
        d['release_year'] = int(d['release_year'])
    if d['duration'] != '':
        d['duration'] = int(d['duration'].split(" ")[0])
    if d['date_added'] != '':
        d['date_added'] = datetime.datetime.strptime(d["date_added"].strip(), "%B %d, %Y").date
()
        dataset.append(d) # Remove the data not having date_added

print("The number of data in Netflix Movies and TV Shows is", len(dataset))
```

```
The number of data in Netflix Movies and TV Shows is 6223
```

In [2]:

```
#Import the csv file: Movies on Netflix, Prime Video, Hulu and Disney+
file_path2 = "MoviesOnStreamingPlatforms_updated/MoviesOnStreamingPlatforms_updated.csv"
file2 = open(file_path2, encoding='utf-8')
all_lines = csv.reader(file2, delimiter = ",")

#Set header for a dataset
dataset_rating = []
header_rating = next(all_lines)
header_rating[0] = "count"

for line in all_lines:
    d = dict(zip(header_rating, line))
    if d['IMDb'] != '':
        d['IMDb'] = float(d['IMDb'])
    else:
        d['IMDb'] = float("0")
    if d['Rotten Tomatoes'] != '':
        d['Rotten Tomatoes'] = int(d['Rotten Tomatoes'].replace('%', ''))
    else:
        d['Rotten Tomatoes'] = 0
    dataset_rating.append(d)

print("The number of data in Movies on Netflix, Prime Video, Hulu and Disney+ is", len(dataset_r
ating))
```

```
The number of data in Movies on Netflix, Prime Video, Hulu and Disney+ is 16744
```

In [3]:

```
# Get ratings from IMDb and Rotten Tomatoes when they exist

for data in dataset:
    data["IMDb"] = 0.0
    data["Rotten Tomatoes"] = 0
    if data["type"] == "Movie":
        for tmp in dataset_rating:
            if data["title"] == tmp["Title"]:
                data["IMDb"] = tmp["IMDb"]
                data["Rotten Tomatoes"] = tmp["Rotten Tomatoes"]
```

In [4]:

```
# Define a new dataset of movies on Netflix having ratings of IMDb and Rotten Tomatoes
dataset_Netflix_Rating = []

for data in dataset:
    if data["Rotten Tomatoes"] != 0 and data["IMDb"] != 0.0:
        dic = {'title':data['title'], 'country':data['country'],'date_added':data['date_added'],
               'release_year':data['release_year'], 'rating':data['rating'], 'duration':data['du
ration'],
               'IMDb':data["IMDb"], 'Rotten Tomatoes':data["Rotten Tomatoes"]}
        dataset_Netflix_Rating.append(dic)

print("The number of movies on Netflix having ratings of IMDb and (or) Rotten Tomatoes is", len(
dataset_Netflix_Rating))
```

```
The number of movies on Netflix having ratings of IMDb and (or) Rotten Tomatoes is
1165
```

Simple Statistics

In [5]:

```
print("Total numbers of datasets are listed below:¥n",
     "The number of data in Netflix Movies and TV Shows is", len(dataset), "¥n",
     "The number of data in Movies on Netflix, Prime Video, Hulu and Disney+ is", len(dataset_ra
ting), "¥n",
     "The number of movies on Netflix having ratings of IMDb and (or) Rotten Tomatoes is", len(d
ataset_Netflix_Rating))
```

```
Total numbers of datasets are listed below:
 The number of data in Netflix Movies and TV Shows is 6223
 The number of data in Movies on Netflix, Prime Video, Hulu and Disney+ is 16744
 The number of movies on Netflix having ratings of IMDb and (or) Rotten Tomatoes i
s 1165
```

In [7]:

```
# the Number of release years with ratings
data_release_year_ratings = [d["release_year"] for d in dataset_Netflix_Rating]
data_release_year_ratings.sort(reverse = True)

from collections import defaultdict
release_year_cnt = defaultdict(int)

for year in data_release_year_ratings:
    release_year_cnt[year] += 1

print("The number of release years with ratings is", len(release_year_cnt.keys()))

import numpy as np
release_year_arr = np.array([d['release_year'] for d in dataset_Netflix_Rating])
print("Average of release years: ", np.average(release_year_arr))
```

```
The number of release years with ratings is 47
Average of release years:  2013.462660944206
```

In [8]:

```
#Average of IMDb / Rotten Tomatoes ratings on Netflix
IMDb_ratings = np.array([d['IMDb'] for d in dataset_Netflix_Rating])
Tomatoes_ratings = np.array([d['Rotten Tomatoes'] for d in dataset_Netflix_Rating])

print("Average of IMDb ratings: ", np.average(IMDb_ratings))
print("Average of Rotten Tomatoes ratings (%): ", np.average(Tomatoes_ratings))
```

```
Average of IMDb ratings:  6.43244635193133
Average of Rotten Tomatoes ratings (%):  66.8283261802575
```

In [9]:

```
#Max and Min values of IMDb / Rotten Tomatoes ratings on Netflix

print("Maximum value of IMDb ratings: ", np.amax(IMDb_ratings))
print("Minimum value of IMDb ratings: ", np.amin(IMDb_ratings))
print("Maximum value of Rotten Tomatoes ratings (%): ", np.amax(Tomatoes_ratings))
print("Minimum value of Rotten Tomatoes ratings: ", np.amin(Tomatoes_ratings))
```

```
Maximum value of IMDb ratings:  8.8
Minimum value of IMDb ratings:  2.5
Maximum value of Rotten Tomatoes ratings (%):  100
Minimum value of Rotten Tomatoes ratings:  3
```

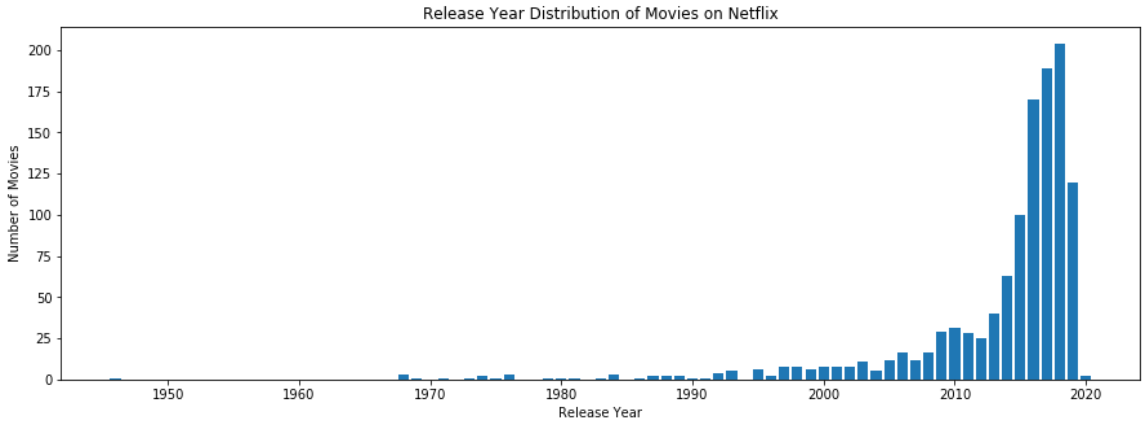
Data Visualization

In [10]:

```
import matplotlib.pyplot as plt
from matplotlib import colors
```

In [11]:

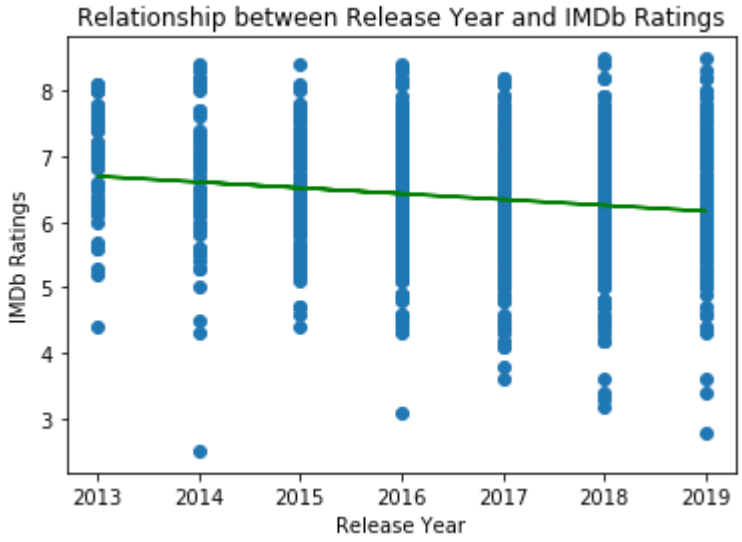
```
# Histogram to observe # of data per released year
X = list(release_year_cnt.keys())
Y = list(release_year_cnt.values())

plt.figure(figsize=(15, 5))
plt.gca().set(xlabel='Release Year', ylabel='Number of Movies',
       title='Release Year Distribution of Movies on Netflix')
plt.bar(X, Y)
plt.show()
```

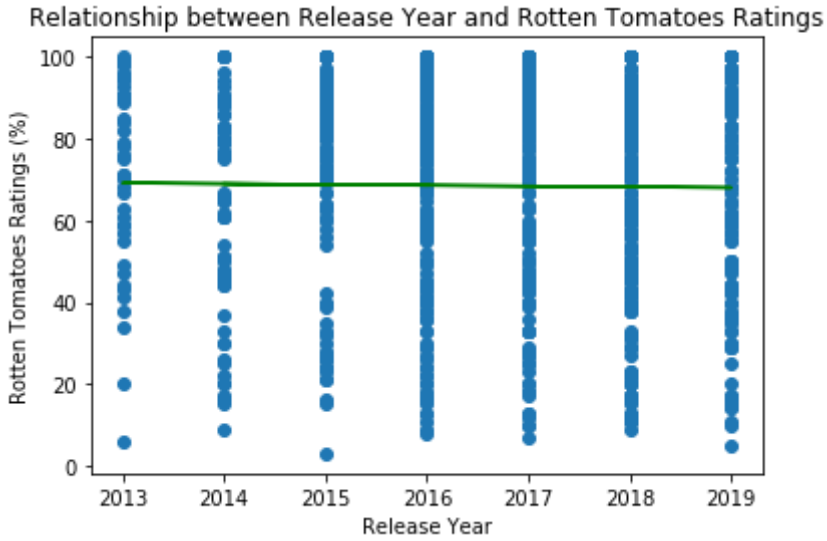


In [12]:

```
print("Because the average of release years is about 2013 and # of data in 2020 is very small,¥n
visualizations below focus on the release year from 2013 to 2019.")
# Scatter plot of IMDb ratings and Rotten Tamatoes ratings per release years
release_year_target = [2013, 2014, 2015, 2016, 2017, 2018, 2019]
year_target = [d['release_year'] for d in dataset_Netflix_Rating if d['release_year'] in releas
e_year_target]
ratings_IMDb_target = [d['IMDb'] for d in dataset_Netflix_Rating if d['release_year'] in releas
e_year_target]
ratings_Tomatoes_target = [d['Rotten Tomatoes'] for d in dataset_Netflix_Rating if d['release_y
ear'] in release_year_target]

# IMDb ratings
x = np.array(year_target)
y = np.array(ratings_IMDb_target)

plt.scatter(x,y)
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x + b, color='g')
plt.gca().set(xlabel='Release Year', ylabel='IMDb Ratings',
       title='Relationship between Release Year and IMDb Ratings')
plt.show()

# Rotten Tomatoes ratings
y = np.array(ratings_Tomatoes_target)

plt.scatter(x,y)
m, b = np.polyfit(x, y, 1)
plt.plot(x, m*x + b, color='g')
plt.gca().set(xlabel='Release Year', ylabel='Rotten Tomatoes Ratings (%)',
       title='Relationship between Release Year and Rotten Tomatoes Ratings')
plt.show()
```

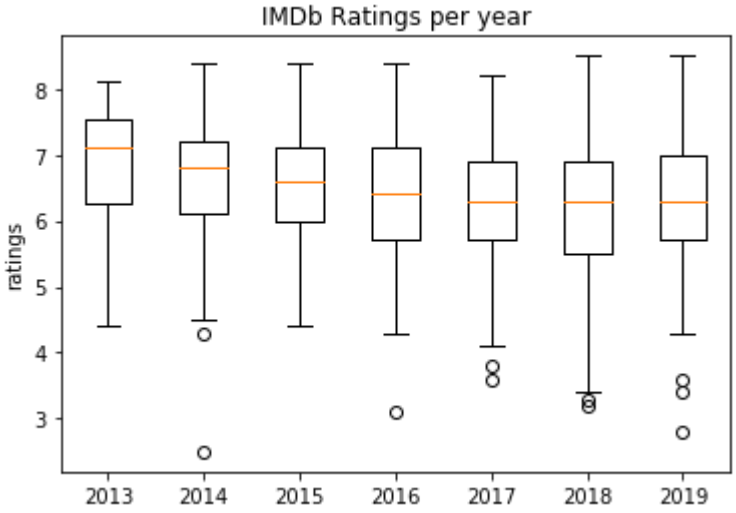
Because the average of release years is about 2013 and # of data in 2020 is very s
mall,
visualizations below focus on the release year from 2013 to 2019.

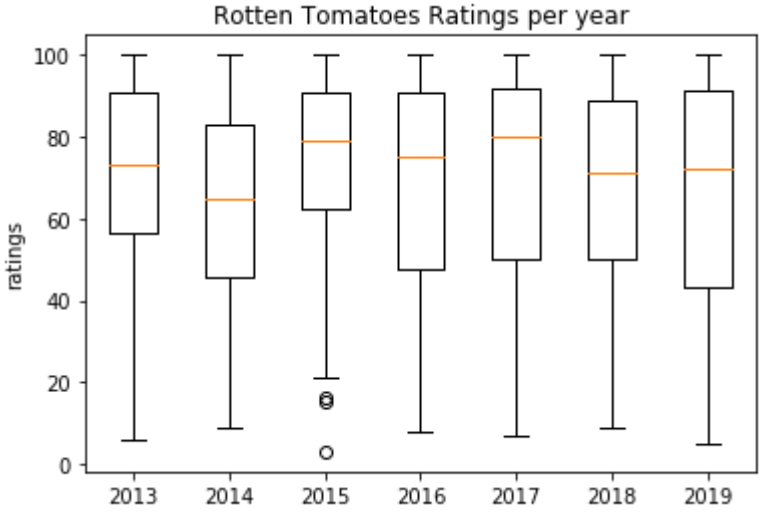




In [13]:

```
# box plot from 2015 to 2019 per ratings of IMDb and Rotten Tomatoes
#IMDb Ratings
IMDb_ratings_dic = defaultdict(list)
for d in dataset_Netflix_Rating:
    IMDb_ratings_dic[d["release_year"]].append(d["IMDb"])

import collections
od_IMDb_ratings_dic = collections.OrderedDict(sorted(IMDb_ratings_dic.items()))

val_IMDb_list = []
for year in release_year_target:
    val_IMDb_list.append(od_IMDb_ratings_dic[year])

plt.boxplot(val_IMDb_list, labels=["2013", "2014", "2015", "2016", "2017", "2018", "2019"])
plt.gca().set(title="IMDb Ratings per year", ylabel="ratings")
plt.show()

#Rotten Tomatoes Ratings
Tomatoes_ratings_dic = defaultdict(list)
for d in dataset_Netflix_Rating:
    Tomatoes_ratings_dic[d["release_year"]].append(d["Rotten Tomatoes"])

od_Tomatoes_ratings_dic = collections.OrderedDict(sorted(Tomatoes_ratings_dic.items()))

val_Tomatoes_list = []
for year in release_year_target:
    val_Tomatoes_list.append(od_Tomatoes_ratings_dic[year])

plt.boxplot(val_Tomatoes_list, labels=["2013", "2014", "2015", "2016", "2017", "2018", "2019"])
plt.gca().set(title="Rotten Tomatoes Ratings per year", ylabel="ratings")
plt.show()
```


IMDb Ratings per year
ratings
8
7
6
5
4
3
2013
2014
2015
2016
2017
2018
2019


Rotten Tomatoes Ratings per year
ratings
100
80
60
40
20
0
2013
2014
2015
2016
2017
2018
2019