NETFLIX

GUIDED BY,
PROF.SHUBHANGI KALE, COURSE CHAMPION

ARYAN MESHRAM-202201050010
SHREYA BORLE-202201050006
SNEHAL CHAVAN-202201040022

INTRODUCTION

Data analytics is the process of examining vast volumes of data to extract meaningful patterns, trends, and correlations. In this dataset of netflix, data analysis becomes a window through which we can do various analysis such as data manipulation, data visualization ,etc.

By this process we will understand the mass amount of users of netflix and there preferences , the most trending series movies and their quality of content and reviews regarding the same, etc within the same dataset.

MOTIVATION

Netflix is the most trending platform now a days to watch the series and the movies which have gained much popularity in the youth from 2020 containing a total of 5.5 million paying members and a accumulated over a total of 25 million users . As consists of such mass population the quality content by the producers and the reviews of the critics, user is the main part of the dataset. Also it gives a chance to predict the preferences of the people on large scale

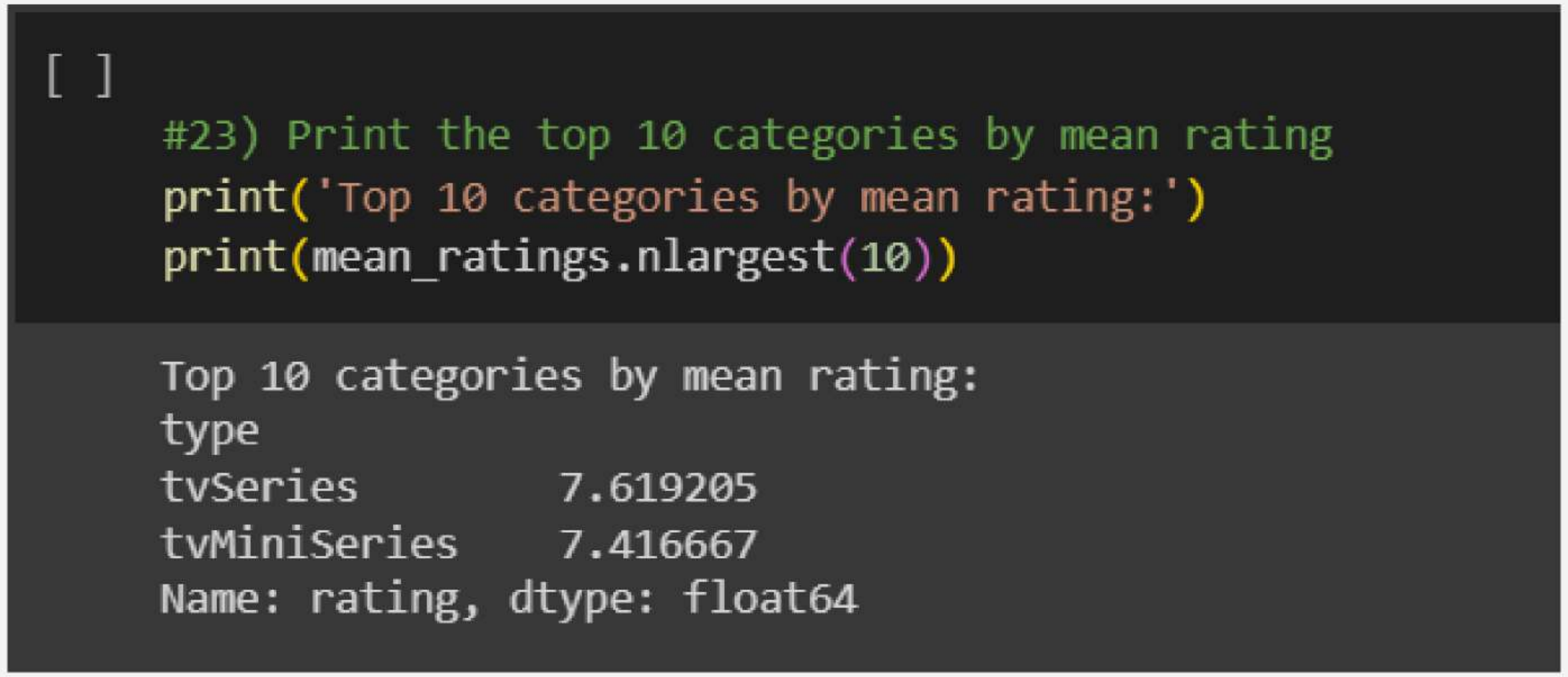
DETAILS OF DATASET

Name of Dataset:-Netflix
Number of Features:-19
Number of Records:-6650

Data Manipulation

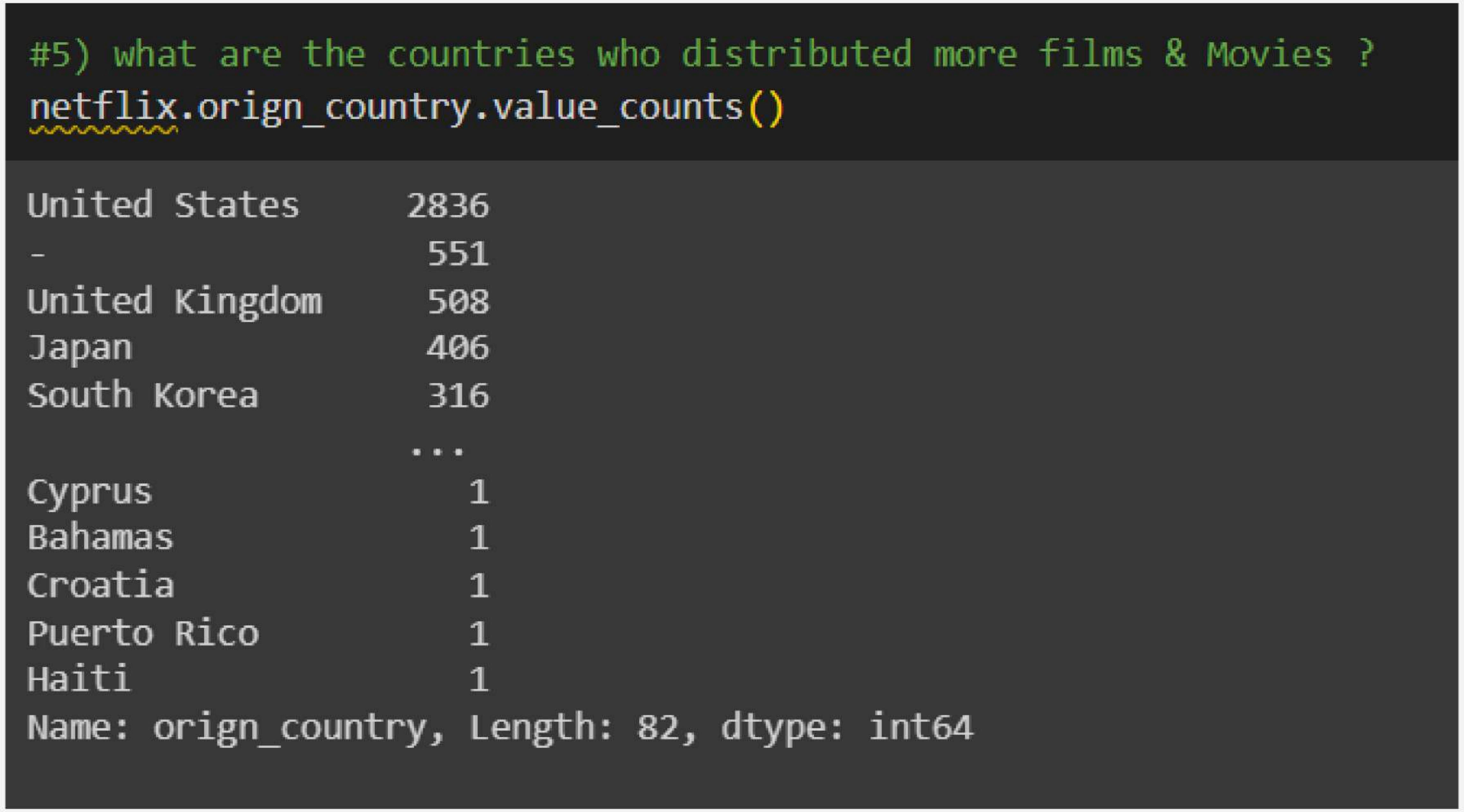
Data manipulation refers to the process of modifying, transforming, or reorganizing data to extract meaningful insights or prepare it for further analysis. It involves various operations performed on the data, such as filtering, sorting, aggregating, merging, and reshaping, among others.

Data manipulation is an essential step in the data analysis workflow as it helps to clean, preprocess, and transform raw data into a format that is suitable for analysis or visualization. It allows data scientists and analysts to extract valuable information, discover patterns, and derive meaningful insights from the data.



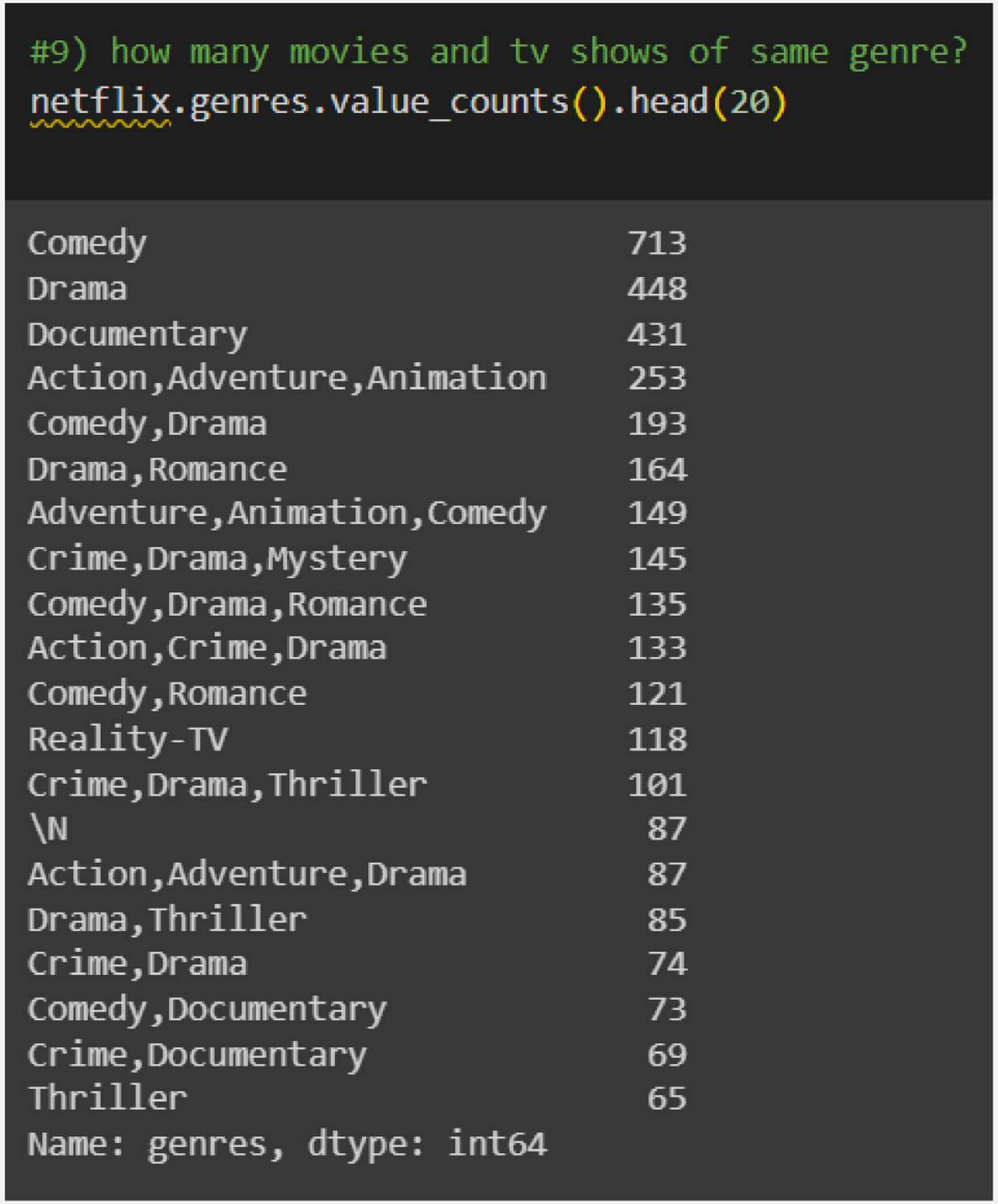
```
[ ]
    #23) Print the top 10 categories by mean rating
    print('Top 10 categories by mean rating:')
    print(mean_ratings.nlargest(10))
```

```
Top 10 categories by mean rating:
type
tvSeries        7.619205
tvMiniSeries    7.416667
Name: rating, dtype: float64
```



```
#5) what are the countries who distributed more films & Movies ?
netflix.orign_country.value_counts()
```

```
United States     2836
-                  551
United Kingdom     508
Japan              406
South Korea        316
                  ...
Cyprus               1
Bahamas              1
Croatia              1
Puerto Rico          1
Haiti                1
Name: orign_country, Length: 82, dtype: int64
```



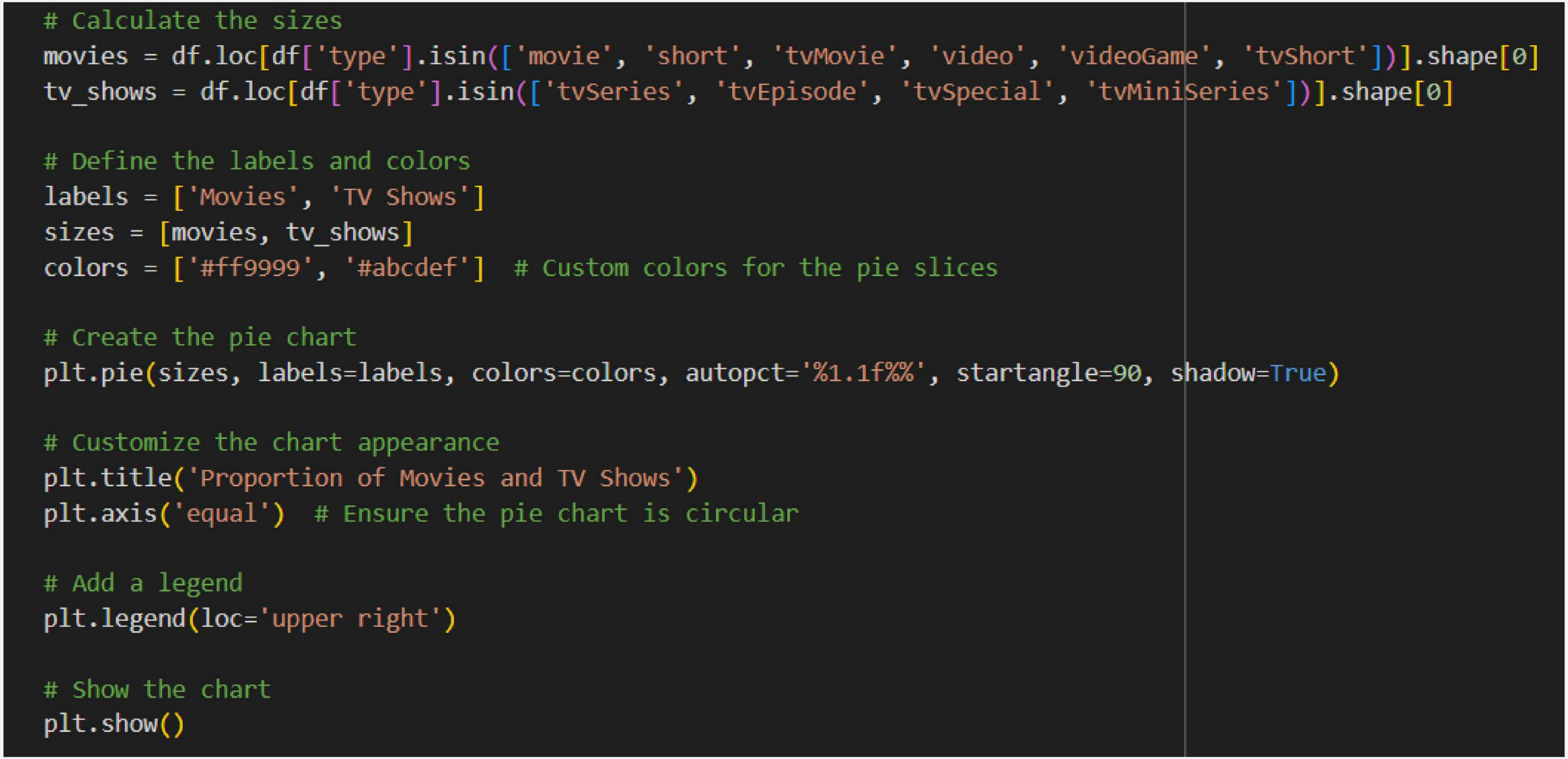
```
#9) how many movies and tv shows of same genre?
netflix.genres.value_counts().head(20)
```

```
Comedy                          713
Drama                           448
Documentary                     431
Action,Adventure,Animation      253
Comedy,Drama                    193
Drama,Romance                   164
Adventure,Animation,Comedy      149
Crime,Drama,Mystery             145
Comedy,Drama,Romance            135
Action,Crime,Drama              133
Comedy,Romance                  121
Reality-TV                      118
Crime,Drama,Thriller            101
\N                               87
Action,Adventure,Drama           87
Drama,Thriller                   85
Crime,Drama                      74
Comedy,Documentary               73
Crime,Documentary                69
Thriller                         65
Name: genres, dtype: int64
```

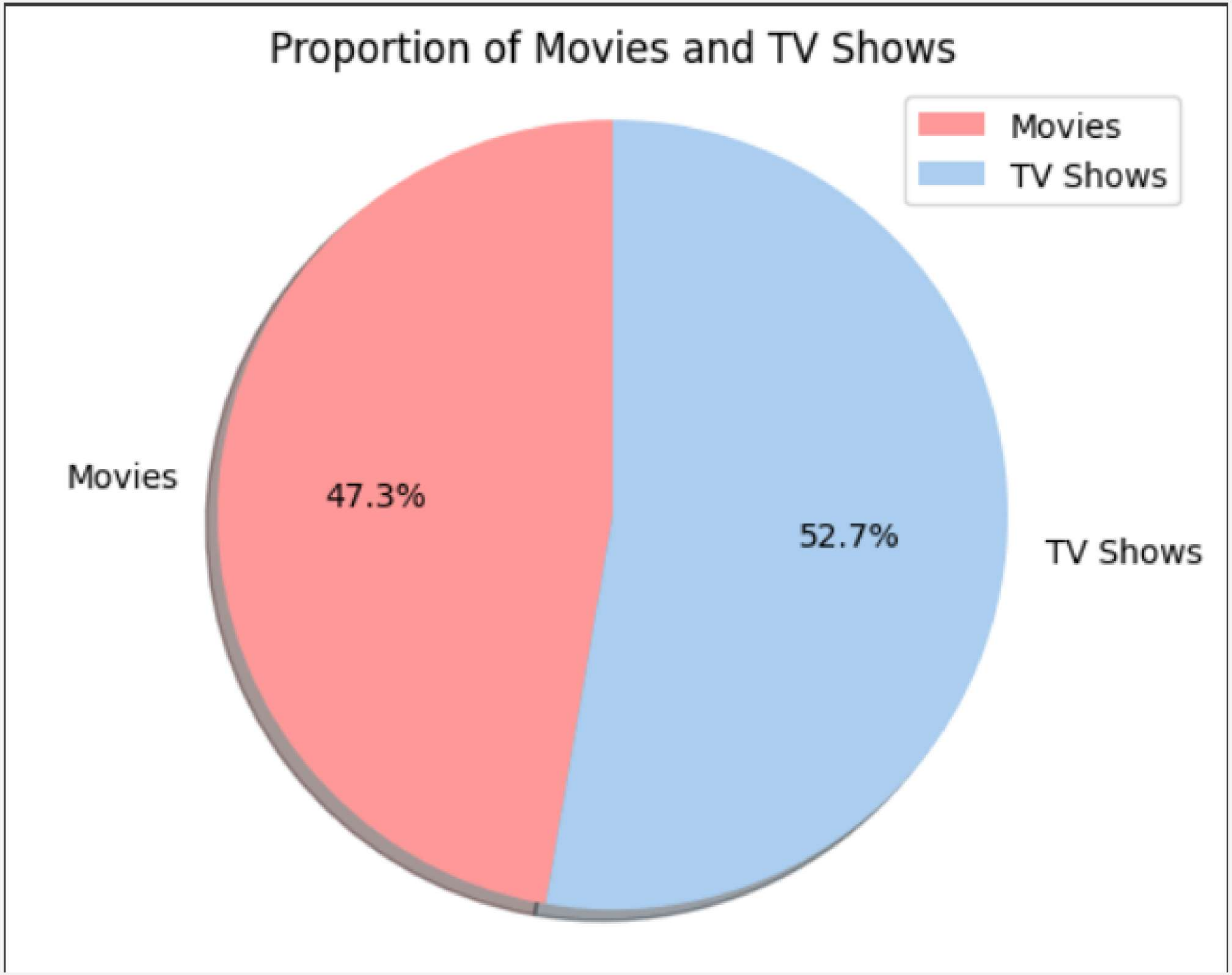
Data Visualization

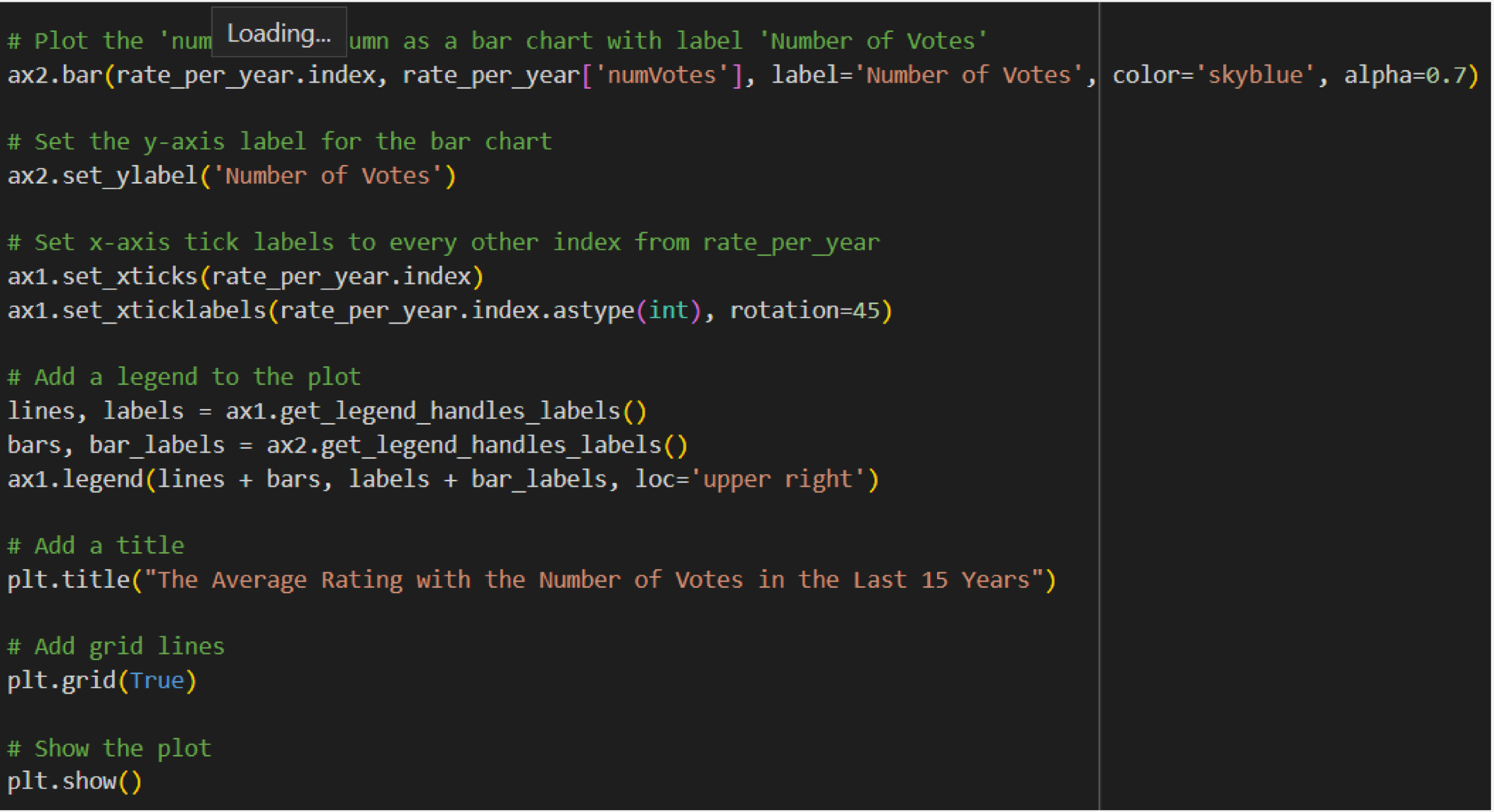
Data visualization refers to the graphical representation of data using visual elements such as charts, graphs, and plots. It is a powerful tool for exploring, analyzing, and communicating data patterns, trends, and insights.

In Python, there are several libraries available for data visualization, with Matplotlib being one of the most popular and widely used. Matplotlib provides a flexible and comprehensive set of functions and methods for creating various types of visualizations.

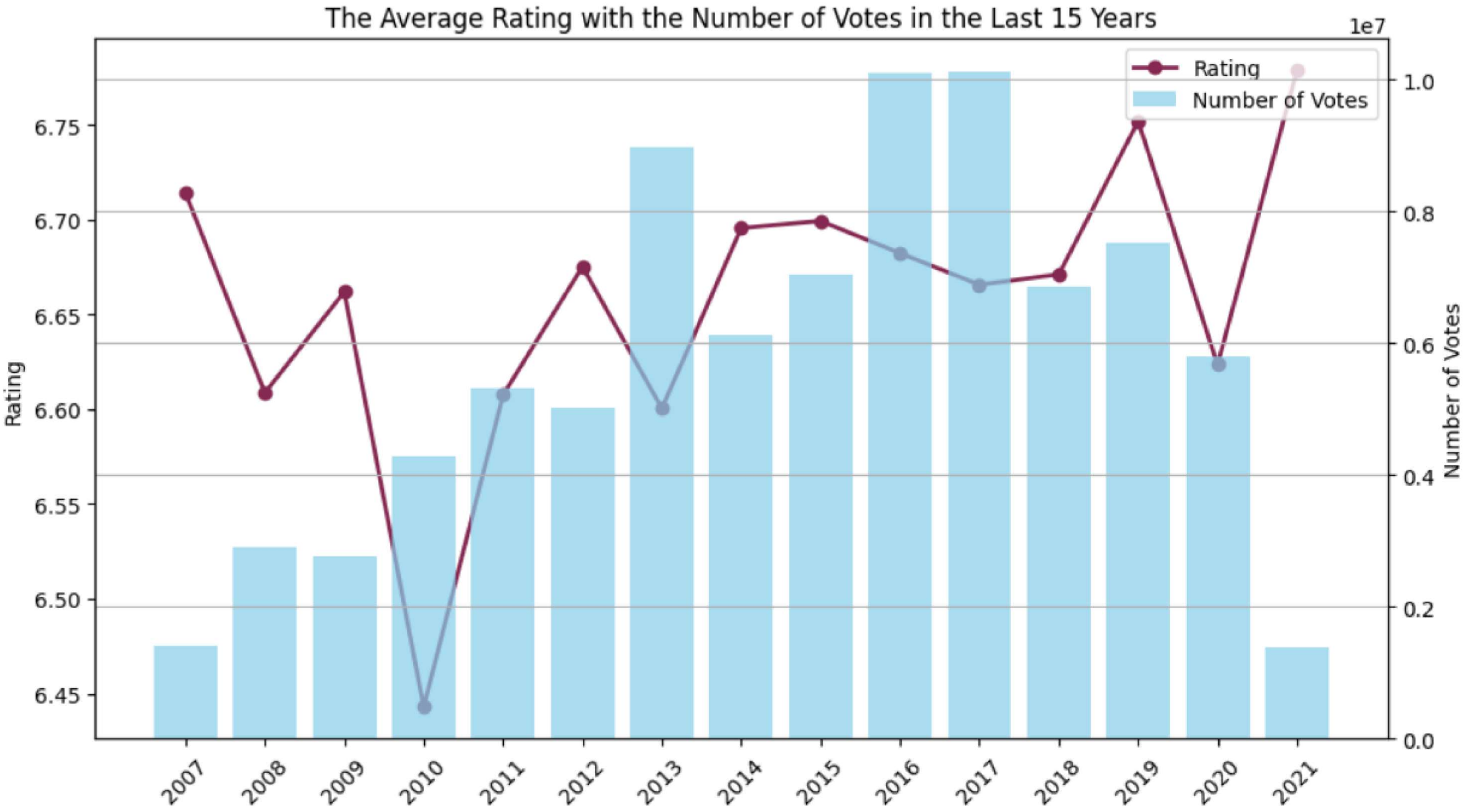


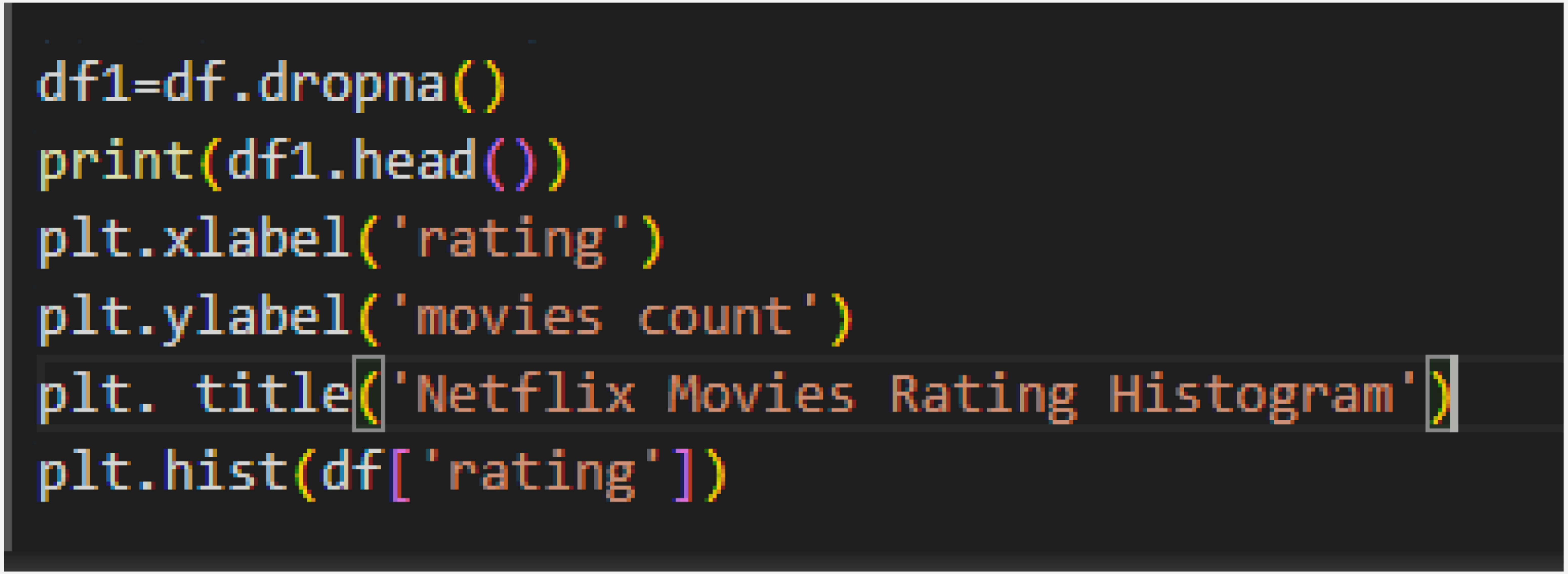
```python
# Calculate the sizes
movies = df.loc[df['type'].isin(['movie', 'short', 'tvMovie', 'video', 'videoGame', 'tvShort'])].shape[0]
tv_shows = df.loc[df['type'].isin(['tvSeries', 'tvEpisode', 'tvSpecial', 'tvMiniSeries'])].shape[0]

# Define the labels and colors
labels = ['Movies', 'TV Shows']
sizes = [movies, tv_shows]
colors = ['#ff9999', '#abcdef']  # Custom colors for the pie slices

# Create the pie chart
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90, shadow=True)

# Customize the chart appearance
plt.title('Proportion of Movies and TV Shows')
plt.axis('equal')  # Ensure the pie chart is circular

# Add a legend
plt.legend(loc='upper right')

# Show the chart
plt.show()
```


Proportion of Movies and TV Shows
🟥 Movies
🟦 TV Shows
Movies
47.3%
52.7%
TV Shows

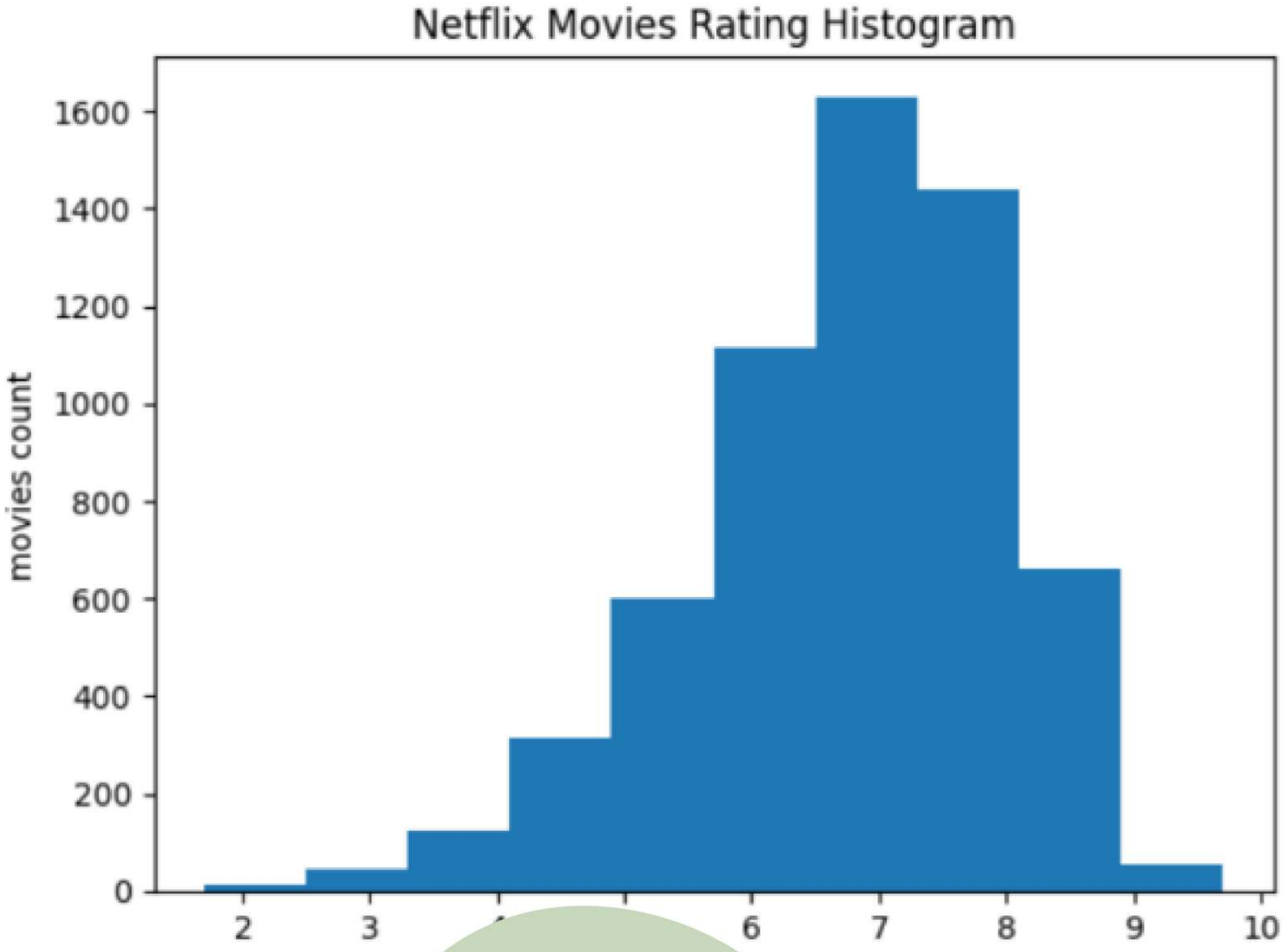


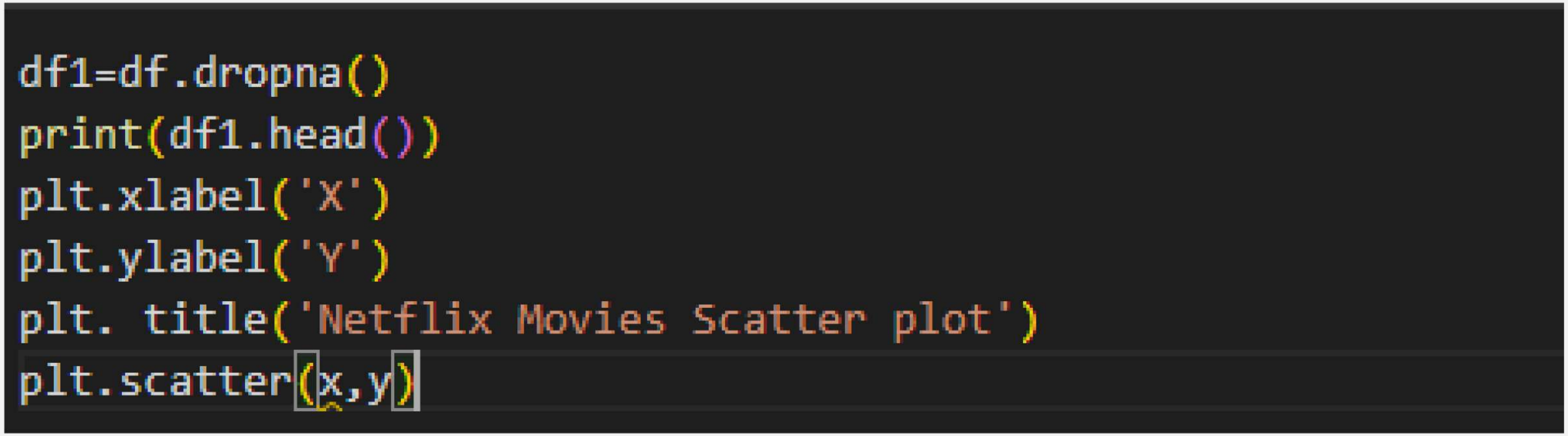
```
# Plot the 'num          umn as a bar chart with label 'Number of Votes'
ax2.bar(rate_per_year.index, rate_per_year['numVotes'], label='Number of Votes', color='skyblue', alpha=0.7)

# Set the y-axis label for the bar chart
ax2.set_ylabel('Number of Votes')

# Set x-axis tick labels to every other index from rate_per_year
ax1.set_xticks(rate_per_year.index)
ax1.set_xticklabels(rate_per_year.index.astype(int), rotation=45)

# Add a legend to the plot
lines, labels = ax1.get_legend_handles_labels()
bars, bar_labels = ax2.get_legend_handles_labels()
ax1.legend(lines + bars, labels + bar_labels, loc='upper right')

# Add a title
plt.title("The Average Rating with the Number of Votes in the Last 15 Years")

# Add grid lines
plt.grid(True)

# Show the plot
plt.show()
```
]


The Average Rating with the Number of Votes in the Last 15 Years
Rating
Number of Votes
Rating
Number of Votes
1e7
6.45
6.50
6.55
6.60
6.65
6.70
6.75
0.0
0.2
0.4
0.6
0.8
1.0
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

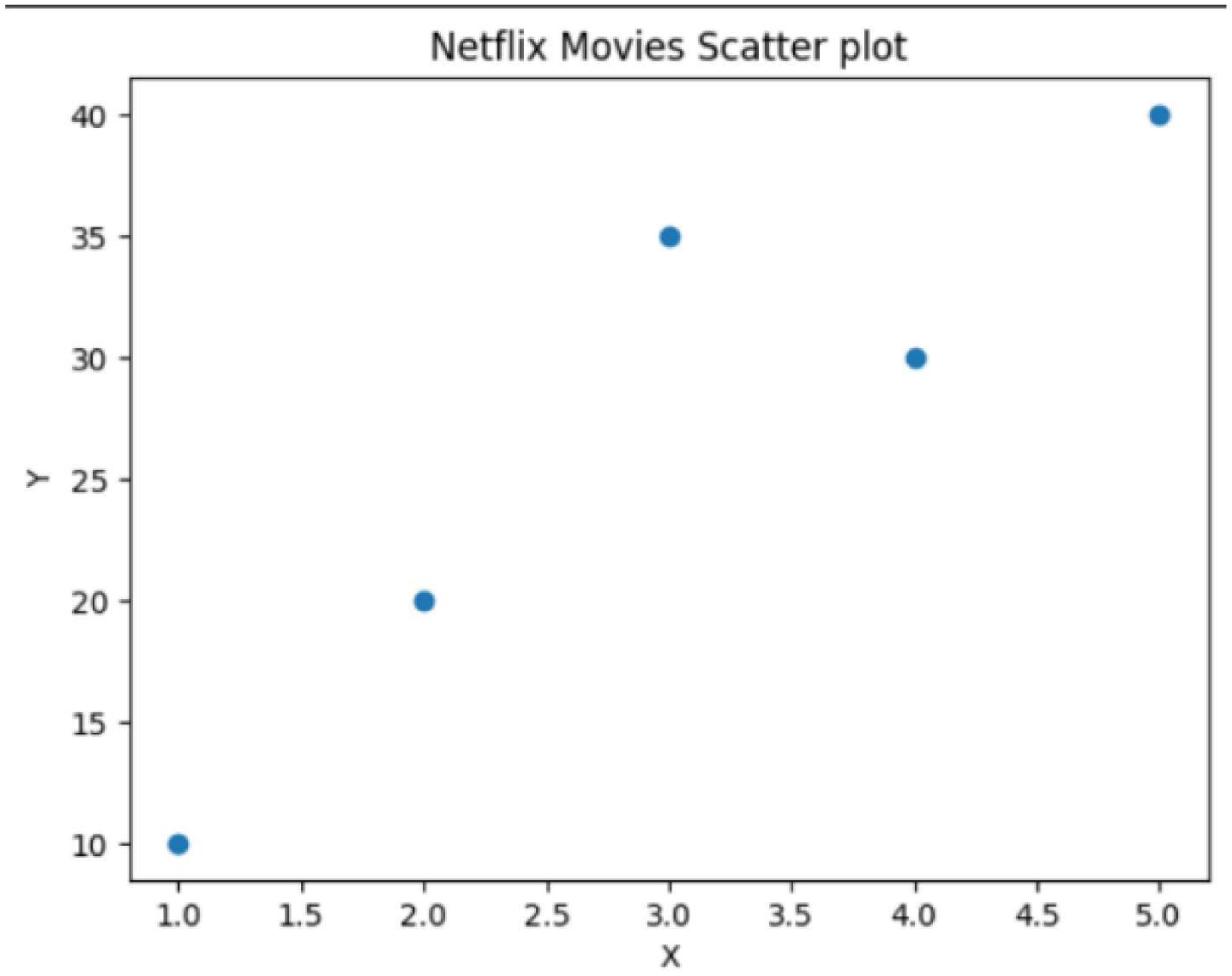


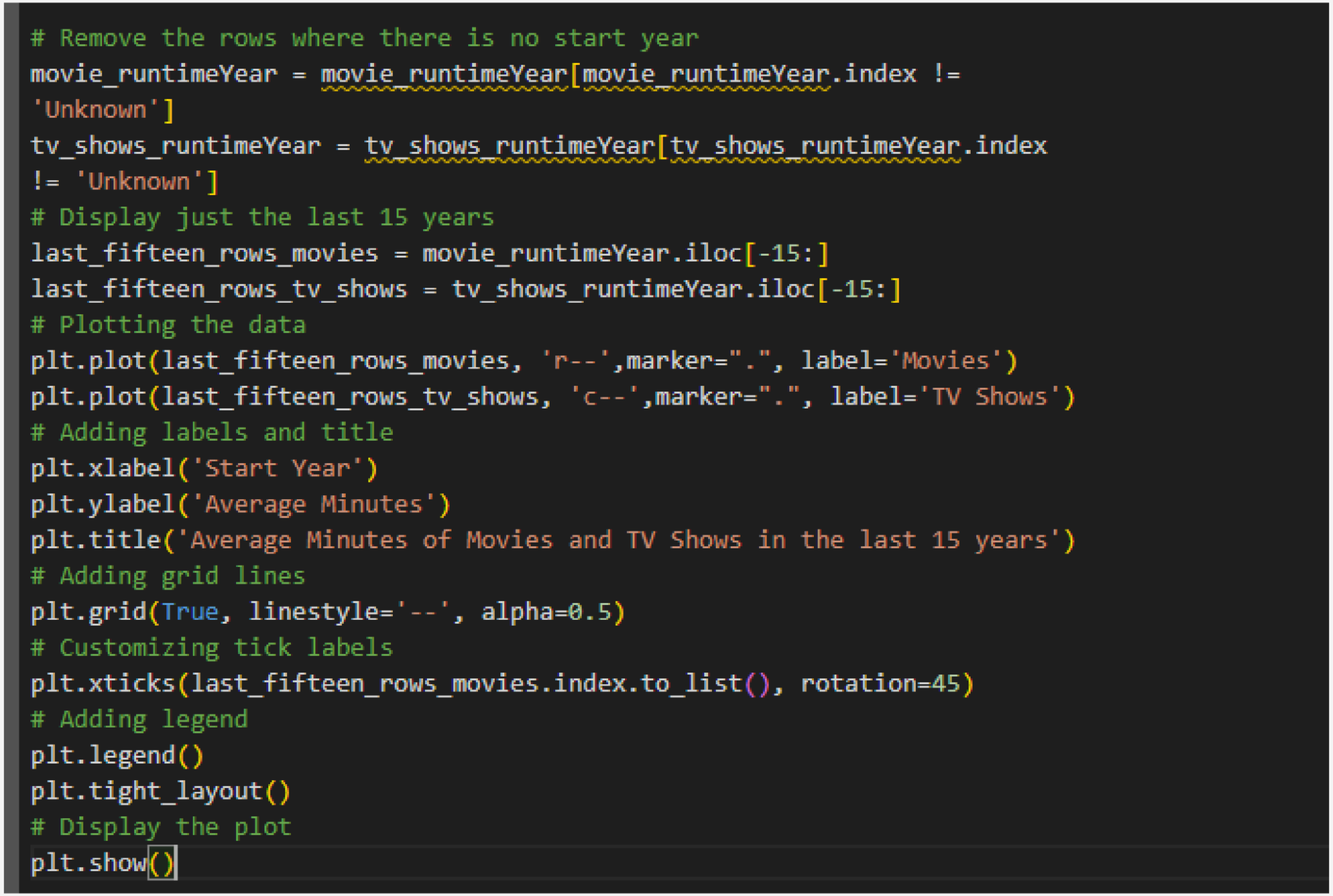
```
df1=df.dropna()
print(df1.head())
plt.xlabel('rating')
plt.ylabel('movies count')
plt. title('Netflix Movies Rating Histogram')
plt.hist(df['rating'])
```


Netflix Movies Rating Histogram
movies count
1600
1400
1200
1000
800
600
400
200
0
2
3
6
7
8
9
10

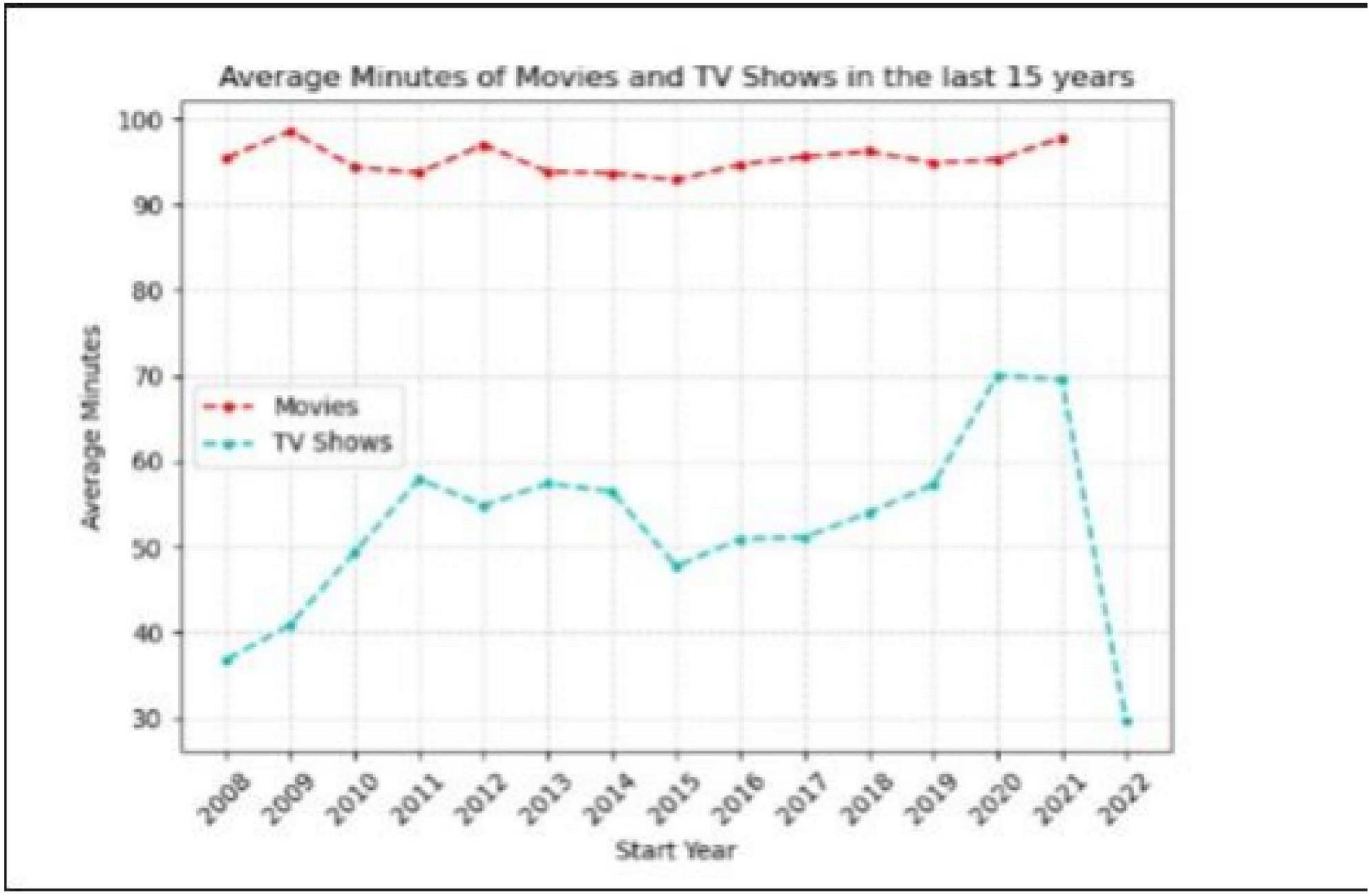


```
df1=df.dropna()
print(df1.head())
plt.xlabel('X')
plt.ylabel('Y')
plt. title('Netflix Movies Scatter plot')
plt.scatter(x,y)
```

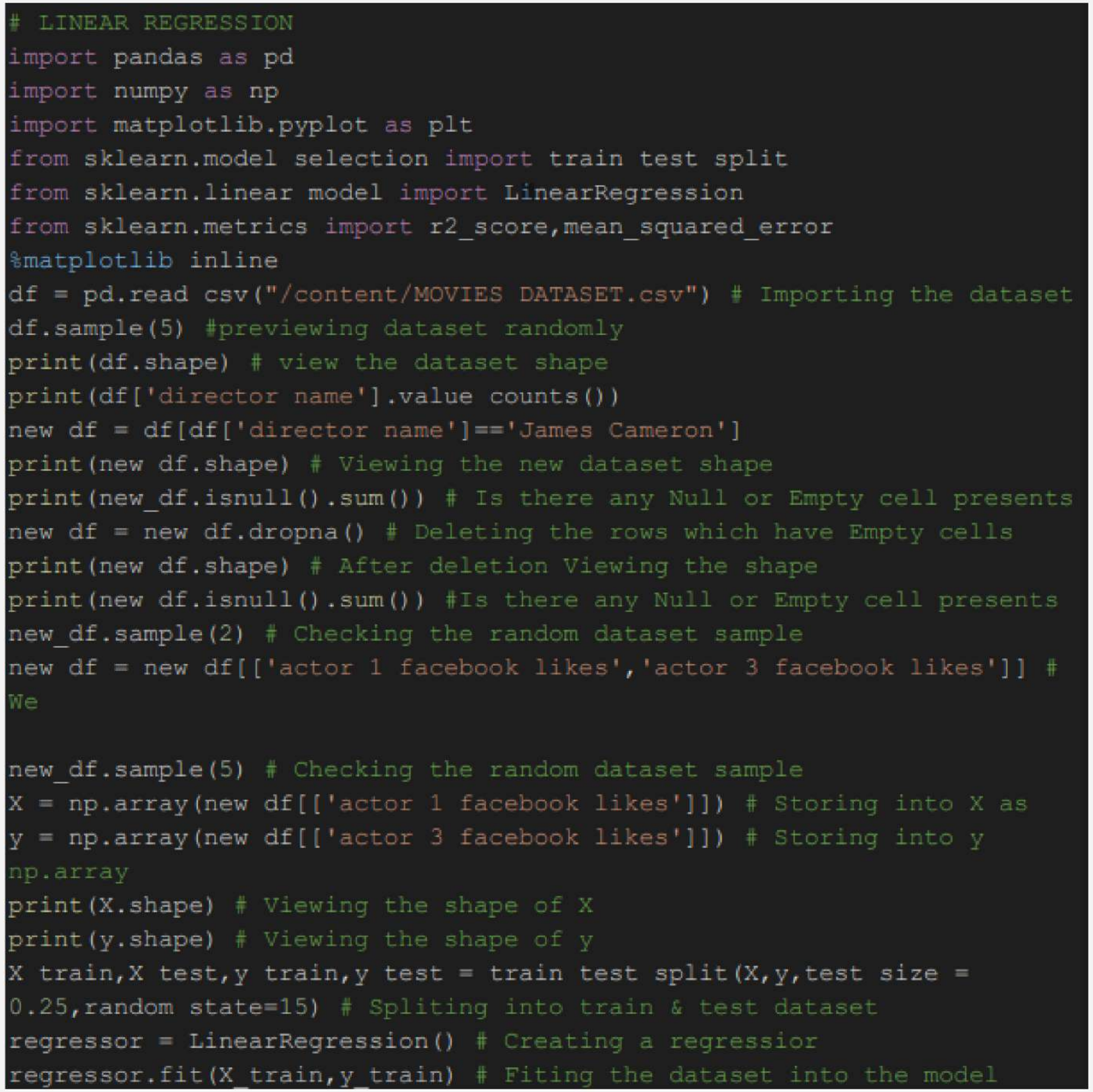

Netflix Movies Scatter plot
40
35
30
25
20
15
10
Y
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
X



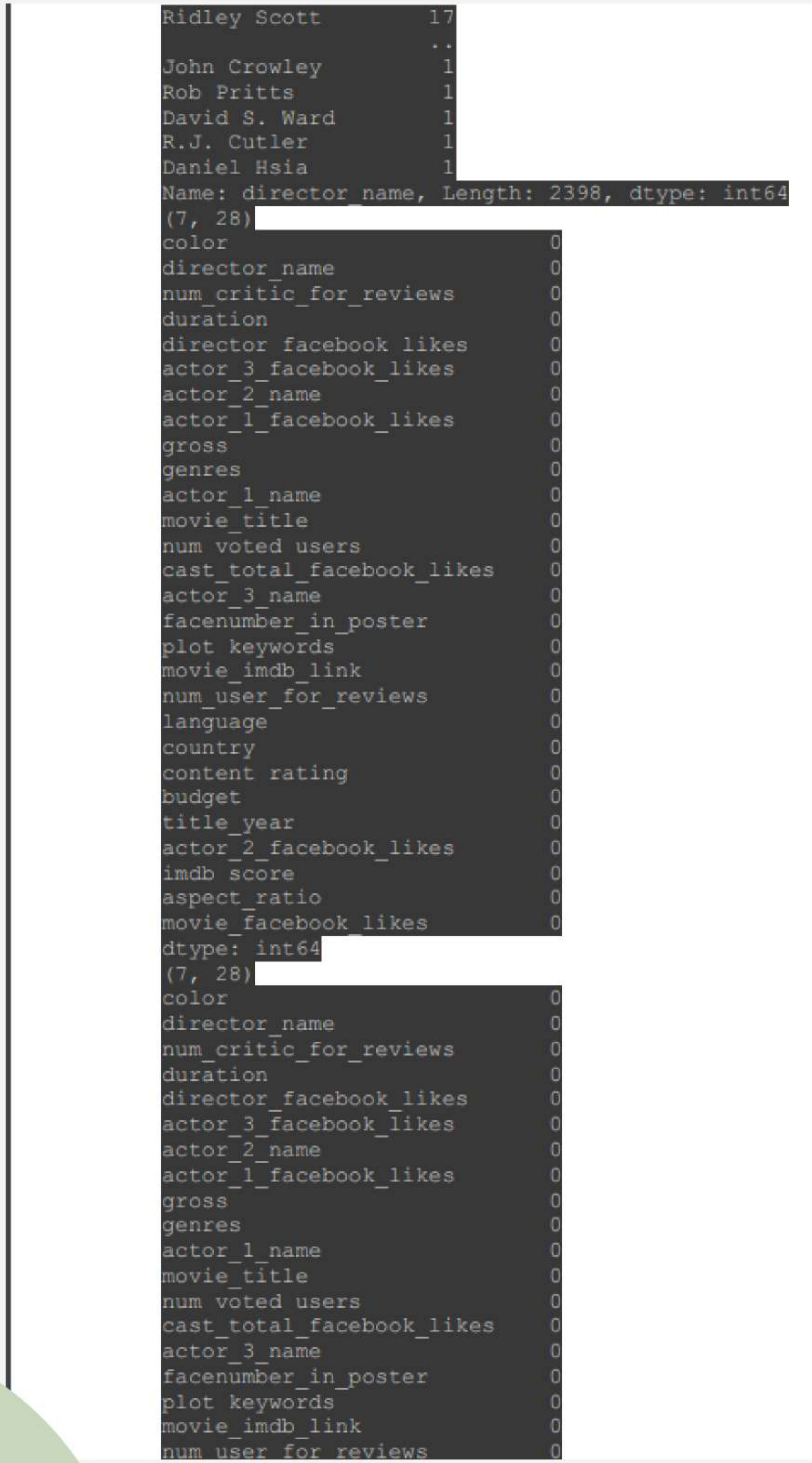
```
# Remove the rows where there is no start year
movie_runtimeYear = movie_runtimeYear[movie_runtimeYear.index !=
'Unknown']
tv_shows_runtimeYear = tv_shows_runtimeYear[tv_shows_runtimeYear.index
!= 'Unknown']
# Display just the last 15 years
last_fifteen_rows_movies = movie_runtimeYear.iloc[-15:]
last_fifteen_rows_tv_shows = tv_shows_runtimeYear.iloc[-15:]
# Plotting the data
plt.plot(last_fifteen_rows_movies, 'r--',marker=".", label='Movies')
plt.plot(last_fifteen_rows_tv_shows, 'c--',marker=".", label='TV Shows')
# Adding labels and title
plt.xlabel('Start Year')
plt.ylabel('Average Minutes')
plt.title('Average Minutes of Movies and TV Shows in the last 15 years')
# Adding grid lines
plt.grid(True, linestyle='--', alpha=0.5)
# Customizing tick labels
plt.xticks(last_fifteen_rows_movies.index.to_list(), rotation=45)
# Adding legend
plt.legend()
plt.tight_layout()
# Display the plot
plt.show()
```



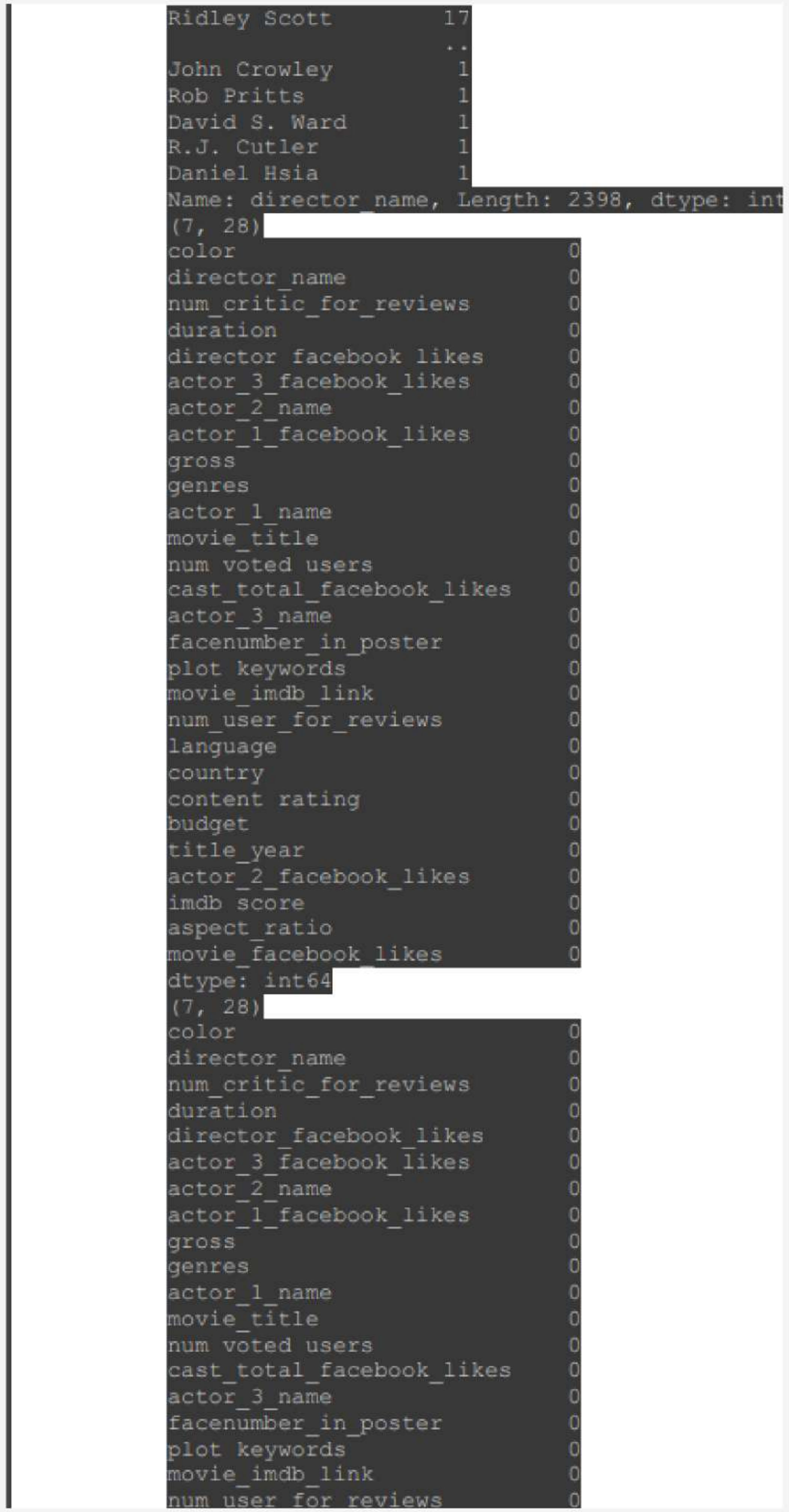
Average Minutes of Movies and TV Shows in the last 15 years
Average Minutes
100
90
80
70
60
50
40
30
Movies
TV Shows
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Start Year



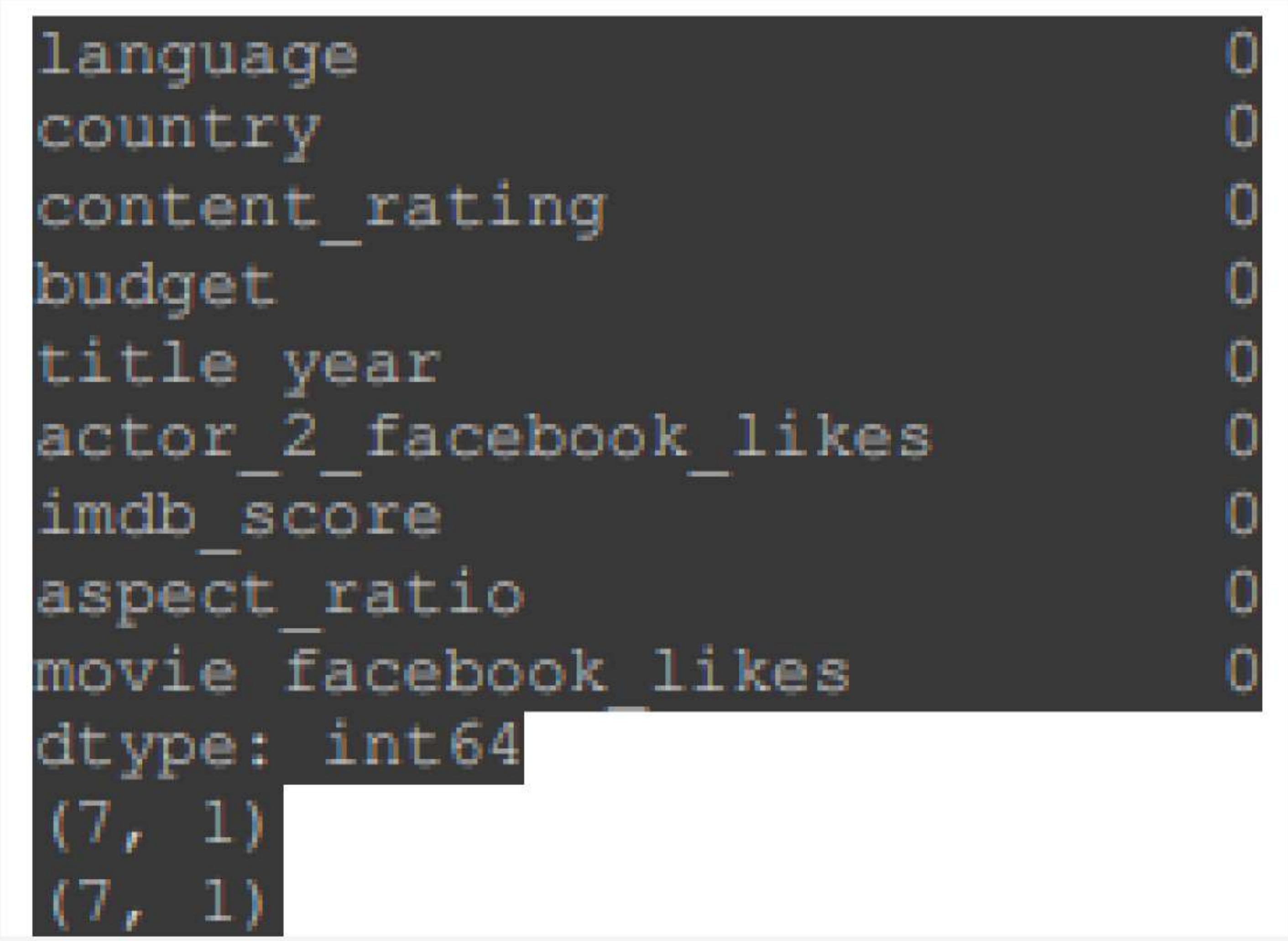
```
# LINEAR REGRESSION
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model selection import train test split
from sklearn.linear model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error
%matplotlib inline
df = pd.read csv("/content/MOVIES DATASET.csv") # Importing the dataset
df.sample(5) #previewing dataset randomly
print(df.shape) # view the dataset shape
print(df['director name'].value counts())
new df = df[df['director name']=='James Cameron']
print(new df.shape) # Viewing the new dataset shape
print(new_df.isnull().sum()) # Is there any Null or Empty cell presents
new df = new df.dropna() # Deleting the rows which have Empty cells
print(new df.shape) # After deletion Viewing the shape
print(new df.isnull().sum()) #Is there any Null or Empty cell presents
new_df.sample(2) # Checking the random dataset sample
new df = new df[['actor 1 facebook likes','actor 3 facebook likes']] #
We

new_df.sample(5) # Checking the random dataset sample
X = np.array(new df[['actor 1 facebook likes']]) # Storing into X as
y = np.array(new df[['actor 3 facebook likes']]) # Storing into y
np.array
print(X.shape) # Viewing the shape of X
print(y.shape) # Viewing the shape of y
X train,X test,y train,y test = train test split(X,y,test size =
0.25,random state=15) # Spliting into train & test dataset
regressor = LinearRegression() # Creating a regressior
regressor.fit(X_train,y_train) # Fiting the dataset into the model
```



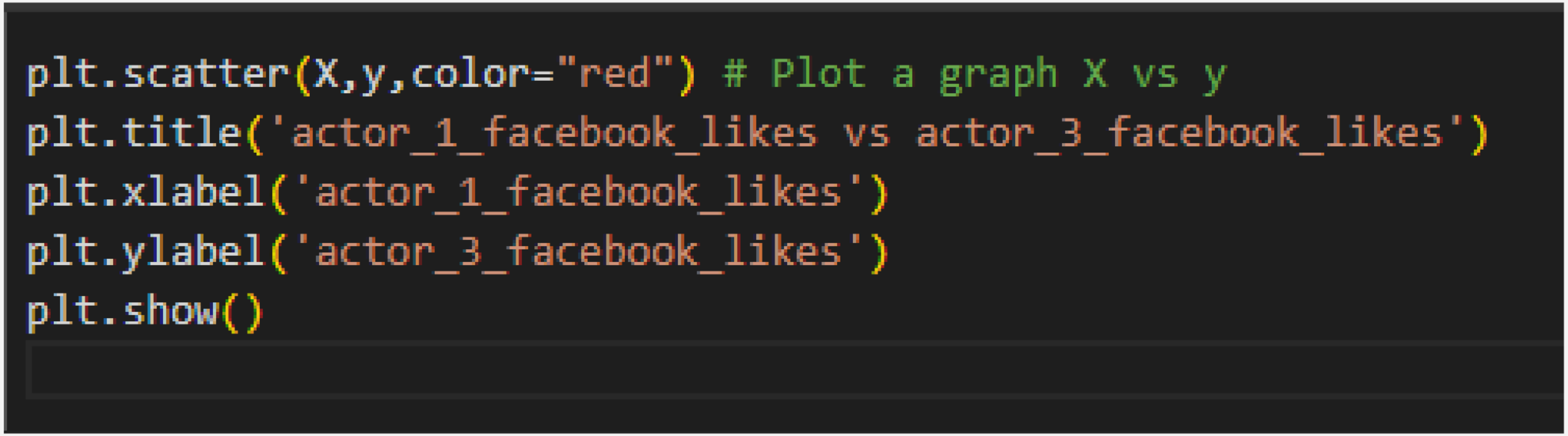
```
Ridley Scott        17
                    ..
John Crowley         1
Rob Pritts           1
David S. Ward        1
R.J. Cutler          1
Daniel Hsia          1
Name: director_name, Length: 2398, dtype: int64
(7, 28)
color                        0
director_name                0
num_critic_for_reviews       0
duration                     0
director facebook likes      0
actor_3_facebook_likes       0
actor_2_name                 0
actor_1_facebook_likes       0
gross                        0
genres                       0
actor_1_name                 0
movie_title                  0
num voted users              0
cast_total_facebook_likes    0
actor_3_name                 0
facenumber_in_poster         0
plot keywords                0
movie_imdb_link              0
num_user_for_reviews         0
language                     0
country                      0
content rating               0
budget                       0
title_year                   0
actor_2_facebook_likes       0
imdb score                   0
aspect_ratio                 0
movie_facebook_likes         0
dtype: int64
(7, 28)
color                        0
director_name                0
num_critic_for_reviews       0
duration                     0
director_facebook_likes      0
actor_3_facebook_likes       0
actor_2_name                 0
actor_1_facebook_likes       0
gross                        0
genres                       0
actor_1_name                 0
movie_title                  0
num voted users              0
cast_total_facebook_likes    0
actor_3_name                 0
facenumber_in_poster         0
plot keywords                0
movie_imdb_link              0
num user for reviews         0
```



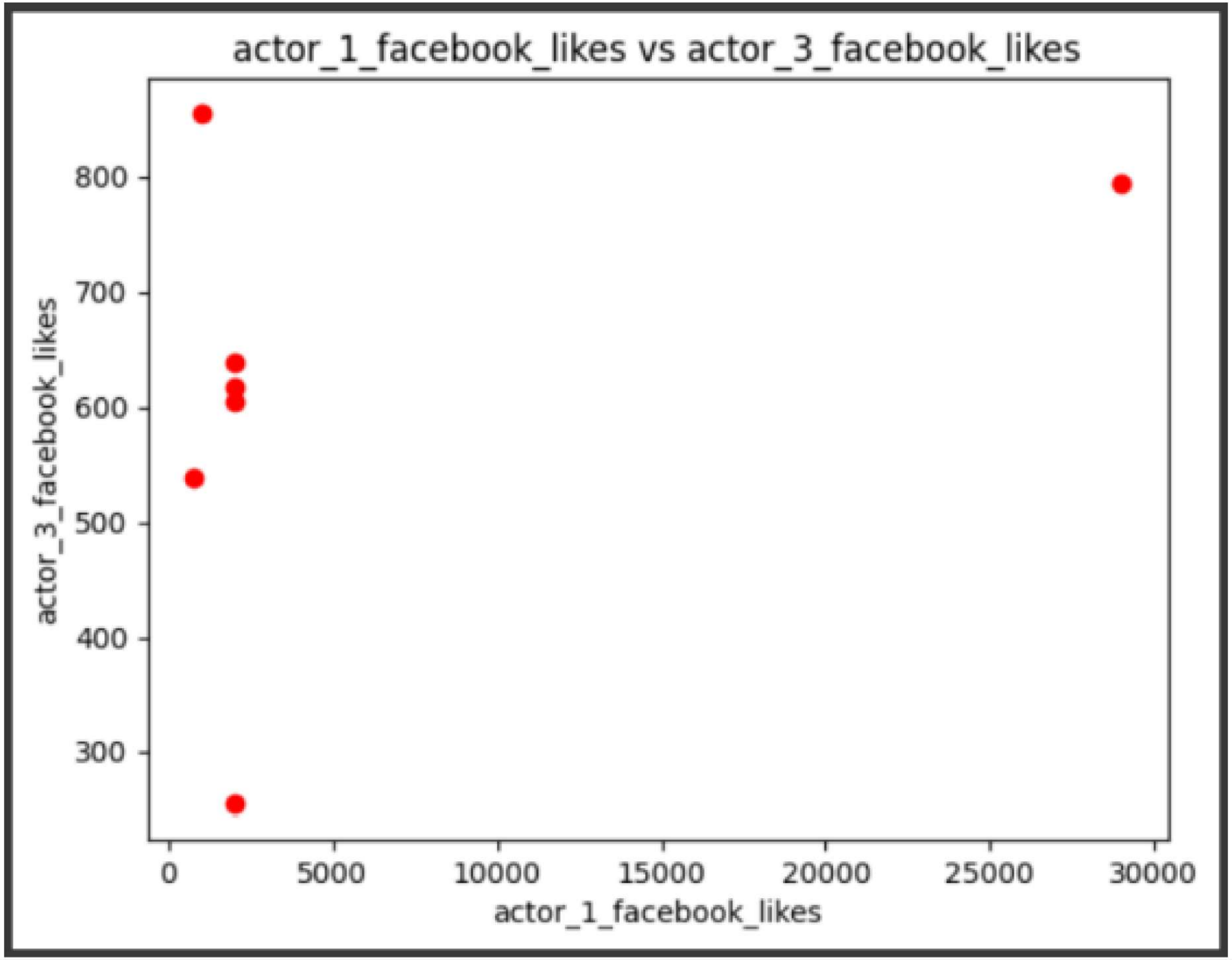
```
Ridley Scott        17
                    ..
John Crowley         1
Rob Pritts           1
David S. Ward        1
R.J. Cutler          1
Daniel Hsia          1
Name: director_name, Length: 2398, dtype: int
(7, 28)
color                        0
director_name                0
num_critic_for_reviews       0
duration                     0
director_facebook_likes      0
actor_3_facebook_likes       0
actor_2_name                 0
actor_1_facebook_likes       0
gross                        0
genres                       0
actor_1_name                 0
movie_title                  0
num_voted_users              0
cast_total_facebook_likes    0
actor_3_name                 0
facenumber_in_poster         0
plot_keywords                0
movie_imdb_link              0
num_user_for_reviews         0
language                     0
country                      0
content_rating               0
budget                       0
title_year                   0
actor_2_facebook_likes       0
imdb_score                   0
aspect_ratio                 0
movie_facebook_likes         0
dtype: int64
(7, 28)
color                        0
director_name                0
num_critic_for_reviews       0
duration                     0
director_facebook_likes      0
actor_3_facebook_likes       0
actor_2_name                 0
actor_1_facebook_likes       0
gross                        0
genres                       0
actor_1_name                 0
movie_title                  0
num_voted_users              0
cast_total_facebook_likes    0
actor_3_name                 0
facenumber_in_poster         0
plot_keywords                0
movie_imdb_link              0
num_user_for_reviews         0
```

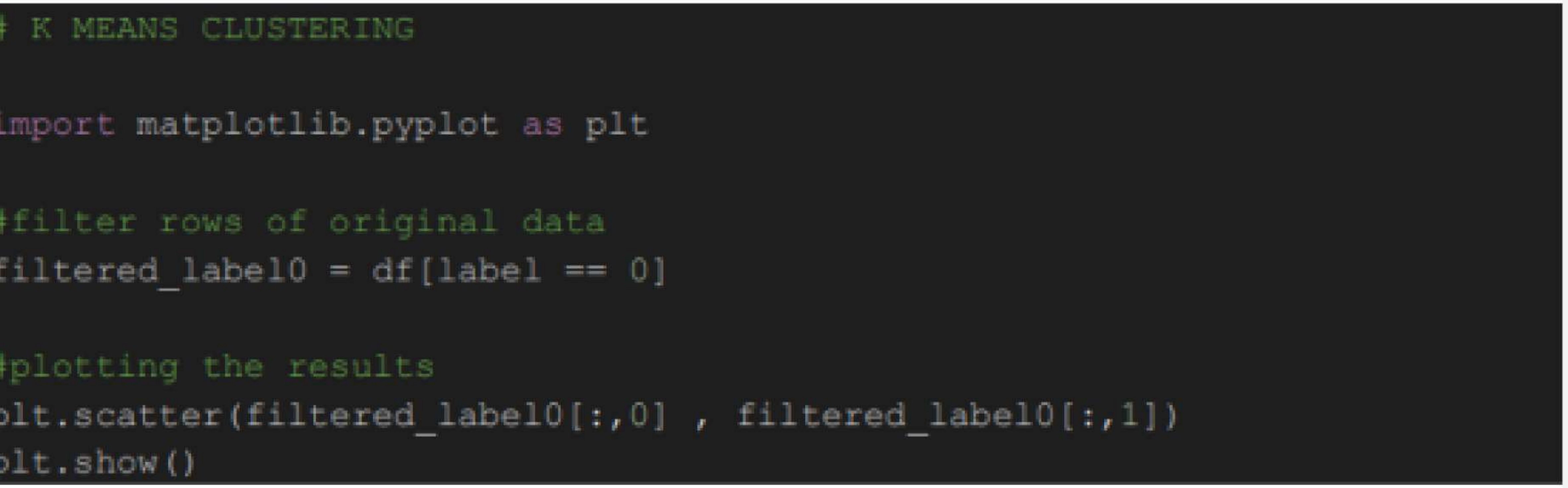


```
language                     0
country                      0
content_rating               0
budget                       0
title_year                   0
actor_2_facebook_likes       0
imdb_score                   0
aspect_ratio                 0
movie_facebook_likes         0
dtype: int64
(7, 1)
(7, 1)
```

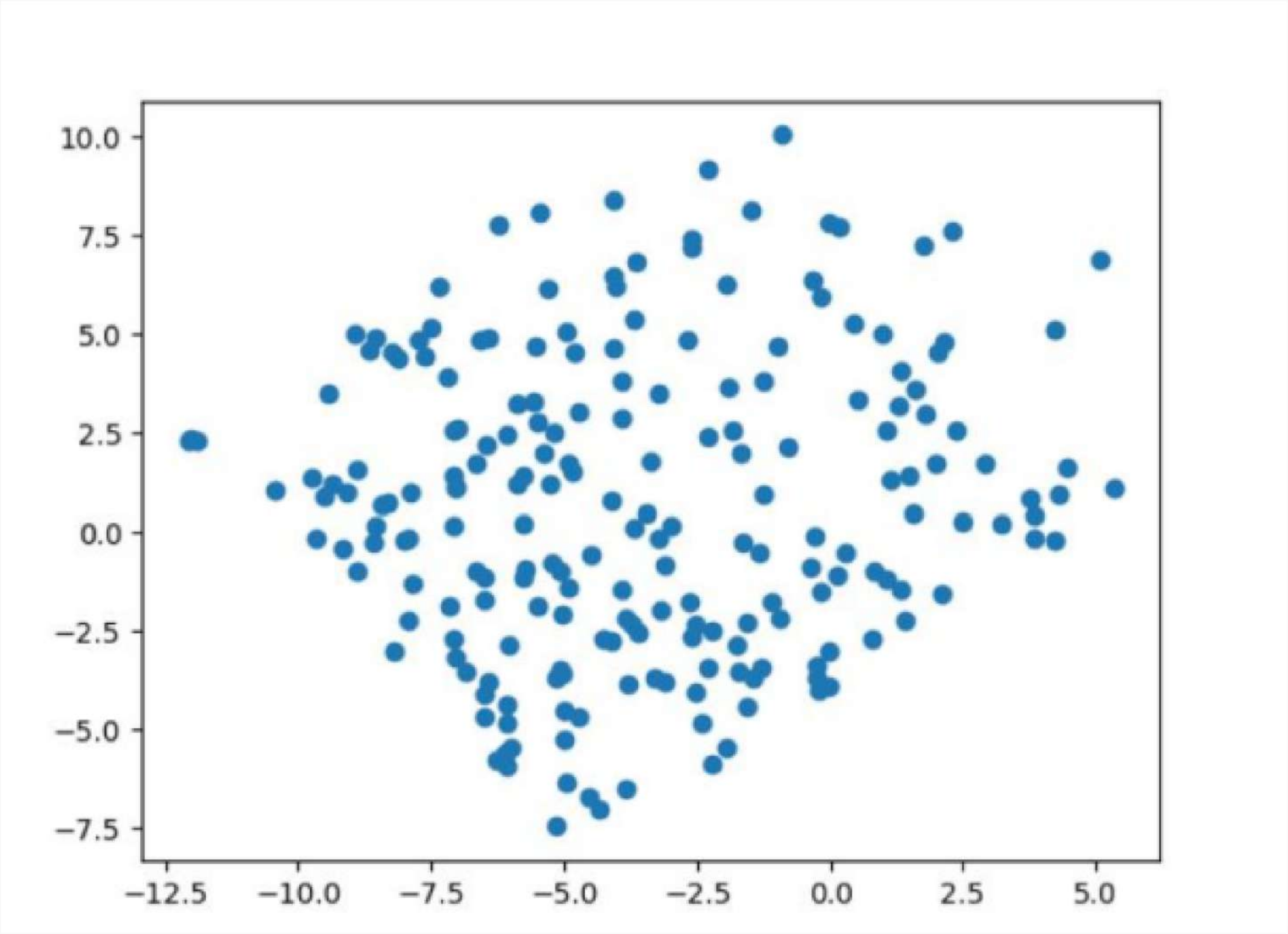


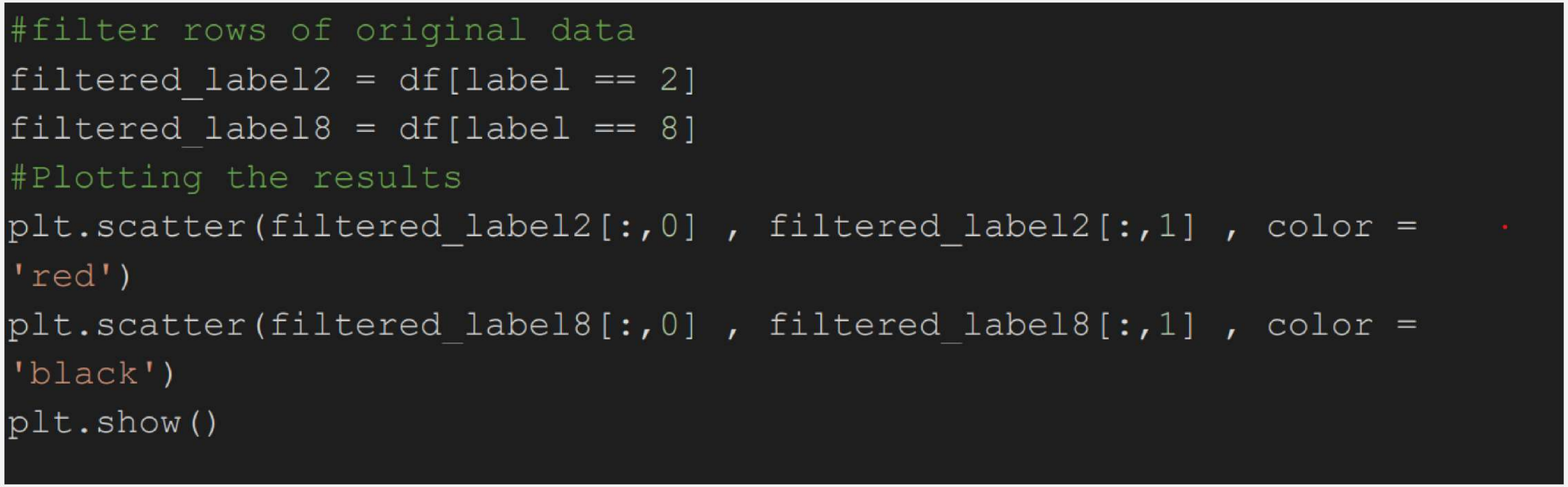
```
plt.scatter(X,y,color="red") # Plot a graph X vs y
plt.title('actor_1_facebook_likes vs actor_3_facebook_likes')
plt.xlabel('actor_1_facebook_likes')
plt.ylabel('actor_3_facebook_likes')
plt.show()
```


actor_1_facebook_likes vs actor_3_facebook_likes
actor_3_facebook_likes
800
700
600
500
400
300
0
5000
10000
15000
20000
25000
30000
actor_1_facebook_likes

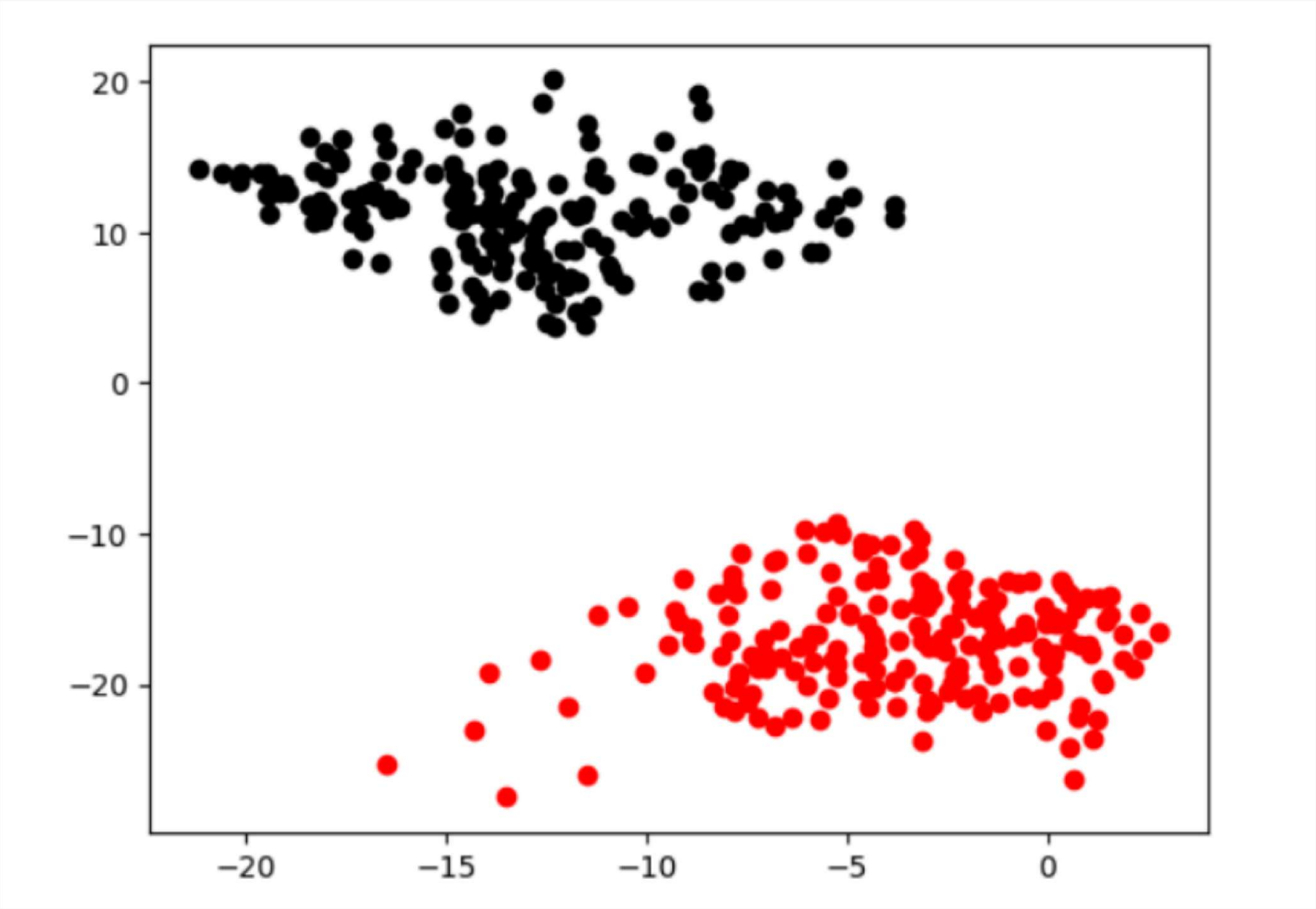


```
# K MEANS CLUSTERING

import matplotlib.pyplot as plt

#filter rows of original data
filtered_label0 = df[label == 0]

#plotting the results
plt.scatter(filtered_label0[:,0] , filtered_label0[:,1])
plt.show()
```





```
#filter rows of original data
filtered_label2 = df[label == 2]
filtered_label8 = df[label == 8]
#Plotting the results
plt.scatter(filtered_label2[:,0] , filtered_label2[:,1] , color =
'red')
plt.scatter(filtered_label8[:,0] , filtered_label8[:,1] , color =
'black')
plt.show()
```


20
10
0
−10
−20
−20
−15
−10
−5
0

Some insight and conclusion :

- This dataset set almost contain same number of films and series
- The USA has been exporting most number of films and TV
- The genre that has dominates are comedy,drama and docummentaries.

Thank You