

Assignment :05

403 Vaishnavi Amate

407 Pratik Bangar

410 Dhanashree Borkar

Dataset : Netflix

```
import pandas as pd import numpy as np
import matplotlib.pyplot as plt df =
pd.read_csv('/content/netflix_list.csv')
df.head()
```

```
missing_values = df.isnull().sum()
```

#1) Graph

```
df['startYear'] = df['startYear'].fillna('Unknown')
df['episodes'] = df['episodes'].fillna('No Data')
df['certificate'] = df['certificate'].fillna('No certificate')
df['numVotes'] = df['numVotes'].fillna('No rate')
df['rating'] = df['rating'].fillna('No rate') df['plot'] =
df['certificate'].fillna('No Data') df['language'] =
df['language'].fillna('Unknown') df['genres'] =
df['genres'].fillna('No Genre') df['type'] =
df['type'].fillna('No Type')

df['runtime'] = df['runtime'].fillna('Unknown')
```

```
# 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

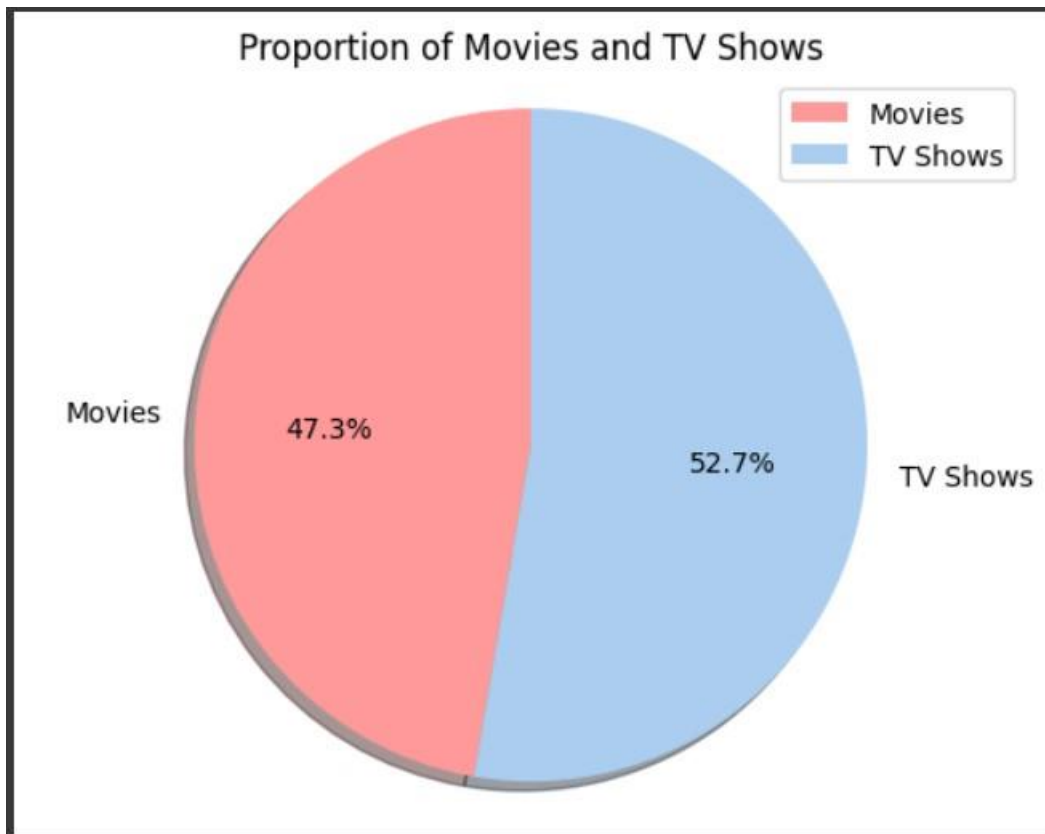
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=90,
shadow=True)


# 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()
```



#2) Graph

# Filter and aggregate the data

```
# Filter out rows where the 'rating' column is 'No rate' df.rating  
= df.rating[df.rating != 'No rate']
```

```
# Filter out rows where the 'numVotes' column is 'No rate' df.numVotes  
= df.numVotes[df.numVotes != 'No rate']
```

```
# Filter out rows where the 'startYear' column is 'Unknown' df.startYear  
= df.startYear[df.startYear != 'Unknown']
```

```
# Group the filtered data by 'startYear' and calculate the mean of 'rating' and  
the sum of 'numVotes'
```

```
rate_per_year = df.groupby('startYear').agg({'rating':'mean','numVotes':'sum'})
```

```
# Select just the last 15 years until 2021 rate_per_year
```

```
= rate_per_year.iloc[:15].tail(15)
```

```
# Create the figure object and plot the data fig,
```

```
ax1 = plt.subplots(figsize=(11, 6))
```

```
# Plot the 'rating' column as a line chart with label 'Rating'
```

```
ax1.plot(rate_per_year['rating'], label='Rating', color='#852852', marker='o',  
linestyle='-', linewidth=2)
```

```
# Set the y-axis label for the line chart ax1.set_ylabel('Rating')
```

```
# Create a second y-axis for the bar chart ax2
```

```
= ax1.twinx()
```

```
# Plot the 'numVotes' column 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()
```

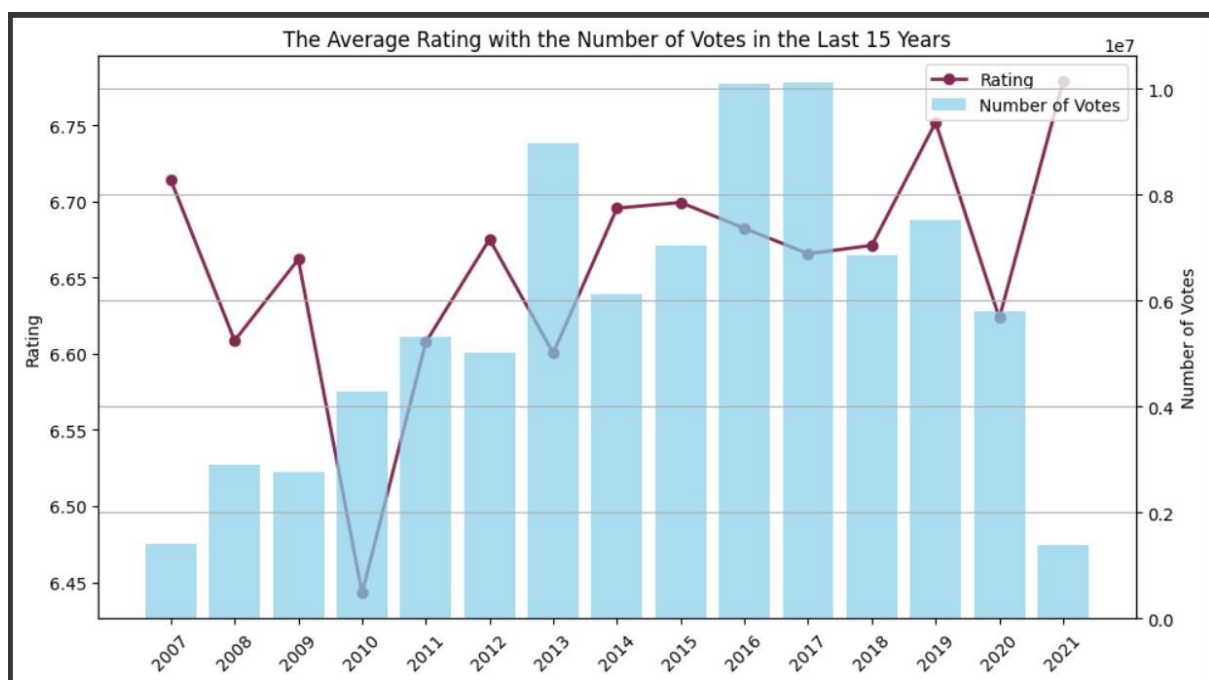
```
ax1.legend(lines + bars, labels + bar_labels, loc='upper right')
```

```
# 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()
```



#3) Graph

```
#group the dataframe by start year ,and count how many rows we have for each year
```

```
df_StartYear = df.groupby('startYear')['imdb_id'].count()
```

```
#we can choose any columns instead of imdb_id column #
remove the rows where the start year is UNKNOWN
df_StartYear = df_StartYear[df_StartYear.index != 'Unknown']
#sort from the first year to last year
df_StartYear = df_StartYear.sort_index() years
= df_StartYear.index.to_list()

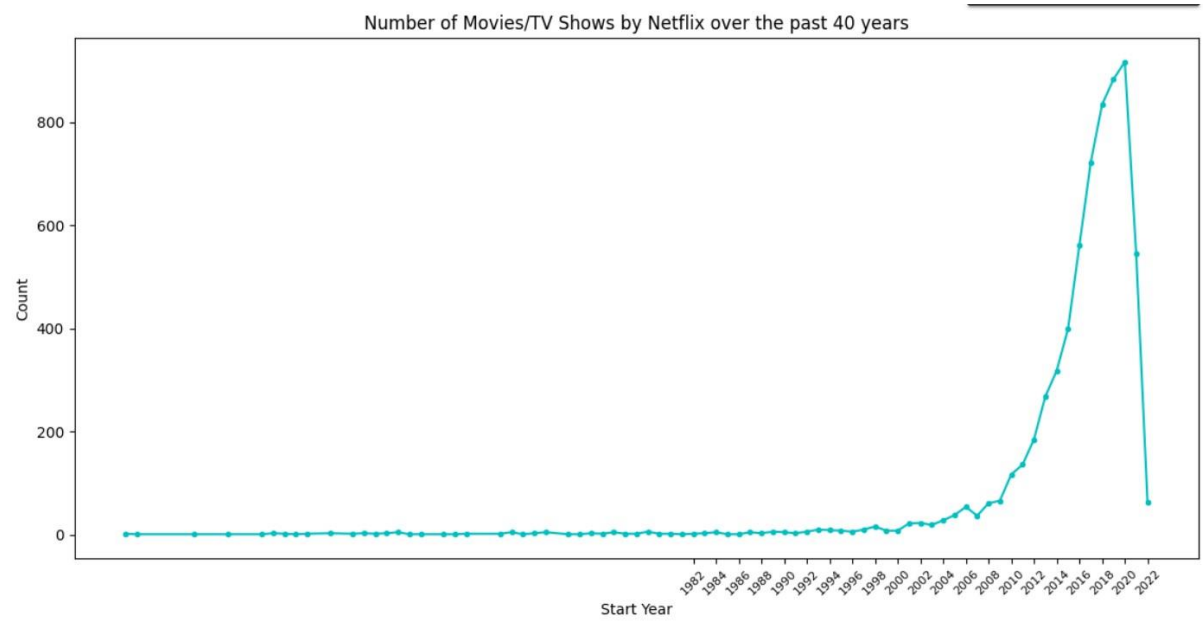
# Create the figure and set the figure size plt.figure(figsize=(11.5,
6))

# Plot the data
plt.plot(df_StartYear[36:], 'c-', marker='.')

# Customize the plot
plt.title('Number of Movies/TV Shows by Netflix over the past 40 years')
plt.xlabel('Start Year') plt.ylabel('Count')

# Adjust x-axis tick spacing
plt.xticks( years[36::2], rotation=45, fontsize=8)

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



#### #4) Graph

# Remove the unknown runtime rows (we have just 2 rows; we can easily remove them without any change for our graph) # +(Store movies and TV shows in a separate variable)

```
movies = df.loc[(df['runtime'] != 'Unknown') & (df['type'].isin(['movie', 'short', 'tvMovie', 'video', 'videoGame', 'tvShort']))]
```

```
tv_shows = df.loc[(df['runtime'] != 'Unknown') & (df['type'].isin(['tvSeries', 'tvEpisode', 'tvSpecial', 'tvMiniSeries']))]
```

# Convert the runtime column to float type using .loc movies.loc[:,

'runtime'] = movies['runtime'].astype(float) tv\_shows.loc[:,

'runtime'] = tv\_shows['runtime'].astype(float)

# Group the dataframe by start year and show the runtime for each year

```
movie_runtimeYear = movies.groupby('startYear')[['runtime']].mean()
```

```
tv_shows_runtimeYear = tv_shows.groupby('startYear')[['runtime']].mean() #
```

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)

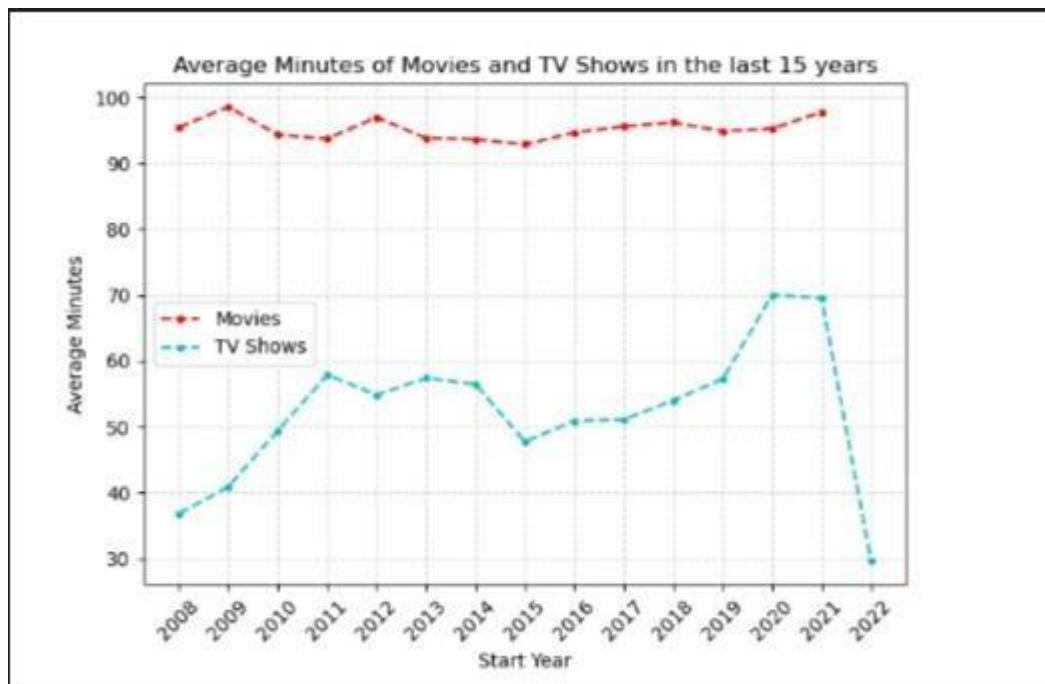
# Customizing tick labels
plt.xticks(last_fifteen_rows_movies.index.to_list(), rotation=45)

# Adding legend
plt.legend()
plt.tight_layout() #
```



Display the plot

plt.show()



#5) Graph df.info() #Plot line

chart data=pd.DataFrame(df)

fig=plt.figure()

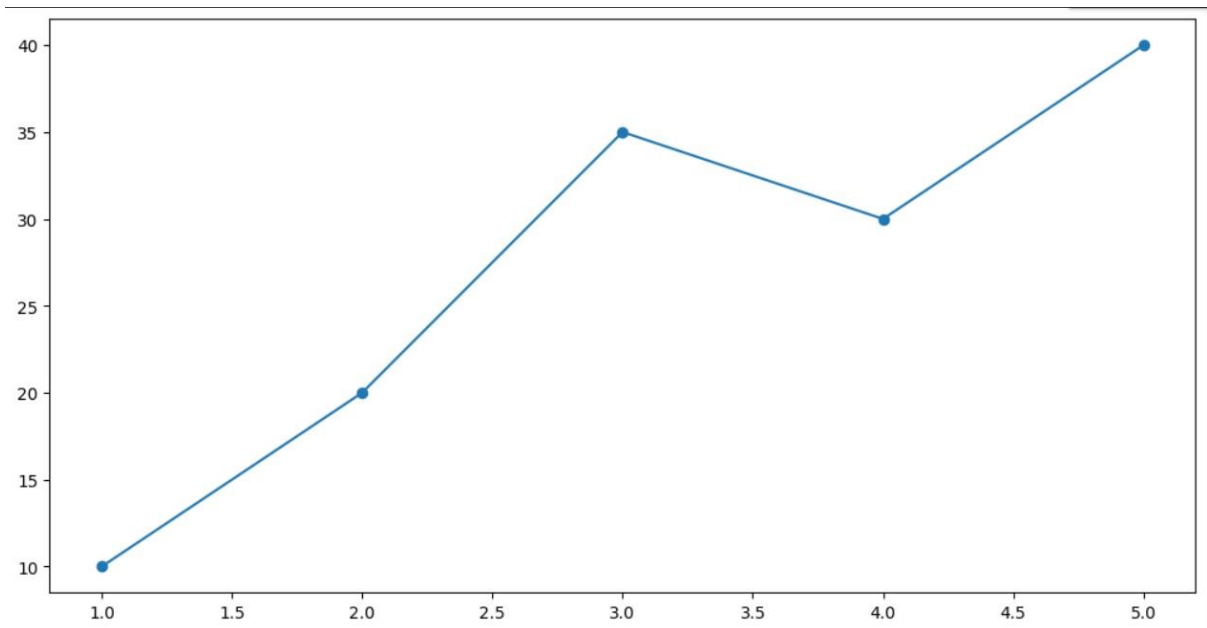
ax=fig.add\_axes([0,0,1.5,1.0])

x=[1,2,3,4,5]

y=[10,20,35,30,40]

plt.plot(x,y,marker='o')

plt.show



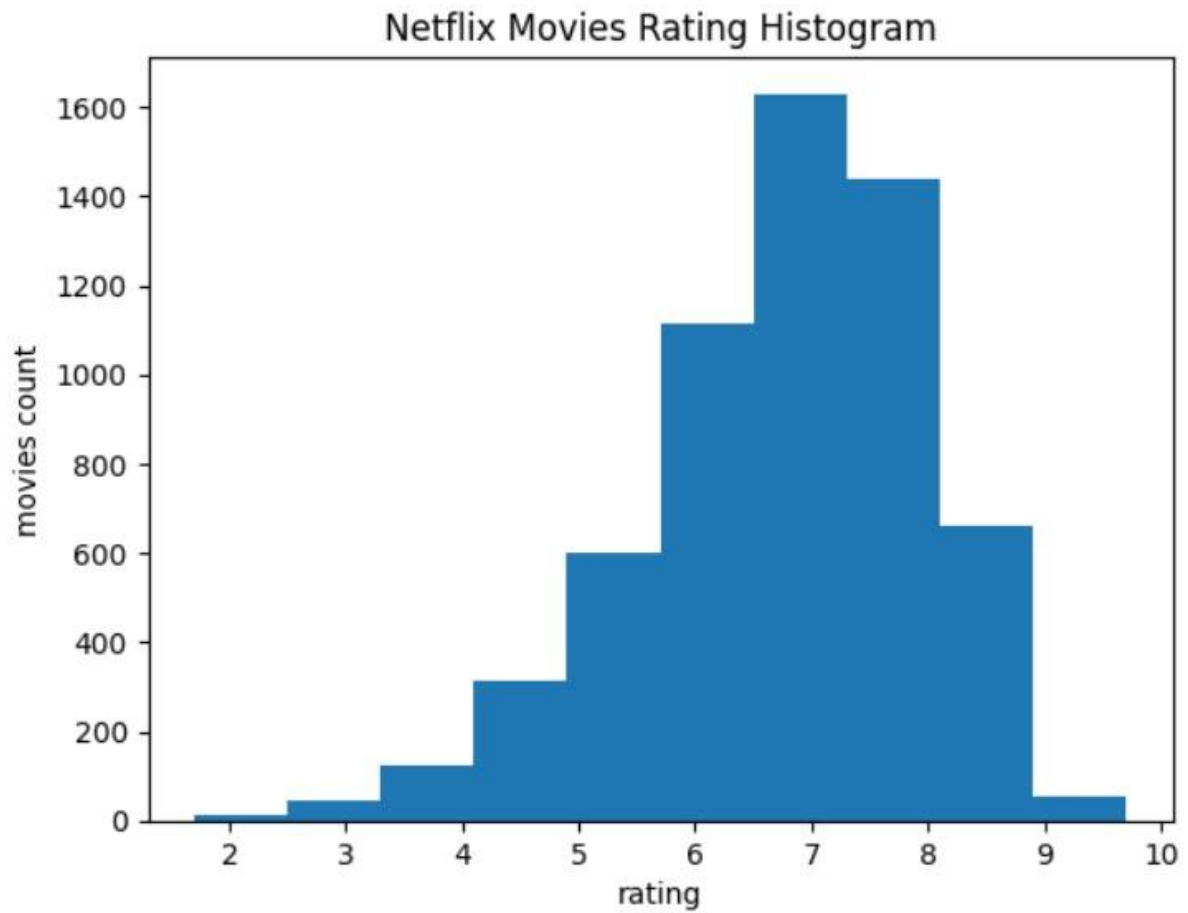
#6) Graph `df1=df.dropna()`

`print(df1.head())`

`plt.xlabel('rating')`

`plt.ylabel('movies count')`

`plt. title('Netflix Movies Rating Histogram') plt.hist(df['rating'])`



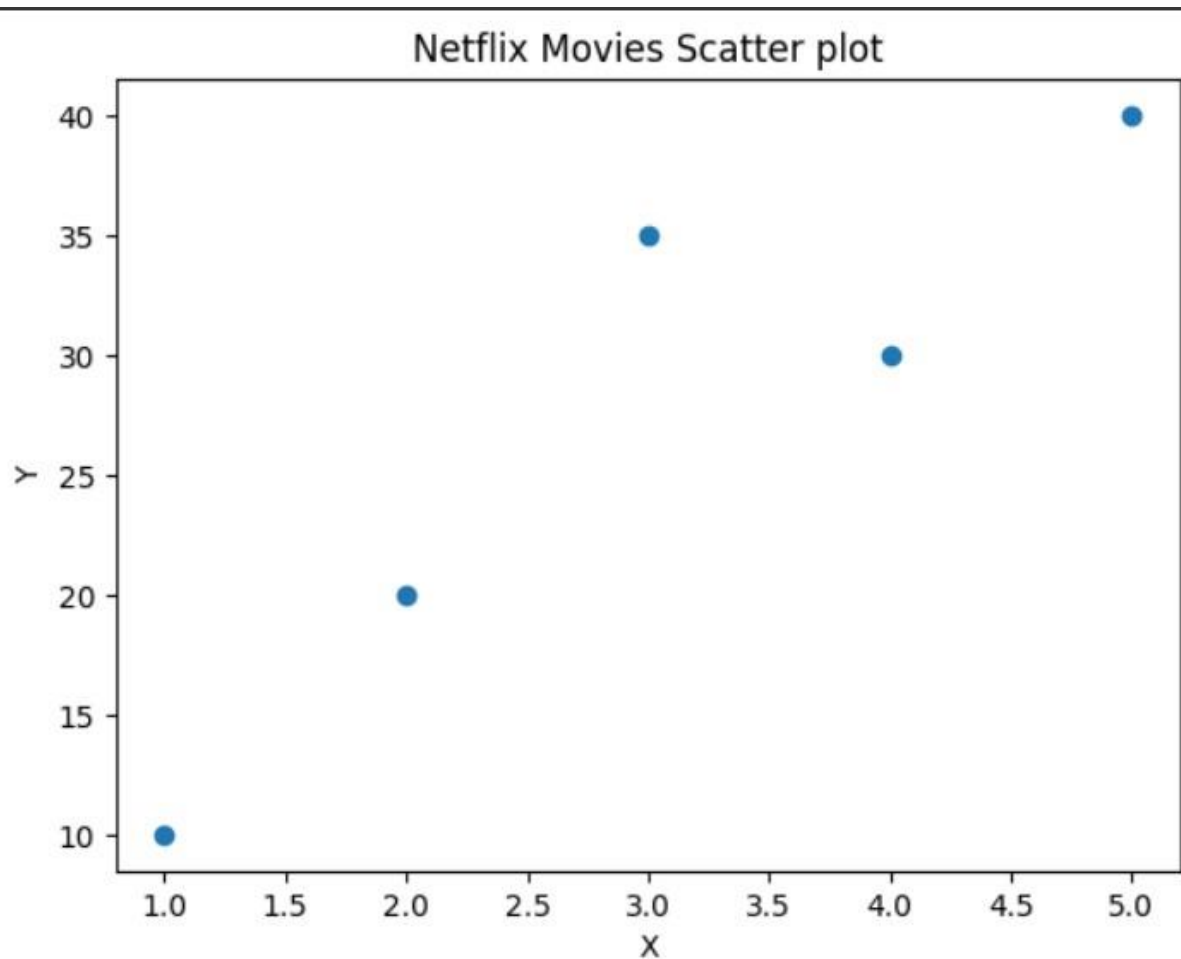
#7) Graph `df1=df.dropna()`

`print(df1.head())`

`plt.xlabel('X')`

`plt.ylabel('Y')`

`plt. title('Netflix Movies Scatter plot') plt.scatter(x,y)`



#8) Graph

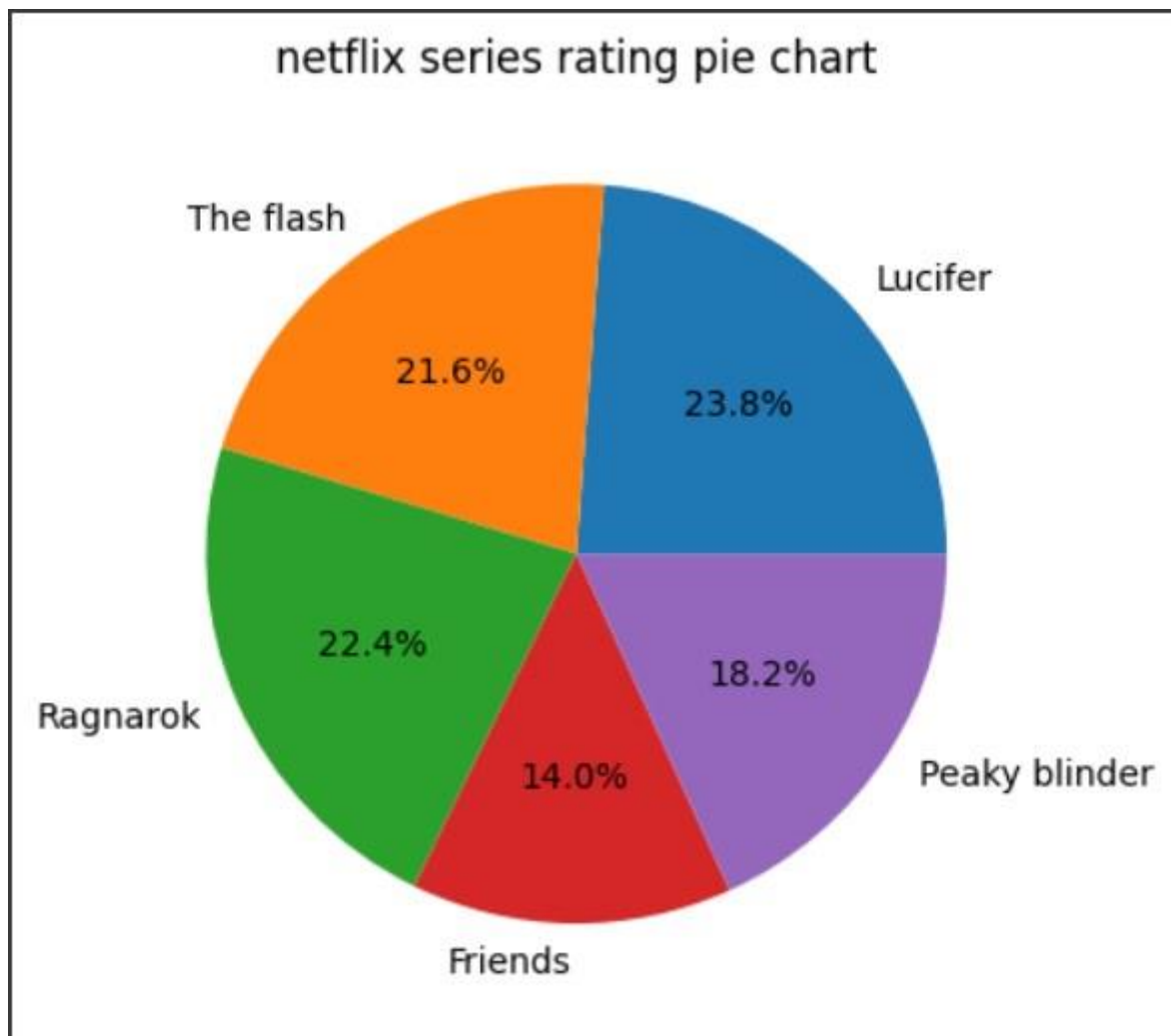
```
title=['Lucifer','The flash','Ragnarok','Friends','Peaky blinder']
```

```
Rating=[8.5,7.7,8,5,6.5] #plotting
```

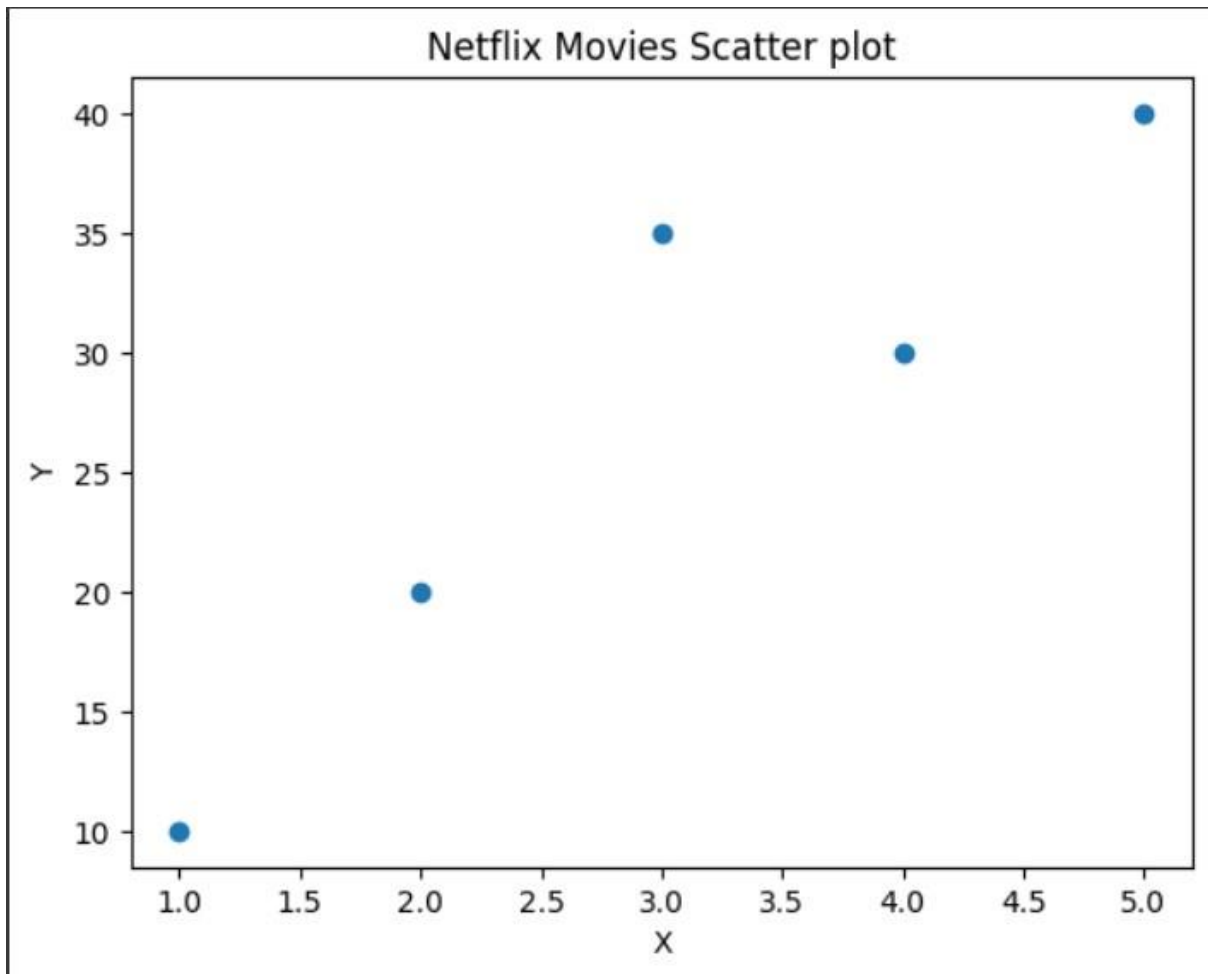
the pie chart

```
plt.title('netflix series rating pie chart')
```

```
plt.pie(Rating,labels=title,autopct='%1.1f%%') plt.show()
```



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



#10) Graph

```
title=['Army of the Dead','The Woman in the Window','The Mitchells vs the  
Machines','Trouble','Blue Miracle']
```

```
Rating=[5.8,5.7,7.8,5.9,6.7] #plotting
```

the pie chart

```
plt.title('netflix movies rating pie chart')
```

```
plt.pie(Rating,labels=title,autopct='%1.1f%%') plt.show()
```

netflix movies rating pie chart

