[ ]:
```
import pandas as pd
import numpy as np
import random as rnd
import seaborn as sns
import matplotlib.pyplot as plt
```

[ ]:
```
movie_df= pd.read_csv('/content/movie_dataset.csv',encoding="latin1")
movie_df.head()
```

[ ]:

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1	Actor 2	Actor 3
0		NaN	NaN	Drama	NaN	NaN	J.S. Randhawa	Manmauji	Birbal	Rajendra Bhatia
1	#Gadhvi (He thought he was Gandhi)	-2019.0	109 min	Drama	7.0	8	Gaurav Bakshi	Rasika Dugal	Vivek Ghamande	Arvind Jangid
2	#Homecoming	-2021.0	90 min	Drama, Musical	NaN	NaN	Soumyajit Majumdar	Sayani Gupta	Plabita Borthakur	Roy Angana
3	#Yaaram	-2019.0	110 min	Comedy, Romance	4.4	35	Ovais Khan	Prateik	Ishita Raj	Siddhant Kapoor
4	…And Once Again	-2010.0	105 min	Drama	NaN	NaN	Amol Palekar	Rajat Kapoor	Rituparna Sengupta	Antara Mali

[ ]:
```
movie_df.shape
```

[ ]: (15509, 10)

[ ]:
```
movie_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15509 entries, 0 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      15509 non-null  object
 1   Year      14981 non-null  float64
 2   Duration  7240 non-null   object
 3   Genre     13632 non-null  object
 4   Rating    7919 non-null   float64
 5   Votes     7920 non-null   object
 6   Director  14984 non-null  object
 7   Actor 1   13892 non-null  object
 8   Actor 2   13125 non-null  object
 9   Actor 3   12365 non-null  object
dtypes: float64(2), object(8)
memory usage: 1.2+ MB
```

calculate the statistics value

[ ]:
```
movie_df.describe()
```

[ ]:

	Year	Rating
count	14981.000000	7919.000000
mean	-1987.012215	5.841621
std	25.416689	1.381777
min	-2022.000000	1.100000
25%	-2009.000000	4.900000
50%	-1991.000000	6.000000
75%	-1968.000000	6.800000
max	-1913.000000	10.000000

Data Cleaning

[ ]:
```
movie_df.isnull().sum()
```

[ ]:
```
Name           0
Year         528
Duration    8269
Genre       1877
Rating      7590
Votes       7589
Director     525
Actor 1     1617
Actor 2     2384
```

```
Actor 3     3144
dtype: int64
```

[ ]:
```
movie_df.dropna(subset=["Rating"], inplace = True)
```

[ ]:
```
movie_df.isnull().sum()
```

[ ]:
```
Name           0
Year           0
Duration    2068
Genre        102
Rating         0
Votes          0
Director       5
Actor 1      125
Actor 2      200
Actor 3      292
dtype: int64
```

[ ]:
```
movie_df.dropna(subset=['Actor 1','Actor 2','Actor␣
↪3','Director','Genre'],inplace=True)
```

[ ]:
```
movie_df.isnull().sum()
```

[ ]:
```
Name           0
Year           0
Duration    1899
Genre          0
Rating         0
Votes          0
Director       0
Actor 1        0
Actor 2        0
Actor 3        0
dtype: int64
```

[ ]:
```
# convert votes columns
movie_df['Votes'] = movie_df['Votes'].str.replace(',','').astype(int)
```

[ ]:
```
# convert year columns
if movie_df['Year'].dtype == object:
    movie_df['Year'] = movie_df['Year'].str.strip('()').astype(int)
```

[ ]:
```
# convert duration columns
movie_df['Duration'] = movie_df['Duration'].str.strip('min')
```

[ ]:
```
movie_df['Duration'].fillna(movie_df['Duration'].median(),inplace=True)
```

[ ]:
```
movie_df.isnull().sum()
```

[ ]:
```
Name        0
Year        0
Duration    0
Genre       0
Rating      0
Votes       0
Director    0
Actor 1     0
Actor 2     0
Actor 3     0
dtype: int64
```

[ ]:
```
movie_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7558 entries, 1 to 15508
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Name      7558 non-null   object
 1   Year      7558 non-null   float64
 2   Duration  7558 non-null   object
 3   Genre     7558 non-null   object
 4   Rating    7558 non-null   float64
 5   Votes     7558 non-null   int64
 6   Director  7558 non-null   object
 7   Actor 1   7558 non-null   object
 8   Actor 2   7558 non-null   object
 9   Actor 3   7558 non-null   object
dtypes: float64(2), int64(1), object(7)
memory usage: 649.5+ KB
```

[ ]:
```
movie_df.head()
```

[ ]:
```
                                  Name    Year Duration  \
1  #Gadhvi (He thought he was Gandhi) -2019.0     109
3                             #Yaaram -2019.0     110
5                …Aur Pyaar Ho Gaya -1997.0     147
6                           …Yahaan -2005.0     142
8                 ?: A Question Mark -2012.0      82
```

```
                      Genre  Rating  Votes        Director          Actor 1  \
1                     Drama     7.0      8   Gaurav Bakshi     Rasika Dugal
3           Comedy, Romance     4.4     35      Ovais Khan          Prateik
5    Comedy, Drama, Musical     4.7    827    Rahul Rawail       Bobby Deol
6       Drama, Romance, War     7.4   1086  Shoojit Sircar  Jimmy Sheirgill
8  Horror, Mystery, Thriller     5.6    326   Allyson Patel        Yash Dave

                  Actor 2          Actor 3
1          Vivek Ghamande    Arvind Jangid
3              Ishita Raj  Siddhant Kapoor
5  Aishwarya Rai Bachchan    Shammi Kapoor
6          Minissha Lamba   Yashpal Sharma
8          Muntazir Ahmad     Kiran Bhatia
```

Now data are clean and inputed.

[ ]:
```
print(movie_df.columns)
```

```
Index(['Name', 'Year', 'Duration', 'Genre', 'Rating', 'Votes', 'Director',
       'Actor 1', 'Actor 2', 'Actor 3'],
      dtype='object')
```

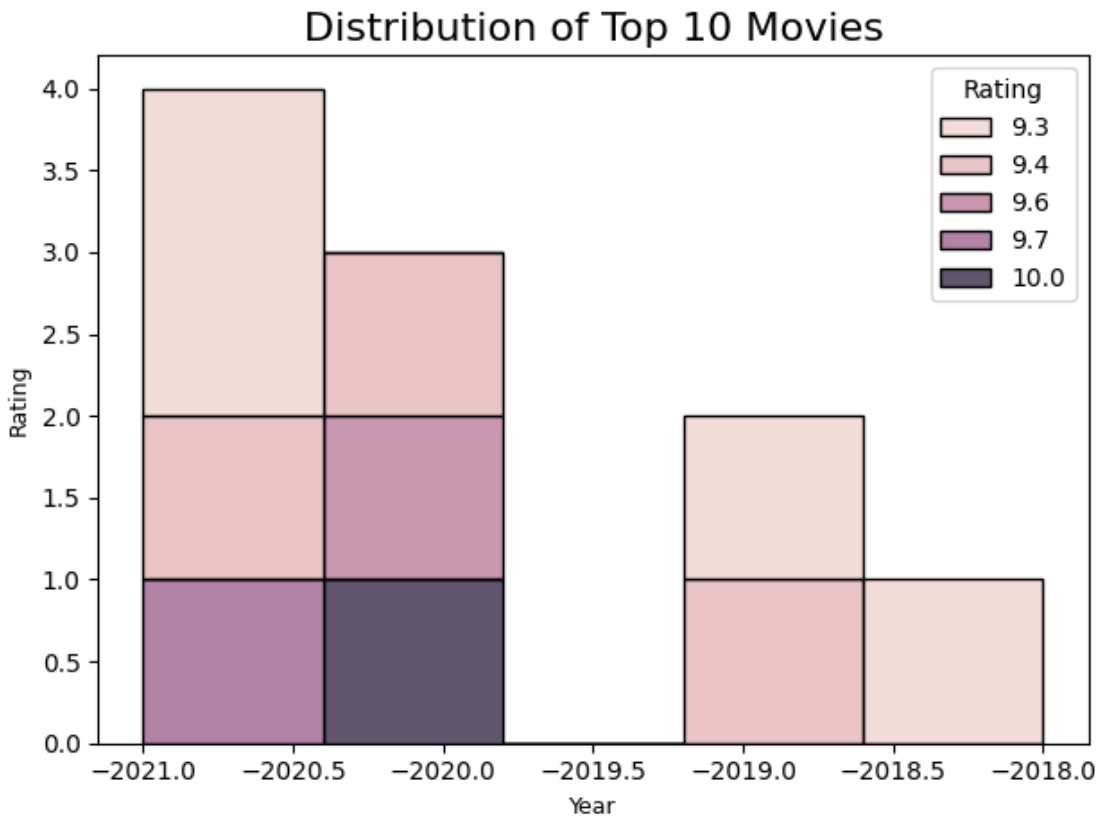
[ ]:
```
# find top 10 movies based on rating
top_movie = movie_df.loc[movie_df['Rating'].sort_values(ascending=False)[:10].
 ↪index]
top_movie
```

[ ]:
```
                     Name    Year Duration                   Genre  Rating  \
8339       Love Qubool Hai -2020.0       94          Drama, Romance    10.0
5410            Half Songs -2021.0       79          Music, Romance     9.7
2563                 Breed -2020.0    135.0                   Drama     9.6
14222  The Reluctant Crime -2020.0      113                   Drama     9.4
5077          Gho Gho Rani -2019.0      105        History, Romance     9.4
6852                  June -2021.0       93                   Drama     9.4
12673   Secrets of Sinauli -2021.0       56    Documentary, History     9.3
5125           God of gods -2019.0       90             Documentary     9.3
8344          Love Sorries -2021.0      101  Comedy, Drama, Romance     9.3
1314          Ashok Vatika -2018.0       97                   Drama     9.3

       Votes         Director           Actor 1          Actor 2  \
8339       5  Saif Ali Sayeed          Ahaan Jha   Mahesh Narayan
5410       7      Sriram Raja       Raj Banerjee  Emon Chatterjee
2563      48      Bobby Kumar        Bobby Kumar           Ashfaq
14222     16    Arvind Pratap    Dharmendra Ahir   Awanish Kotnal
5077      47     Munni Pankaj  Nishi Neha Mishra     Pankaj Kamal
6852      18   Suhrud Godbole     Vaibhav Khisti   Nilesh Divekar
```

```
12673   1373   Raghav Jairath     Manoj Bajpayee       R.S. Bhist
5125      46     Venkatesh Bk  Tejaswini Manogna    Triyug Mantri
8344      79     Gautam Joshi   Prashant Chaubey  Puneet Chouksey
1314       7    Rahul Mallick         Kunj Anand   Sanjay Bishnoi

                  Actor 3
8339   Rajasree Rajakumari
5410     Purshottam Mulani
2563        Fasih Choudhry
14222         Rakhi Mansha
5077           Akash Kumar
6852        Jitendra Joshi
12673           K.N. Dixit
5125       Raj Singh Verma
8344         Amitabh Gupta
1314          Paras Zutshi
```

Distribution of Top 5 movies wrt Year

[ ]:
```
sns.histplot(data=top_movie, x="Year",hue="Rating", multiple="stack")
plt.title('Distribution of Top 10 Movies', fontsize=16)
plt.xlabel('Year', fontsize=9)
plt.ylabel('Rating', fontsize=9)
plt.tight_layout()
plt.show()
```

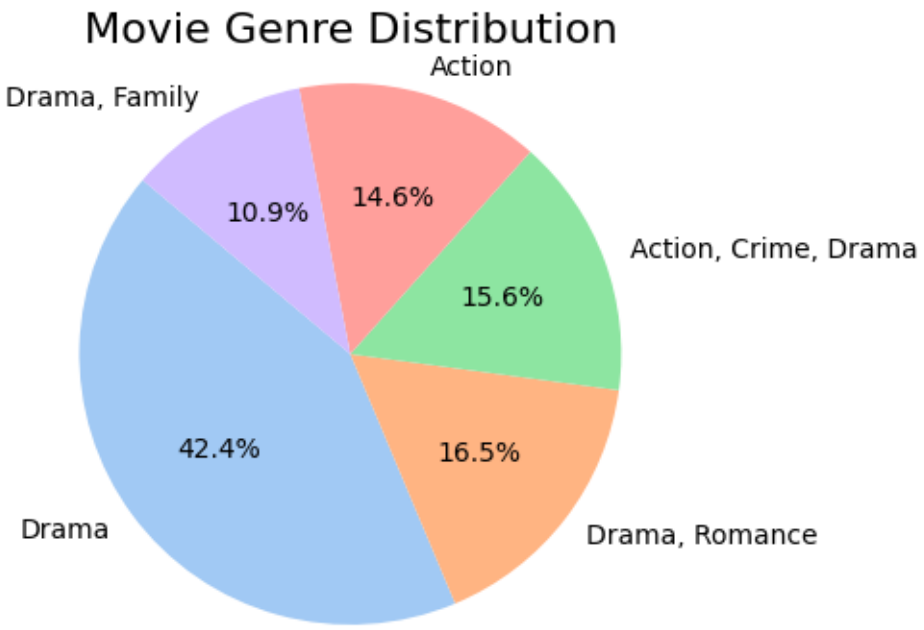


[ ]:
```
genre_counts = movie_df['Genre'].value_counts().reset_index()
genre_counts.columns = ['Genre', 'Count']

# Select the top N genres (e.g., top 5)
top_n_genres = genre_counts.head(5)
top_n_genres
```

[ ]:
```
                  Genre  Count
0                 Drama   1137
1        Drama, Romance    443
2  Action, Crime, Drama    417
3                Action    391
4         Drama, Family    291
```

[ ]:
```
plt.figure(figsize=(4, 4))
plt.pie(top_n_genres['Count'], labels=top_n_genres['Genre'], autopct='%1.1f%%',␣
 ↪startangle=140, colors=sns.color_palette('pastel'))
plt.title('Movie Genre Distribution', fontsize=16)
plt.axis('equal')
```

```
plt.show()
```



Distribution of Top directors by average rating

[ ]:
```
# Group the data by director and calculate the average rating
director_avg_rating = movie_df.groupby('Director')['Rating'].mean().
 ↪reset_index()

director_avg_rating = director_avg_rating.sort_values(by='Rating',␣
 ↪ascending=False)

top_directors = director_avg_rating.head()
top_directors
```

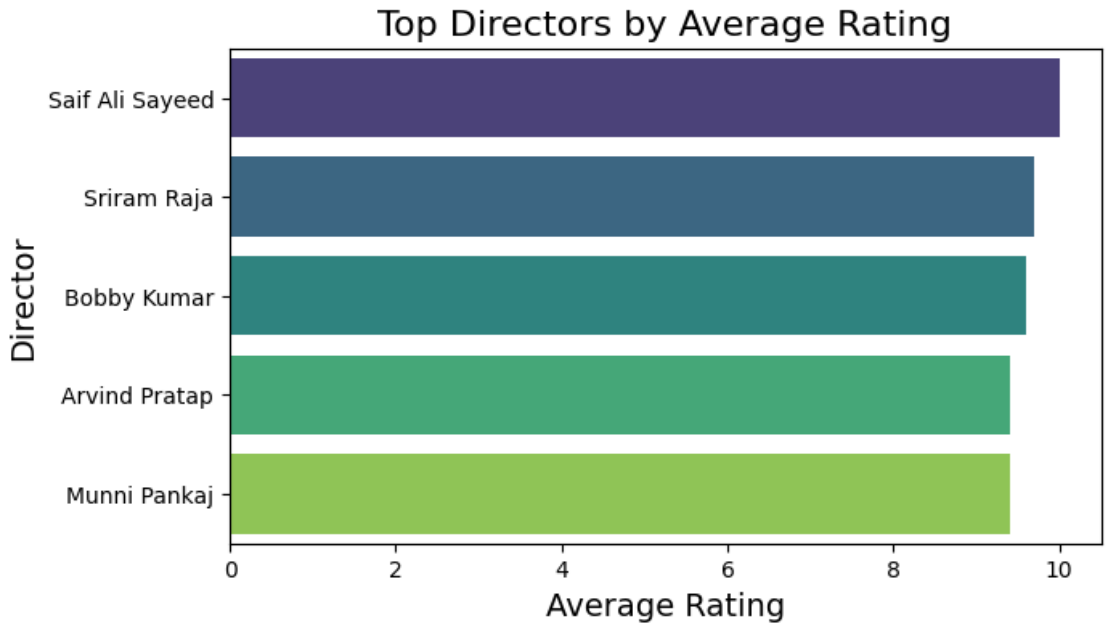
[ ]:

	Director	Rating
2243	Saif Ali Sayeed	10.0
2560	Sriram Raja	9.7
504	Bobby Kumar	9.6
322	Arvind Pratap	9.4
1513	Munni Pankaj	9.4

[ ]:
```
plt.figure(figsize=(7, 4))
sns.barplot(data=top_directors, x='Rating', y='Director', palette='viridis')

plt.title('Top Directors by Average Rating', fontsize=16)
```

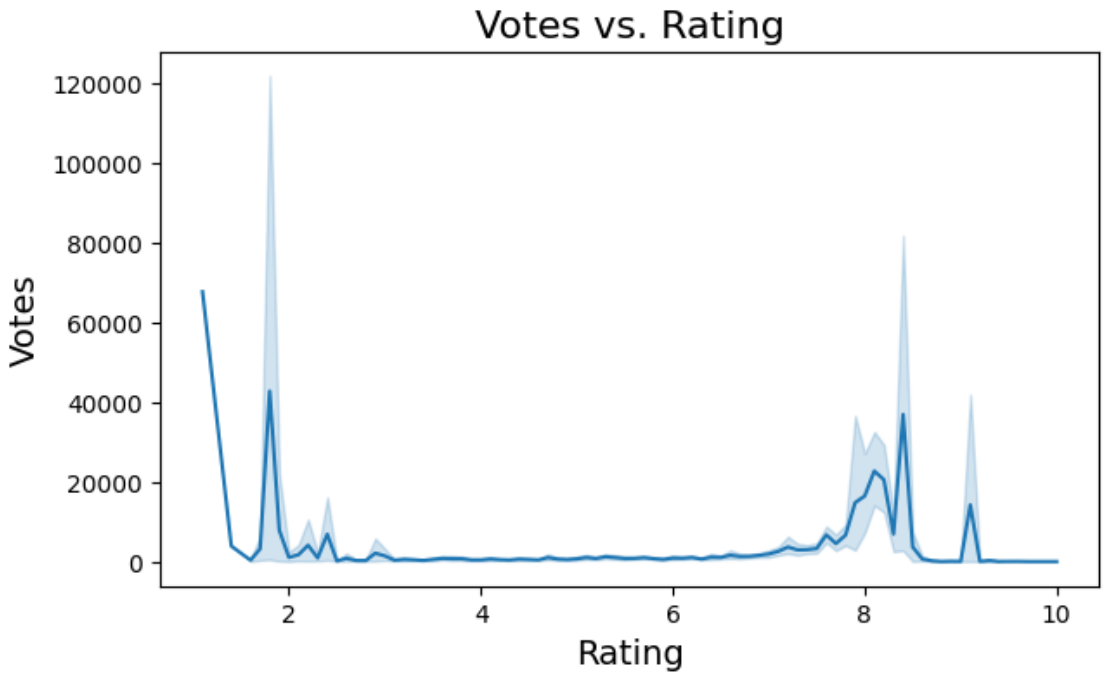
```
plt.xlabel('Average Rating', fontsize=14)
plt.ylabel('Director', fontsize=14)

plt.show()
```

```
<ipython-input-79-bd877bbaefbb>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(data=top_directors, x='Rating', y='Director', palette='viridis')
```



Relationship between the number of votes and movie ratings

[ ]:
```
plt.figure(figsize=(7, 4))
sns.lineplot(data=movie_df, x='Rating', y='Votes')

plt.title('Votes vs. Rating', fontsize=16)
plt.xlabel('Rating', fontsize=14)
plt.ylabel('Votes', fontsize=14)

plt.show()
```

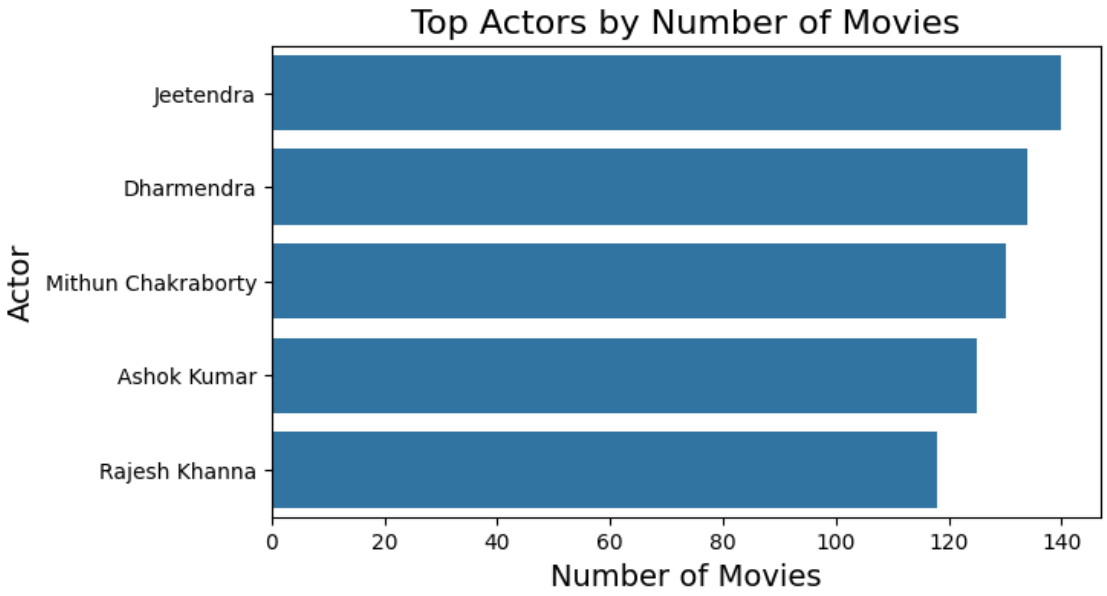


Distribution of top actors by number of movie

[ ]:
```
actor_counts = movie_df['Actor 1'].value_counts().reset_index()
actor_counts.columns = ['Actor', 'MovieCount']

top_n_actors = actor_counts.head()
top_n_actors
```

[ ]:
```
                Actor  MovieCount
0           Jeetendra         140
1          Dharmendra         134
2  Mithun Chakraborty         130
3         Ashok Kumar         125
4       Rajesh Khanna         118
```

[ ]:
```
plt.figure(figsize=(7, 4))
sns.barplot(data=top_n_actors, x='MovieCount', y='Actor', orient='h')

# Set plot labels and title
plt.title('Top Actors by Number of Movies', fontsize=16)
plt.xlabel('Number of Movies', fontsize=14)
plt.ylabel('Actor', fontsize=14)

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



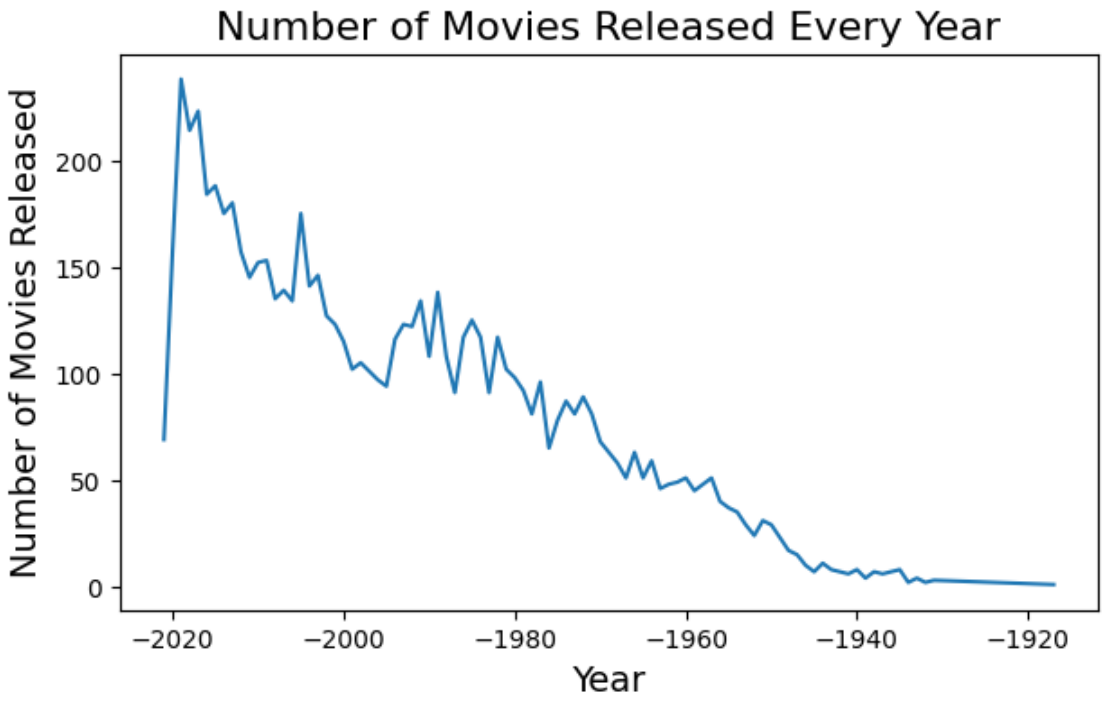
Distribution of number of movie released every year

[ ]:
```
yearly_movie_counts = movie_df['Year'].value_counts().reset_index()
yearly_movie_counts.columns = ['Year', 'MovieCount']

yearly_movie_counts = yearly_movie_counts.sort_values(by='Year')
yearly_movie_counts
```

[ ]:
```
       Year  MovieCount
49 -2021.0          69
9  -2020.0         157
0  -2019.0         238
2  -2018.0         214
1  -2017.0         223
..     …           …
90 -1934.0           2
86 -1933.0           4
89 -1932.0           2
88 -1931.0           3
91 -1917.0           1

[92 rows x 2 columns]
```

[ ]:
```
plt.figure(figsize=(7, 4))
sns.lineplot(data=yearly_movie_counts, x='Year', y='MovieCount')

plt.title('Number of Movies Released Every Year', fontsize=16)
```

```
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Movies Released', fontsize=14)

plt.show()
```



Distribution of Movies with rating greater than 8 and votes greater than 10000

[ ]:
```
filtered_df = movie_df[(movie_df['Rating'] > 8) & (movie_df['Votes'] > 10000)]
filtered_df.head(10)
```

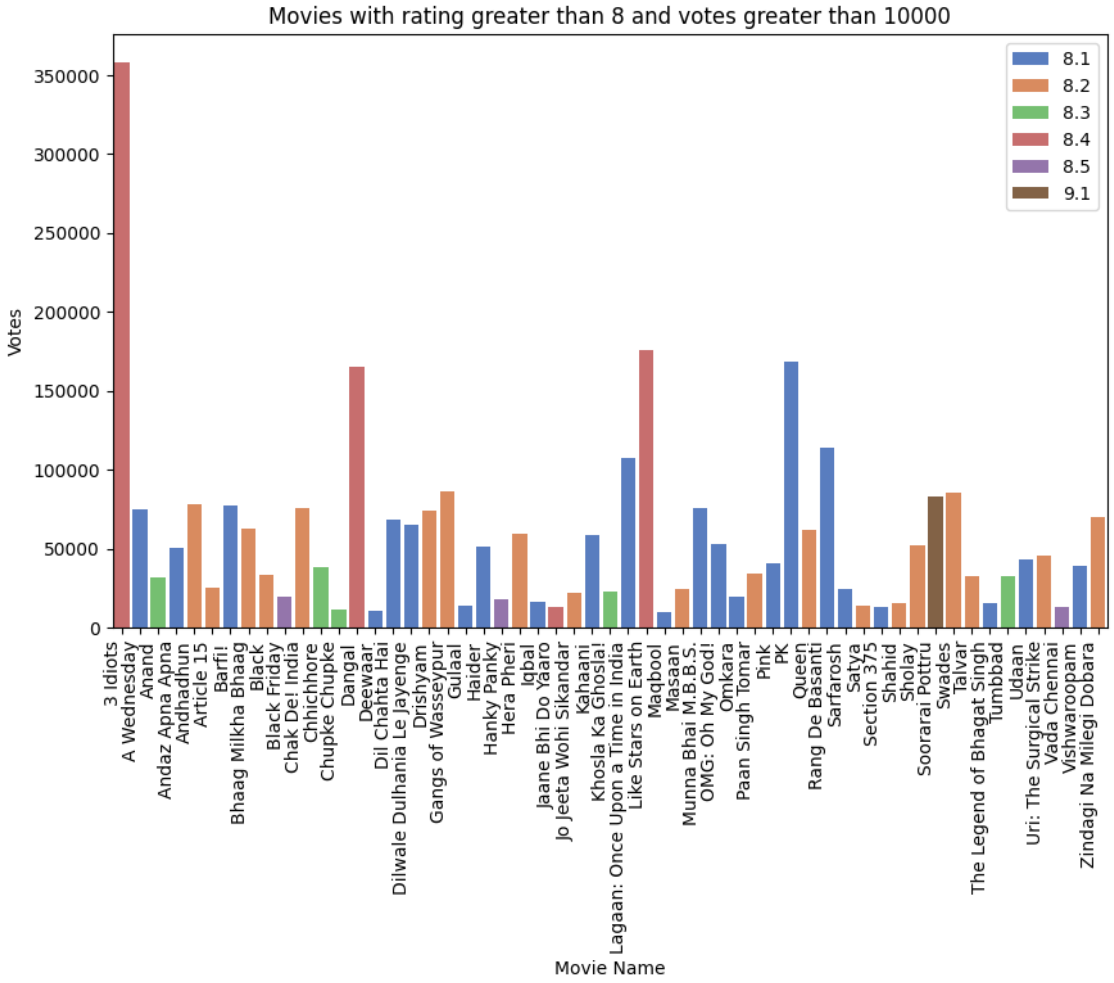
[ ]:

	Name	Year	Duration	Genre	Rating	Votes	Director	Actor 1
75	3 Idiots	-2009.0	170	Comedy, Drama	8.4	357889	Rajkumar Hirani	Aamir Khan
173	A Wednesday	-2008.0	104	Action, Crime, Drama	8.1			
981	Anand	-1971.0	122	Drama, Musical	8.3			
1009	Andaz Apna Apna	-1994.0	160	Action, Comedy, Romance	8.1			
1019	Andhadhun	-2018.0	139	Crime, Drama, Music	8.2			
1285	Article 15	-2019.0	130	Crime, Drama, Mystery	8.2			
1877	Barfi!	-2012.0	151	Comedy, Drama, Romance	8.1			
2065	Bhaag Milkha Bhaag	-2013.0	186	Biography, Drama, Sport	8.2			
2412	Black	-2005.0	122	Drama	8.2			
2425	Black Friday	-2004.0	143	Action, Crime, Drama	8.5			

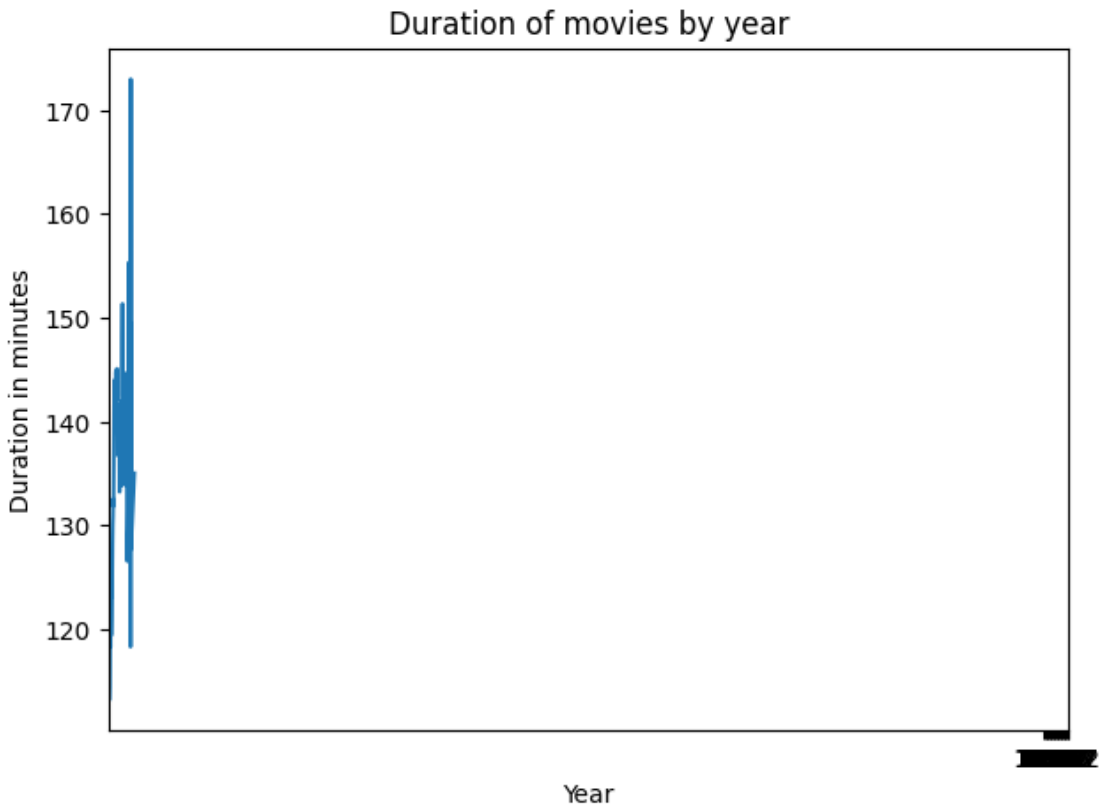
```
173    75118            Neeraj Pandey         Anupam Kher
981    31937     Hrishikesh Mukherjee       Rajesh Khanna
1009   50810        Rajkumar Santoshi          Aamir Khan
1019   77901          Sriram Raghavan  Ayushmann Khurrana
1285   25706            Anubhav Sinha  Ayushmann Khurrana
1877   77377              Anurag Basu       Ranbir Kapoor
2065   62636  Rakeysh Omprakash Mehra       Farhan Akhtar
2412   33782    Sanjay Leela Bhansali    Amitabh Bachchan
2425   19493            Anurag Kashyap       Kay Kay Menon

                    Actor 2           Actor 3
75                 Madhavan        Mona Singh
173         Naseeruddin Shah   Jimmy Sheirgill
981         Amitabh Bachchan     Sumita Sanyal
1009             Salman Khan    Raveena Tandon
1019                    Tabu      Radhika Apte
1285                  Nassar       Manoj Pahwa
1877  Priyanka Chopra Jonas     Ileana D'Cruz
2065            Sonam Kapoor    Pawan Malhotra
2412            Rani Mukerji     Shernaz Patel
2425         Pawan Malhotra  Aditya Srivastav
```

[ ]:
```
plt.figure(figsize=(10, 6))
ax=sns.
 ↪barplot(data=filtered_df,x='Name',y='Votes',hue='Rating',dodge=False,width=0.
 ↪8,palette='muted')

ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha='right')
ax.legend(loc='upper right')
ax.set_xlabel('Movie Name')
ax.set_ylabel('Votes')
ax.set_title('Movies with rating greater than 8 and votes greater than 10000')

plt.show()
```

```
<ipython-input-86-7820e5098d77>:4: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha='right')
```



[ ]:
```
movie_df['Duration'] = movie_df['Duration'].astype(int)
movie_df['Year'] = movie_df['Year'].astype(int)


plt.figure(figsize=(7, 5))
sns.lineplot(data=movie_df,x='Year',y='Duration',errorbar=None)
plt.xlabel('Year')
plt.ylabel('Duration in minutes')
plt.title('Duration of movies by year')
plt.xticks(np.arange(1917,2023,5))
plt.show()
```



Distribution of Number of movies each genre

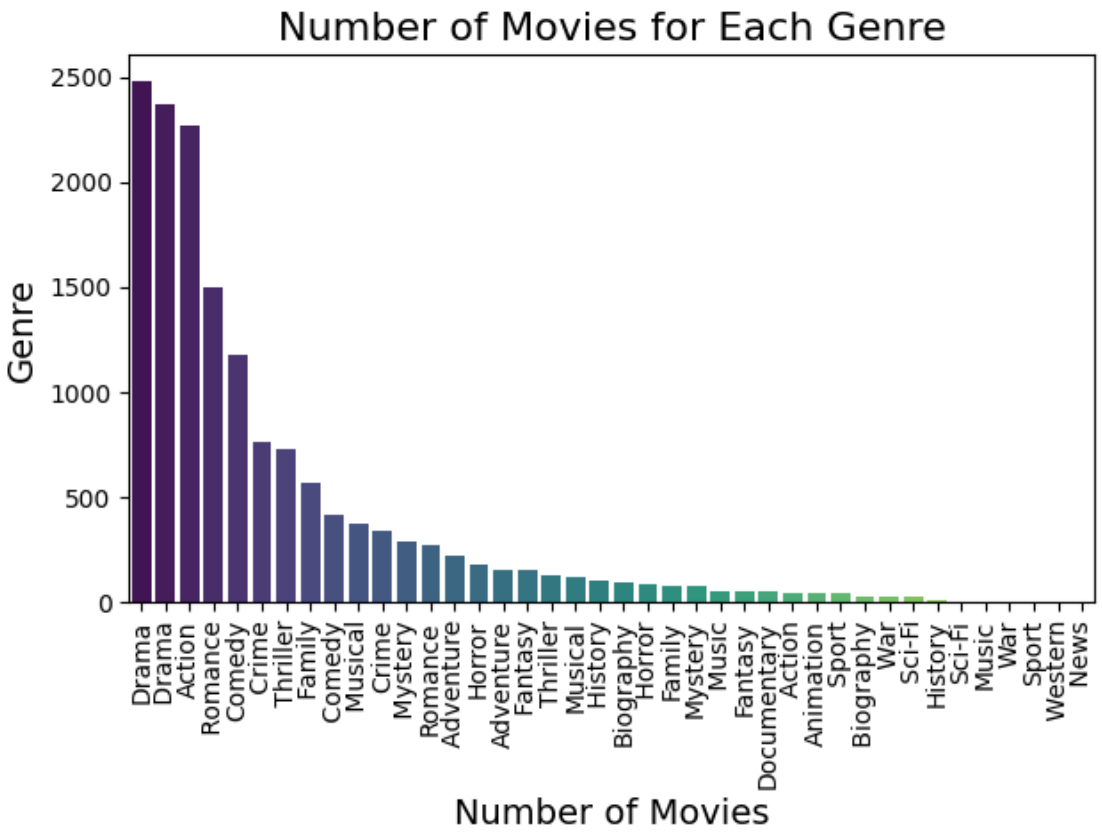
```
[ ]: movie_df['Genre'] = movie_df['Genre'].str.split(',')

     # Create a new DataFrame with one row for each genre
     genre_df = movie_df.explode('Genre')
     genre_df
```

```
[ ]:                                    Name  Year  Duration     Genre  Rating  \
     1      #Gadhvi (He thought he was Gandhi) -2019       109     Drama     7.0
     3                                 #Yaaram -2019       110    Comedy     4.4
     3                                 #Yaaram -2019       110   Romance     4.4
     5                   …Aur Pyaar Ho Gaya -1997       147    Comedy     4.7
     5                   …Aur Pyaar Ho Gaya -1997       147     Drama     4.7
     …                                     …    …        …        …     …
     15504                Zulm Ko Jala Doonga -1988       135    Action     4.6
     15505                             Zulmi -1999       129    Action     4.5
     15505                             Zulmi -1999       129     Drama     4.5
     15508                       Zulm-O-Sitam -1998       130    Action     6.2
     15508                       Zulm-O-Sitam -1998       130     Drama     6.2
```

```
       Votes       Director           Actor 1                 Actor 2  \
1          8  Gaurav Bakshi      Rasika Dugal          Vivek Ghamande
3         35     Ovais Khan           Prateik              Ishita Raj
3         35     Ovais Khan           Prateik              Ishita Raj
5        827   Rahul Rawail        Bobby Deol  Aishwarya Rai Bachchan
5        827   Rahul Rawail        Bobby Deol  Aishwarya Rai Bachchan
…        …            …                 …                       …
15504     11  Mahendra Shah  Naseeruddin Shah           Sumeet Saigal
15505    655     Kuku Kohli      Akshay Kumar          Twinkle Khanna
15505    655     Kuku Kohli      Akshay Kumar          Twinkle Khanna
15508     20   K.C. Bokadia        Dharmendra              Jaya Prada
15508     20   K.C. Bokadia        Dharmendra              Jaya Prada

                Actor 3
1         Arvind Jangid
3       Siddhant Kapoor
3       Siddhant Kapoor
5         Shammi Kapoor
5         Shammi Kapoor
…                   …
15504    Suparna Anand
15505      Aruna Irani
15505      Aruna Irani
15508      Arjun Sarja
15508      Arjun Sarja

[15380 rows x 10 columns]
```

[ ]:
```
plt.figure(figsize=(7, 4))
sns.countplot(data=genre_df, x='Genre', order=genre_df['Genre'].value_counts().
 ↪index, palette='viridis')

plt.title('Number of Movies for Each Genre', fontsize=16)
plt.xlabel('Number of Movies', fontsize=14)
plt.ylabel('Genre', fontsize=14)
plt.xticks(rotation=90)

plt.show()
```

```
<ipython-input-89-862a84832a9e>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(data=genre_df, x='Genre',
order=genre_df['Genre'].value_counts().index, palette='viridis')
```



Distribution of Average rating of movies in each genre

[ ]:
```
average_rating_by_genre = genre_df.groupby('Genre')['Rating'].mean().
  ↪reset_index()
average_rating_by_genre = average_rating_by_genre.sort_values(by='Rating',␣
  ↪ascending=False)
```

[ ]:
```
plt.figure(figsize=(7,4))
sns.
  ↪barplot(data=average_rating_by_genre,y="Rating",x='Genre',palette='coolwarm')
plt.xlabel('Genre')
plt.ylabel('Average Rating')
plt.title('Average rating of movies in each genre')
plt.xticks(rotation=90)

plt.show()
```

```
<ipython-input-91-4c508a015f0a>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
```

effect.

```
  sns.barplot(data=average_rating_by_genre,y="Rating",x='Genre',palette='coolwar
m')
```