aclivtc3o

October 19, 2025

[30]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[31]:
```
df = pd.read_csv('/content/netflix_Project.csv',lineterminator='\n')
```

[32]:
```
df.head()
```

[32]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to…	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.org/t/p/original/1g0dhYtq4i…
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u…	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org/t/p/original/74xTEgt7R3…
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin…	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl…
3	2021-11-24	Encanto	The tale of an extraordinary family, the Madri…	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org/t/p/original/4j0PNHkMr5…
4	2021-12-22	The King's Man	As a collection of history's worst tyrants and…	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org/t/p/original/aq4Pwv5Xeu…

[33]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Release_Date       9827 non-null   object 
 1   Title              9827 non-null   object 
 2   Overview           9827 non-null   object 
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   int64  
 5   Vote_Average       9827 non-null   float64
 6   Original_Language  9827 non-null   object 
 7   Genre              9827 non-null   object 
 8   Poster_Url         9827 non-null   object 
dtypes: float64(2), int64(1), object(6)
memory usage: 691.1+ KB
```

[34]:
```
df.duplicated().sum()
```

[34]:
```
np.int64(0)
```

[35]:
```
df.describe()
```

[35]:

	Popularity	Vote_Count	Vote_Average
count	9827.000000	9827.000000	9827.000000
mean	40.326088	1392.805536	6.439534
std	108.873998	2611.206907	1.129759
min	13.354000	0.000000	0.000000
25%	16.128500	146.000000	5.900000
50%	21.199000	444.000000	6.500000
75%	35.191500	1376.000000	7.100000
max	5083.954000	31077.000000	10.000000

[36]:
```
df['Release_Date']= pd.to_datetime(df['Release_Date'])
```

[37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype         
---  ------             --------------  -----         
 0   Release_Date       9827 non-null   datetime64[ns]
 1   Title              9827 non-null   object        
 2   Overview           9827 non-null   object        
 3   Popularity         9827 non-null   float64
```

```
 4   Vote_Count          9827 non-null   int64
 5   Vote_Average        9827 non-null   float64
 6   Original_Language   9827 non-null   object
 7   Genre               9827 non-null   object
 8   Poster_Url          9827 non-null   object
dtypes: datetime64[ns](1), float64(2), int64(1), object(5)
memory usage: 691.1+ KB
```

[38]:
```
df['Release_Date']=df['Release_Date'].dt.year
```

[39]:
```
df.head()
```

[39]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
0	2021	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to…	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.org/t/p/original/1g0dhYtq4i…
1	2022	The Batman	In his second year of fighting crime, Batman u…	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org/t/p/original/74xTEgt7R3…
2	2022	No Exit	Stranded at a rest stop in the mountains durin…	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl…
3	2021	Encanto	The tale of an extraordinary family, the Madri…	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org/t/p/original/4j0PNHkMr5…
4	2021	The King's Man	As a collection of history's worst tyrants and…	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org/t/p/original/aq4Pwv5Xeu…

[40]:
```
df.drop(['Overview','Original_Language','Poster_Url'],axis=1,inplace=True)
```

[41]:
```
df.head()
```

[41]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	8.1	Crime, Mystery, Thriller

```
2          2022                  No Exit    2618.087         122
3          2021                  Encanto    2402.201        5076
4          2021          The King's Man    1895.511        1793

   Vote_Average                                Genre
0           8.3  Action, Adventure, Science Fiction
1           8.1           Crime, Mystery, Thriller
2           6.3                            Thriller
3           7.7  Animation, Comedy, Family, Fantasy
4           7.0    Action, Adventure, Thriller, War
```

[42]:
```
def cotigorise_col(df,col,labels):
  edge_description = df[col].describe()
  edge = [edge_description['min'],
          edge_description['25%'],
          edge_description['50%'],
          edge_description['75%'],
          edge_description['max']]

  df[col]=pd.cut(df[col],bins=edge,labels=labels,duplicates='drop')
  return df
```

categorizing **Vote_Average** column

We would cut the **Vote_Average** values and make 4 categories: **popular**, **average**, **below_avg**, **not_popular** to describe it more using **catigorize_col()** function provided above.

[43]:
```
labels = ['not_popular', 'below_avg', 'average', 'popular']
cotigorise_col(df, 'Vote_Average', labels)
df.head()
```

[43]:
```
   Release_Date                  Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2022               The Batman    3827.658        1151      popular
2          2022                  No Exit    2618.087         122    below_avg
3          2021                  Encanto    2402.201        5076      popular
4          2021          The King's Man    1895.511        1793      average

                               Genre
0  Action, Adventure, Science Fiction
1           Crime, Mystery, Thriller
2                           Thriller
3  Animation, Comedy, Family, Fantasy
4    Action, Adventure, Thriller, War
```

[44]:
```
df['Vote_Average'].value_counts()
```

[44]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

[45]:
```
df.dropna(inplace=True)
df.isna().sum()
```

[45]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

[46]:
```
df['Genre']=df['Genre'].str.split(', ')

df=df.explode('Genre').reset_index(drop=True)
df.head()
```

[46]:
```
   Release_Date                  Title  Popularity  Vote_Count Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940      popular
1          2021  Spider-Man: No Way Home    5083.954        8940      popular
2          2021  Spider-Man: No Way Home    5083.954        8940      popular
3          2022               The Batman    3827.658        1151      popular
4          2022               The Batman    3827.658        1151      popular

            Genre
0          Action
1       Adventure
2  Science Fiction
3           Crime
4         Mystery
```

[47]:
```
# casting column into category

df['Genre']=df['Genre'].astype('category')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
```

```
 0   Release_Date  25552 non-null  int32
 1   Title         25552 non-null  object
 2   Popularity    25552 non-null  float64
 3   Vote_Count    25552 non-null  int64
 4   Vote_Average  25552 non-null  category
 5   Genre         25552 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(1)
memory usage: 749.6+ KB
```

[48]:
```
df.nunique()
```

[48]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

1 DATA visualization

[49]:
```
sns.set_style('whitegrid')
```

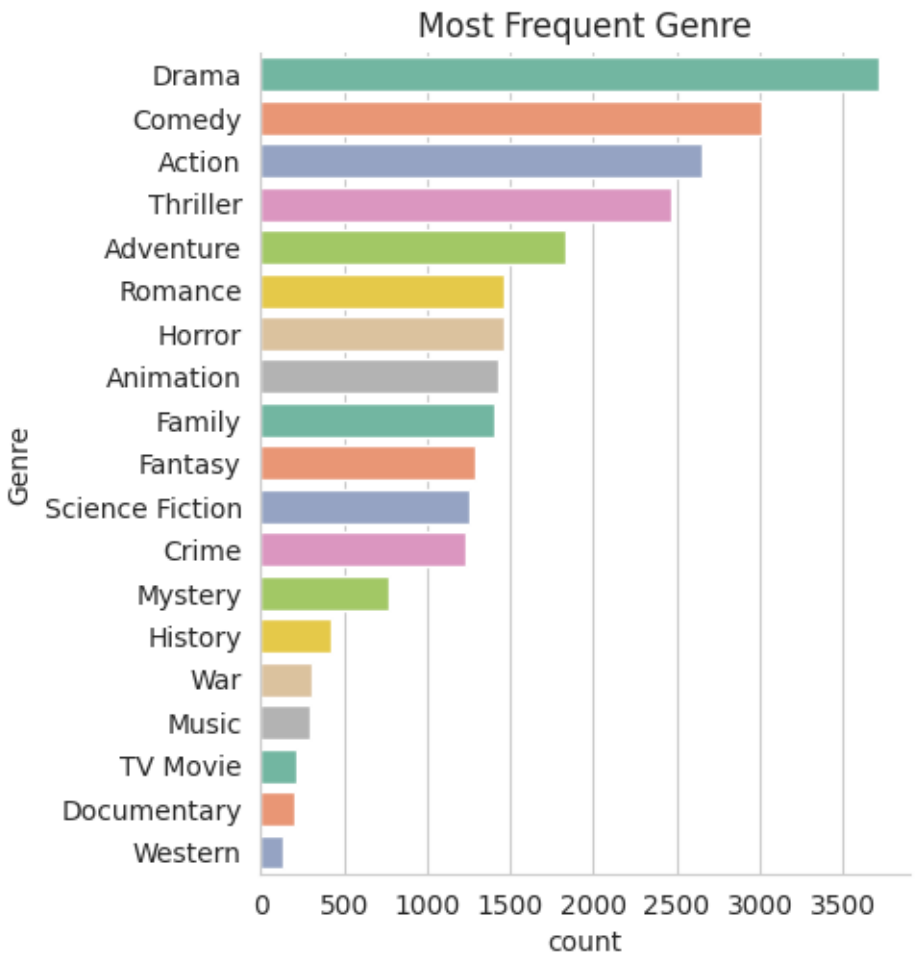
#1. What is the most frequent genre of movies released on Netflix?

[50]:
```
df['Genre'].describe()
```

[50]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

[51]:
```
sns.catplot(y='Genre',kind='count',data=df,order=df['Genre'].value_counts().
  ↪index,palette='Set2')
plt.title('Most Frequent Genre')
plt.show()
```

```
/tmp/ipython-input-3150560572.py:1: 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.catplot(y='Genre',kind='count',data=df,order=df['Genre'].value_counts().in
dex,palette='Set2')
```



#2. Which has highest votes in vote avg column?

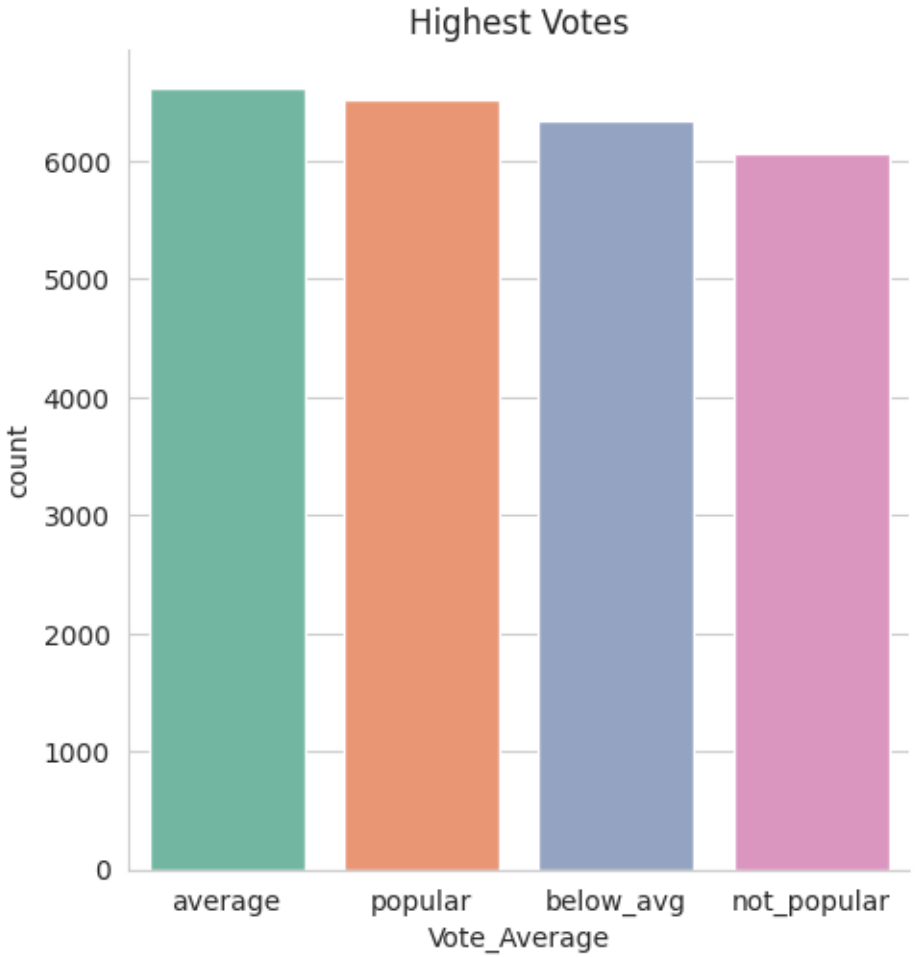
[52]:
```
df['Vote_Average'].describe()
```

[52]:
```
count       25552
unique          4
top       average
freq         6613
Name: Vote_Average, dtype: object
```

[53]:
```
sns.catplot(x='Vote_Average',kind='count',data=df,order=df['Vote_Average'].
  ↪value_counts().index,palette='Set2')
plt.title('Highest Votes')
plt.show()
```

```
/tmp/ipython-input-2773569032.py:1: 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.catplot(x='Vote_Average',kind='count',data=df,order=df['Vote_Average'].val
ue_counts().index,palette='Set2')
```



2 3. What movie got the highest popularity? what's its genre?

[54]:
```
df[df['Popularity']==df['Popularity'].max()]
```

[54]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	\
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	

```
             Genre
0           Action
1        Adventure
2  Science Fiction
```

#4. What movie got the lowest popularity? what's its genre?

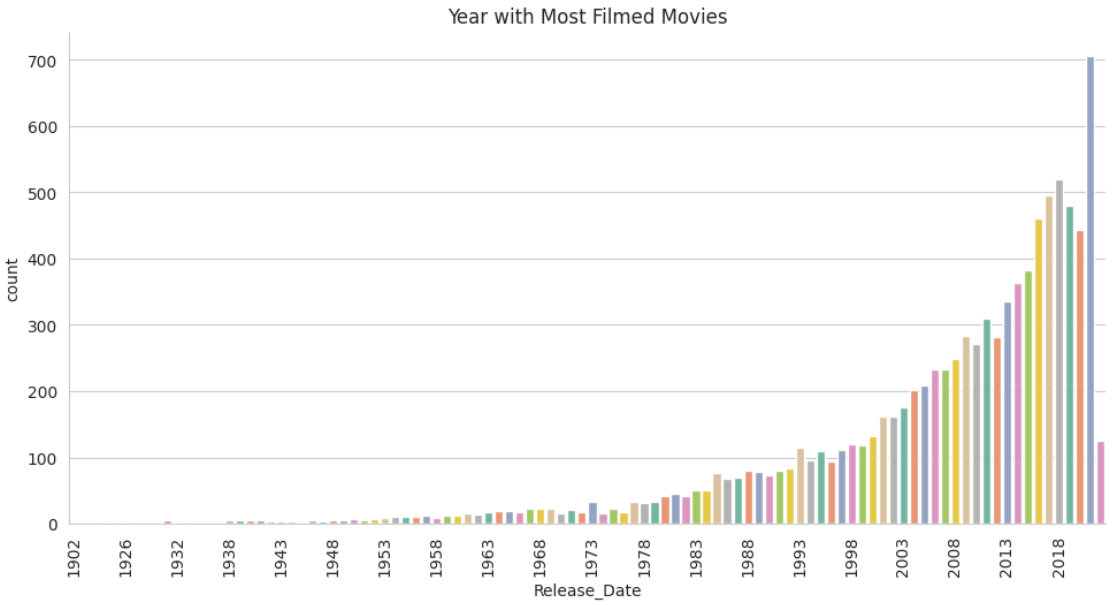
[55]:
```
df[df['Popularity']==df['Popularity'].min()]
```

[55]:
```
       Release_Date                                Title  Popularity  \
25546          2021  The United States vs. Billie Holiday      13.354
25547          2021  The United States vs. Billie Holiday      13.354
25548          2021  The United States vs. Billie Holiday      13.354
25549          1984                               Threads      13.354
25550          1984                               Threads      13.354
25551          1984                               Threads      13.354

       Vote_Count Vote_Average            Genre
25546         152      average            Music
25547         152      average            Drama
25548         152      average          History
25549         186      popular              War
25550         186      popular            Drama
25551         186      popular  Science Fiction
```

#5. Which year has the most filmmed movies?

[56]:
```
def yearBYmovie(df):
    unique_movies = df.drop_duplicates(subset='Title', keep='first')
    return unique_movies[['Title', 'Release_Date']]

Myear=yearBYmovie(df)

order = sorted(Myear['Release_Date'].unique())

g= sns.catplot(
    x='Release_Date',
    kind='count',
    data=Myear,
    order=order,
    palette='Set2',
    height=5, aspect=2
)
g.ax.set_xticklabels([label.get_text() if i%5==0 else '' for i,label in␣
 ↪enumerate(g.ax.get_xticklabels())])
g.set_xticklabels(rotation=90)
plt.title('Year with Most Filmed Movies')
plt.show()
```

```
/tmp/ipython-input-1034976008.py:9: 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.

  g= sns.catplot(
/tmp/ipython-input-1034976008.py:17: UserWarning: set_ticklabels() should only
be used with a fixed number of ticks, i.e. after set_ticks() or using a
FixedLocator.
  g.ax.set_xticklabels([label.get_text() if i%5==0 else '' for i,label in
enumerate(g.ax.get_xticklabels())])
```



2.1 Conclusion

1. **Most Frequent Genre:** The most frequent genre of movies in this dataset is **Drama**.
2. **Distribution of Vote Average:** ” The top category of votes is average
3. **Highest and Lowest Popularity:** The movie with the highest popularity is **Spider-Man: No Way Home**, which belongs to the [Action, Adventure, and Science Fiction] genres.
4. **Highest and Lowest Popularity:** The movies with the lowest popularity include **The United States vs. Billie Holiday** (Music, Drama, History) and **Threads** (War, Drama, Science Fiction).
5. **Year with Most Filmed Movies:** The histogram shows that the year with the most filmed movies is **2021**.

[56]: