

```
# importing lib.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Authenticate with GitHub
from google.colab import auth
auth.authenticate_user()

# Set up Git (replace with your details)
!git config --global user.name "Prateek Kumar Yadav"
!git config --global user.email "prateekkumary73@gmail.com"

# Clone the repository (if you haven't already)
!git clone https://github.com/prateekkumary/netflix-movies-analysis.git

# Change directory to the repository
%cd netflix-movies-analysis

# Add your changes (this assumes you modified analyze_netflix_movies.ipynb)
!git add /content/analyze_netflix_movies.ipynb

# Commit the changes with a message
!git commit -m "Import necessary libraries: numpy, pandas, matplotlib, and seaborn for analysis and visualization in analyze

# Push the changes to GitHub
!git push https://github.com/prateekkumary/netflix-movies-analysis.git
```

```
Cloning into 'netflix-movies-analysis'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
/content/netflix-movies-analysis
fatal: /content/analyze_netflix_movies.ipynb: '/content/analyze_netflix_movies.ipynb' is outside repository at '/content
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
fatal: could not read Username for 'https://github.com': No such device or address
```

```
df = pd.read_csv('mymoviedb.csv', lineterminator=
'\n')
df.head()
```

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
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.tmbd.org
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.tmbd.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmbd.org/t/4
			The tale of an					Animation,	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
# viewing dataset info
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

```

• looks like our dataset has no NaNs! • Overview, Original_Language and Poster_Url wouldn't be so useful during analysis • Release_Date column needs to be casted into date time and to extract only the year value

```

# exploring genres column
df['Genre'].head()

```

```

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

dtype: object

```

genres are saperated by commas followed by whitespaces.

```

# check for duplicated rows
df.duplicated().sum()

```

```

↗ np.int64(0)

```

our dataset has no duplicated rows either.

```

# exploring summary statistics
df.describe()

```

```

↗
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

```

📊 Exploratory Summary

💡 Popularity Mean: 40.33 — indicates the average popularity score.

Median (50%): 21.20 — most movies are moderately popular.

Standard Deviation: 108.87 — very high variability, suggesting some extremely popular movies are skewing the data.

Min–Max: 13.35 to 5083.95 — wide range, with a few outliers having very high popularity.

Observation: Popularity is highly right-skewed due to some blockbuster titles.

💡 Vote Count

Mean: 1392.81 — average number of votes per movie.

Median: 444 — half of the movies have fewer than 444 votes.

Standard Deviation: 2611.21 — very high spread.

Min–Max: 0 to 31,077 — some movies have no votes, while others are extremely popular.

Observation: Like popularity, vote count is heavily skewed with a few movies dominating the count.

◆ Vote Average

Mean: 6.44 — average user rating is fairly high.

Median: 6.5 — the typical movie scores between 6 and 7.

Standard Deviation: 1.13 — low variability, showing most movies cluster around the mean.

Min–Max: 0 to 10 — covers the full rating spectrum.

Observation: Vote average is fairly normally distributed, centered around 6–7, indicating consistent user ratings.

📌 Conclusion

Popularity and Vote Count are highly skewed, indicating a few movies dominate attention.


Vote Average is more evenly distributed, suggesting viewers rate movies consistently, regardless of their popularity.

Ideal for plotting histograms and box plots to visualize outliers and skewness.

Data Cleaning

Casting Release_Date column and extracing year values

df.head()



	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
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.tmbd.org
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.tmbd.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmbd.org/t/
			The tale of an					Animation,	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)


```
# casting column a
df['Release_Date'] = pd.to_datetime(df['Release_Date'])
# confirming changes
print(df['Release_Date'].dtypes)

datetime64[ns]

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
```

df.head()



	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
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.tmbd.org
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.tmbd.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmbd.org/t/
			The tale of an					Animation,	


Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

Dropping Overview, Original_Language and Poster-Url

```
# making list of column to be dropped
cols = ['Overview', 'Original_Language', 'Poster_Url']
# dropping columns and confirming changes
df.drop(cols, axis = 1, inplace = True)
df.columns

Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
      'Genre'],
      dtype='object')

df.head()
```



	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
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.tmbd.org
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.tmbd.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmbd.org/t/
			The tale of an					Animation,	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

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.

Double-click (or enter) to edit

```
def catigorize_col (df, col, labels):
    edges = [df[col].describe()['min'],
             df[col].describe()['25%'],
             df[col].describe()['50%'],
             df[col].describe()['75%'],
             df[col].describe()['max']]

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

# define labels for edges
```

```
labels = ['not_popular', 'below_avg', 'average', 'popular']
# categorize column based on labels and edges
catigorize_col(df, 'Vote_Average', labels)
# confirming changes
df['Vote_Average'].unique()

['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

```
df.head()
```

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	popular	en	Action, Adventure, Science Fiction	https://image.tmdb.org
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	popular	en	Crime, Mystery, Thriller	https://image.tmdb.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	below_avg	en	Thriller	https://image.tmdb.org/t/f
			The tale of an					Animation,	

Next steps:

Generate code with df

View recommended plots

New interactive sheet

```
# exploring column
df['Vote_Average'].value_counts()
```

	count
Vote_Average	
not_popular	2467
popular	2450
average	2412
below_avg	2398

dtype: int64

```
# dropping NaNs
df.dropna(inplace = True)
# confirming
df.isna().sum()
```

	0
Release_Date	0
Title	0
Overview	0
Popularity	0
Vote_Count	0
Vote_Average	0
Original_Language	0
Genre	0
Poster_Url	0

dtype: int64

```
df.head()
```

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	popular	en	Action, Adventure, Science Fiction	https://image.tmdb.org
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	popular	en	Crime, Mystery, Thriller	https://image.tmdb.org/t
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	below_avg	en	Thriller	https://image.tmdb.org/t/f
			The tale of an					Animation,	

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

we'd split genres into a list and then explode our dataframe to have only one genre per row for each movie

```
# split the strings into lists
df['Genre'] = df['Genre'].str.split(',')
# explode the lists
df = df.explode('Genre').reset_index(drop=True)
df.head()
```

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	popular	en	Action	https://image.tmdb.org/t/p/o
1	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	popular	en	Adventure	https://image.tmdb.org/t/p/o
			Peter						

Next steps:

[Generate code with df](#)[View recommended plots](#)[New interactive sheet](#)

```
# casting column into category
df['Genre'] = df['Genre'].astype('category')
# confirming changes
df['Genre'].dtypes
```

```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation', 'Comedy', 'Crime',
                             'Documentary', 'Drama', 'Family', 'Fantasy', 'History',
                             'Horror', 'Music', 'Mystery', 'Romance', 'Science Fiction',
                             'TV Movie', 'Thriller', 'War', 'Western'],
                  ordered=False, categories_dtype=object)
```

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Release_Date          25552 non-null  datetime64[ns]
1   Title                 25552 non-null  object
2   Overview              25552 non-null  object
3   Popularity            25552 non-null  float64
4   Vote_Count            25552 non-null  int64
5   Vote_Average          25552 non-null  category
6   Original_Language     25552 non-null  object
7   Genre                 25552 non-null  category
8   Poster_Url            25552 non-null  object
dtypes: category(2), datetime64[ns](1), float64(1), int64(1), object(4)
memory usage: 1.4+ MB
```

df.nunique()

```
↵
```

	0
Release_Date	5846
Title	9415
Overview	9722
Popularity	8088
Vote_Count	3265
Vote_Average	4
Original_Language	42
Genre	19
Poster_Url	9727

dtype: int64

Now that our dataset is clean and tidy, we are left with a total of 6 columns and 25551 rows to dig into during our analysis

Data Visualization

here, we'd use Matplotlib and seaborn for making some informative visuals to gain insights about our data

```
# setting up seaborn configurations
sns.set_style('whitegrid')
```

Q1: What is the most frequent genre in the dataset?

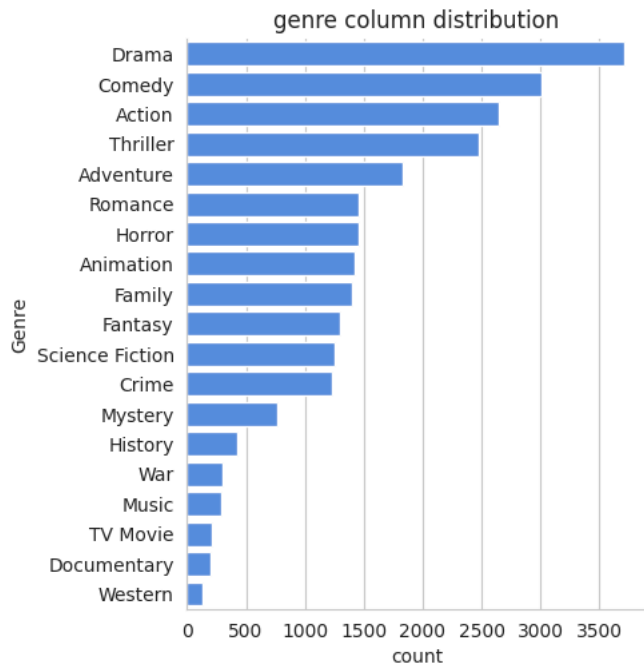
```
# showing stats. on genre column
df['Genre'].describe()
```

```
↵
```

	Genre
count	25552
unique	19
top	Drama
freq	3715

dtype: object

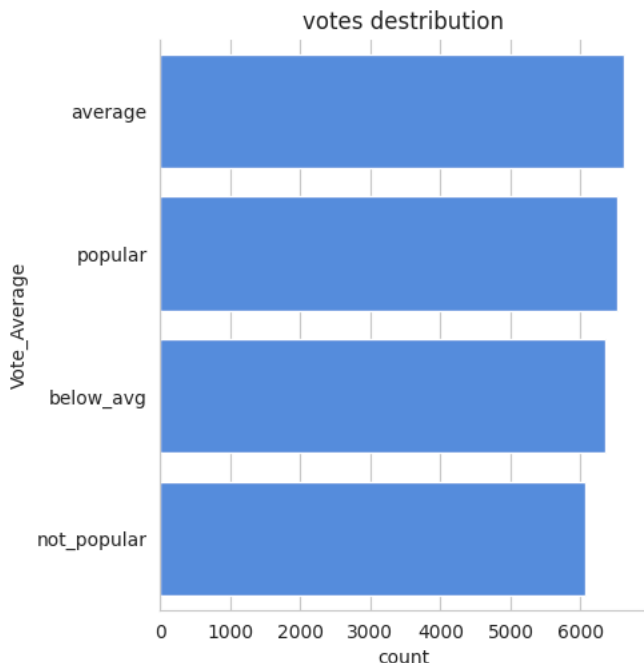
```
# visualizing genre column
sns.catplot(y = 'Genre', data = df, kind = 'count',
order = df['Genre'].value_counts().index,
color = '#4287f5')
plt.title('genre column distribution')
plt.show()
```



- we can notice from the above visual that Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the times among 19 other genres.

Q2: What genres has highest votes?

```
# visualizing vote_average column
sns.catplot(y = 'Vote_Average', data = df, kind = 'count',
order = df['Vote_Average'].value_counts().index,
color = '#4287f5')
plt.title('votes destribution')
plt.show()
```



Q3: What movie got the highest popularity? what's its genre?

```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].max()]
```


	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	popular	en	Action	https://image.tmdb.org/t/p/ori
		Spider-Man: ...	Peter Parker is						

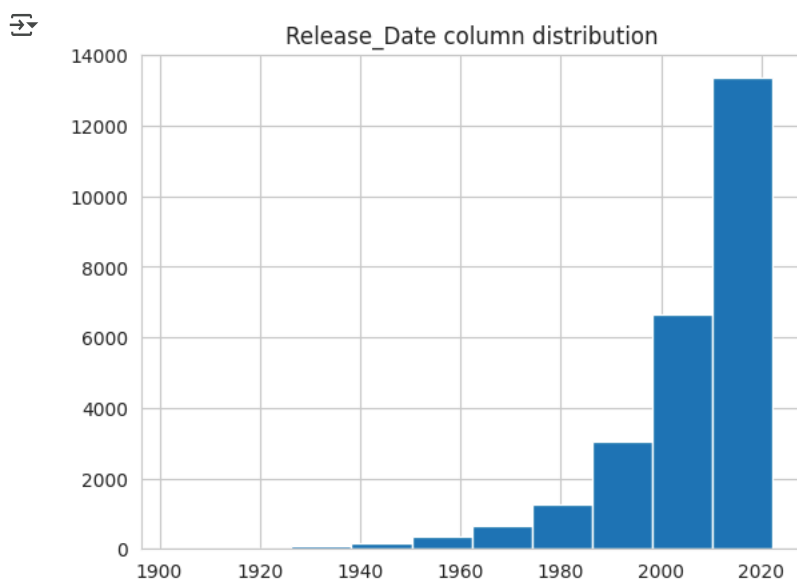
Q4: What movie got the lowest popularity? what's its genre?

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

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
25546	2021-03-31	The United States vs. Billie Holiday	Billie Holiday spent much of her career being ...	13.354	152	average	en	Music	https://image.tmdb.org
25547	2021-03-31	The United States vs. Billie Holiday	Billie Holiday spent much of her career being ...	13.354	152	average	en	Drama	https://image.tmdb.org
25548	2021-03-31	The United States vs. Billie Holiday	Billie Holiday spent much of her career being ...	13.354	152	average	en	History	https://image.tmdb.org
25549	1984-09-23	Threads	Documentary style account of a nuclear holocau...	13.354	186	popular	en	War	https://image.tmdb.org
		Documentary							

Q5: Which year has the most filmed movies?

```
df['Release_Date'].hist()
plt.title('Release_Date column distribution')
plt.show()
```



Conclusion

Q1: What is the most frequent genre in the dataset?

Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the times among 19 other genres.

Q2: What genres has highest votes ?

we have 25.5% of our dataset with popular vote (6520 rows). Drama again gets the highest popularity among fans by being having more than 18.5% of movies popularities.

Q3: What movie got the highest popularity ? what's its genre ?

Spider-Man: No Way Home has the highest popularity rate in our dataset and it has genres of Action , Adventure and Sience Fiction .

Q3: What movie got the lowest popularity ? what's its genre ?

The united states, thread' has the highest lowest rate in our dataset and it has genres of music , drama , 'war', 'sci-fi' and history`.

Q4: Which year has the most filmed movies?

year 2020 has the highest filmming rate in our dataset.



This is the movies analysis project.

This is the movies analysis project.

.....

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.