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

```
df = pd.read_csv("C:\Users\HP\Downloads\netflix1.csv", lineterminator = '\n')
```

df.head()

| | 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/1g0dhYtq4... |
| 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... | 2619.087 | 122 | 6.3 | en | Thriller | https://image.tmdb.org/t/p/original/vDHLHOWH... |
| 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... |

```
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(3), int64(1), object(5)
memory usage: 691.1+ KB
```

```
df['Genre'].head()
```

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

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.describe()
```

| | Popularity | Vote_Count | Vote_Average |
|---|---|---|---|
| count | 9827.000000 | 9827.000000 | 9827.000000 |
| mean | 40.326498 | 1392.805536 | 6.439534 |
| std | 108.873998 | 2417.203927 | 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 |

```
# Exploration Summary
# We have a dataframe consisting of 9827 rows and 9 columns.
# Our dataset looks a bit tidy with neither NaNs nor duplicate values.
# Release_Date column needs to be changed to type datetime and only the year value
# Overview, Original_Language, Poster_Url wouldn't be so useful during analysis, so we'll drop them.
# There is noticable outlier in Popular column.
# Vote_Average column can be categorized for better analysis.
# Genre column has comma separated values and white spaces that needs to be handled and casted into category.
```

df.head()

| | 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/1g0dhYtq4... |
| 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... | 2619.087 | 122 | 6.3 | en | Thriller | https://image.tmdb.org/t/p/original/vDHLHOWH... |
| 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... |

```
df['Release_Date'] = pd.to_datetime(df['Release_Date'])
print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

```
df['Release_Date'] = df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

```
dtype('int32')
```

df.head()

| | 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/1g0dhYtq4... |
| 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... | 2619.087 | 122 | 6.3 | en | Thriller | https://image.tmdb.org/t/p/original/vDHLHOWH... |
| 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... |

Dropping the columns

```
cols = ['Overview','Original_Language','Poster_Url']

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 | 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 | 2619.087 | 122 | 6.3 | Thriller |
| 3 | 2021 | Encanto | 2402.201 | 5076 | 7.7 | Animation, Comedy, Family, Fantasy |
| 4 | 2021 | The King's Man | 1895.511 | 1793 | 7.0 | Action, Adventure, Thriller, War |

Categorized Vote_Average column.

```
### We will use the Vote_Average values and make four categories popular, average, below_average and not_popular. We use categorize_col() function for this.

# Expects pandas as df

def categorize_col(df, lnb, labels):
    mean = df.loc[:, lnb].describe()
    edges = [mean['min'], mean['25%'], mean['50%'], mean['75%'], mean['max']]
    df[lnb] = pd.cut(df[lnb], edges, labels=labels, duplicates='drop', include_lowest=True)
    return df

labels = ['not_popular', 'below_avg', 'average', 'popular']

# Call the function
df = categorize_col(df, 'Vote_Average', labels)

# Check unique categories
print(df['Vote_Average'].unique())
```

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

df.head()

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action, Adventure, Science Fiction |
| 1 | 2022 | The Batman | 3827.658 | 1151 | popular | Crime, Mystery, Thriller |
| 2 | 2022 | No Exit | 2619.087 | 122 | below_avg | Thriller |
| 3 | 2021 | Encanto | 2402.201 | 5076 | popular | Animation, Comedy, Family, Fantasy |
| 4 | 2021 | The King's Man | 1895.511 | 1793 | average | Action, Adventure, Thriller, War |

```
df["Vote_Average"].value_counts()
```

```
Vote_Average
popular        2467
average        2412
below_avg      2412
not_popular    2412
Name: count, dtype: int64
```

```
df.dropna(inplace = True)
df.isnull().sum()
```

```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

df.head()

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action, Adventure, Science Fiction |
| 1 | 2022 | The Batman | 3827.658 | 1151 | popular | Crime, Mystery, Thriller |
| 2 | 2022 | No Exit | 2619.087 | 122 | below_avg | Thriller |
| 3 | 2021 | Encanto | 2402.201 | 5076 | popular | Animation, Comedy, Family, Fantasy |
| 4 | 2021 | The King's Man | 1895.511 | 1793 | average | Action, Adventure, Thriller, War |

Splitting genres into a list and then obtaining DataFrame to only have one genre per row for each movie.

```
df['Genre'] = df['Genre'].str.split(',')
df = df.explode('Genre').reset_index(drop=True)
df.head()
```

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action |
| 1 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Adventure |
| 2 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Science Fiction |
| 3 | 2022 | The Batman | 3827.658 | 1151 | popular | Crime |
| 4 | 2022 | The Batman | 3827.658 | 1151 | popular | Mystery |

```
# Converting column into category

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

```
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: 25753 entries, 0 to 25752
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Release_Date  25753 non-null  int32
 1   Title         25753 non-null  object
 2   Popularity    25753 non-null  float64
 3   Vote_Count    25753 non-null  int64
 4   Vote_Average  25753 non-null  category
 5   Genre         25753 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(1)
memory usage: 756.7+ KB
```

```
df.nunique()
```

```
Release_Date     102
Title           9513
Popularity      8160
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

df.head()

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action |
| 1 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Adventure |
| 2 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Science Fiction |
| 3 | 2022 | The Batman | 3827.658 | 1151 | popular | Crime |
| 4 | 2022 | The Batman | 3827.658 | 1151 | popular | Mystery |

```
sns.set_style('whitegrid')
```

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

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

```
count     25753
unique       19
top       Drama
freq       3744
Name: Genre, dtype: object
```

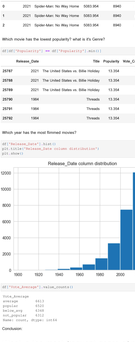
```
sns.catplot(y = 'Genre', data = df, kind = 'count',
            order = df['Genre'].value_counts().index,
            color = '#4287f5')
plt.title('Genre column distribution')
plt.show()
```


which category has highest count in Vote_Average column?

df.head()

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action |
| 1 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Adventure |
| 2 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Science Fiction |
| 3 | 2022 | The Batman | 3827.658 | 1151 | popular | Crime |
| 4 | 2022 | The Batman | 3827.658 | 1151 | popular | Mystery |

```
sns.catplot(y = 'Vote_Average', data = df, kind = 'count',
            order = df['Vote_Average'].value_counts().index,
            color = '#4287f5')
plt.title('Category Distribution')
plt.show()
```




Which movie has the highest popularity? what is it's Genre?

df.head(1)

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action |
| 1 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Adventure |

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

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 0 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Action |
| 1 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Adventure |
| 2 | 2021 | Spider-Man: No Way Home | 5083.954 | 8940 | popular | Science Fiction |

Which movie has the lowest popularity? what is it's Genre?

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

| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| 25767 | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | average | Music |
| 25768 | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | average | Drama |
| 25769 | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | average | History |
| 25790 | 1984 | Threads | 13.354 | 186 | popular | War |
| 25791 | 1984 | Threads | 13.354 | 186 | popular | Drama |
| 25792 | 1984 | Threads | 13.354 | 186 | popular | Science Fiction |

Which year has the most filmmed movies?

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


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

```
Vote_Average
average      6413
popular      6320
below_avg    6320
not_popular  6320
Name: count, dtype: int64
```

Conclusion

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

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

## Which movie has the highest popularity? what is it's Genre?

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

## Which movie has the lowest popularity? what is it's Genre?

The United States vs. Billie Holiday, Threads has the lowest popularity in our dataset.It has genres 'Music, Drama , History','War, Drama, Science Fiction'.

## Which year has the most filmmed movies?

Year 2020 has the highest filming rate in our dataset.

## Which Genre has the highest votes?

We have 25.5% of our dataset with popular votes (6320 rows). Drama again gets the highest popularity among fans by having more than 18.5% of movies popularities.