


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

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/t/p/original/1
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/p/original/74
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/p/original/vDF
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.tmbd.org/t/p/original/4jC
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.tmbd.org/t/p/original/aq4

Next steps:

Generate code with df

 View recommended plots

New interactive sheet

```
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
```

```
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

df.describe()

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



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

```
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	

Change the format of Release Date, remove unnecessary white spaces and get rid of unwanted files

```
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/t/p/original/1
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/p/original/74
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/p/original/vDH
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.tmbd.org/t/p/original/4jC
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.tmbd.org/t/p/original/aq4

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
df['Release_Date'] = pd.to_datetime(df['Release_Date'])
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['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	
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/1
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/74
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/vDf
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/4jC
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/aq4

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

Now, 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	2618.087	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

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

print(df.columns.tolist())

```
['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average', 'Genre']
```

```
def categorize_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
```

```
labels = ['not_popular' , 'below_average' , 'average' , 'popular']
```

```
categorize_col(df , 'Vote_Average' , labels)
```

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

```
['popular', 'below_average', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_average' < '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	2618.087	122	below_average	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

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

df['Vote_Average'].value_counts()

	count
Vote_Average	
not_popular	2467
popular	2450
average	2412
below_average	2398

df['Vote_Average'].value_counts()

df.dropna(inplace = True)

df.isna().sum().sum()

np.int64(0)

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	2618.087	122	below_average	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

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

We will split the genres into lists

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

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

Casting Genre column as category

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

df.nunique()



```
0
Release_Date    100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average     4
Genre           19
```

df: int64

Data Visualisation

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

Most Frequent visited genre

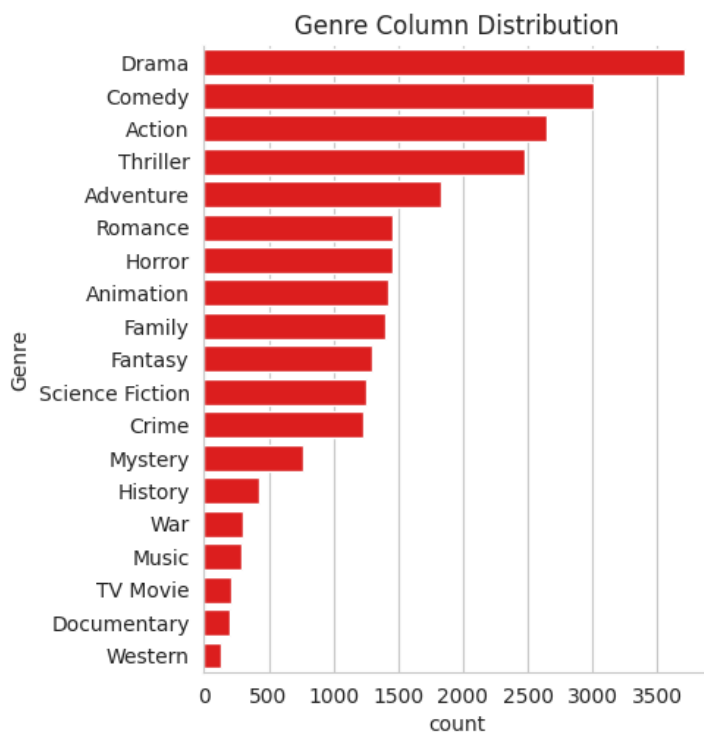
```
df['Genre'].describe()
```



```
Genre
count    25552
unique      19
top      Drama
freq      3715
```

df: object

```
sns.catplot(y = 'Genre' , kind = 'count' , data = df ,
            order = df['Genre'].value_counts().index ,
            color = 'red')
plt.title("Genre Column Distribution")
plt.show()
```



Which genre has highest votes

```
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



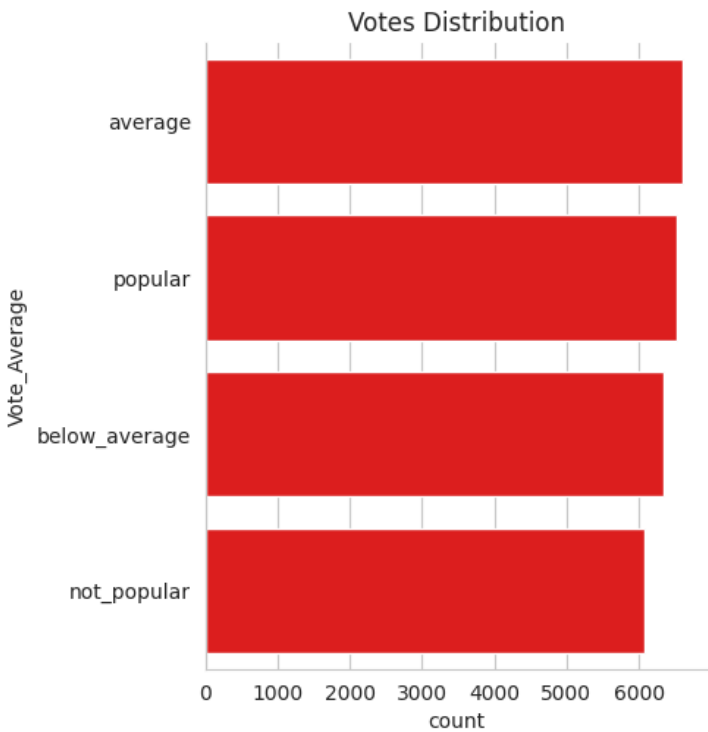
Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
sns.catplot(y = 'Vote_Average' , kind = 'count' , data = df ,
            order = df['Vote_Average'].value_counts().index ,
            color = 'red')
plt.title("Votes Distribution")
plt.show()
```



What movie has highest popularity? What genre?

df.head(2)



	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



Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

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



Lowest Popularity?

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

	Release_Date		Title	Popularity	Vote_Count	Vote_Average	Genre
25546	2021		The United States vs. Billie Holiday	13.354	152	average	Music
25547	2021		The United States vs. Billie Holiday	13.354	152	average	Drama
25548	2021		The United States vs. Billie Holiday	13.354	152	average	History
25549	1984		Threads	13.354	186	popular	War
25550	1984		Threads	13.354	186	popular	Drama
25551	1984		Threads	13.354	186	popular	Science Fiction

Year with Most days to film a movie

```
df['Release_Date'].hist()  
plt.title("Release Date Distribution")  
plt.show()
```

