# TCR Innovation project

DV with Py internship program

project name: Data analysis and visualization on IMDb Movies

Group Member’s Name: Subhash kumar Mahato , Rita Pradhan, Mamta Prajapati

batch\_name: DV with python sept. Batch

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#importing python libraries.

import pandas as pd

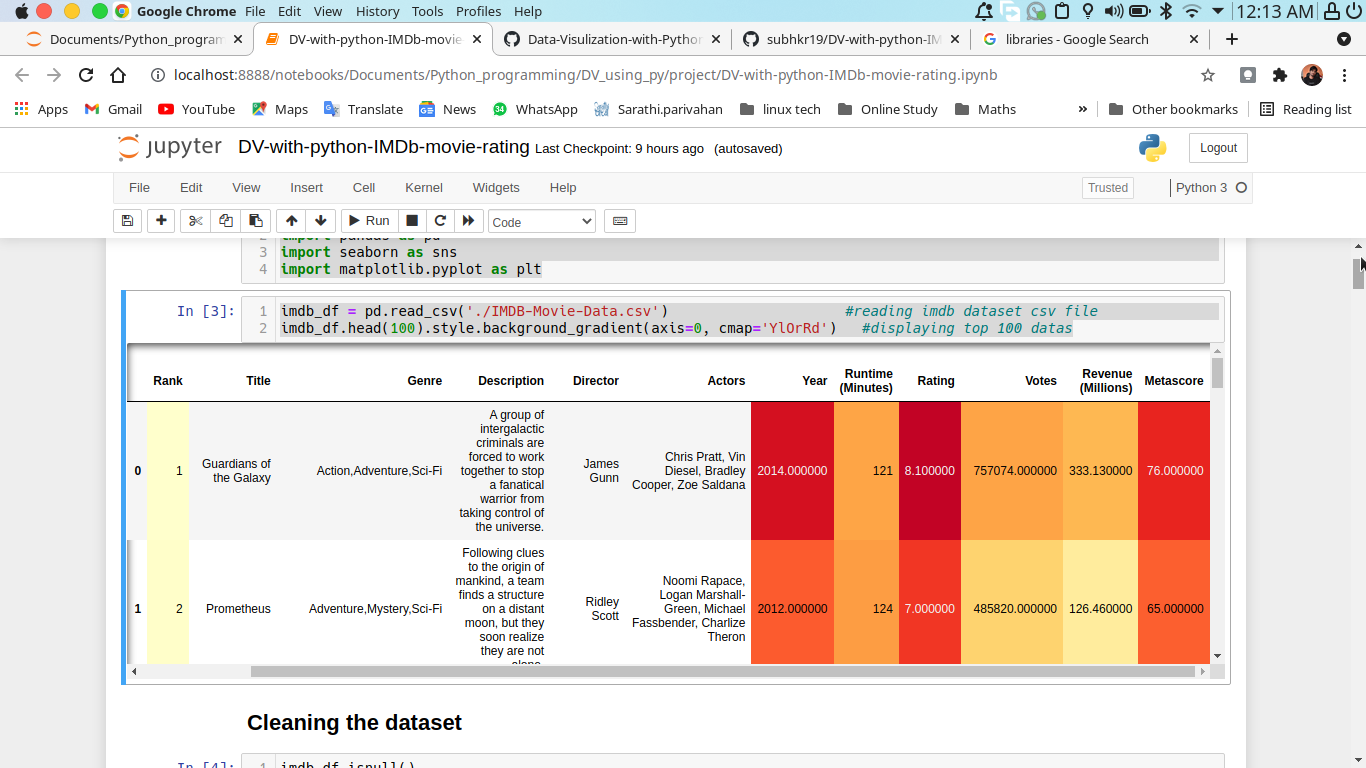
import seaborn as sns

import matplotlib.pyplot as plt

imdb\_df = pd.read\_csv('./IMDB-Movie-Data.csv') #reading imdb dataset csv file

imdb\_df.head(100).style.background\_gradient(axis=0, cmap='YlOrRd') #displaying top 100 data

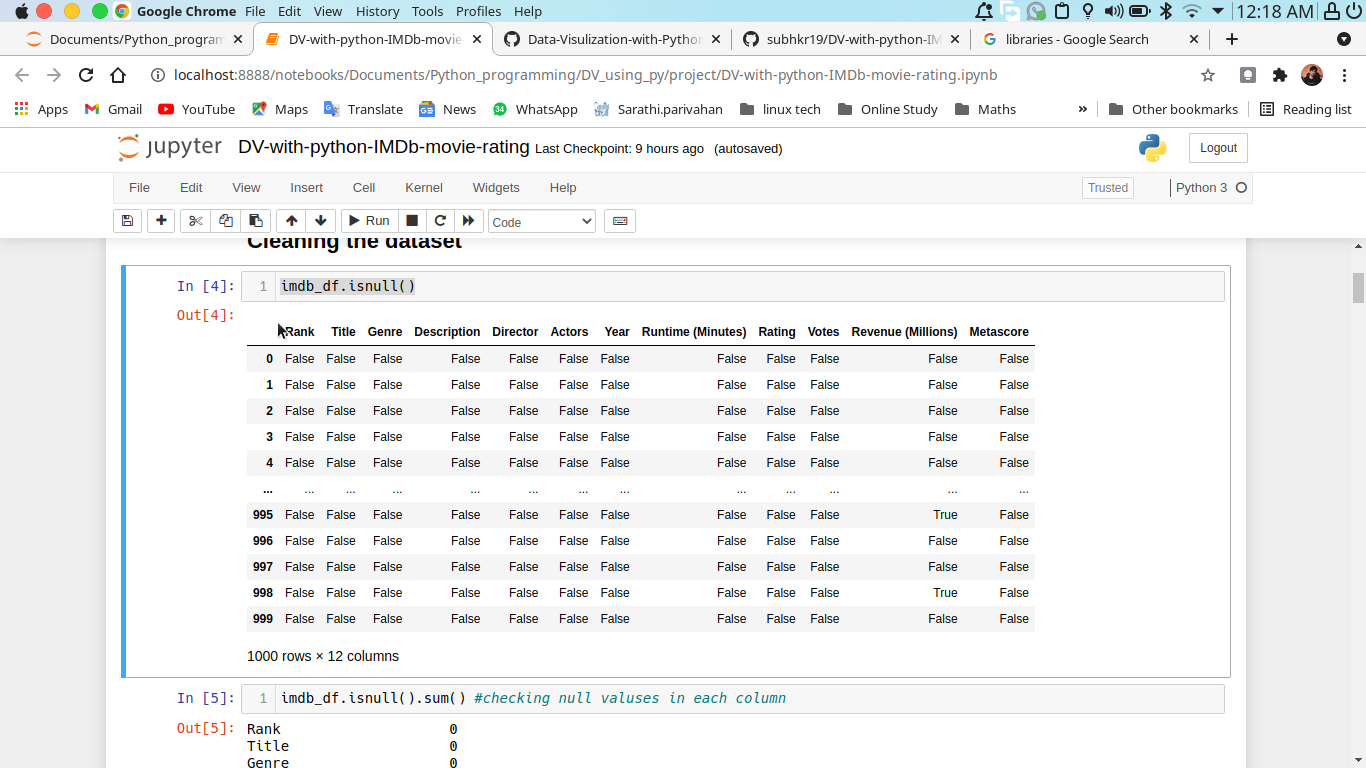
output:



**Cleaning the dataset**

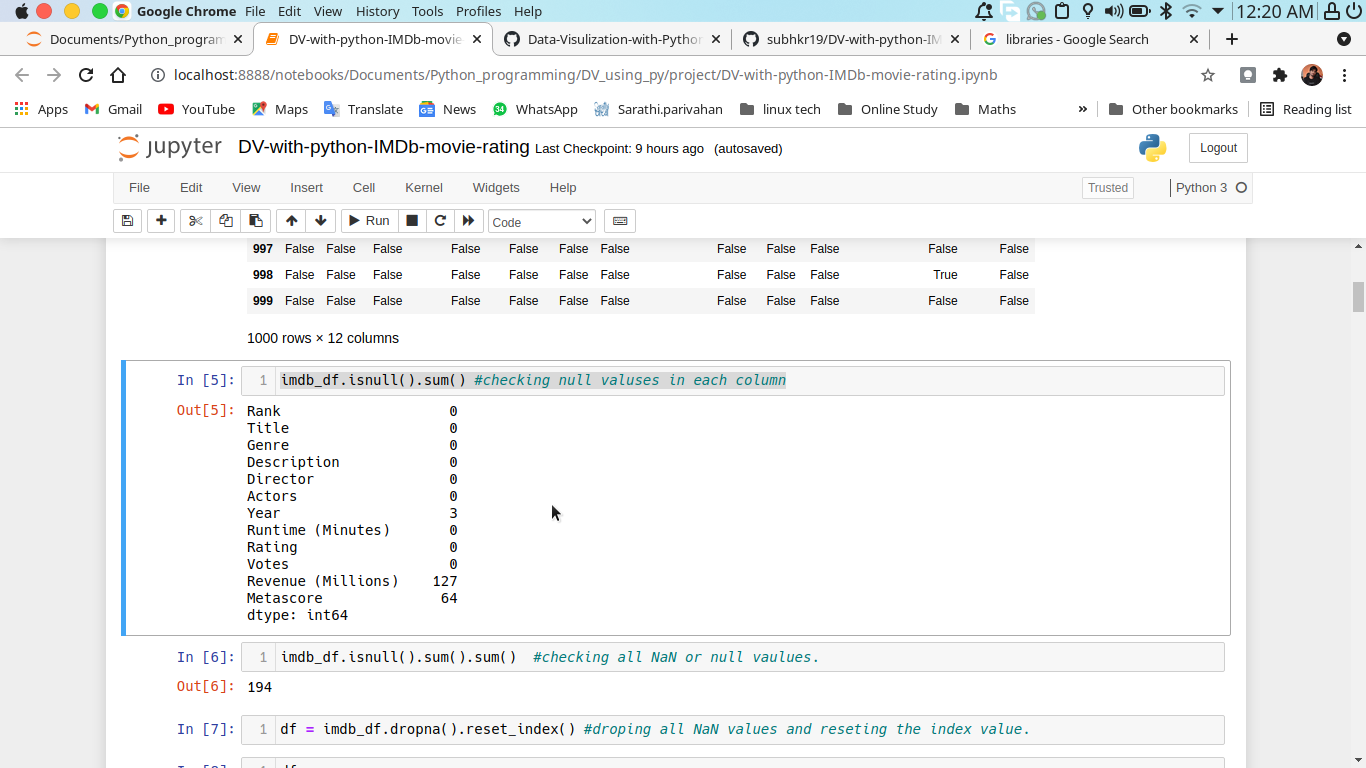
imdb\_df.isnull()

Output:



imdb\_df.isnull().sum() #checking null valuses in each column

Output:



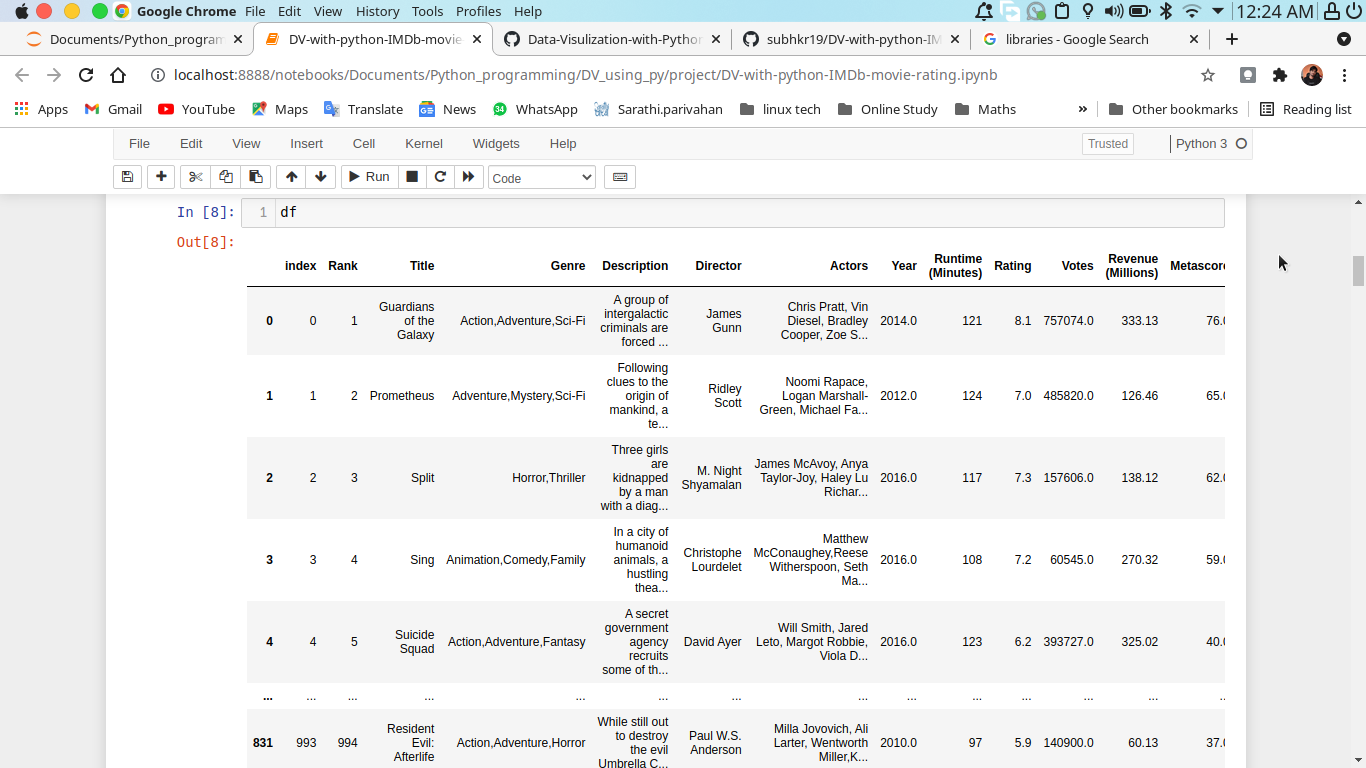
imdb\_df.isnull().sum().sum() #checking all NaN or null vaulues.

Output:

194

df = imdb\_df.dropna().reset\_index() #droping all NaN values and reseting the index value.

Output:

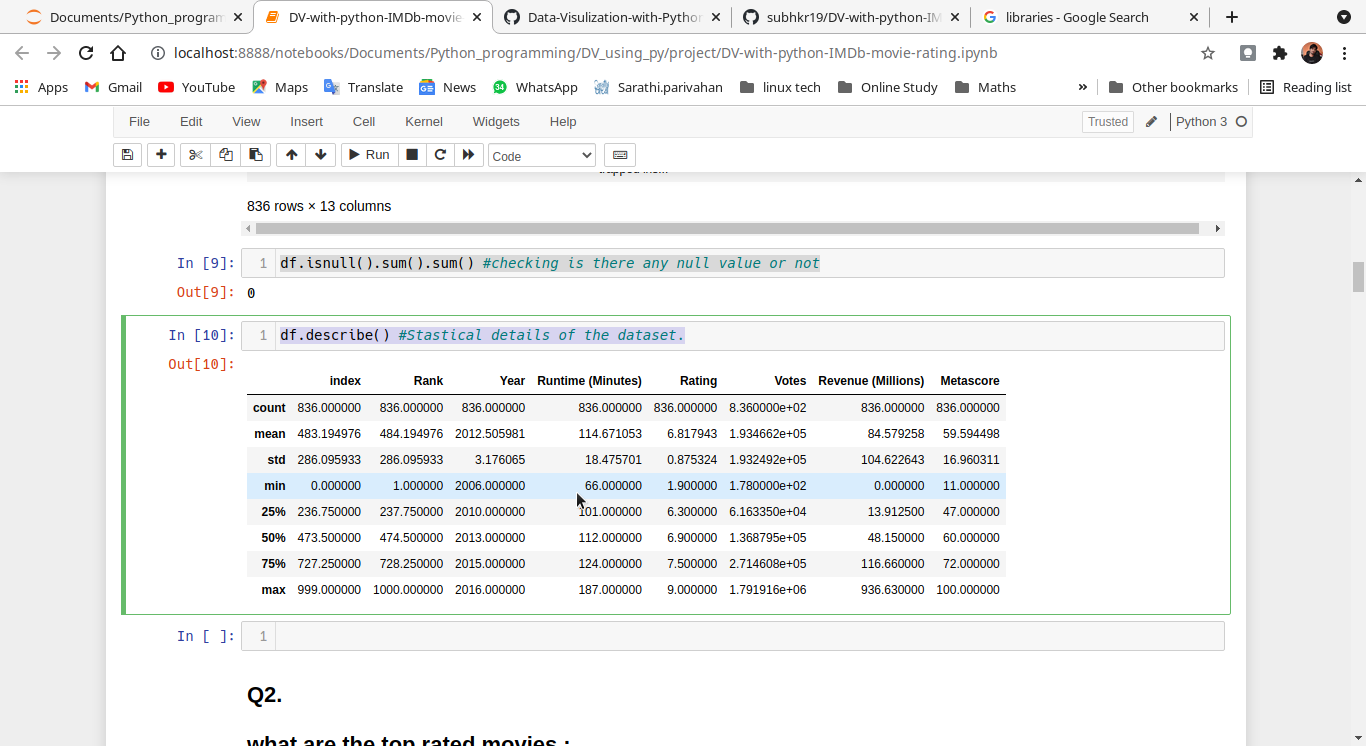


df.isnull().sum().sum() #checking is there any null value or not

Output:

0

df.describe() #Stastical details of the dataset.

Output:

**Q2.**

**what are the top rated movies :**

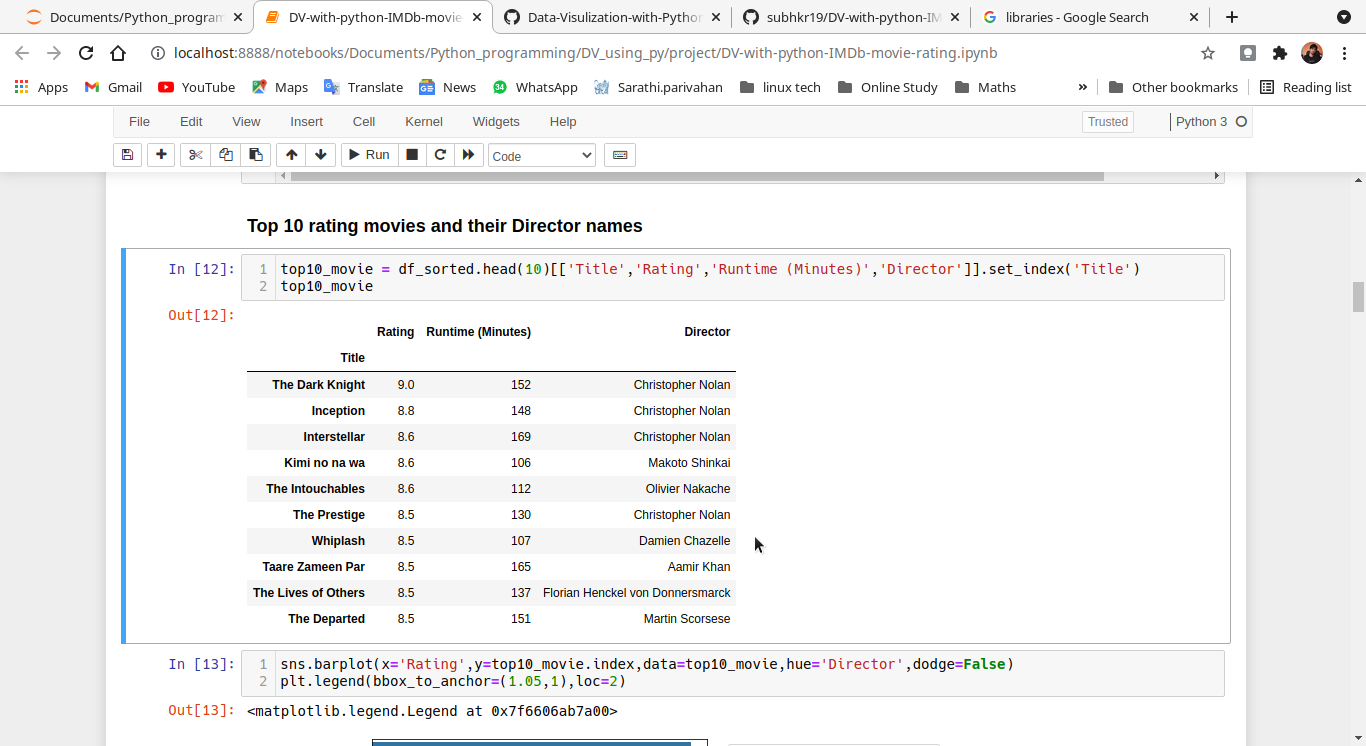
df\_sorted = df.sort\_values('Rating',ascending=False) #sorting the DataFrame in decending order according to the movie's rating

**Top 10 rating movies and their Director names**

top10\_movie = df\_sorted.head(10)[['Title','Rating','Runtime (Minutes)','Director']].set\_index('Title')

top10\_movie

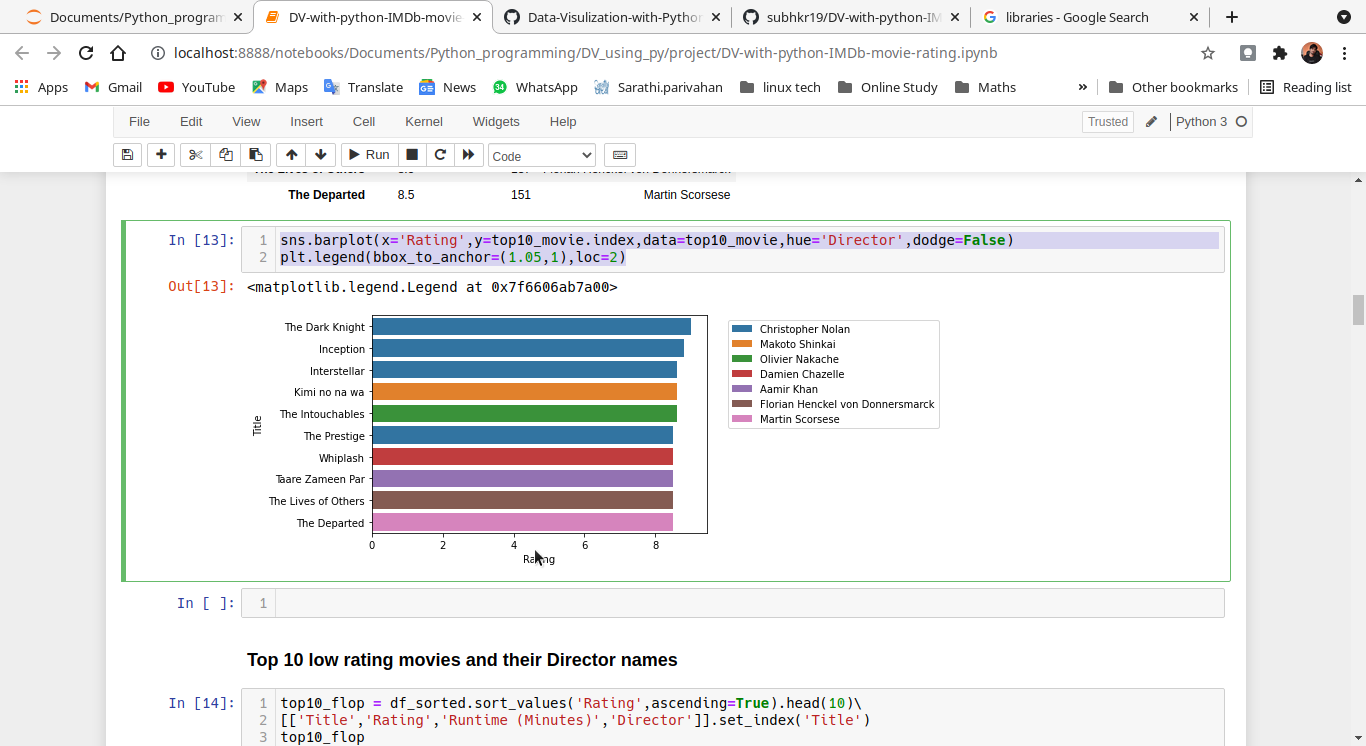
output:



sns.barplot(x='Rating',y=top10\_movie.index,data=top10\_movie,hue='Director',dodge=False)

plt.legend(bbox\_to\_anchor=(1.05,1),loc=2)

output:

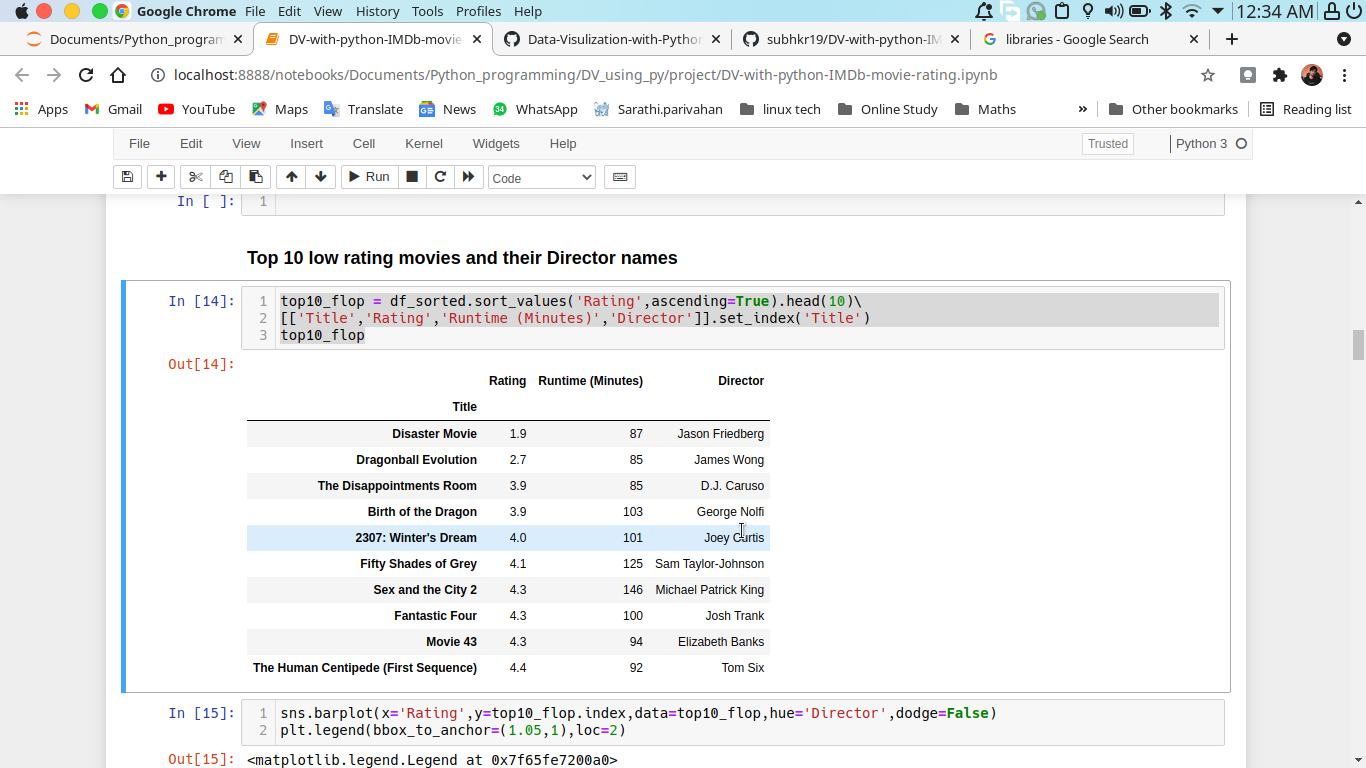


**Top 10 low rating movies and their Director names**

top10\_flop = df\_sorted.sort\_values('Rating',ascending=True).head(10)\

[['Title','Rating','Runtime (Minutes)','Director']].set\_index('Title')

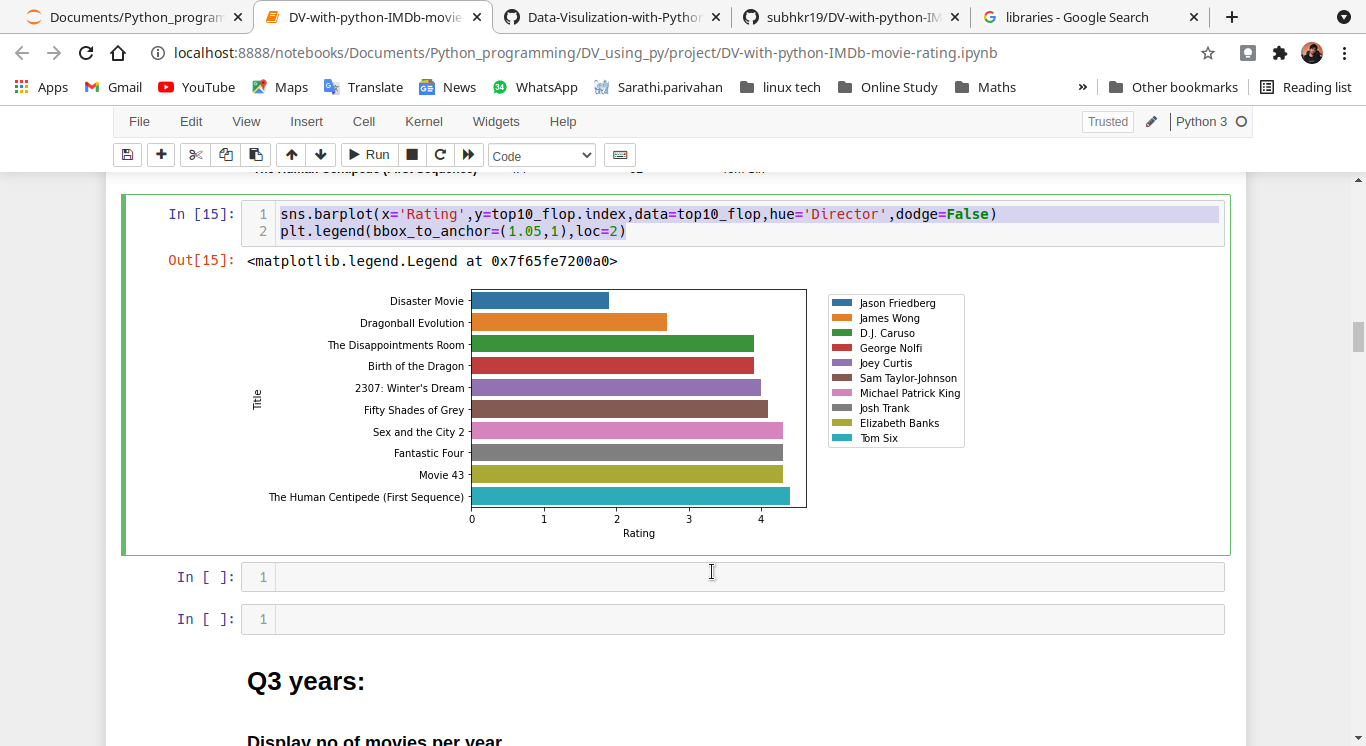
top10\_flop



sns.barplot(x='Rating',y=top10\_flop.index,data=top10\_flop,hue='Director',dodge=False)

plt.legend(bbox\_to\_anchor=(1.05,1),loc=2)

Output:



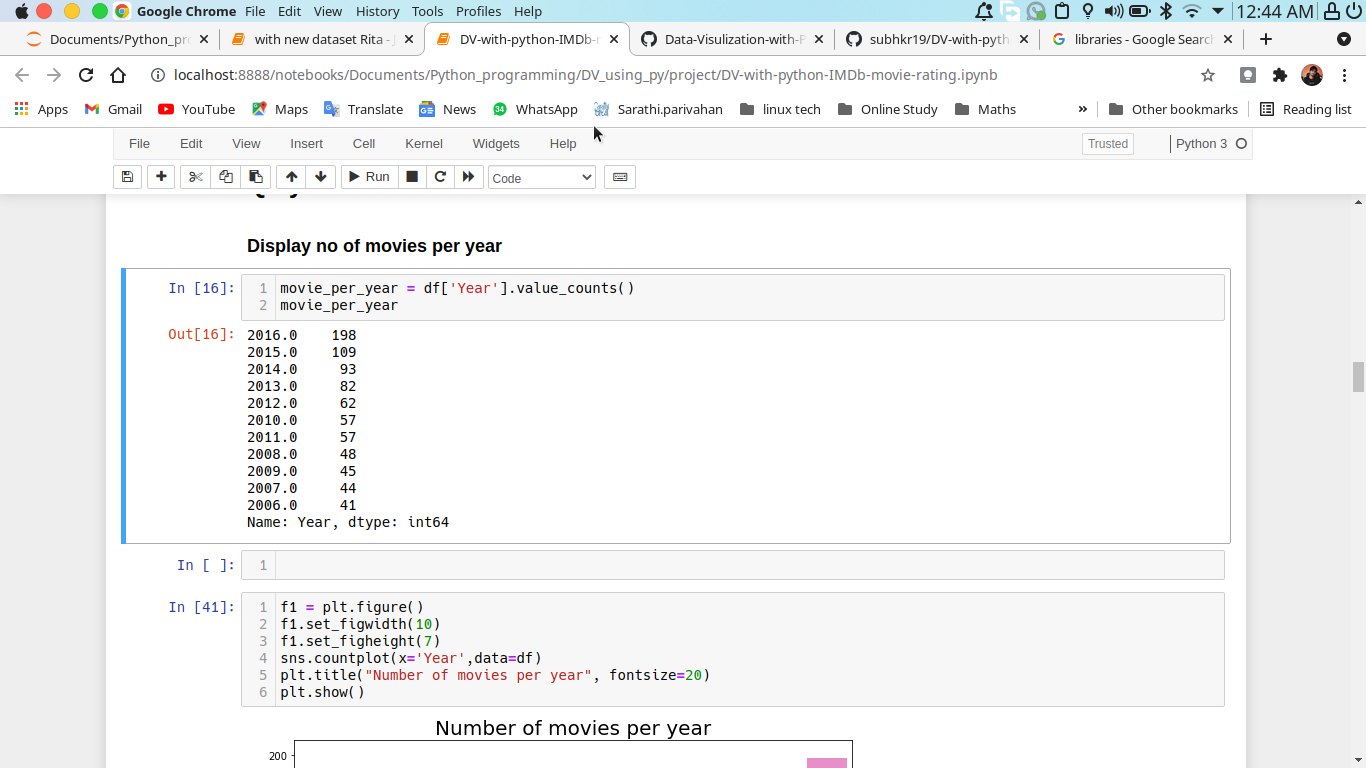
**Q3 years:**

**Display no of movies per year**

movie\_per\_year = df['Year'].value\_counts()

movie\_per\_year

Output:



f1 = plt.figure()

f1.set\_figwidth(10)

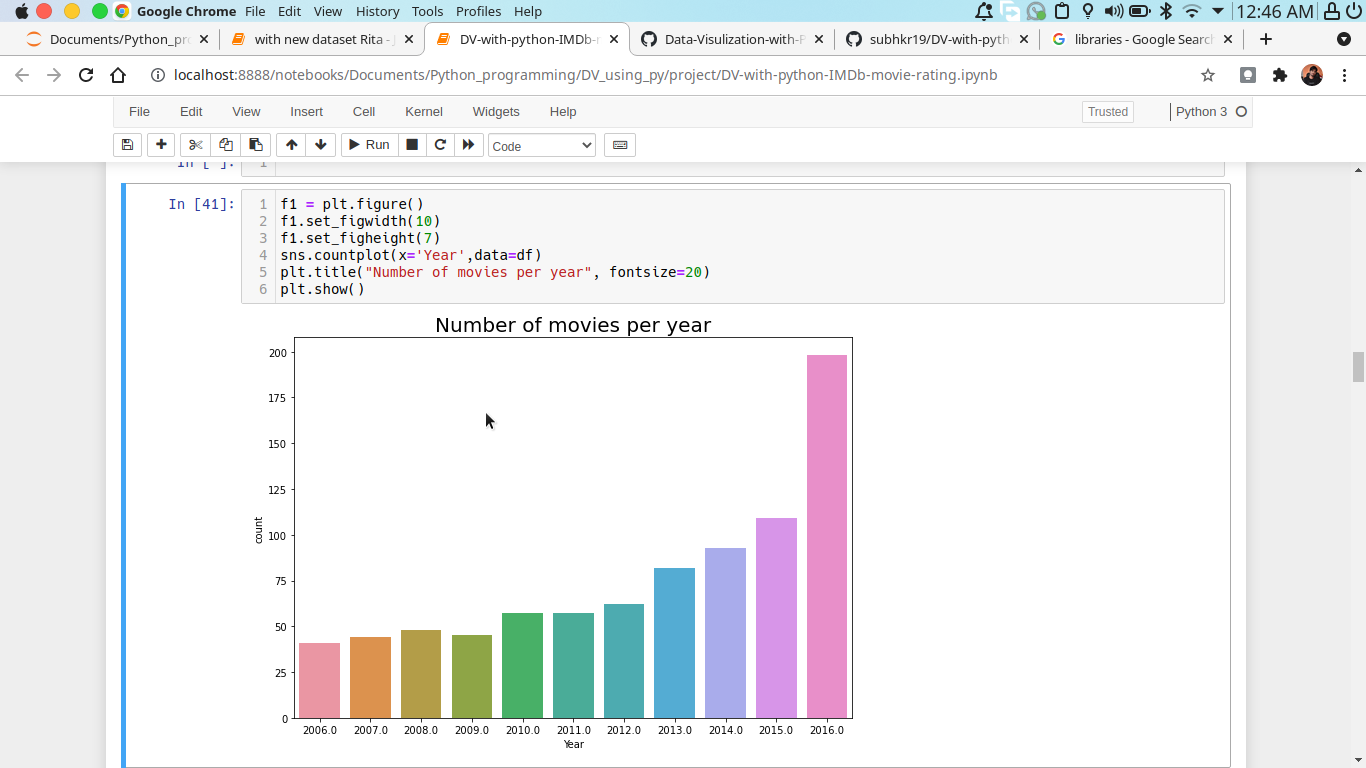
f1.set\_figheight(7)

sns.countplot(x='Year',data=df)

plt.title("Number of movies per year", fontsize=20)

plt.show()

output:



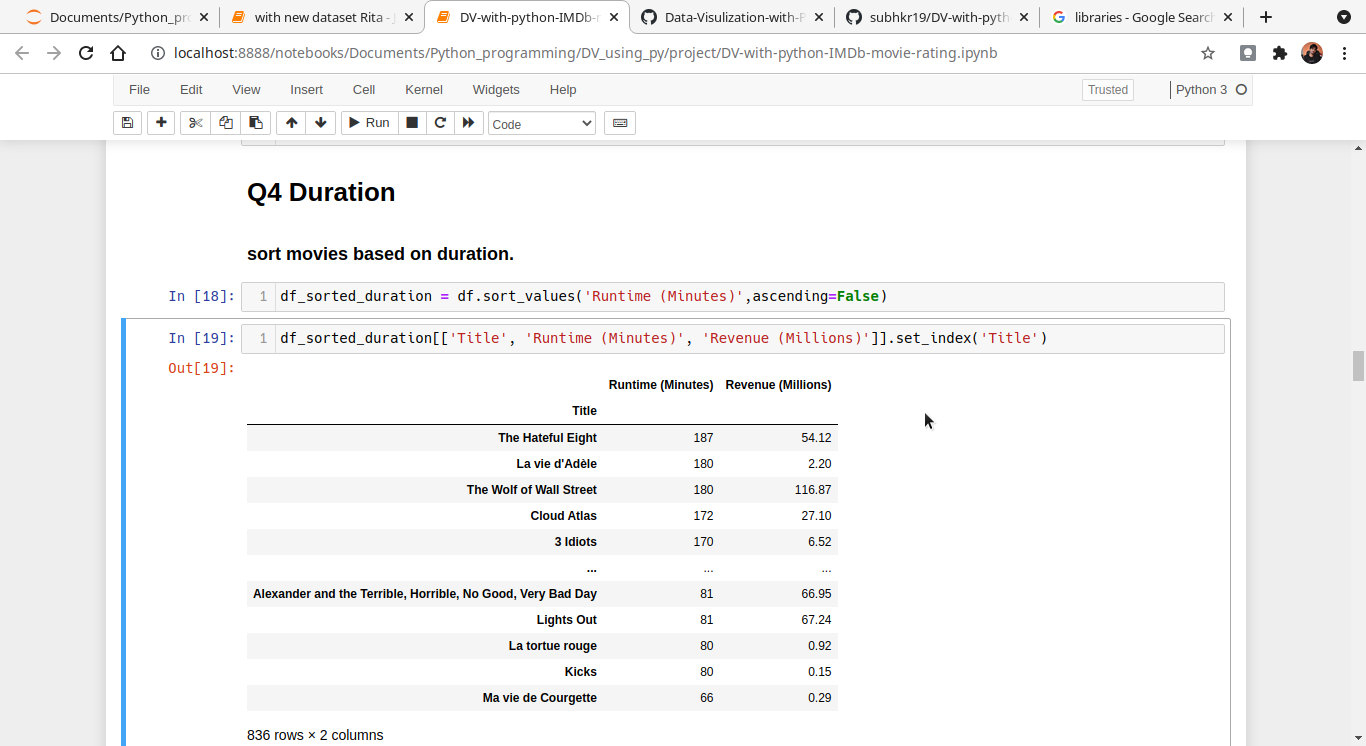
**Q4 Duration**

**sort movies based on duration.**

df\_sorted\_duration = df.sort\_values('Runtime (Minutes)',ascending=False)

df\_sorted\_duration[['Title', 'Runtime (Minutes)', 'Revenue (Millions)']].set\_index('Title')

output:

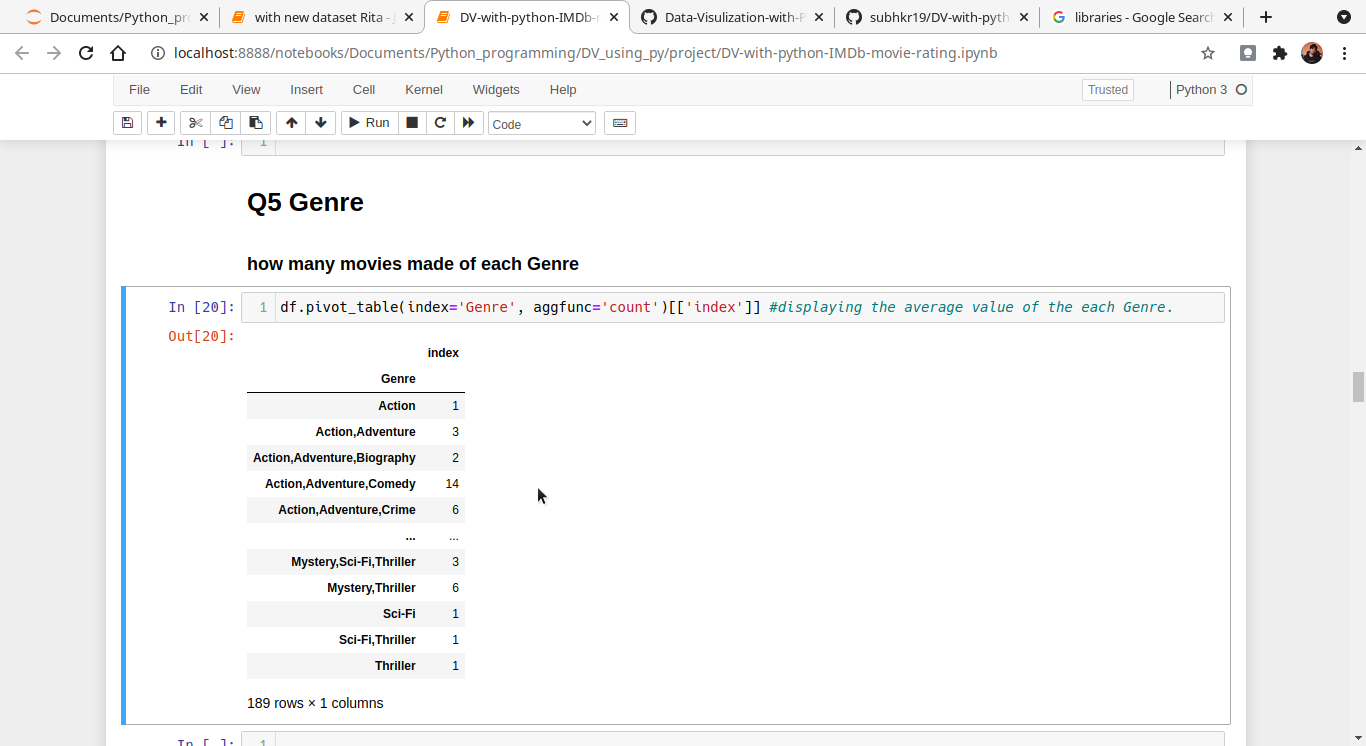


**Q5 Genre**

**how many movies made of each Genre**

df.pivot\_table(index='Genre', aggfunc='count')[['index']] #displaying the average value of the each Genre.

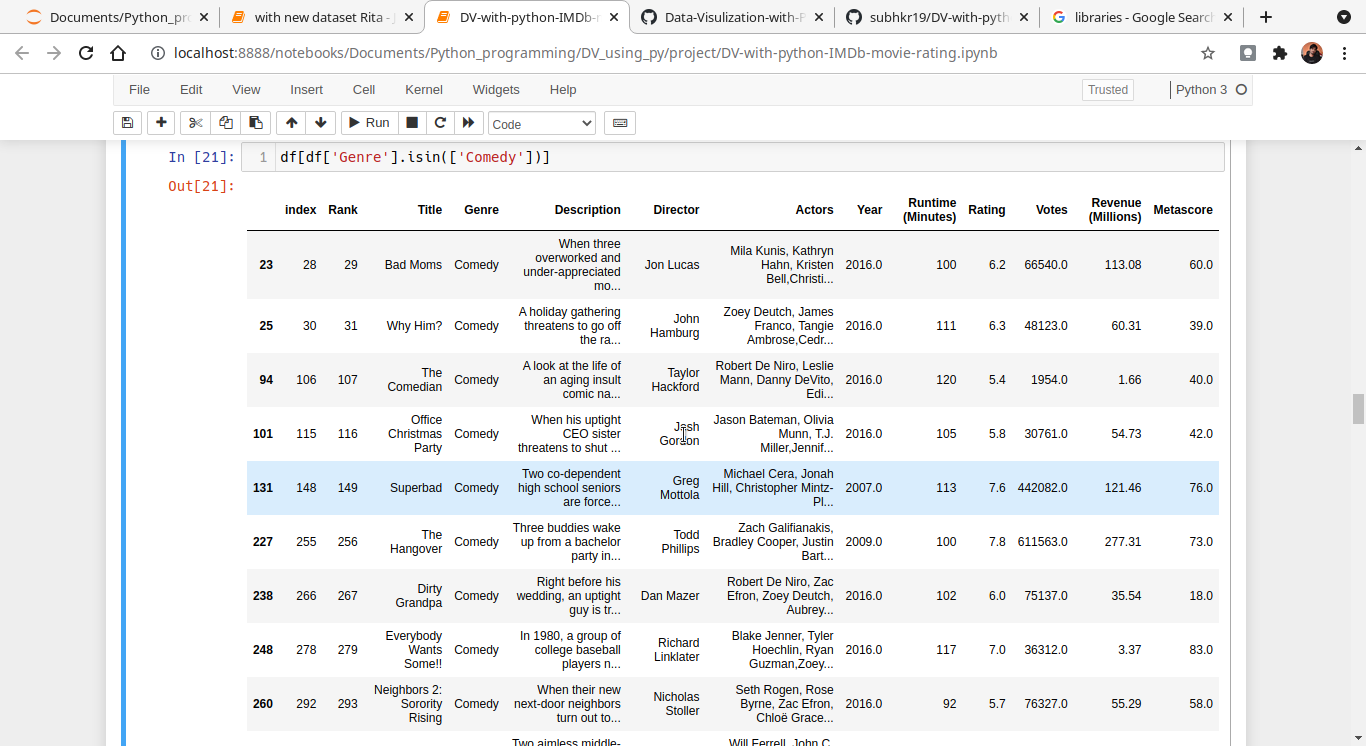
Output:



**Display movie details made of Genre is equal to comedy**

df[df['Genre'].isin(['Comedy'])]

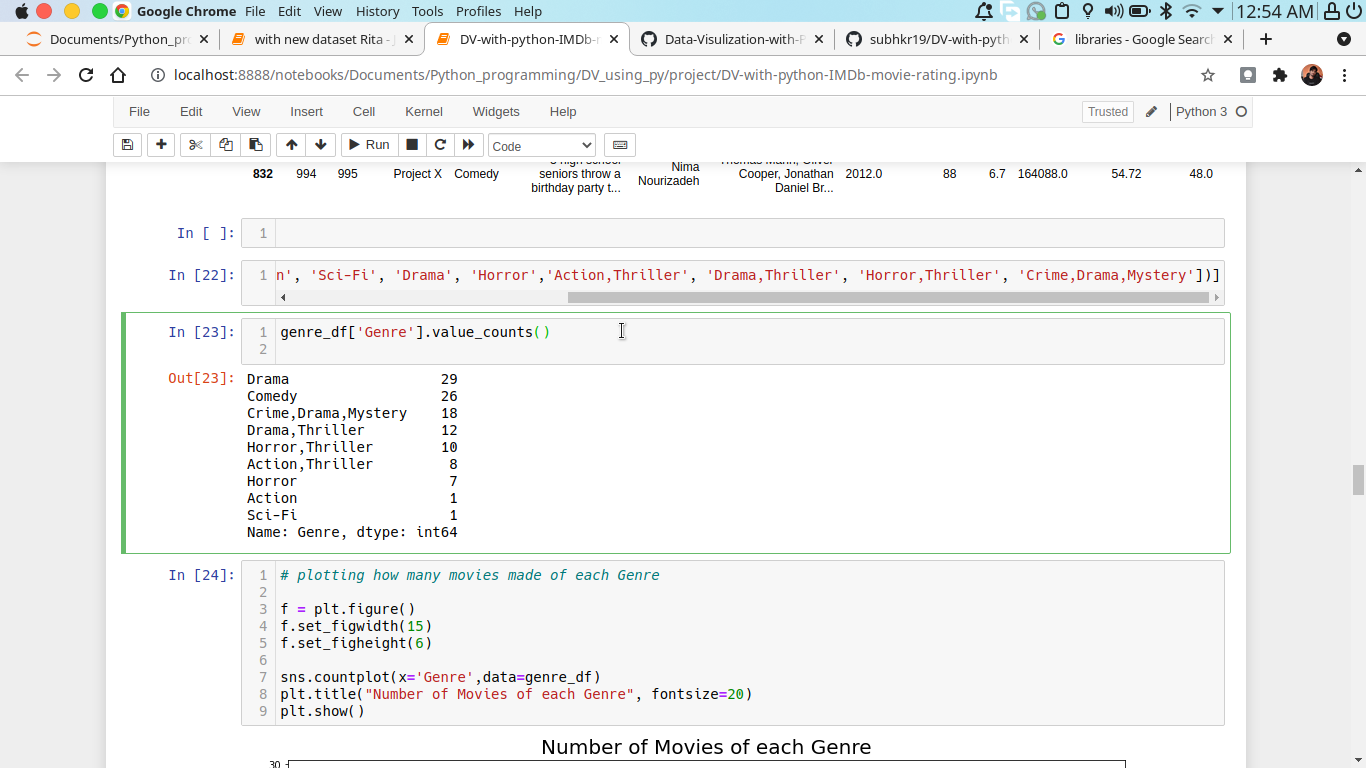
output:



genre\_df = df[df['Genre'].isin(['Comedy', 'Action', 'Sci-Fi', 'Drama', 'Horror','Action,Thriller', 'Drama,Thriller', 'Horror,Thriller', 'Crime,Drama,Mystery'])]

genre\_df['Genre'].value\_counts()

output:



# plotting how many movies made of each Genre

f = plt.figure()

f.set\_figwidth(15)

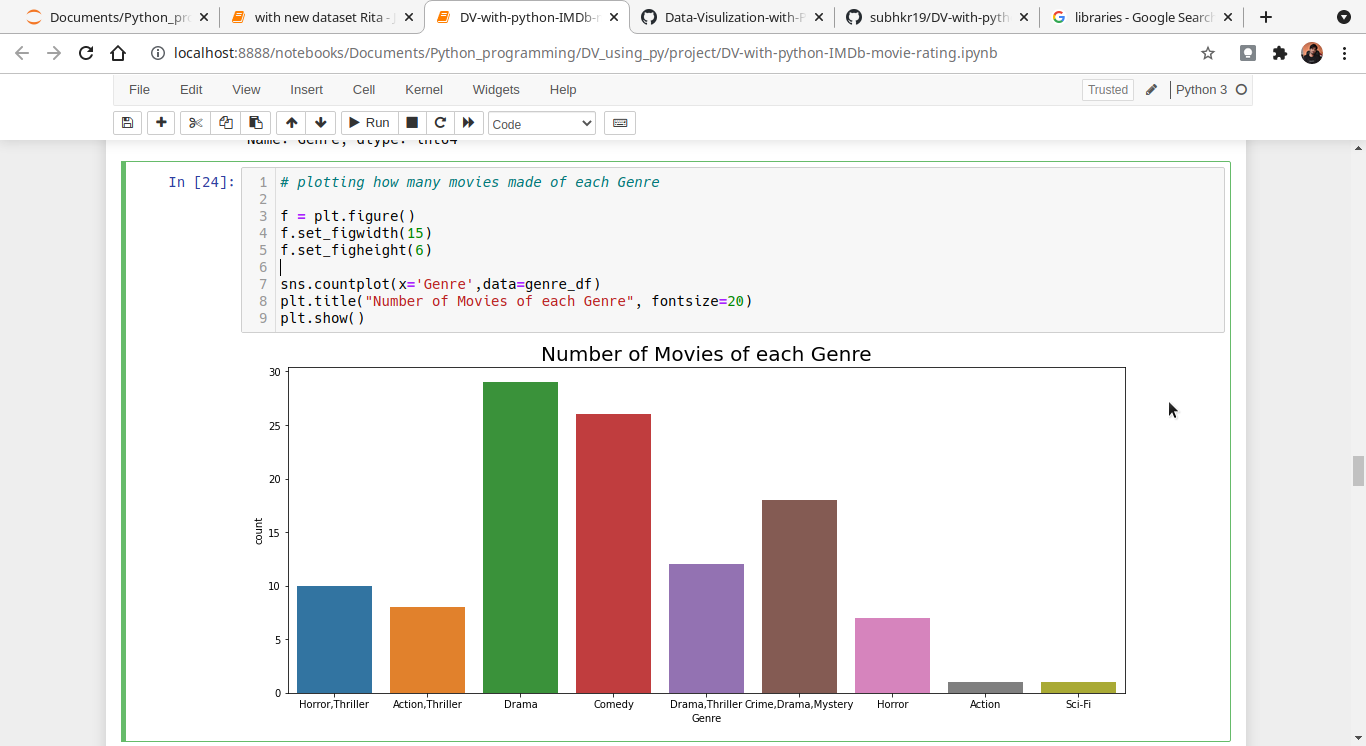
f.set\_figheight(6)

sns.countplot(x='Genre',data=genre\_df)

plt.title("Number of Movies of each Genre", fontsize=20)

plt.show()

Output:



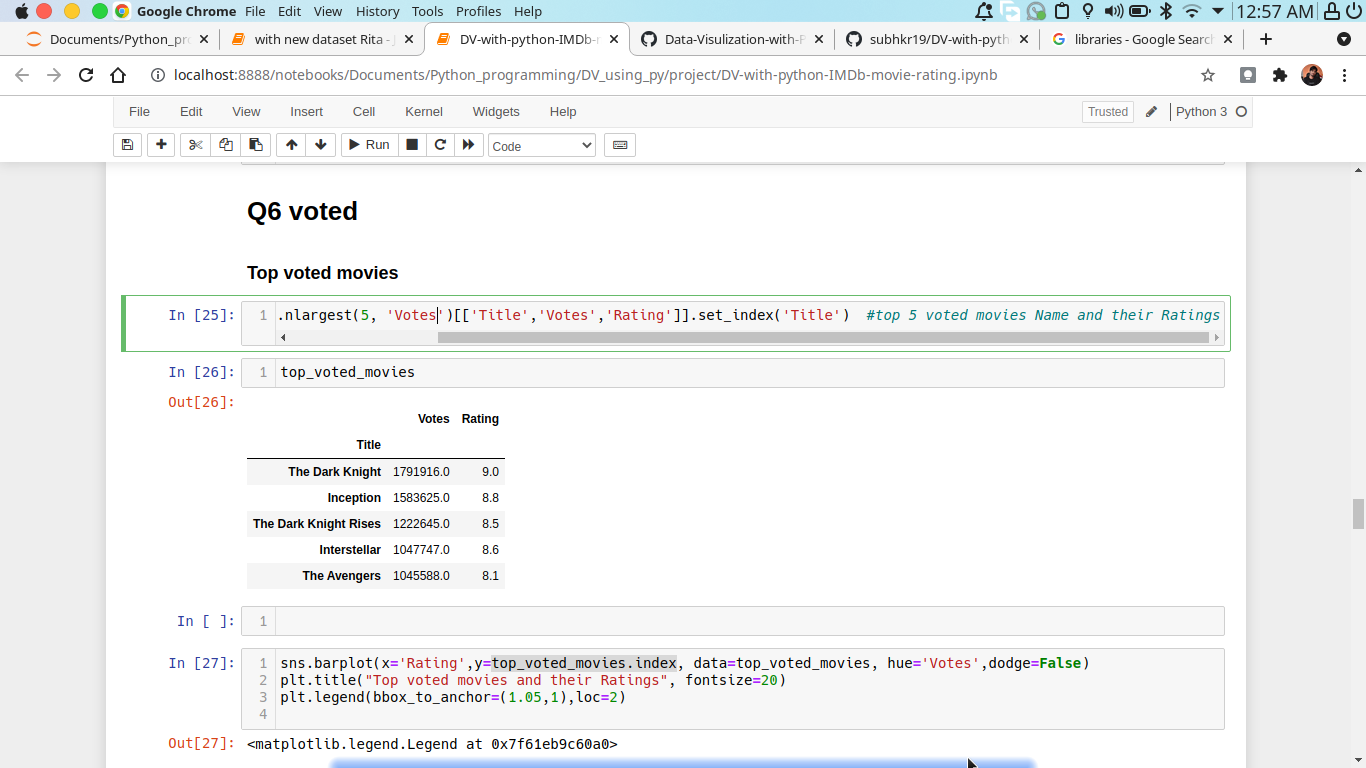
**Q6 voted**

**Top voted movies**

top\_voted\_movies = df.nlargest(5, 'Votes')[['Title','Votes','Rating']].set\_index('Title') #top 5 voted movies Name and their Ratings

top\_voted\_movies

output:

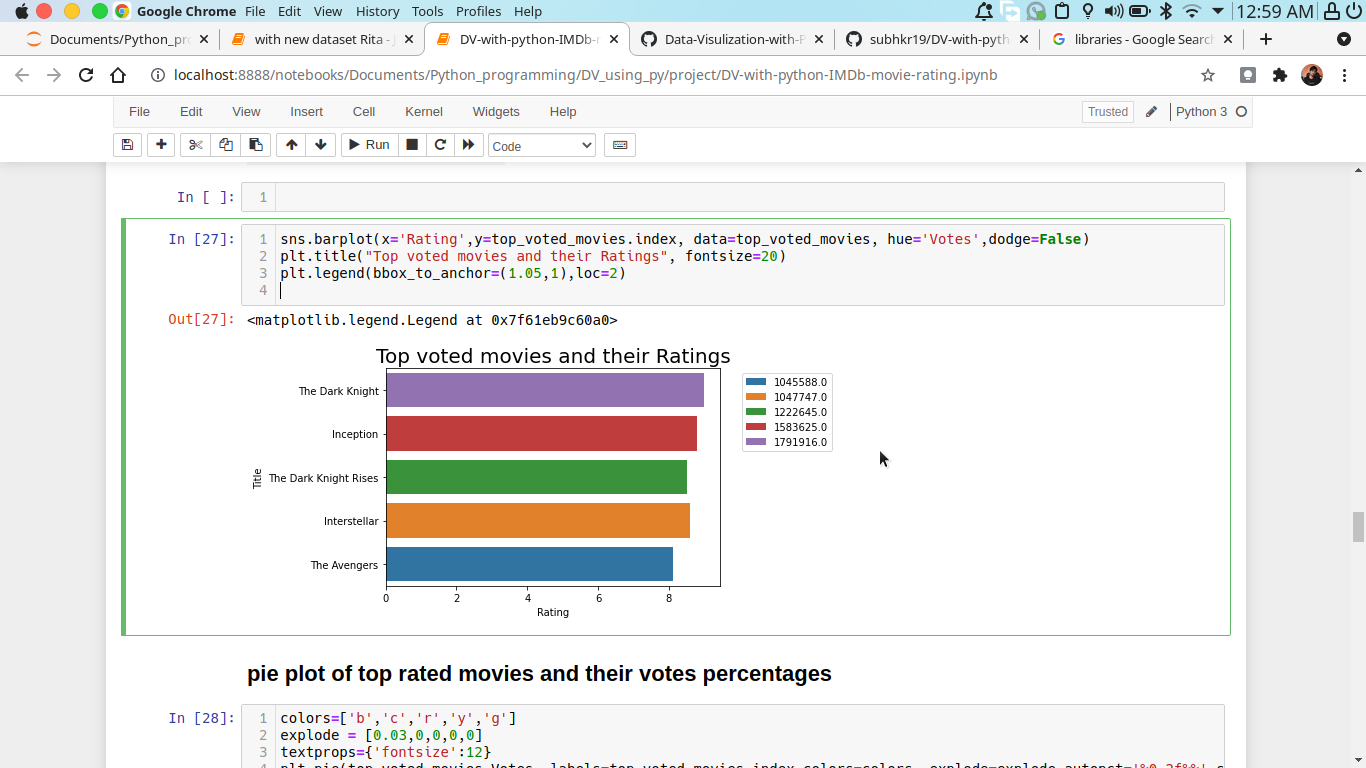


sns.barplot(x='Rating',y=top\_voted\_movies.index, data=top\_voted\_movies, hue='Votes',dodge=False)

plt.title("Top voted movies and their Ratings", fontsize=20)

plt.legend(bbox\_to\_anchor=(1.05,1),loc=2)

output:



**pie plot of top rated movies and their votes percentages**

colors=['b','c','r','y','g']

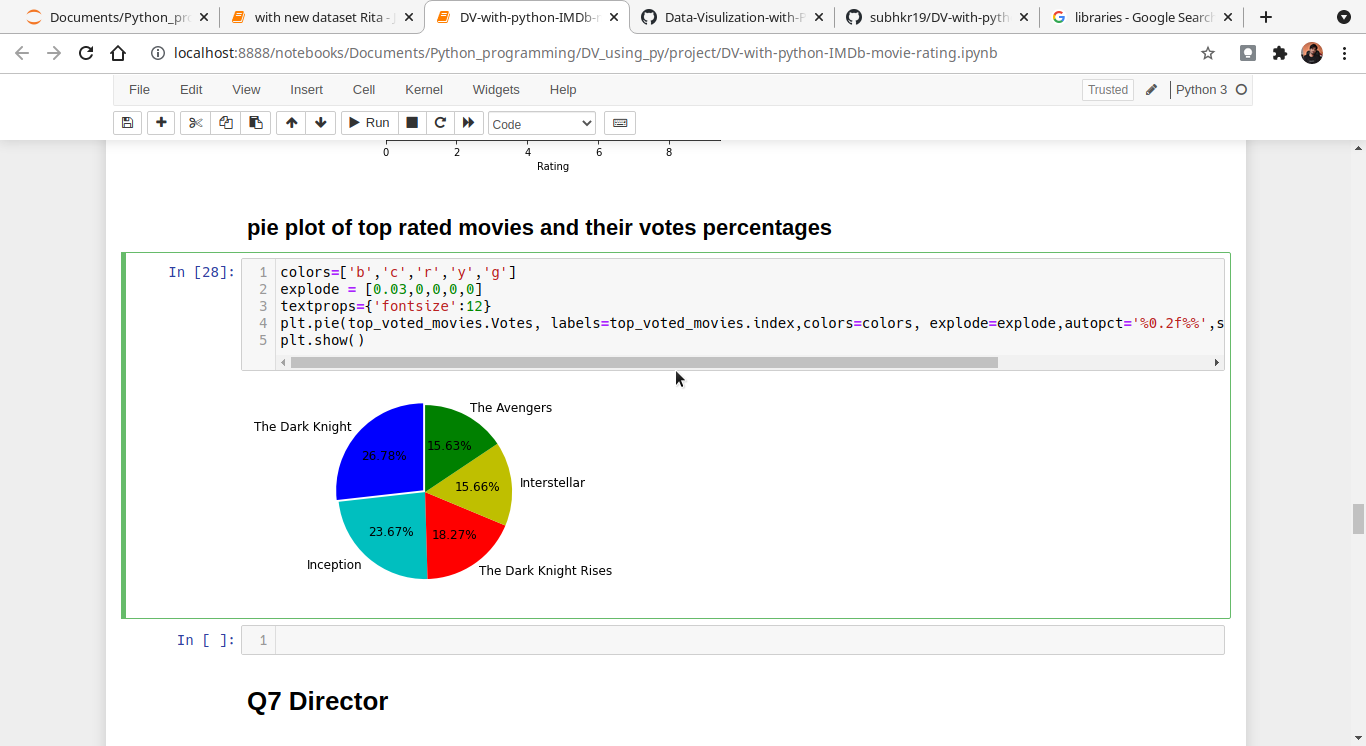
explode = [0.03,0,0,0,0]

textprops={'fontsize':12}

plt.pie(top\_voted\_movies.Votes, labels=top\_voted\_movies.index,colors=colors, explode=explode,autopct='%0.2f%%',startangle=90,textprops=textprops)

plt.show()

Output:

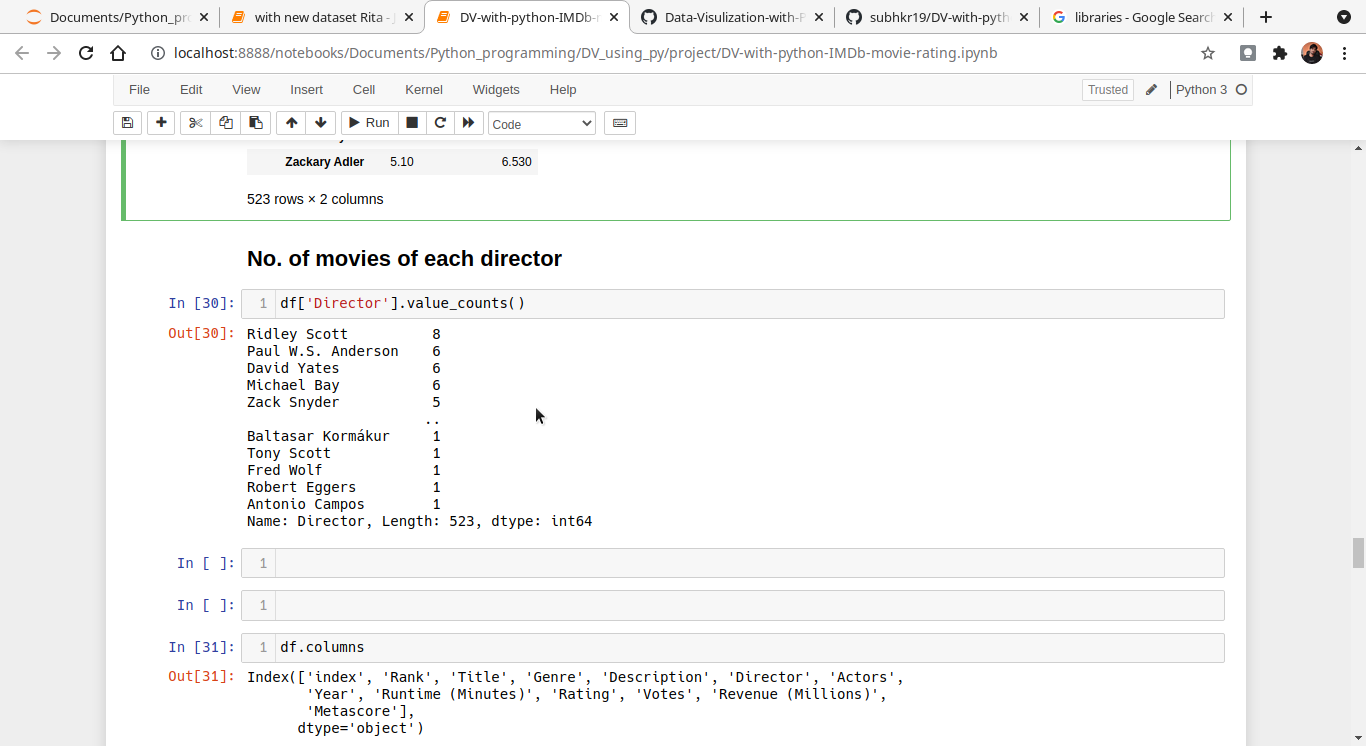


**Q7 Director**

**Average rating and revenue of each Director.**

df.pivot\_table(index='Director', aggfunc='mean')[['Rating', 'Revenue (Millions)']]

output:

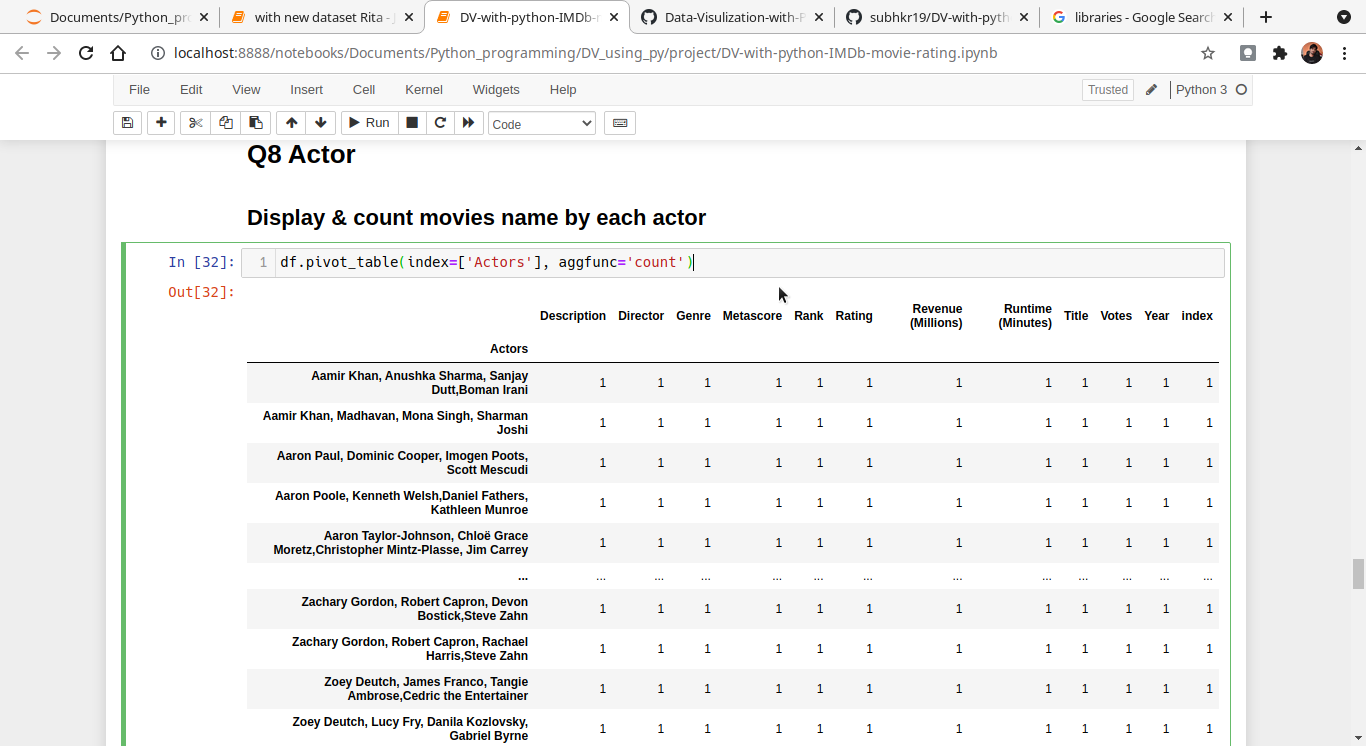


**Q8 Actor**

**Display & count movies name by each actor**

df.pivot\_table(index=['Actors'], aggfunc='count')

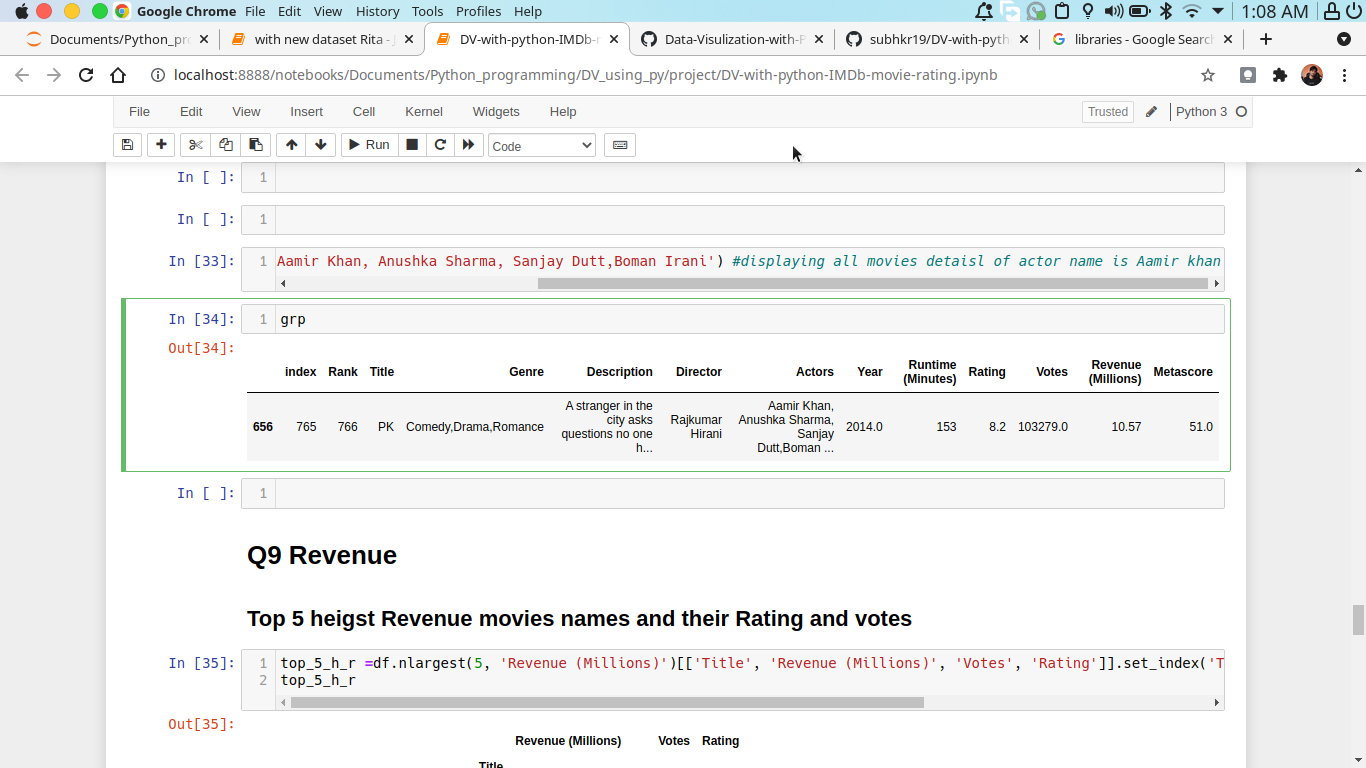
output:



grp = df.groupby(by='Actors').get\_group('Aamir Khan, Anushka Sharma, Sanjay Dutt,Boman Irani') #displaying all movies detaisl of actor name is Aamir khan

grp

output:



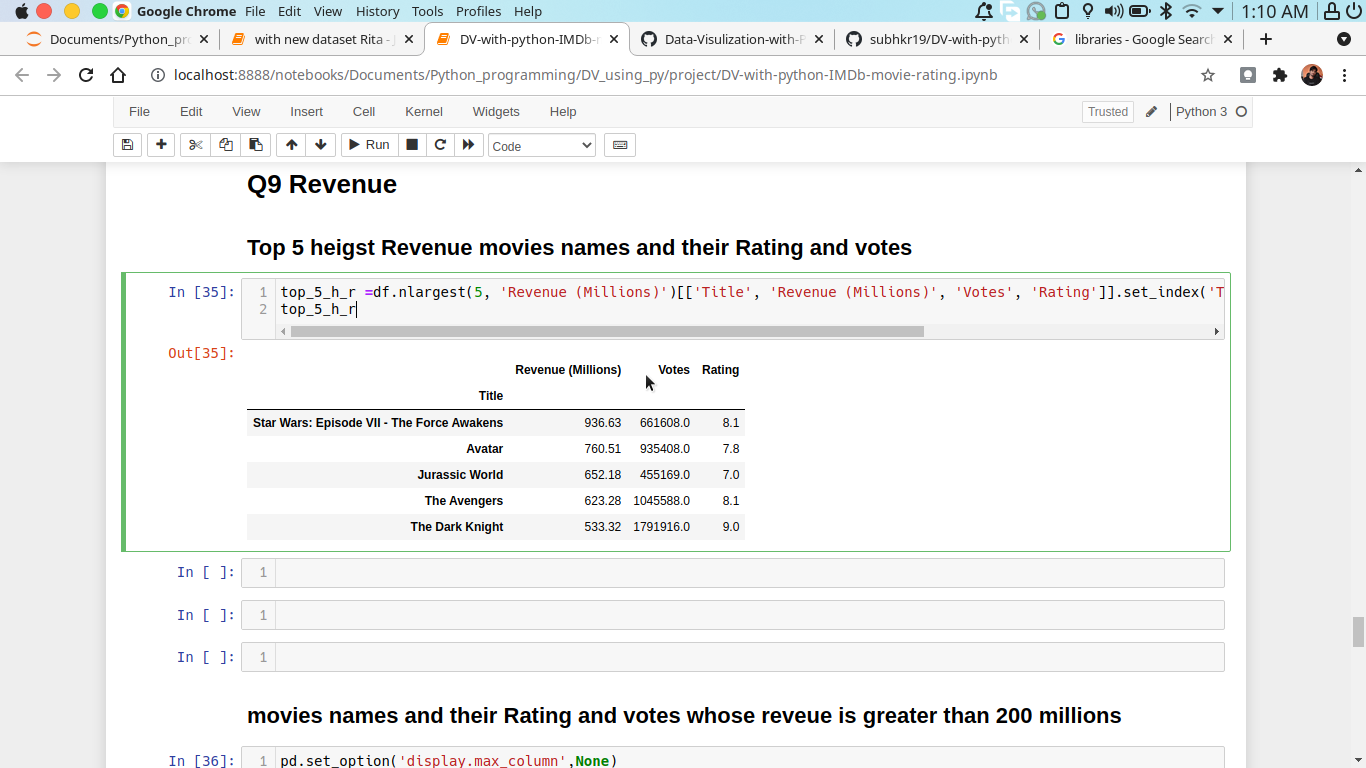
**Q9 Revenue**

**Top 5 heigst Revenue movies names and their Rating and votes**

top\_5\_h\_r =df.nlargest(5, 'Revenue (Millions)')[['Title', 'Revenue (Millions)', 'Votes', 'Rating']].set\_index('Title') #top 5 voted movies Name and their Ratings

top\_5\_h\_r

Output:

****

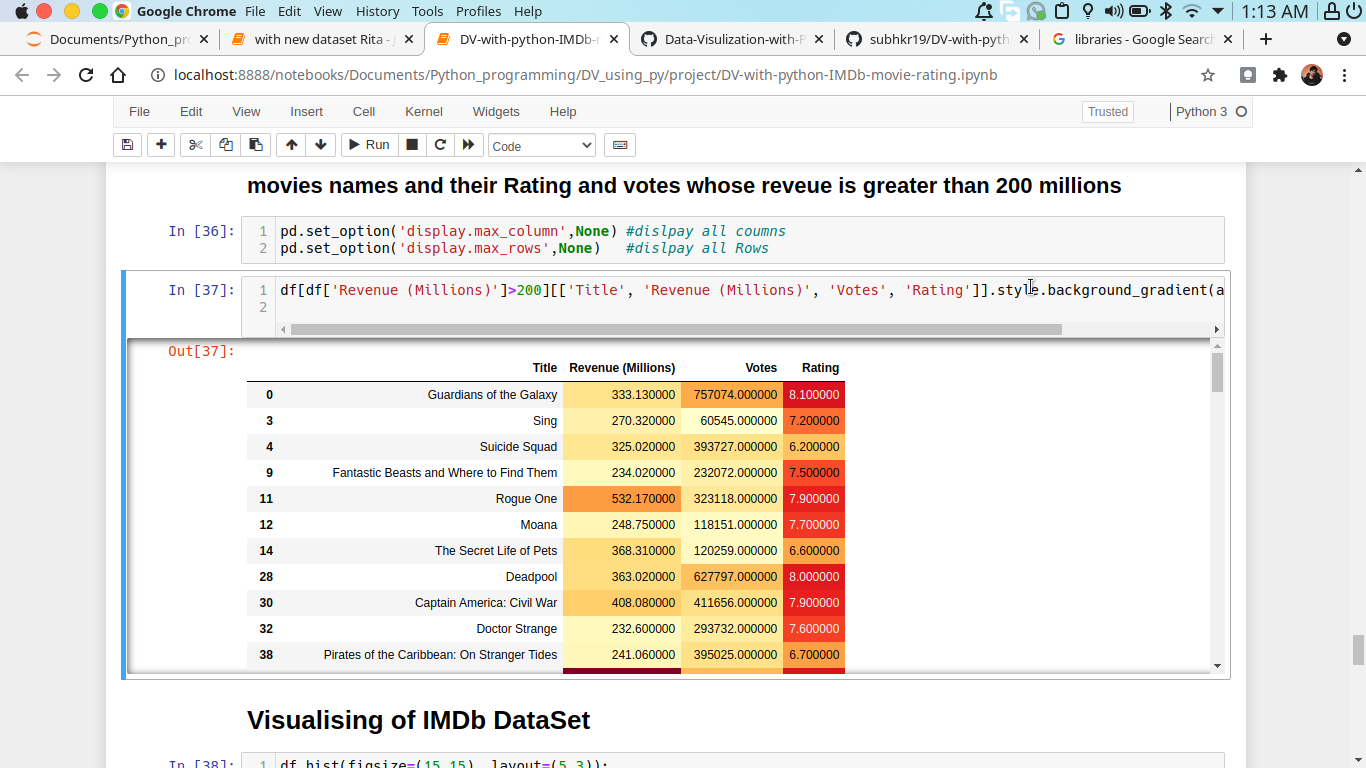
**movies names and their Rating and votes whose reveue is greater than 200 millions**

pd.set\_option('display.max\_column',None)

pd.set\_option('display.max\_rows',None)

df[df['Revenue (Millions)']>200][['Title', 'Revenue (Millions)', 'Votes', 'Rating']].style.background\_gradient(axis=0, cmap='YlOrRd')

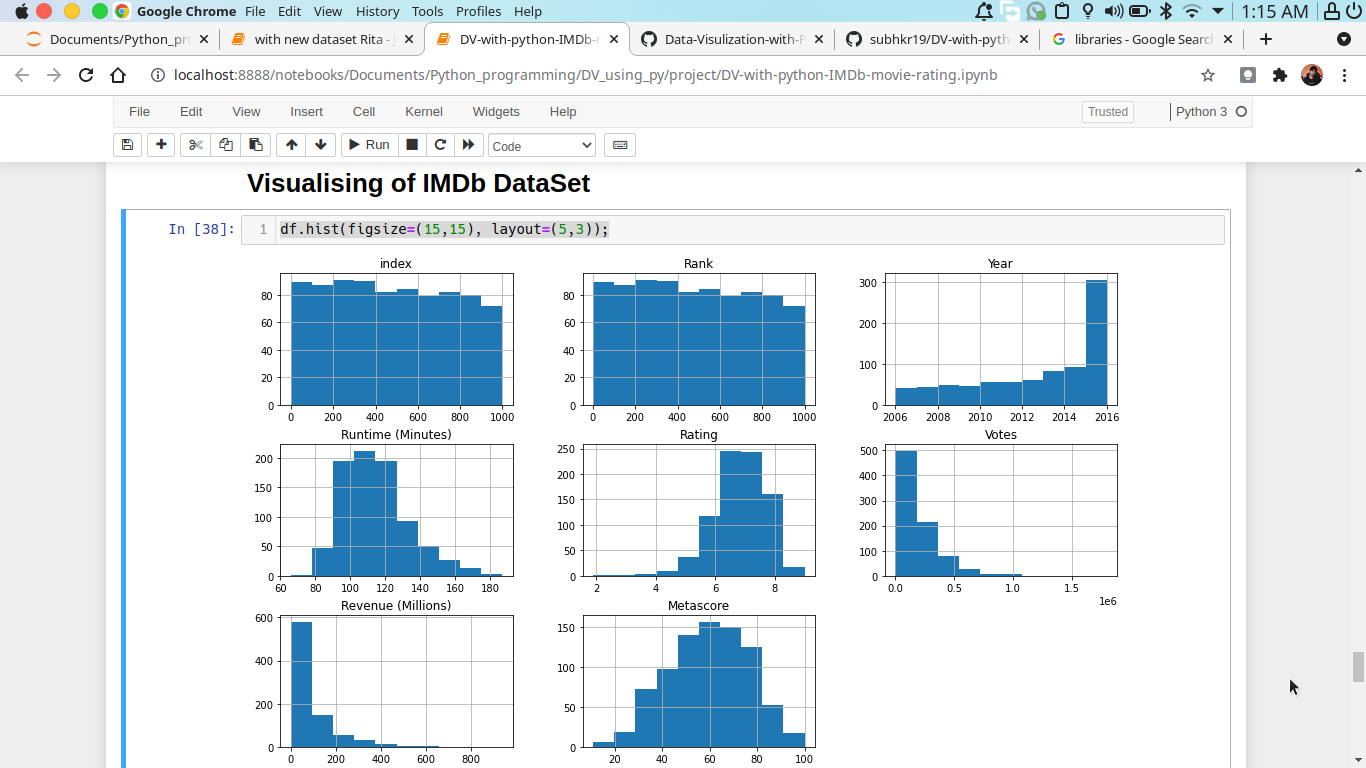
output:



**Visualising of IMDb DataSet**

df.hist(figsize=(15,15), layout=(5,3));

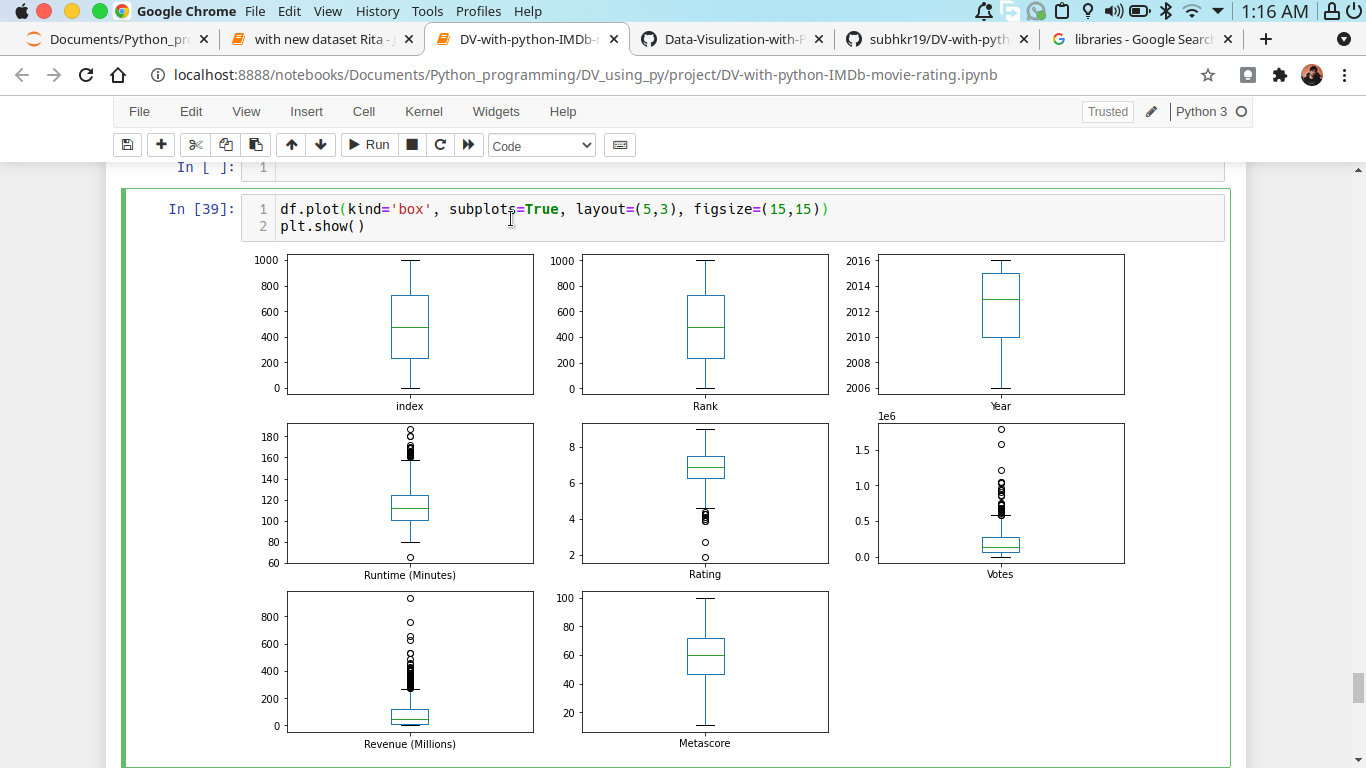
output:



df.plot(kind='box', subplots=True, layout=(5,3), figsize=(15,15))

plt.show()

output:



sns.pairplot(data=df)

output:

