MINI PROJECT JUNE ML BATCH 3

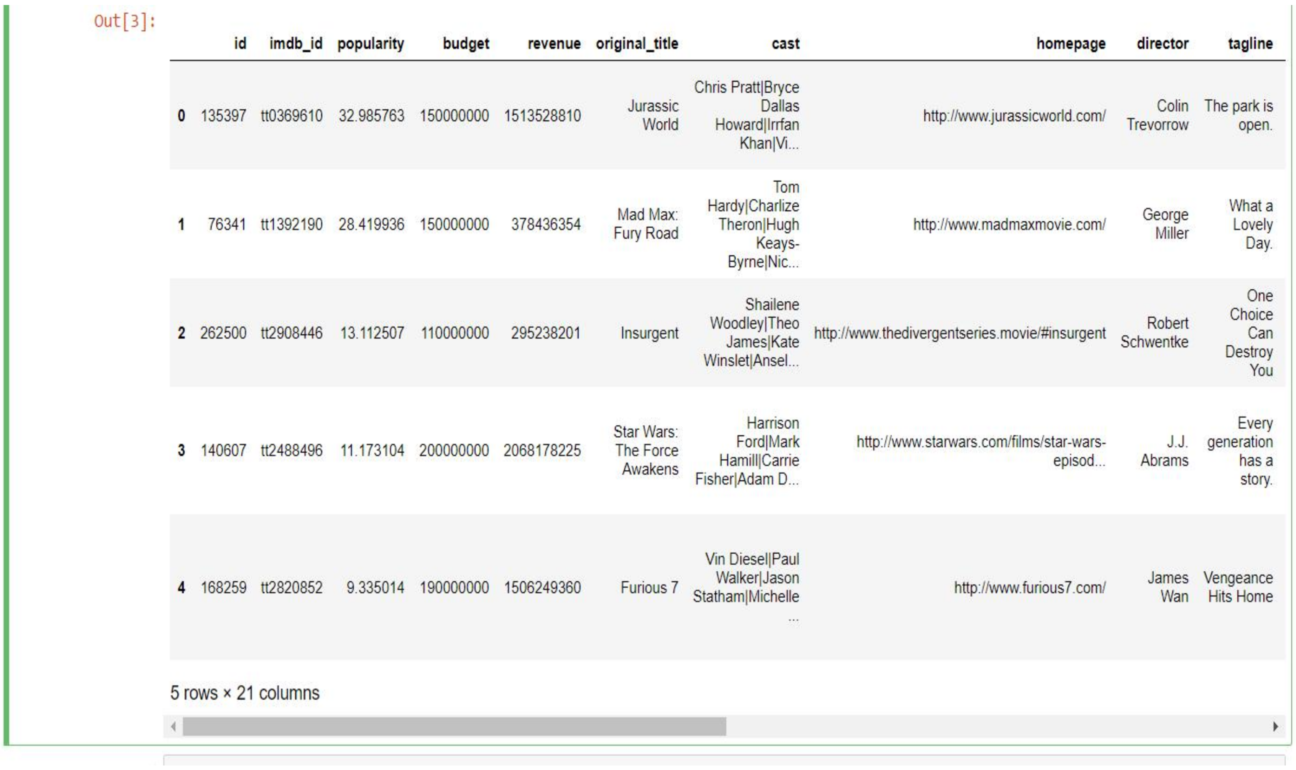
IMPORTING MODULES:

1) **import pandas as pd**
   import matplotlib.pyplot as plt
   %matplotlib inline
   import seaborn as sns

READING FILE:

2) **movies_df = pd.read_csv(r'C:\Users\vamsh\Downloads\tmdb-movies.csv' , encoding = 'utf8')**

3) **movies_df.head()**


Out[3]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...	J.J. Abrams	Every generation has a story.
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/	James Wan	Vengeance Hits Home

5 rows × 21 columns

DROPPING COLUMNS:

4) **movies_df = movies_df[['original_title', 'genres', 'release_year', 'cast', 'budget','production_companies', 'revenue','runtime']]**

DROPPING NULL VALUES:

5) **movies_df=movies_df.dropna()**

1) Which are the movies with the third lowest and third highest budget?

Ans)

```
movies_df_2=movies_df.loc[(movies_df['budget']!=0)
#(not considering movies with zero budget)

movies_df_2=movies_df_2.sort_values('budget',ascending=False)

list1=[]
for i in movies_df['budget']:
    list1.append(i)
n=sorted(set(list1))
k=len(n)-3
```

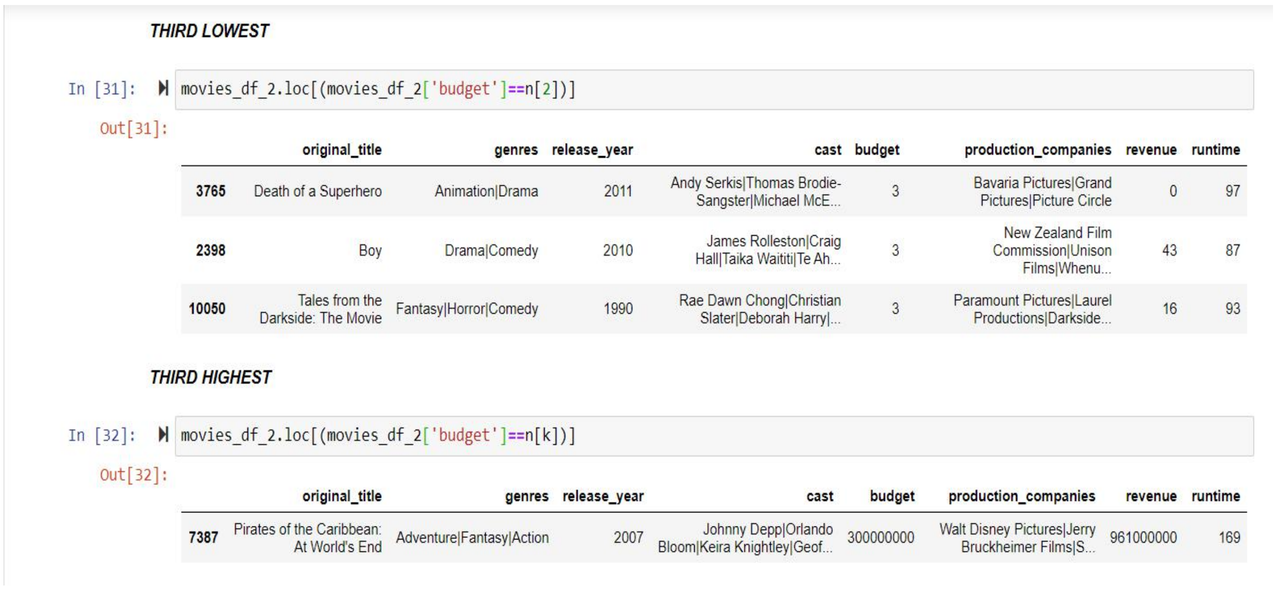
<u>THIRD LOWEST:</u>

```
    movies_df_2.loc[(movies_df_2['budget']==n[2])]
```

<u>THIRD HIGHEST:</u>

```
    movies_df_2.loc[(movies_df_2['budget']==n[k])]
```

<u>OUTPUT:</u>



THIRD LOWEST

```
In [31]:  movies_df_2.loc[(movies_df_2['budget']==n[2])]
```

Out[31]:

	original_title	genres	release_year	cast	budget	production_companies	revenue	runtime
3765	Death of a Superhero	Animation|Drama	2011	Andy Serkis|Thomas Brodie-Sangster|Michael McE...	3	Bavaria Pictures|Grand Pictures|Picture Circle	0	97
2398	Boy	Drama|Comedy	2010	James Rolleston|Craig Hall|Taika Waititi|Te Ah...	3	New Zealand Film Commission|Unison Films|Whenu...	43	87
10050	Tales from the Darkside: The Movie	Fantasy|Horror|Comedy	1990	Rae Dawn Chong|Christian Slater|Deborah Harry|...	3	Paramount Pictures|Laurel Productions|Darkside...	16	93

THIRD HIGHEST

```
In [32]:  movies_df_2.loc[(movies_df_2['budget']==n[k])]
```

Out[32]:

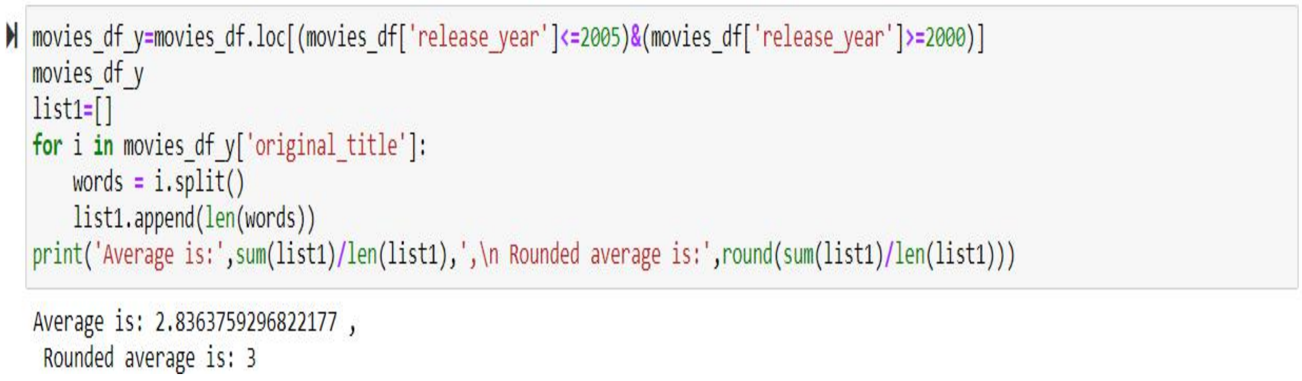
	original_title	genres	release_year	cast	budget	production_companies	revenue	runtime
7387	Pirates of the Caribbean: At World's End	Adventure|Fantasy|Action	2007	Johnny Depp|Orlando Bloom|Keira Knightley|Geof...	300000000	Walt Disney Pictures|Jerry Bruckheimer Films|S...	961000000	169

2) What is the average number of words in movie titles between the year 2000-2005?

Ans)

```
movies_df_y=movies_df.loc[(movies_df['release_year']<=2005)
              &(movies_df['release_year']>=2000)]
movies_df_y
list1=[]
for i in movies_df_y['original_title']:
    words = i.split()
    list1.append(len(words))
print('Average is:',sum(list1)/len(list1),',\n Rounded average
              is:',round(sum(list1)/len(list1)))
```

<u>OUTPUT:</u>



```
movies_df_y=movies_df.loc[(movies_df['release_year']<=2005)&(movies_df['release_year']>=2000)]
movies_df_y
list1=[]
for i in movies_df_y['original_title']:
    words = i.split()
    list1.append(len(words))
print('Average is:',sum(list1)/len(list1),',\n Rounded average is:',round(sum(list1)/len(list1)))
```

```
Average is: 2.8363759296822177 ,
 Rounded average is: 3
```

3) What is the most common Genre for Vin Diesel & Emma Watson movies?

Ans)

```
def get_key(my_dict,val):
  list=[]
  for key, value in my_dict.items():
          if val == value:
               list.append(key)
  return list
```

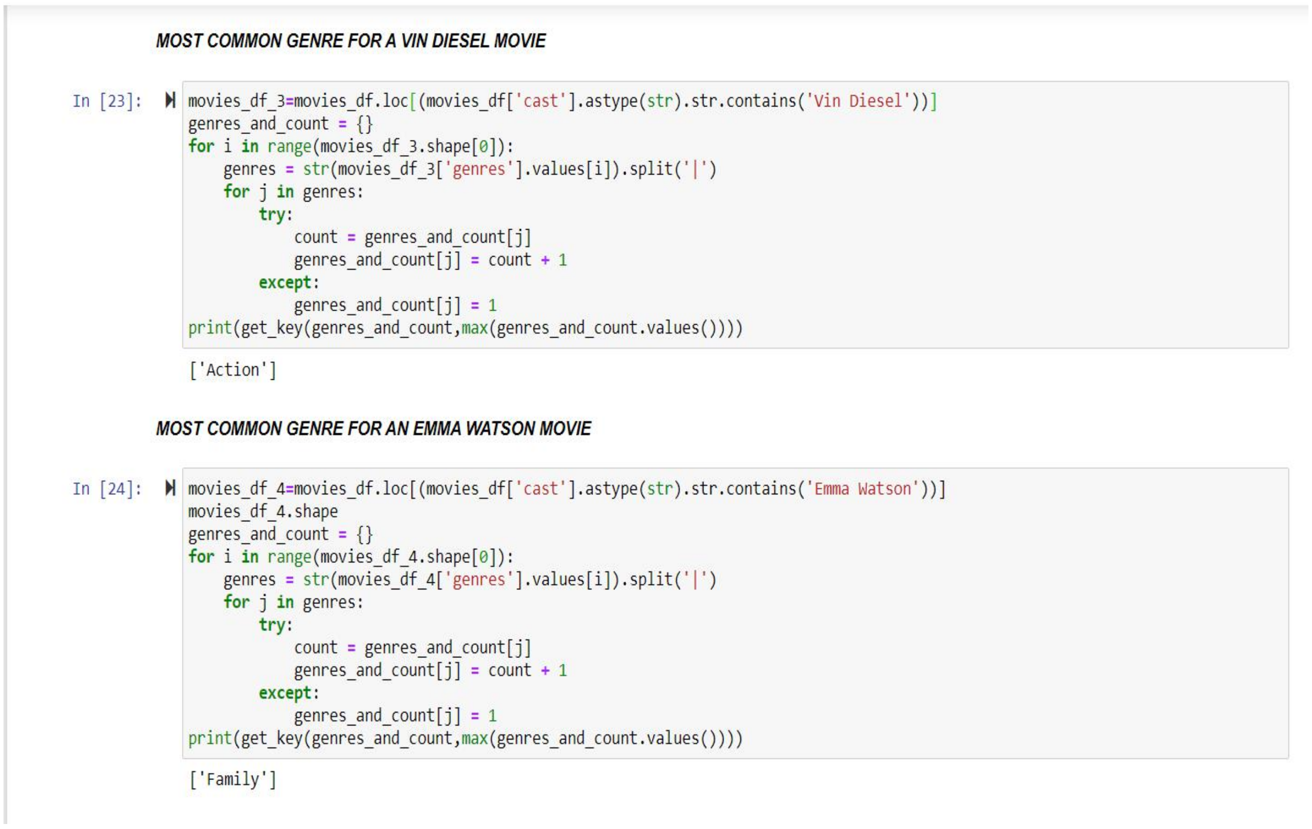
<u>MOST COMMON GENRE FOR VIN DIESEL MOVIES</u>

```
movies_df_3=movies_df.loc[(movies_df['cast'].astype(str).str.
                    contains('Vin Diesel'))]
genres_and_count = {}
for i in range(movies_df_3.shape[0]):
      genres = str(movies_df_3['genres'].values[i]).split('|')
      for j in genres:
             try:
                   count = genres_and_count[j]
                   genres_and_count[j] = count + 1
             except:
                   genres_and_count[j] = 1
print(get_key(genres_and_count,max(genres_and_count.values())
                                                            ))
```

<u>MOST COMMON GENRE FOR EMMA WATSON MOVIES:</u>

```
movies_df_4=movies_df.loc[(movies_df['cast'].astype(str).str.con
                         tains('Emma Watson'))]
movies_df_4.shape
genres_and_count = {}
for i in range(movies_df_4.shape[0]):
      genres = str(movies_df_4['genres'].values[i]).split('|')
      for j in genres:
             try:
                   count = genres_and_count[j]
                   genres_and_count[j] = count + 1
             except:
                   genres_and_count[j] = 1
print(get_key(genres_and_count,max(genres_and_count.values())
                                                            ))
```

OUTPUT:



MOST COMMON GENRE FOR A VIN DIESEL MOVIE

```
In [23]:  movies_df_3=movies_df.loc[(movies_df['cast'].astype(str).str.contains('Vin Diesel'))]
          genres_and_count = {}
          for i in range(movies_df_3.shape[0]):
              genres = str(movies_df_3['genres'].values[i]).split('|')
              for j in genres:
                  try:
                      count = genres_and_count[j]
                      genres_and_count[j] = count + 1
                  except:
                      genres_and_count[j] = 1
          print(get_key(genres_and_count,max(genres_and_count.values())))

          ['Action']
```

MOST COMMON GENRE FOR AN EMMA WATSON MOVIE

```
In [24]:  movies_df_4=movies_df.loc[(movies_df['cast'].astype(str).str.contains('Emma Watson'))]
          movies_df_4.shape
          genres_and_count = {}
          for i in range(movies_df_4.shape[0]):
              genres = str(movies_df_4['genres'].values[i]).split('|')
              for j in genres:
                  try:
                      count = genres_and_count[j]
                      genres_and_count[j] = count + 1
                  except:
                      genres_and_count[j] = 1
          print(get_key(genres_and_count,max(genres_and_count.values())))

          ['Family']
```

4) Which are the movies with most and least earned revenue?

Ans)

movies_df_5=movies_df.loc[(movies_df['revenue']!=0)]
#(not considering movies with zero revenue)

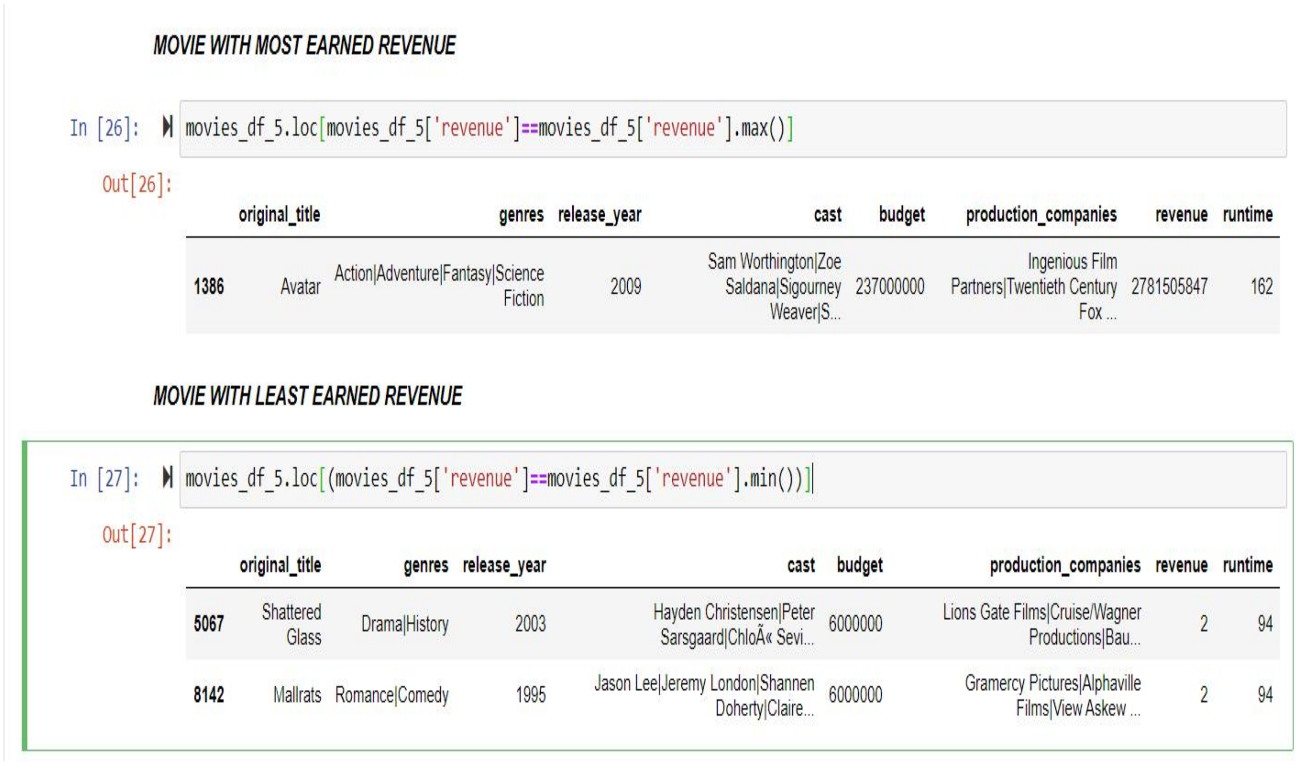
MOVIES WITH MOST EARNED REVENUE:

movies_df_5.loc[movies_df_5['revenue']==movies_df_5['revenue'].max()]

MOVIES WITH LEAST EARNED REVENUE:

movies_df_5.loc[(movies_df_5['revenue']==movies_df_5['revenue'].min())]

OUTPUT:



MOVIE WITH MOST EARNED REVENUE

```
In [26]: movies_df_5.loc[movies_df_5['revenue']==movies_df_5['revenue'].max()]
Out[26]:
```

	original_title	genres	release_year	cast	budget	production_companies	revenue	runtime
1386	Avatar	Action|Adventure|Fantasy|Science Fiction	2009	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	237000000	Ingenious Film Partners|Twentieth Century Fox ...	2781505847	162

MOVIE WITH LEAST EARNED REVENUE

```
In [27]: movies_df_5.loc[(movies_df_5['revenue']==movies_df_5['revenue'].min())]
Out[27]:
```

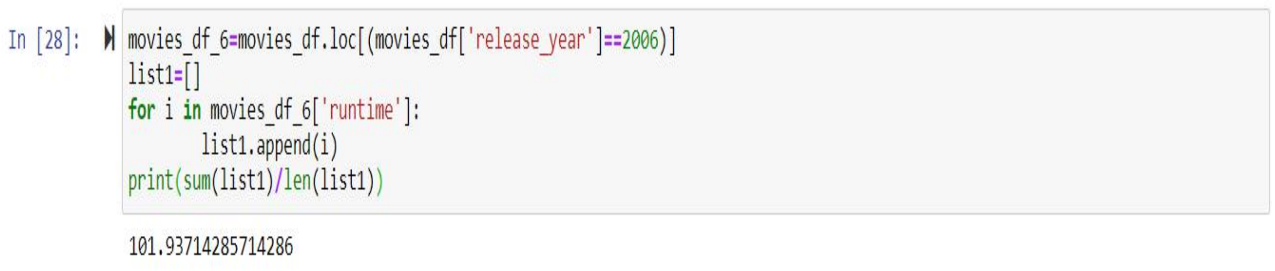
	original_title	genres	release_year	cast	budget	production_companies	revenue	runtime
5067	Shattered Glass	Drama|History	2003	Hayden Christensen|Peter Sarsgaard|ChloÃ« Sevi...	6000000	Lions Gate Films|Cruise/Wagner Productions|Bau...	2	94
8142	Mallrats	Romance|Comedy	1995	Jason Lee|Jeremy London|Shannen Doherty|Claire...	6000000	Gramercy Pictures|Alphaville Films|View Askew ...	2	94

5) What is the average runtime of movies in the year 2006?

Ans)

```
movies_df_6=movies_df.loc[(movies_df['release_year']==2006)
]
list1=[]
for i in movies_df_6['runtime']:
        list1.append(i)
print(sum(list1)/len(list1))
```

OUTPUT:



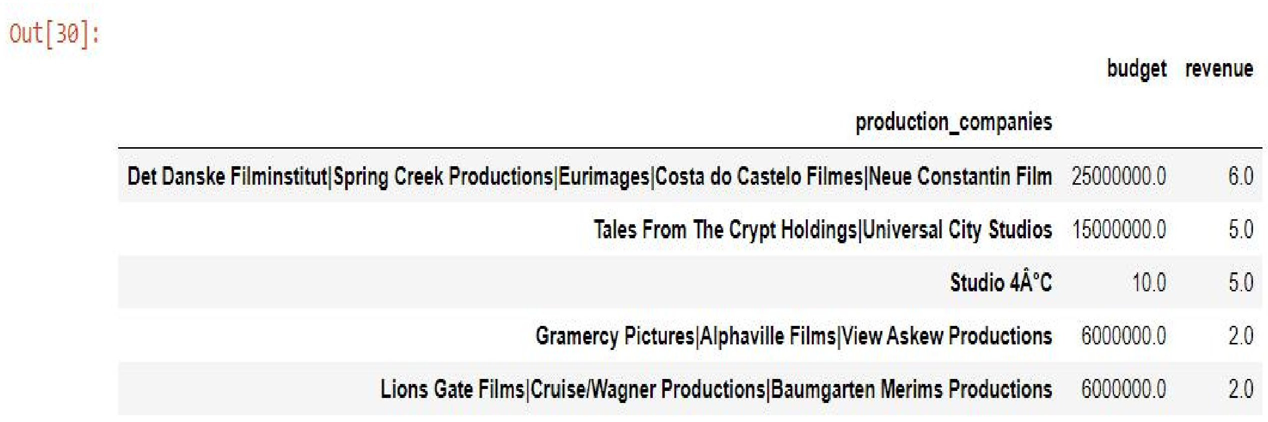
```
In [28]: movies_df_6=movies_df.loc[(movies_df['release_year']==2006)]
         list1=[]
         for i in movies_df_6['runtime']:
                 list1.append(i)
         print(sum(list1)/len(list1))

         101.93714285714286
```

6) Name any three production companies which have invested money in worse revenue movies?

Ans)

```
movies_df_br=movies_df_2.loc[(movies_df_2['revenue']!=0)]
#(not considering movies with zero budget and zero revenue)
movies_df_7=movies_df_br.groupby('production_companies').
                                                mean()
movies_df_8=movies_df_7.sort_values('revenue',ascending=
                                                False)
movies_df_8[['budget','revenue']].tail(5)
```

OUTPUT:



Out[30]:

	budget	revenue
production_companies		
Det Danske Filminstitut|Spring Creek Productions|Eurimages|Costa do Castelo Filmes|Neue Constantin Film	25000000.0	6.0
Tales From The Crypt Holdings|Universal City Studios	15000000.0	5.0
Studio 4Â°C	10.0	5.0
Gramercy Pictures|Alphaville Films|View Askew Productions	6000000.0	2.0
Lions Gate Films|Cruise/Wagner Productions|Baumgarten Merims Productions	6000000.0	2.0

DONE BY:

VAMSHISAI PERUMANDLA