Final Project Submission

Please fill out:

- Student name: *Patel Pinalben Nikhilkumar*
- Student pace: self paced / part time / full time *Part Time*
- Scheduled project review date/time: *20/12/2022*
- Instructor name: *Hardik*
- Blog post URL: *https://github.com/pinalnikhil/dsc4project.git*

In [1]:
```
# Your code here - remember to use markdown cells for comments as well!
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import time
import seaborn as sns
sns.set_style('white')
```

In [2]:
```
# gatting my first dataset,checking and describe the dataset
df_1= pd.read_csv('zippedData/bom.movie_gross.csv.gz')
df_1.head(10)
```

Out[2]:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010
5	The Twilight Saga: Eclipse	Sum.	300500000.0	398000000	2010
6	Iron Man 2	Par.	312400000.0	311500000	2010
7	Tangled	BV	200800000.0	391000000	2010
8	Despicable Me	Uni.	251500000.0	291600000	2010
9	How to Train Your Dragon	P/DW	217600000.0	277300000	2010

In [3]:
```
df_1.describe()
```

Out[3]:

	domestic_gross	year
count	3.359000e+03	3387.000000
mean	2.874585e+07	2013.958075
std	6.698250e+07	2.478141
min	1.000000e+02	2010.000000
25%	1.200000e+05	2012.000000
50%	1.400000e+06	2014.000000
75%	2.790000e+07	2016.000000
max	9.367000e+08	2018.000000

In [4]:
```
df_1.shape
```

Out[4]:
```
(3387, 5)
```

In [5]:
```
# gatting my second dataset,checking and describe the dataset
df_2= pd.read_csv('zippedData/imdb.title.ratings.csv.gz')
df_2.head(10)
```

Out[5]:

	tconst	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21
5	tt1069246	6.2	326
6	tt1094666	7.0	1613
7	tt1130982	6.4	571
8	tt1156528	7.2	265
9	tt1161457	4.2	148

In [6]:
```
# gatting my third dataset
df_2.describe()
```

Out[6]:

	averagerating	numvotes
count	73856.000000	7.385600e+04
mean	6.332729	3.523662e+03
std	1.474978	3.029402e+04
min	1.000000	5.000000e+00
25%	5.500000	1.400000e+01
50%	6.500000	4.900000e+01
75%	7.400000	2.820000e+02
max	10.000000	1.841066e+06

In [7]:
```
df_2.shape
```

Out[7]:
```
(73856, 3)
```

In [8]:
```
# gatting my third dataset,checking and describe the dataset
df_3= pd.read_csv('zippedData/imdb.title.basics.csv.gz')
df_3.head(10)
```

Out[8]:

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy
5	tt0111414	A Thin Life	A Thin Life	2018	75.0	Comedy
6	tt0112502	Bigfoot	Bigfoot	2017	NaN	Horror,Thriller
7	tt0137204	Joe Finds Grace	Joe Finds Grace	2017	83.0	Adventure,Animation,Comedy
8	tt0139613	O Silêncio	O Silêncio	2012	NaN	Documentary,History
9	tt0144449	Nema aviona za Zagreb	Nema aviona za Zagreb	2012	82.0	Biography

In [9]:
```
df_3.describe()
```

Out[9]:

	start_year	runtime_minutes
count	146144.000000	114405.000000
mean	2014.621798	86.187247
std	2.733583	166.360590
min	2010.000000	1.000000
25%	2012.000000	70.000000
50%	2015.000000	87.000000
75%	2017.000000	99.000000
max	2115.000000	51420.000000

In [10]:
```
df_3.shape
```

Out[10]:
```
(146144, 6)
```

In [11]:
```
# merge two dataset(df_2, df_3) togather and gatting new dataset
df_4= pd.merge(df_2, df_3, on= 'tconst', how= "outer")
```

In [12]:
```
df_4
```

Out[12]:

	tconst	averagerating	numvotes	primary_title	original_title	start_year	runtime_minutes	genres
0	tt10356526	8.3	31.0	Laiye Je Yaarian	Laiye Je Yaarian	2019	117.0	Romance
1	tt10384606	8.9	559.0	Borderless	Borderless	2019	87.0	Documentary
2	tt1042974	6.4	20.0	Just Inès	Just Inès	2010	90.0	Drama
3	tt1043726	4.2	50352.0	The Legend of Hercules	The Legend of Hercules	2014	99.0	Action,Adventure,Fantasy
4	tt1060240	6.5	21.0	Até Onde?	Até Onde?	2011	73.0	Mystery,Thriller
...	...	...	...	...	...	...	...	...
146139	tt9916538	NaN	NaN	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	Drama
146140	tt9916622	NaN	NaN	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Documentary
146141	tt9916706	NaN	NaN	Dankyavar Danka	Dankyavar Danka	2013	NaN	Comedy
146142	tt9916730	NaN	NaN	6 Gunn	6 Gunn	2017	116.0	NaN
146143	tt9916754	NaN	NaN	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 8 columns

In [13]:
```
df_4.describe()
```

Out[13]:

	averagerating	numvotes	start_year	runtime_minutes
count	73856.000000	7.385600e+04	146144.000000	114405.000000
mean	6.332729	3.523662e+03	2014.621798	86.187247
std	1.474978	3.029402e+04	2.733583	166.360590
min	1.000000	5.000000e+00	2010.000000	1.000000
25%	5.500000	1.400000e+01	2012.000000	70.000000
50%	6.500000	4.900000e+01	2015.000000	87.000000
75%	7.400000	2.820000e+02	2017.000000	99.000000
max	10.000000	1.841066e+06	2115.000000	51420.000000

In [14]:
```
df_4.shape
```

Out[14]:
```
(146144, 8)
```

In [15]:
```
#To find null value in merge dataset df_4
df_4.isnull().sum()
```

Out[15]:
```
tconst                 0
averagerating      72288
numvotes           72288
primary_title          0
original_title        21
start_year             0
runtime_minutes    31739
genres              5408
dtype: int64
```

In [16]:
```
#To find percentage of null values in the merged dataset
percentage_missing_valies_df= ((df_4.isnull().sum())/len(df_4))*100
percentage_missing_valies_df
```

Out[16]:
```
tconst              0.000000
averagerating      49.463543
numvotes           49.463543
primary_title       0.000000
original_title      0.014369
start_year          0.000000
runtime_minutes    21.717621
genres              3.700460
dtype: float64
```

In [17]:
```
#to remove the rows wich have null values in averagerating, numvotes, runtime_minutes and genres in df_4
df_4.dropna(inplace=True)
df_4.columns=df_4.columns.str.title()
df_4
```

Out[17]:

	Tconst	Averagerating	Numvotes	Primary_Title	Original_Title	Start_Year	Runtime_Minutes	Genres
0	tt10356526	8.3	31.0	Laiye Je Yaarian	Laiye Je Yaarian	2019	117.0	Romance
1	tt10384606	8.9	559.0	Borderless	Borderless	2019	87.0	Documentary
2	tt1042974	6.4	20.0	Just Inès	Just Inès	2010	90.0	Drama
3	tt1043726	4.2	50352.0	The Legend of Hercules	The Legend of Hercules	2014	99.0	Action,Adventure,Fantasy
4	tt1060240	6.5	21.0	Até Onde?	Até Onde?	2011	73.0	Mystery,Thriller
...	...	...	...	...	...	...	...	...
73849	tt9768966	8.6	27.0	Plugged in	Plugged in	2019	53.0	Documentary
73851	tt9805820	8.1	25.0	Caisa	Caisa	2018	84.0	Documentary
73852	tt9844256	7.5	24.0	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	Action,Animation,Sci-Fi
73854	tt9886934	7.0	5.0	The Projectionist	The Projectionist	2019	81.0	Documentary
73855	tt9894098	6.3	128.0	Sathru	Sathru	2019	129.0	Thriller

65720 rows × 8 columns

In [18]:
```
df_4.shape
```

Out[18]:
```
(65720, 8)
```

In [19]:
```
df_4.isnull().sum()
```

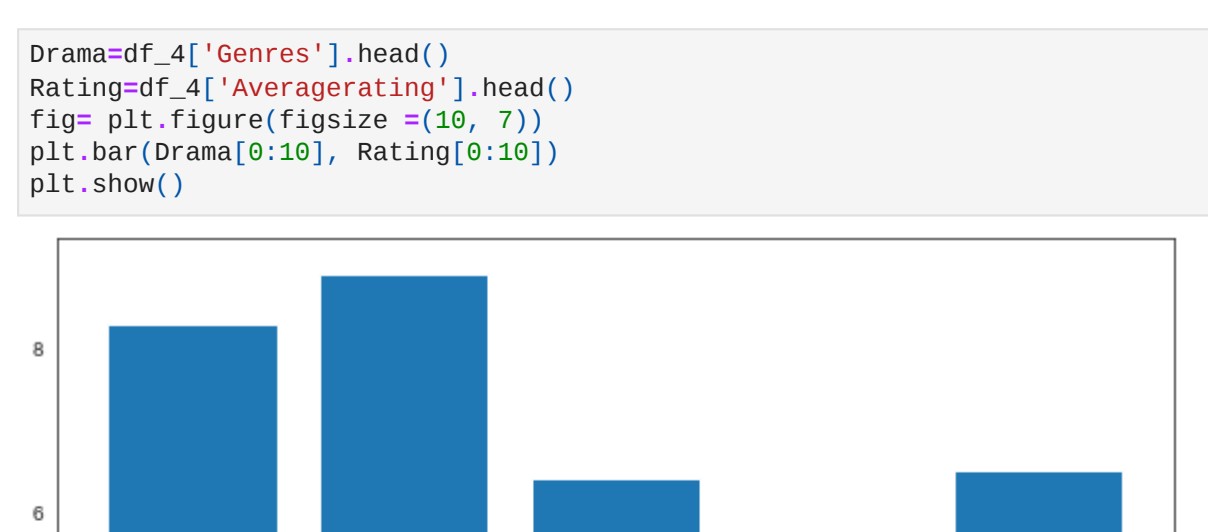
Out[19]:
```
Tconst             0
Averagerating      0
Numvotes           0
Primary_Title      0
Original_Title     0
Start_Year         0
Runtime_Minutes    0
Genres             0
dtype: int64
```

In [20]:
```
df_4.describe()
```

Out[20]:

	Averagerating	Numvotes	Start_Year	Runtime_Minutes
count	65720.000000	6.572000e+04	65720.000000	65720.000000
mean	6.320902	3.954674e+03	2014.258065	94.732273
std	1.458878	3.208823e+04	2.600143	209.377017
min	1.000000	5.000000e+00	2010.000000	3.000000
25%	5.500000	1.600000e+01	2012.000000	81.000000
50%	6.500000	6.200000e+01	2014.000000	91.000000
75%	7.300000	3.520000e+02	2016.000000	104.000000
max	10.000000	1.841066e+06	2019.000000	51420.000000

In [21]:
```
Drama=df_4['Genres'].head()
Rating=df_4['Averagerating'].head()
fig= plt.figure(figsize =(10, 7))
plt.bar(Drama[0:10], Rating[0:10])
plt.show()
```

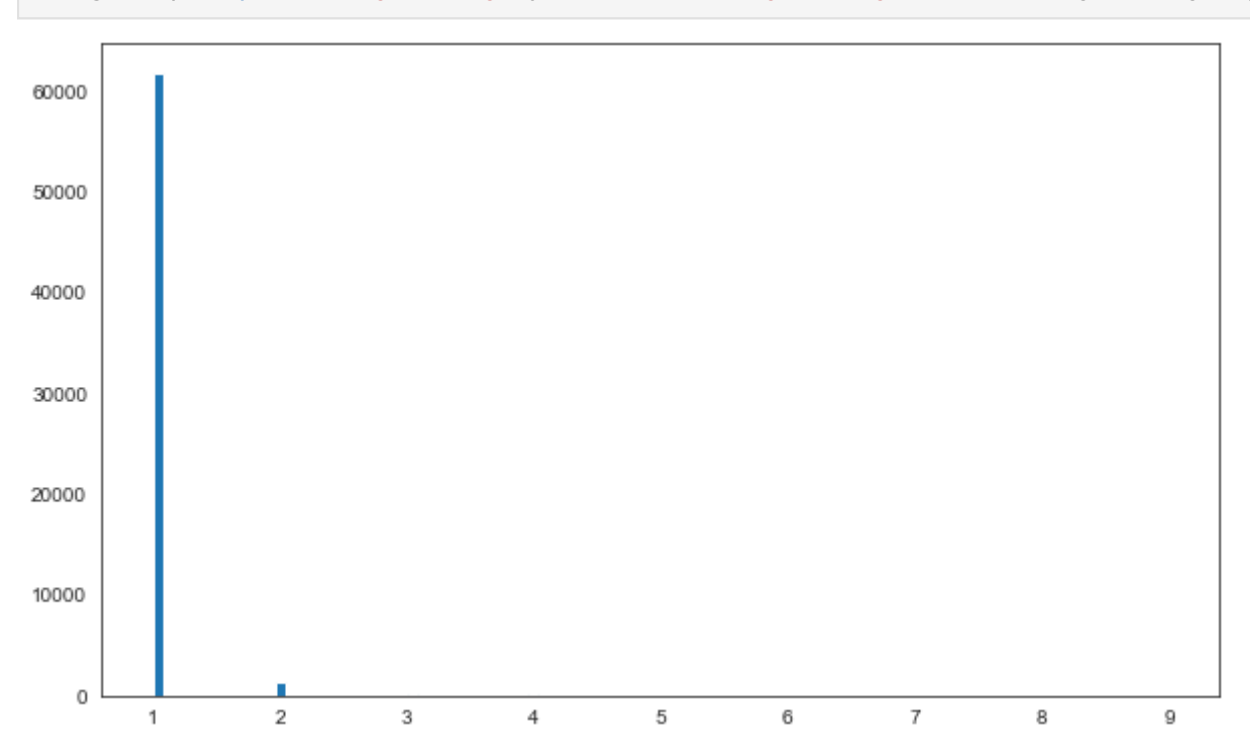


In [49]:
```
df_4.groupby('Original_Title').mean()['Averagerating'].sort_values(ascending=False).head()
df_4.groupby('Original_Title').count()['Averagerating'].sort_values(ascending=False).head()
Averagerating= pd.DataFrame(df_4.groupby('Original_Title').mean()['Averagerating'])
Averagerating['num of Averagerating'] = df_4.groupby('Original_Title').count()['Averagerating']
Averagerating.sort_values(by='Averagerating',ascending=False).head()
```

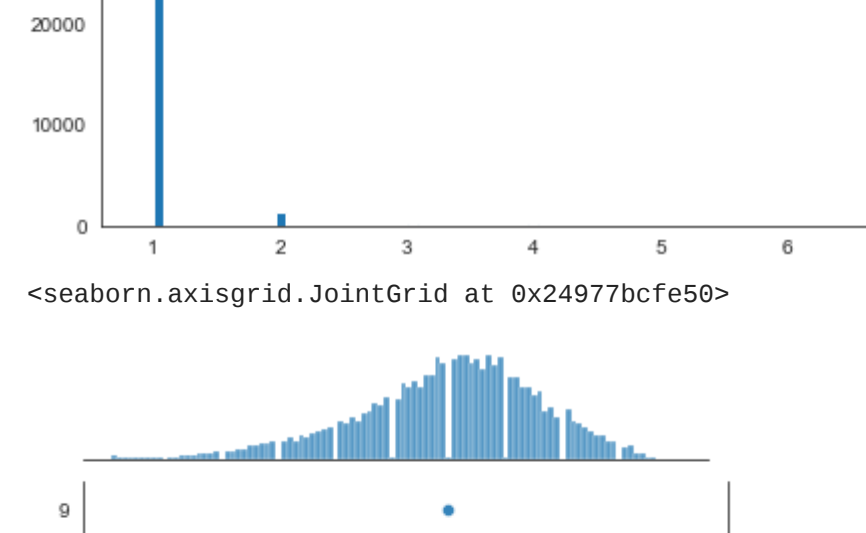
Out[49]:

	Averagerating	num of Averagerating
Original_Title		
I Was Born Yesterday!	10.0	1
Hercule contre Hermès	10.0	1
Requiem voor een Boom	10.0	1
Exteriores: Mulheres Brasileiras na Diplomacia	10.0	1
Fly High: Story of the Disc Dog	10.0	1

In [38]:
```
plt.figure(figsize=(10,6))
plt.hist(Averagerating['num of Averagerating'],bins=100)
plt.show()
sns.jointplot(x='Averagerating',y='num of Averagerating',data=Averagerating,alpha=0.9)
```



Out[38]:
```
<seaborn.axisgrid.JointGrid at 0x24977bcfe50>
```



In [39]:
```
df_1.groupby('foreign_gross').mean()['year'].sort_values(ascending=False).head()
df_1.groupby('foreign_gross').count()['year'].sort_values(ascending=False).head()
```

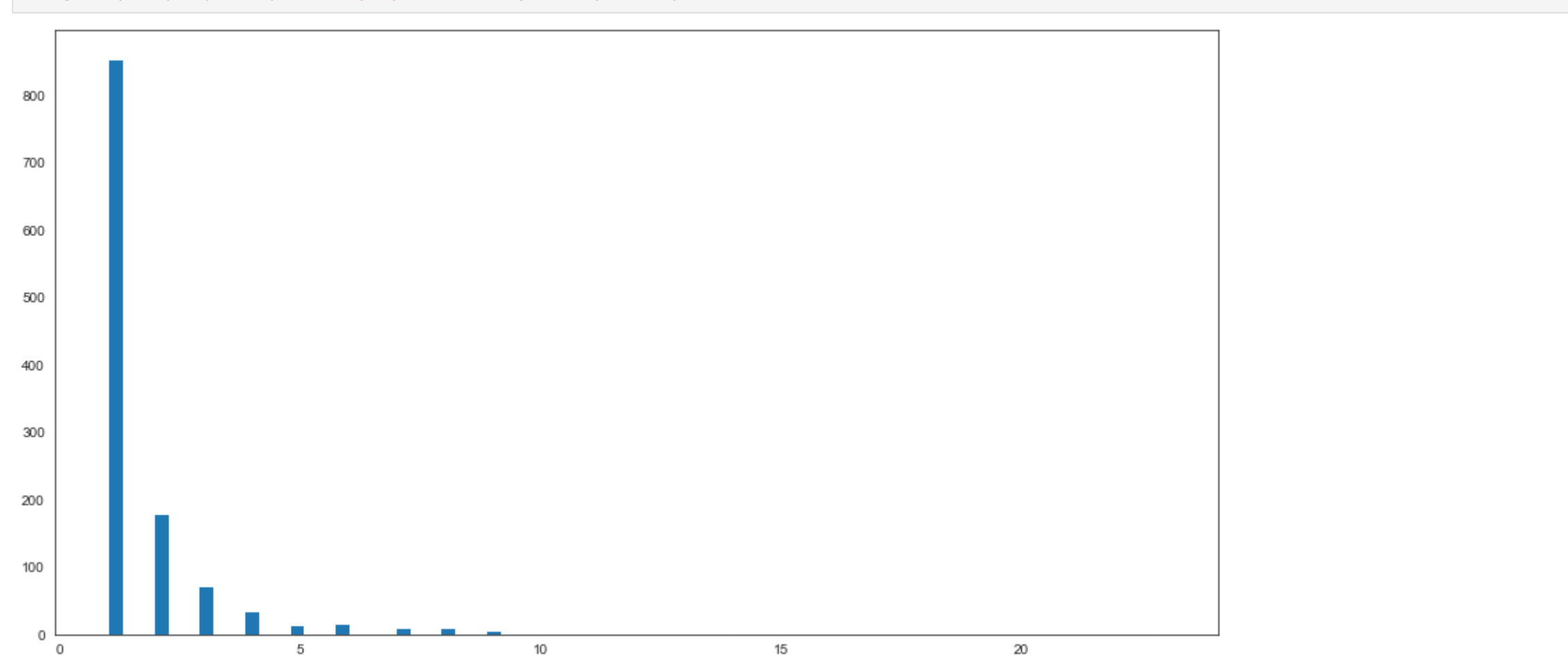
Out[39]:
```
foreign_gross
1200000    23
1100000    14
1900000    12
4200000    12
1300000    11
Name: year, dtype: int64
```

In [50]:
```
year = pd.DataFrame(df_1.groupby('foreign_gross').mean()['year'])
year['num of year'] = df_1.groupby('foreign_gross').count()['year']
year.sort_values(by='year',ascending=False).head()
```

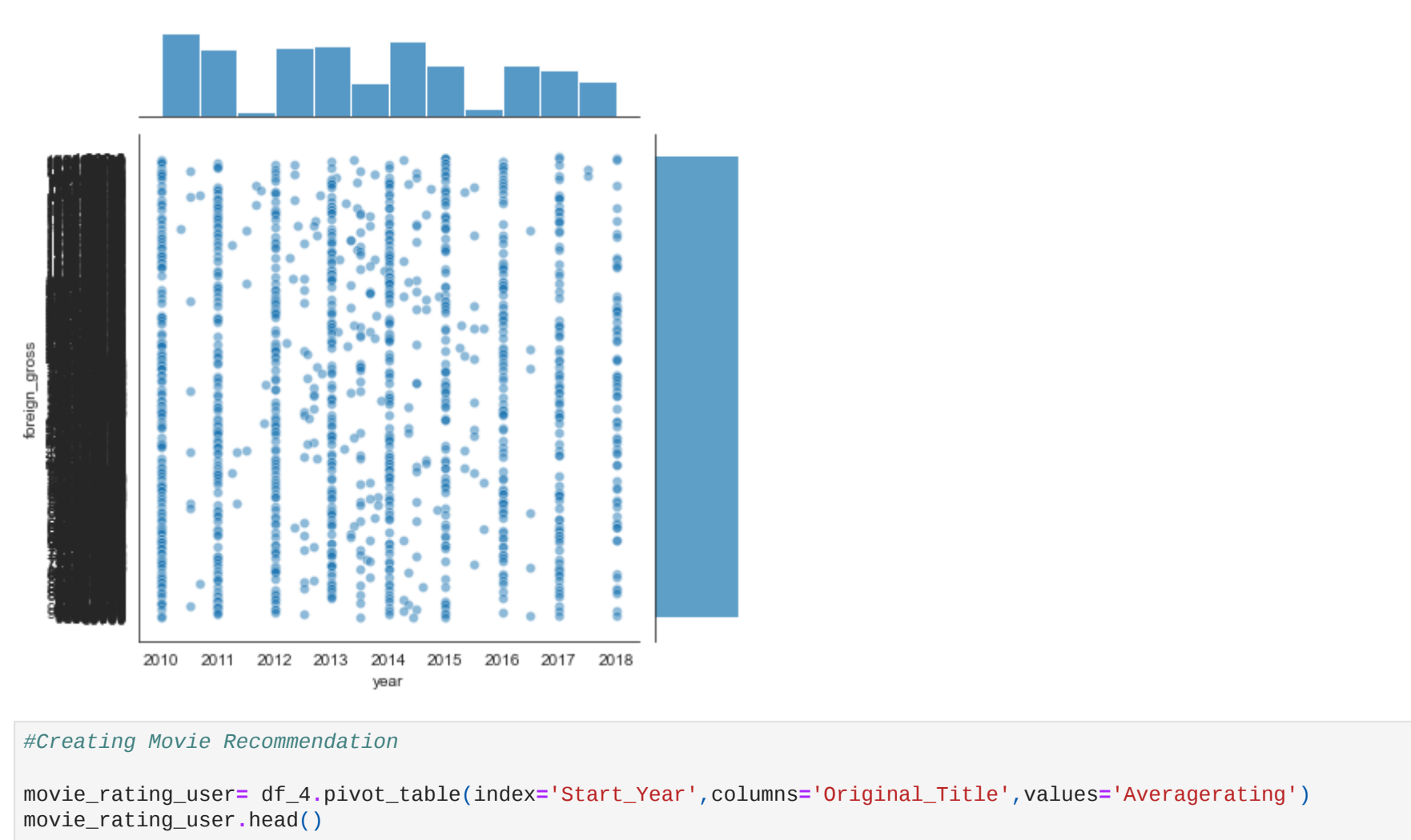
Out[50]:

	year	num of year
foreign_gross		
33000000	2018.0	1
236000000	2018.0	1
218900000	2018.0	1
494100000	2018.0	1
49400000	2018.0	1

In [58]:
```
Gross=df_1['foreign_gross'].head(15)
Title=df_1['title'].head(15)
fig= plt.figure(figsize =(15, 7),)
plt.bar(Gross[0:],Title[0:],)
plt.show()
```



In [42]:
```
plt.figure(figsize=(15,8))
plt.hist(year['num of year'],bins=70)
plt.show()
sns.jointplot(x='year',y='foreign_gross',data=year,alpha=0.5)
```



Out[42]:
```
<seaborn.axisgrid.JointGrid at 0x2497a090040>
```

In [43]:
```
#Creating Movie Recommendation

movie_rating_user= df_4.pivot_table(index='Start_Year',columns='Original_Title',values='Averagerating')
movie_rating_user.head()
```

Out[43]:

Original_Title	#1 Serial Killer	#5	#66	#AbroHilo	#BKKY	#BeRobin the Movie	#Beings	#Captured	#Disneyland60	#EradosGigantes	...	Üvey Evlat	Üç Harfliler 2: Hablis	Üç Harfliler 3: Karabüyü	Üç Harfliler: Adak	Üç Harfliler: Beddua	Üç Harflilerin Musallat Oldugu Büyülü Konakta Ruh Çagiran Gençlerin Hazin Hikayesi	Üç, Iki, Bir... Kestik!
Start_Year																		
2010	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2012	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2013	5.6	6.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2014	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	4.3

5 rows × 63434 columns

In [44]:
```
Delirium_Averagerating= movie_rating_user['Delirium']
Delirium_Averagerating.head()
```

Out[44]:
```
Start_Year
2010     NaN
2011     NaN
2012     NaN
2013    7.85
2014    4.50
Name: Delirium, dtype: float64
```

In [45]:
```
Delirium_Averagerating = movie_rating_user.corrwith(Delirium_Averagerating)
Delirium_Averagerating_corr = pd.DataFrame(Delirium_Averagerating, columns=['correlation'])
Delirium_Averagerating_corr.dropna(inplace=True)
Delirium_Averagerating_corr.head()
```

```
C:\Users\pinal\anaconda3\lib\site-packages\numpy\lib\function_base.py:2683: RuntimeWarning: Degrees of freedom <= 0 for slice
  c = cov(x, y, rowvar, dtype=dtype)
C:\Users\pinal\anaconda3\lib\site-packages\numpy\lib\function_base.py:2542: RuntimeWarning: divide by zero encountered in true_divide
  c *= np.true_divide(1, fact)
```

Out[45]:

	correlation
Original_Title	
1987	1.0
3/4	1.0
37	-1.0
6-5=2	1.0
7th Day	-1.0

In [46]:
```
Delirium_Averagerating_corr.sort_values('correlation', ascending= False).head(5)
```

Out[46]:

	correlation
Original_Title	
1987	1.0
Imago	1.0
Indivisible	1.0
Inhumane	1.0
Island	1.0

In [47]:
```
Delirium_Averagerating_corr= Delirium_Averagerating_corr.join(Averagerating['num of Averagerating'])
Delirium_Averagerating_corr.head()
```

Out[47]:

	correlation	num of Averagerating
Original_Title		
1987	1.0	2
3/4	1.0	2
37	-1.0	4
6-5=2	1.0	2
7th Day	-1.0	2

In [48]:
```
Delirium_Averagerating_corr[Delirium_Averagerating_corr['num of Averagerating']<100].sort_values('correlation', ascending=False)
```

Out[48]:

	correlation	num of Averagerating
Original_Title		
1987	1.0	2
Imago	1.0	2
Indivisible	1.0	2
Inhumane	1.0	2
Island	1.0	3
...	...	...
Grace	-1.0	4
Penthouse North	-1.0	2
The Butler	-1.0	2
Bloodline	-1.0	3
The Book of Life	-1.0	2

551 rows × 2 columns

In [ ]:

In [ ]: