Final Project Submission

Please fill out:

- Student name: ROSALINE WANGARI MUNGAI
- Student pace: part time
- Scheduled project review date/time: 18/2/2024
- Instructor name: NOAH KANDIE % WILLIAM OKOMBA
- Blog post URL:https://github.com/WangariR/Module-1-Final-Project/tree/main (https://github.com/WangariR/Module-1-Final-Project/tree/main)

In [147]:

```
#import the libraries needed to start our data analysis
import pandas as pd
import numpy as np



```

In [4]:

```
# install pandas new version into the kernel(to allow my jupyter note
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\user\anaconda3\lib\sit
e-packages (2.2.0)
Requirement already satisfied: numpy<2,>=1.23.2 in c:\users\user\anacond
a3\lib\site-packages (from pandas) (1.24.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\user\a
naconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\user\anaconda3\l
ib\site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\user\anaconda3
\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\s
ite-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [148]:

```
# importing the first csv file called the Bom movies CSV file
df= pd.read_csv(r"C:\Users\user\Documents\project phase 1\bom.movie_g
print("Moviest data read Successfully!")
print(df)
```

```
Moviest data read Successfully!
                                            title      studio  domestic_
gross  \
0                                     Toy Story 3          BV     415000
000.0
1                      Alice in Wonderland (2010)          BV     334200
000.0
2     Harry Potter and the Deathly Hallows Part 1          WB     296000
000.0
3                                       Inception          WB     292600
000.0
4                             Shrek Forever After        P/DW     238700
000.0
...                                           ...         ...
...
3382                                    The Quake       Magn.          6
200.0
3383                  Edward II (2018 re-release)          FM          4
800.0
3384                                     El Pacto        Sony          2
500.0
3385                                     The Swan  Synergetic          2
400.0
3386                            An Actor Prepares       Grav.          1
700.0

     foreign_gross  year
0        652000000  2010
1        691300000  2010
2        664300000  2010
3        535700000  2010
4        513900000  2010
...            ...   ...
3382           NaN  2018
3383           NaN  2018
3384           NaN  2018
3385           NaN  2018
3386           NaN  2018

[3387 rows x 5 columns]
```

In [149]:

```
#Check how many rows and columns the data set has
df.shape
```

Out[149]:

```
(3387, 5)
```

In [150]:

```
#Check the first 5 rows of the df content to see what we are working
df.head()
```

Out[150]:

	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

In [151]:

```
#Check how bom movies data looks like ie null values,data types,rows
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   title           3387 non-null   object 
 1   studio          3382 non-null   object 
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object 
 4   year            3387 non-null   int64  
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

In [152]:

```
#Drop any dublicates
df.drop_duplicates(inplace=True)
```

In [158]:

```
#Check how df data looks like ie null values,data types,rows and colu
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   title           3387 non-null   object
 1   studio          3387 non-null   object
 2   domestic_gross  3387 non-null   object
 3   year            3387 non-null   int64 
dtypes: int64(1), object(3)
memory usage: 106.0+ KB
```

In [157]:

```
#Drop null values after already filling some null values with a place
df = df.dropna(axis=1)
print(df)
```

```
                                            title      studio domestic_g
ross  \
0                                     Toy Story 3          BV    4150000
00.0
1                      Alice in Wonderland (2010)          BV    3342000
00.0
2     Harry Potter and the Deathly Hallows Part 1          WB    2960000
00.0
3                                       Inception          WB    2926000
00.0
4                             Shrek Forever After        P/DW    2387000
00.0
...                                           ...         ...
...
3382                                    The Quake       Magn.         62
00.0
3383                  Edward II (2018 re-release)          FM         48
00.0
3384                                     El Pacto        Sony         25
00.0
3385                                     The Swan  Synergetic         24
00.0
3386                            An Actor Prepares       Grav.         17
00.0

      year
0     2010
1     2010
2     2010
3     2010
4     2010
...    ...
3382  2018
3383  2018
3384  2018
3385  2018
3386  2018

[3387 rows x 4 columns]
```

In [156]:

```
#filling some null values with a place holder x so as not to loose the
df["studio"].fillna('x', inplace = True)
```

In [154]:

```
#filling some null values with a place holder 0 so as not to loose the
df["domestic_gross"].fillna('0', inplace = True)
```

In [159]:

```
# Read 2nd  data set called  the Title Basics CSV file
df2= pd.read_csv(r"C:\Users\user\Documents\project phase 1\title.basi
print("Title Movies data read Successfully!")
print(df2)
```

```
Title Movies data read Successfully!
           tconst                                    primary_title  \
0       tt0063540                                        Sunghursh
1       tt0066787                  One Day Before the Rainy Season
2       tt0069049                       The Other Side of the Wind
3       tt0069204                                  Sabse Bada Sukh
4       tt0100275                         The Wandering Soap Opera
...           ...                                              ...
146139  tt9916538                              Kuambil Lagi Hatiku
146140  tt9916622  Rodolpho Teóphilo - O Legado de um Pioneiro
146141  tt9916706                                  Dankyavar Danka
146142  tt9916730                                           6 Gunn
146143  tt9916754                   Chico Albuquerque - Revelações

                                     original_title  start_year  \
0                                         Sunghursh        2013
1                                   Ashad Ka Ek Din        2019
2                        The Other Side of the Wind        2018
3                                   Sabse Bada Sukh        2018
4                             La Telenovela Errante        2017
...                                             ...         ...
146139                          Kuambil Lagi Hatiku        2019
146140  Rodolpho Teóphilo - O Legado de um Pioneiro        2015
146141                              Dankyavar Danka        2013
146142                                       6 Gunn        2017
146143               Chico Albuquerque - Revelações        2013

        runtime_minutes                genres
0                 175.0    Action,Crime,Drama
1                 114.0       Biography,Drama
2                 122.0                 Drama
3                   NaN          Comedy,Drama
4                  80.0  Comedy,Drama,Fantasy
...                 ...                   ...
146139            123.0                 Drama
146140              NaN           Documentary
146141              NaN                Comedy
146142            116.0                   NaN
146143              NaN           Documentary

[146144 rows x 6 columns]
```

In [16]:

```
#Check how the data frame  looks like ie null values,data types,rows
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 146144 entries, 0 to 146143
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   tconst           146144 non-null  object 
 1   primary_title    146143 non-null  object 
 2   original_title   146122 non-null  object 
 3   start_year       146144 non-null  int64  
 4   runtime_minutes  114405 non-null  float64
 5   genres           140736 non-null  object 
dtypes: float64(1), int64(1), object(4)
memory usage: 6.7+ MB
```

In [163]:

```
#Check the first 5 rows of the  content  to see what we are workin wi
df2.head()
```

Out[163]:

	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

In [165]:

```
#filling some null values with a place holder y
df2["original_title"].fillna('y', inplace = True)
```

In [167]:

```
#filling some null values with a place holder z
df2["genres"].fillna('z', inplace = True)
```

In [168]:

```
#Drop null values
df2 = df2.dropna(axis=1)
print(df2)
```

```
           tconst                               original_title  start_ye
ar  \
0       tt0063540                                    Sunghursh        20
13
1       tt0066787                              Ashad Ka Ek Din        20
19
2       tt0069049                   The Other Side of the Wind        20
18
3       tt0069204                              Sabse Bada Sukh        20
18
4       tt0100275                        La Telenovela Errante        20
17
...           ...                                          ...
...
146139  tt9916538                          Kuambil Lagi Hatiku        20
19
146140  tt9916622  Rodolpho Teóphilo - O Legado de um Pioneiro        20
15
146141  tt9916706                              Dankyavar Danka        20
13
146142  tt9916730                                       6 Gunn        20
17
146143  tt9916754               Chico Albuquerque - Revelações        20
13

                      genres
0         Action,Crime,Drama
1            Biography,Drama
2                      Drama
3               Comedy,Drama
4       Comedy,Drama,Fantasy
...                      ...
146139                 Drama
146140           Documentary
146141                Comedy
146142                     z
146143           Documentary

[146144 rows x 4 columns]
```

In [169]:

```
#Drop dublicates
df2.drop_duplicates(inplace=True)
```

In [170]:

```
# Read the  3rd data frame called  Title ratings CSV file
df3= pd.read_csv(r"C:\Users\user\Documents\project phase 1\title.rati
print(df3)
```

```
          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
...           ...            ...       ...
73851   tt9805820            8.1        25
73852   tt9844256            7.5        24
73853   tt9851050            4.7        14
73854   tt9886934            7.0         5
73855   tt9894098            6.3       128

[73856 rows x 3 columns]
```

In [171]:

```
#Check how df data looks like ie null values,data types,rows and colu
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73856 entries, 0 to 73855
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   tconst         73856 non-null  object 
 1   averagerating  73856 non-null  float64
 2   numvotes       73856 non-null  int64  
dtypes: float64(1), int64(1), object(1)
memory usage: 1.7+ MB
```

In [172]:

```
#Drop dublicates
df3.drop_duplicates(inplace=True)
```

In [173]:

```
#Checking the contents of the first 5 rows
df3.head()
```

Out[173]:

	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

In [174]:

```
#Merging the title basics data frame with the title ratings data fram
new_df=pd.merge(df2, df3, on='tconst',how='inner')
print(new_df)
```

```
          tconst              original_title  start_year  \
0      tt0063540                   Sunghursh        2013   
1      tt0066787             Ashad Ka Ek Din        2019   
2      tt0069049  The Other Side of the Wind        2018   
3      tt0069204             Sabse Bada Sukh        2018   
4      tt0100275       La Telenovela Errante        2017   
...          ...                         ...         ...   
73851  tt9913084            Diabolik sono io        2019   
73852  tt9914286           Sokagin Çocuklari        2019   
73853  tt9914642                   Albatross        2017   
73854  tt9914942  La vida sense la Sara Amat        2019   
73855  tt9916160                  Drømmeland        2019   

                     genres  averagerating  numvotes  
0        Action,Crime,Drama            7.0        77  
1           Biography,Drama            7.2        43  
2                     Drama            6.9      4517  
3              Comedy,Drama            6.1        13  
4      Comedy,Drama,Fantasy            6.5       119  
...                     ...            ...       ...  
73851           Documentary            6.2         6  
73852          Drama,Family            8.7       136  
73853           Documentary            8.5         8  
73854                     z            6.6         5  
73855           Documentary            6.5        11  

[73856 rows x 6 columns]
```

In [175]:

```
#renaming  start_year column name to year to allow merge with unique
new_df.rename({'start_year':'year'},axis=1,inplace=True)
```

In [176]:

```
#checking information on the columns and data type
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73856 entries, 0 to 73855
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   tconst          73856 non-null  object 
 1   original_title  73856 non-null  object 
 2   year            73856 non-null  int64  
 3   genres          73856 non-null  object 
 4   averagerating   73856 non-null  float64
 5   numvotes        73856 non-null  int64  
dtypes: float64(1), int64(2), object(3)
memory usage: 3.4+ MB
```

In [179]:

```
#Merging the 3 data frames to provide one data frame to work with the
final_df=pd.merge(df, new_df, on='year',how='inner')
print(final_df)
```

```
                      title studio domestic_gross  year     tconst  \
0               Toy Story 3     BV    415000000.0  2010  tt0146592   
1               Toy Story 3     BV    415000000.0  2010  tt0154039   
2               Toy Story 3     BV    415000000.0  2010  tt0162942   
3               Toy Story 3     BV    415000000.0  2010  tt0230212   
4               Toy Story 3     BV    415000000.0  2010  tt0312305   
...                     ...    ...            ...   ...        ...   
27090564  An Actor Prepares  Grav.         1700.0  2018  tt9899840   
27090565  An Actor Prepares  Grav.         1700.0  2018  tt9899880   
27090566  An Actor Prepares  Grav.         1700.0  2018  tt9903952   
27090567  An Actor Prepares  Grav.         1700.0  2018  tt9904014   
27090568  An Actor Prepares  Grav.         1700.0  2018  tt9908960   

                                  original_title                      ge
nres  \
0                                    Pál Adrienn                       D
rama   
1                                 Oda az igazság                     His
tory   
2                       A zöld sárkány gyermekei                       D
rama   
3                              The Final Journey                       D
rama   
4         Quantum Quest: A Cassini Space Odyssey  Adventure,Animation,Sc
i-Fi   
...                                          ...                        
...   
27090564                       Khaleh Ghurbagheh     Adventure,Comedy,Fa
mily   
27090565                                Columbus                      Co
medy   
27090566             BADMEN with a good behavior               Comedy,Ho
rror   
27090567                         Lost in Klessin                        
War   
27090568                                 Pliusas                      Co
medy   

          averagerating  numvotes  
0                   6.8       451  
1                   4.6        64  
2                   6.9       120  
3                   8.8         8  
4                   5.1       287  
...                 ...       ...  
27090564            6.2         6  
27090565            5.8         5  
27090566            9.2         5  
27090567            7.3        12  
27090568            4.2        13  

[27090569 rows x 9 columns]
```

In [181]:

```
#Showing how my desired ouput of the 3 data sets looks like
final_df.head()
```

Out[181]:

	title	studio	domestic_gross	year	tconst	original_title	genres	av
0	Toy Story 3	BV	415000000.0	2010	tt0146592	Pál Adrienn	Drama	
1	Toy Story 3	BV	415000000.0	2010	tt0154039	Oda az igazság	History	
2	Toy Story 3	BV	415000000.0	2010	tt0162942	A zöld sárkány gyermekei	Drama	
3	Toy Story 3	BV	415000000.0	2010	tt0230212	The Final Journey	Drama	
4	Toy Story 3	BV	415000000.0	2010	tt0312305	Quantum Quest: A Cassini Space Odyssey	Adventure,Animation,Sci-Fi	

In [182]:

```
#Drop dublicates
final_df.drop_duplicates(inplace=True)
```

In [225]:

```
final_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27090569 entries, 0 to 27090568
Data columns (total 9 columns):
 #   Column          Dtype  
---  ------          -----  
 0   title           object 
 1   studio          object 
 2   domestic_gross  object 
 3   year            int64  
 4   tconst          object 
 5   original_title  object 
 6   genres          object 
 7   averagerating   float64
 8   numvotes        int64  
dtypes: float64(1), int64(2), object(6)
memory usage: 1.8+ GB
```

In [187]:

```
#sorting the values in ascending order using the num votes to see the
   final_df.sort_values(by="numvotes", ascending=False )
```

Out[187]:

	title	studio	domestic_gross	year	tconst	original_title	
598748	The Next Three Days	LGF	21100000.0	2010	tt1375666	Inception	Action,A
1610756	The Salvation Poem (Poema de Salvacion)	CZ	915000.0	2010	tt1375666	Inception	Action,A
979100	Country Strong	SGem	20200000.0	2010	tt1375666	Inception	Action,A
1128524	Looking for Eric	IFC	55800.0	2010	tt1375666	Inception	Action,A
795716	My Name is Khan	FoxS	4000000.0	2010	tt1375666	Inception	Action,A
...	...	...	...	...	...	...	
4552287	Illegal (2011)	FM	700.0	2011	tt1950377	Thank You for Judging	Comedy,Docur
10003105	Therese	MPI	102000.0	2013	tt2996696	Reject	Docu
3855741	Carancho	Strand	85500.0	2011	tt1780916	My Dinner with A.J.	C
5275005	Brave	BV	237300000.0	2012	tt2643342	The Greatest Wish	
9261189	New World (2013)	WGUSA	458000.0	2013	tt3247664	Behind the Freedom Curtain	

27090569 rows × 9 columns

In [ ]:

```
#sorting the values in ascending order using the domestic_gross to se
   final_df.sort_values(by="domestic_gross ", ascending=False )
```

In [188]:

```
# find top 10 movies in the bo  using numvotes
final_df.nlargest(10, ['numvotes'])
```

Out[188]:

	title	studio	domestic_gross	year	tconst	original_title	gen
1052	Toy Story 3	BV	415000000.0	2010	tt1375666	Inception	Action,Adventure,S
7844	Alice in Wonderland (2010)	BV	334200000.0	2010	tt1375666	Inception	Action,Adventure,S
14636	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	2010	tt1375666	Inception	Action,Adventure,S
21428	Inception	WB	292600000.0	2010	tt1375666	Inception	Action,Adventure,S
28220	Shrek Forever After	P/DW	238700000.0	2010	tt1375666	Inception	Action,Adventure,S
35012	The Twilight Saga: Eclipse	Sum.	300500000.0	2010	tt1375666	Inception	Action,Adventure,S
41804	Iron Man 2	Par.	312400000.0	2010	tt1375666	Inception	Action,Adventure,S
48596	Tangled	BV	200800000.0	2010	tt1375666	Inception	Action,Adventure,S
55388	Despicable Me	Uni.	251500000.0	2010	tt1375666	Inception	Action,Adventure,S
62180	How to Train Your Dragon	P/DW	217600000.0	2010	tt1375666	Inception	Action,Adventure,S

In [224]:

```
# find the most used production cmpany in studio column
final_df['studio'].mode()
```

Out[224]:

```
0    IFC
Name: studio, dtype: object
```

In [223]:

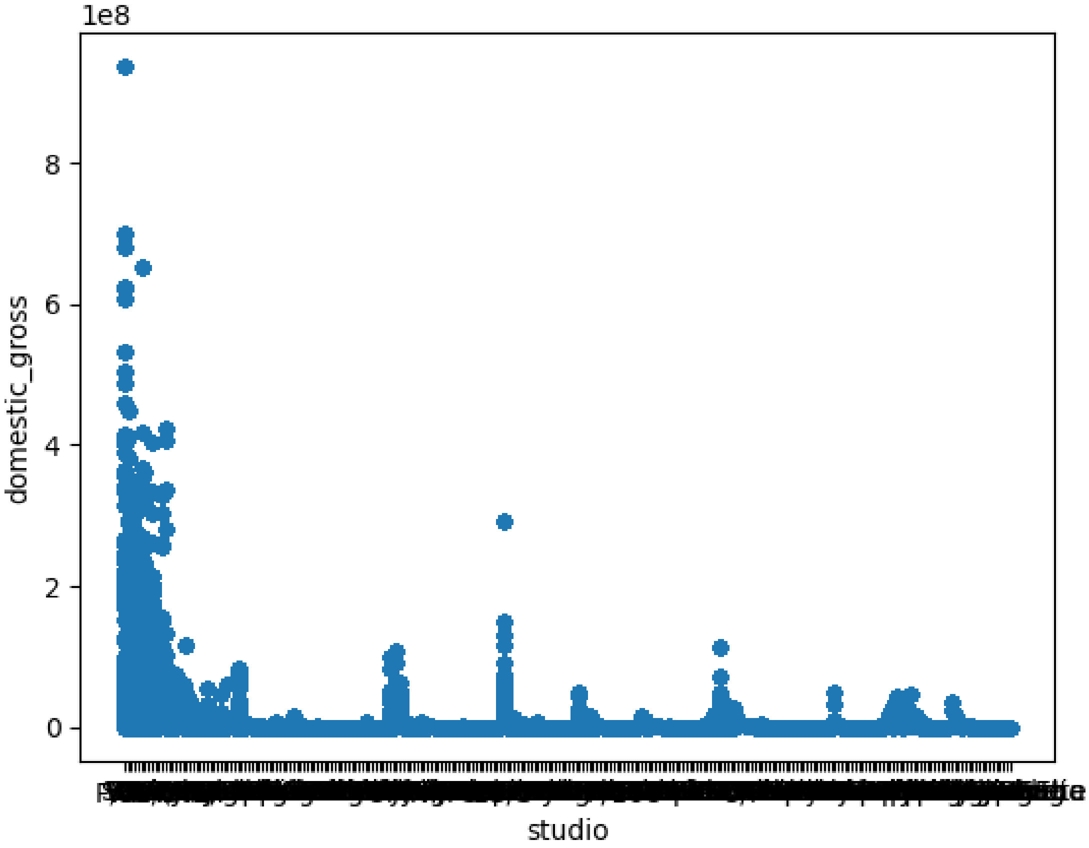
```
#find movie title  that is used most
final_df['title'].value_counts().idxmax()
```

Out[223]:

```
'Bluebeard'
```

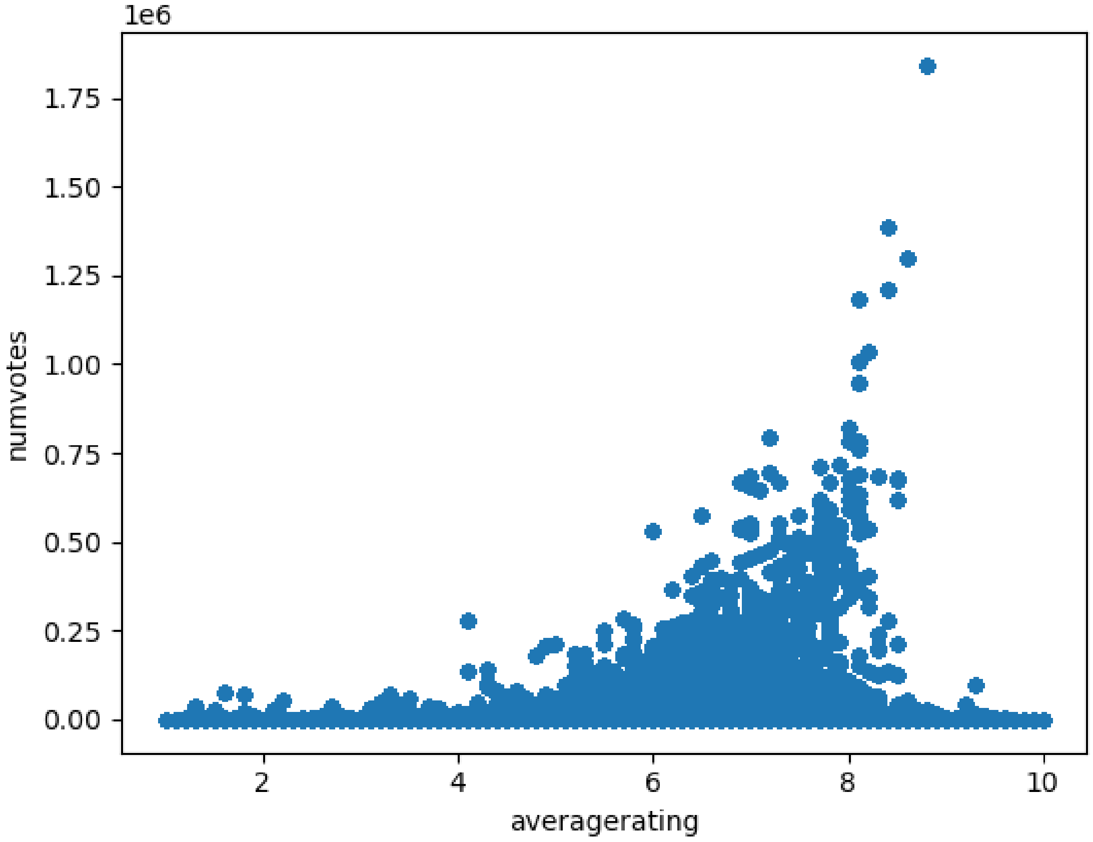
In [210]:

```
#Making a scatter plot to show correlation between the type of studio
import pandas as pd
import matplotlib.pyplot as plt
final_df.plot(kind = 'scatter', x = 'studio', y = 'domestic_gross')

plt.show()
```



In [212]:

```
#Making a scatter plot to show correlation between the numvotes and r
import pandas as pd
import matplotlib.pyplot as plt
final_df.plot(kind = 'scatter', x = 'averagerating', y = 'numvotes')

plt.show()
```



In [194]:

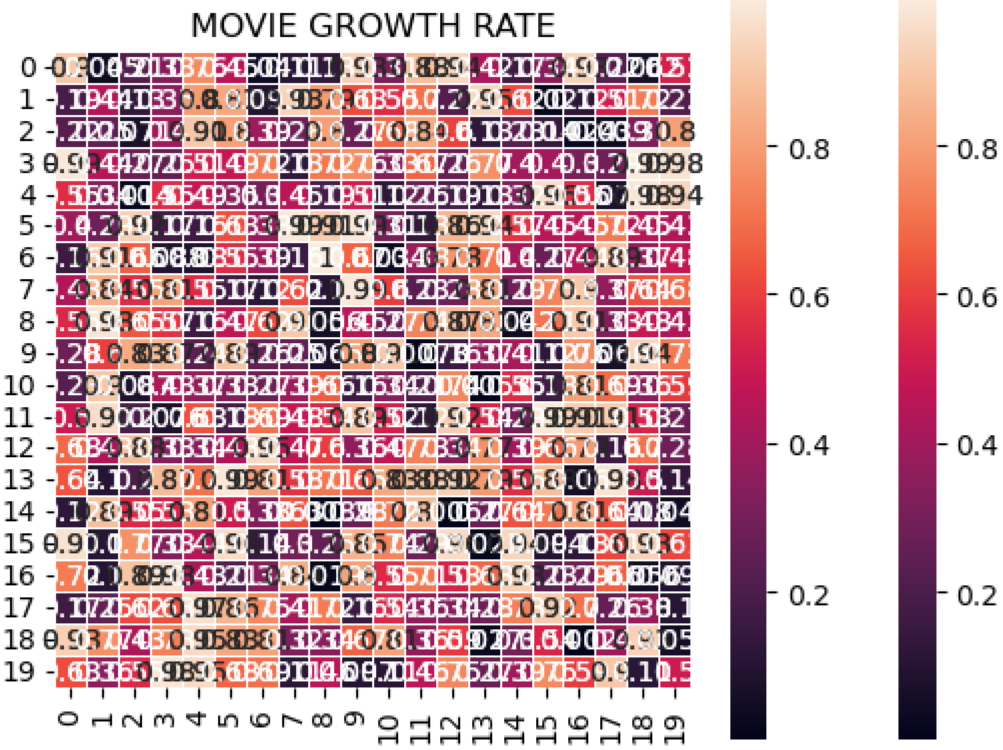
```
#Check correlation between year,average rating and num votes
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

final_df.corr(method='pearson',min_periods=1, numeric_only = True)

```

Out[194]:

	year	averagerating	numvotes
year	1.000000	0.026788	-0.024855
averagerating	0.026788	1.000000	0.046086
numvotes	-0.024855	0.046086	1.000000

In [230]:

```


# Creating a 10x10 array with my data frame information
final_df.corr = np.random.rand(20,20)



# Creating a heatmap using imshow()
plt.imshow(final_df.corr, cmap='hot', interpolation='nearest')

# Turn long format into a wide format-----
final_df.corr = final_df.pivot_table( index='genres', columns='studio
#Showing values inside the heat map
sns.heatmap(final_df.corr, annot=True)
#Changing the width of the middle white lines
sns.heatmap(final_df.corr, annot=True, linewidth=.5)
#NAME THE TITLE
plt.title("MOVIE GROWTH RATE")
plt.show()


```


MOVIE GROWTH RATE