Final Project Submission

Please fill out:

- Student name: Stephen Njoroge
- Student pace: part time
- Scheduled project review date/time: 18/9/2023
- Instructor name: Faith Rotich
- Blog post URL:

Imports

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

#Load the datasets

In [2]:
```
rating = pd.read_csv("zippedData/imdb.title.ratings.csv.gz")
rating
```

Out[2]:

	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

In [3]:
```
gross = pd.read_csv("zippedData/bom.movie_gross.csv.gz")
gross
```

Out[3]:

	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
...	...	...	...	...	...
3382	The Quake	Magn.	6200.0	NaN	2018
3383	Edward II (2018 re-release)	FM	4800.0	NaN	2018
3384	El Pacto	Sony	2500.0	NaN	2018
3385	The Swan	Synergetic	2400.0	NaN	2018
3386	An Actor Prepares	Grav.	1700.0	NaN	2018

3387 rows × 5 columns

In [4]:
```
basic = pd.read_csv("zippedData/imdb.title.basics.csv.gz")
basic
```

Out[4]:

	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
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	Drama
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Documentary
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	NaN	Comedy
146142	tt9916730	6 Gunn	6 Gunn	2017	116.0	NaN
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 6 columns

In [5]:
```
basic = basic.rename(columns = {'start_year' : 'year'})
basic
```

Out[5]:

	tconst	primary_title	original_title	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
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	Drama
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Documentary
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	NaN	Comedy
146142	tt9916730	6 Gunn	6 Gunn	2017	116.0	NaN
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 6 columns

#Merging of the three datasets

In [6]:
```
merged_df = pd.merge(rating, basic, on = 'tconst', how = 'inner' )
merged_df = pd.merge(merged_df, gross, on = 'year', how = 'inner')
merged_df
```

Out[6]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	
...	...	...	...	...	...	...	...	
27090564	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090565	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090566	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090567	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090568	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A

27090569 rows × 12 columns

Data Understanding

1 Dataframe shape 2 Head and tail 3 Columns 4 dtypes 5 Describe 6 Adding of column 7 Info

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

Out[7]: (27090569, 12)

In [8]:
```
merged_df.head()
```

Out[8]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

In [9]:
```
merged_df.tail()
```

Out[9]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	
27090564	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090565	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090566	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090567	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A
27090568	tt9844256	7.5	24	Code Geass: Lelouch of the Rebellion - Glorifi...	Code Geass: Lelouch of the Rebellion Episode III	2018	120.0	A

In [10]:
```
merged_df.columns  # Check column names
```

Out[10]:
```
Index(['tconst', 'averagerating', 'numvotes', 'primary_title',
       'original_title', 'year', 'runtime_minutes', 'genres', 'title',
       'studio', 'domestic_gross', 'foreign_gross'],
      dtype='object')
```

In [11]:
```
merged_df.dtypes # Check columns data types
```

Out[11]:
```
tconst              object
averagerating      float64
numvotes             int64
primary_title       object
original_title      object
year                 int64
runtime_minutes    float64
genres              object
title               object
studio              object
domestic_gross     float64
foreign_gross       object
dtype: object
```

In [12]:
```
merged_df['foreign_gross'] = merged_df['foreign_gross'].str.replace(',', '').asty
```

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

Out[13]:

	averagerating	numvotes	year	runtime_minutes	domestic_gross	foreign_gross
count	2.709057e+07	2.709057e+07	2.709057e+07	2.436889e+07	2.687127e+07	1.602377e+07
mean	6.318168e+00	3.606623e+03	2.014088e+03	9.447218e+01	2.866995e+07	7.621023e+07
std	1.469343e+00	3.074989e+04	2.417634e+00	2.173196e+02	6.719182e+07	1.384432e+08
min	1.000000e+00	5.000000e+00	2.010000e+03	3.000000e+00	1.000000e+02	6.000000e+02
25%	5.400000e+00	1.400000e+01	2.012000e+03	8.100000e+01	1.170000e+05	3.900000e+06
50%	6.500000e+00	5.000000e+01	2.014000e+03	9.000000e+01	1.300000e+06	1.960000e+07
75%	7.300000e+00	2.890000e+02	2.016000e+03	1.030000e+02	2.770000e+07	7.700000e+07
max	1.000000e+01	1.841066e+06	2.018000e+03	5.142000e+04	9.367000e+08	9.605000e+08

Adding another column

In [14]:
```
merged_df['total_gross'] = merged_df['domestic_gross'] + merged_df['foreign_gross
merged_df.head()
```

Out[14]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

In [15]:
```
merged_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27090569 entries, 0 to 27090568
Data columns (total 13 columns):
 #   Column           Dtype  
---  ------           -----  
 0   tconst           object 
 1   averagerating    float64
 2   numvotes         int64  
 3   primary_title    object 
 4   original_title   object 
 5   year             int64  
 6   runtime_minutes  float64
 7   genres           object 
 8   title            object 
 9   studio           object 
 10  domestic_gross   float64
 11  foreign_gross    float64
 12  total_gross      float64
dtypes: float64(5), int64(2), object(6)
memory usage: 2.8+ GB
```

Data Preparation

1 Dropping columns and rows 2 Checking duplicates

In [16]:
```
merged_df.head()
```

Out[16]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

In [17]:
```
merged_df['first_word'] = merged_df['genres'].str.split(',').str[0] #strip genre
merged_df.head()
```

Out[17]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

Dropping of a column

In [18]:
```
merged_df.drop(columns = ["primary_title", "original_title", "genres"])
```

Out[18]:

	tconst	averagerating	numvotes	year	runtime_minutes	title	studio	dome
0	tt1042974	6.4	20	2010	90.0	Toy Story 3	BV	41
1	tt1042974	6.4	20	2010	90.0	Alice in Wonderland (2010)	BV	33
2	tt1042974	6.4	20	2010	90.0	Harry Potter and the Deathly Hallows Part 1	WB	29
3	tt1042974	6.4	20	2010	90.0	Inception	WB	29
4	tt1042974	6.4	20	2010	90.0	Shrek Forever After	P/DW	23
...	...	...	...	...	...	...	...	
27090564	tt9844256	7.5	24	2018	120.0	The Quake	Magn.	
27090565	tt9844256	7.5	24	2018	120.0	Edward II (2018 re-release)	FM	
27090566	tt9844256	7.5	24	2018	120.0	El Pacto	Sony	
27090567	tt9844256	7.5	24	2018	120.0	The Swan	Synergetic	
27090568	tt9844256	7.5	24	2018	120.0	An Actor Prepares	Grav.	

27090569 rows × 11 columns

In [19]:
```
merged_df.drop_duplicates() #dropping of duplicates
```

Out[19]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0
...	...	...	...	...	...	...	...

In [20]:
```
merged_df.shape
```

Out[20]:
```
(27090569, 14)
```

Dealing With Missing Values 1 Check missing values 2 Remove missing values

In [21]:
```
merged_df.isna().sum()
```

Out[21]:
```
tconst                   0
averagerating            0
numvotes                 0
primary_title            0
original_title           0
year                     0
runtime_minutes    2721678
genres              303586
title                    0
studio               38945
domestic_gross      219302
foreign_gross     11066795
total_gross       11286097
first_word          303586
dtype: int64
```

#Dealing With Missing Data 1Find the 5% threshold

In [22]:
```
threshold = len(merged_df)*0.05 # 5% threshold
threshold
```

Out[22]:
```
1354528.4500000002
```

In [23]:
```
cols_to_drop = merged_df.columns[merged_df.isna().sum() <= threshold]
cols_to_drop
```

Out[23]:
```
Index(['tconst', 'averagerating', 'numvotes', 'primary_title',
       'original_title', 'year', 'genres', 'title', 'studio', 'domestic_gross',
       'first_word'],
      dtype='object')
```

In [24]:
```
merged_df.dropna(subset = cols_to_drop, inplace = True)
merged_df.head()
```

Out[24]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

filter again columns with missing values

In [25]:
```
cols_with_missing_values = merged_df.columns[merged_df.isna().sum() > 0]
cols_with_missing_values
```

Out[25]:
```
Index(['runtime_minutes', 'foreign_gross', 'total_gross'], dtype='object')
```

In [26]:
```
merged_df.isna().sum()
```

Out[26]:
```
tconst                   0
averagerating            0
numvotes                 0
primary_title            0
original_title           0
year                     0
runtime_minutes     2589344
genres                   0
title                    0
studio                   0
domestic_gross           0
foreign_gross     10938469
total_gross       10938469
first_word               0
dtype: int64
```

In [27]:
```
studio_runtime = merged_df.groupby('studio')['runtime_minutes'].median()
studio_runtime.head()
```

Out[27]:
```
studio
3D     90.0
A23    90.0
A24    91.0
ADC    91.0
AF     90.0
Name: runtime_minutes, dtype: float64
```

In [28]:
```
runtime_dict = studio_runtime.to_dict()
```

In [29]:
```
merged_df ['runtime_minutes'] = merged_df['runtime_minutes'].fillna(merged_df['st
```

In [30]:
```
merged_df.isna().sum()
```

Out[30]:
```
tconst                   0
averagerating            0
numvotes                 0
primary_title            0
original_title           0
year                     0
runtime_minutes          0
genres                   0
title                    0
studio                   0
domestic_gross           0
foreign_gross      10938469
total_gross        10938469
first_word               0
dtype: int64
```

In [31]:
```
studio_foreign = merged_df.groupby('studio')['foreign_gross'].median()
studio_foreign.head()
```

Out[31]:
```
studio
3D     9900000.0
A23          NaN
A24    9700000.0
ADC          NaN
AF     1750000.0
Name: foreign_gross, dtype: float64
```

In [32]:
```
foreign_dict = studio_foreign.to_dict()
```

In [33]:
```
merged_df ['foreign_gross'] = merged_df['foreign_gross'].fillna(merged_df['studio
```

In [34]:
```
merged_df.isna().sum()
```

Out[34]:
```
tconst                0
averagerating         0
numvotes              0
primary_title         0
original_title        0
year                  0
runtime_minutes       0
genres                0
title                 0
studio                0
domestic_gross        0
foreign_gross       1117497
total_gross        10938469
first_word            0
dtype: int64
```

In [35]:
```
threshold = len(merged_df)*0.05
threshold
```

Out[35]:
```
1327399.4500000002
```

In [36]:
```
cols_to_drop = merged_df.columns[merged_df.isna().sum() <= threshold]
cols_to_drop
```

Out[36]:
```
Index(['tconst', 'averagerating', 'numvotes', 'primary_title',
       'original_title', 'year', 'runtime_minutes', 'genres', 'title',
       'studio', 'domestic_gross', 'foreign_gross', 'first_word'],
      dtype='object')
```

In [37]:
```
merged_df.dropna(subset = cols_to_drop, inplace = True)
merged_df.head()
```

Out[37]:

	tconst	averagerating	numvotes	primary_title	original_title	year	runtime_minutes	genres
0	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
1	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
2	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
3	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama
4	tt1042974	6.4	20	Just Inès	Just Inès	2010	90.0	Drama

In [38]:
```
cols_with_missing_values = merged_df.columns[merged_df.isna().sum() > 0]
cols_with_missing_values
```

Out[38]:
```
Index(['total_gross'], dtype='object')
```

In [39]:
```
merged_df.isna().sum()
```

Out[39]:
```
tconst                0
averagerating         0
numvotes              0
primary_title         0
original_title        0
year                  0
runtime_minutes       0
genres                0
title                 0
studio                0
domestic_gross        0
foreign_gross         0
total_gross        9820972
first_word            0
dtype: int64
```

In [40]:
```
studio_gross = merged_df.groupby('studio')['total_gross'].median()
studio_gross.head()
```

Out[40]:
```
studio
3D     16000000.0
A24    19100000.0
AF      2327500.0
AGF      176800.0
AR     58050000.0
Name: total_gross, dtype: float64
```

In [41]:
```
gross_dict = studio_gross.to_dict()
```

In [42]:
```
merged_df ['total_gross'] = merged_df['total_gross'].fillna(merged_df['studio'].r
```

In [43]:
```
merged_df.isna().sum()
```

Out[43]:
```
tconst             0
averagerating      0
numvotes           0
primary_title      0
original_title     0
year               0
runtime_minutes    0
genres             0
title              0
studio             0
domestic_gross     0
foreign_gross      0
total_gross        0
first_word         0
dtype: int64
```

#Descibe With Summary Statistics

In [44]:
```
merged_df.describe()
```

Out[44]:

	averagerating	numvotes	year	runtime_minutes	domestic_gross	foreign_gross
count	2.543049e+07	2.543049e+07	2.543049e+07	2.543049e+07	2.543049e+07	2.543049e+07
mean	6.315997e+00	3.657963e+03	2.014056e+03	9.416497e+01	2.993948e+07	5.356237e+07
std	1.468610e+00	3.100674e+04	2.423694e+00	2.092711e+02	6.843092e+07	1.143444e+08
min	1.000000e+00	5.000000e+00	2.010000e+03	3.000000e+00	1.000000e+02	6.000000e+02
25%	5.400000e+00	1.400000e+01	2.012000e+03	8.200000e+01	1.310000e+05	2.900000e+06
50%	6.500000e+00	5.100000e+01	2.014000e+03	9.000000e+01	1.600000e+06	9.900000e+06
75%	7.300000e+00	2.960000e+02	2.016000e+03	1.010000e+02	3.050000e+07	4.960000e+07
max	1.000000e+01	1.841066e+06	2.018000e+03	5.142000e+04	9.367000e+08	9.605000e+08

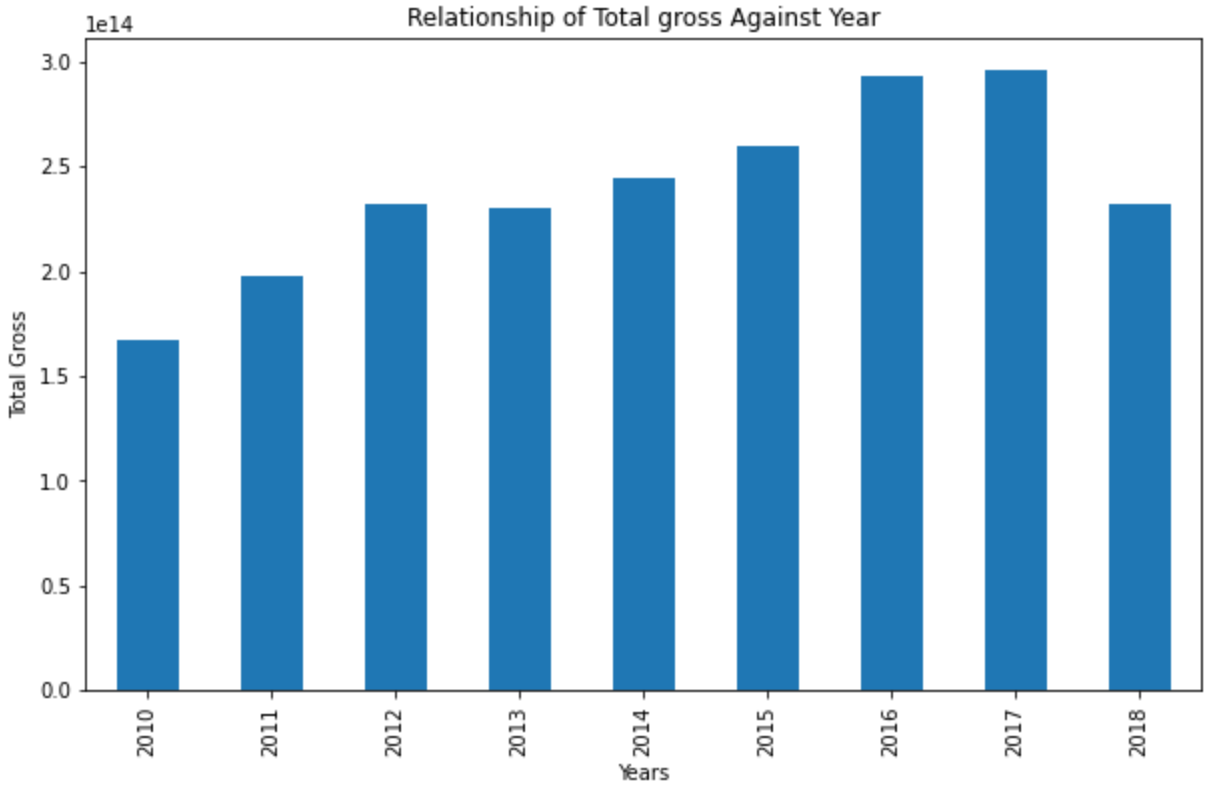
In [ ]:
```

```

#Plot calculations

Q1. Is tota_gross different during different years

In [45]:
```
gross= merged_df.groupby('year')['total_gross'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Total Gross')
plt.set_title('Relationship of Total gross Against Year')
```

Out[45]:
```
Text(0.5, 1.0, 'Relationship of Total gross Against Year')
```



In [46]:
```
total= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
total
```

```
<ipython-input-46-e72d477ab609>:1: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list instea
d.
  total= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
```

Out[46]:

year	domestic_gross	foreign_gross
2010	6.806063e+13	9.775215e+13
2011	7.319210e+13	1.225105e+14
2012	8.267151e+13	1.468275e+14
2013	8.526550e+13	1.426771e+14
2014	8.542614e+13	1.563084e+14
2015	9.276127e+13	1.633392e+14
2016	9.681405e+13	1.910681e+14
2017	9.440813e+13	1.964745e+14
2018	8.277638e+13	1.451599e+14

In [47]:
```
gross= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
gross
```

```
<ipython-input-47-5e7cbeeeb756>:1: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list instea
d.
  gross= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
```

Out[47]:

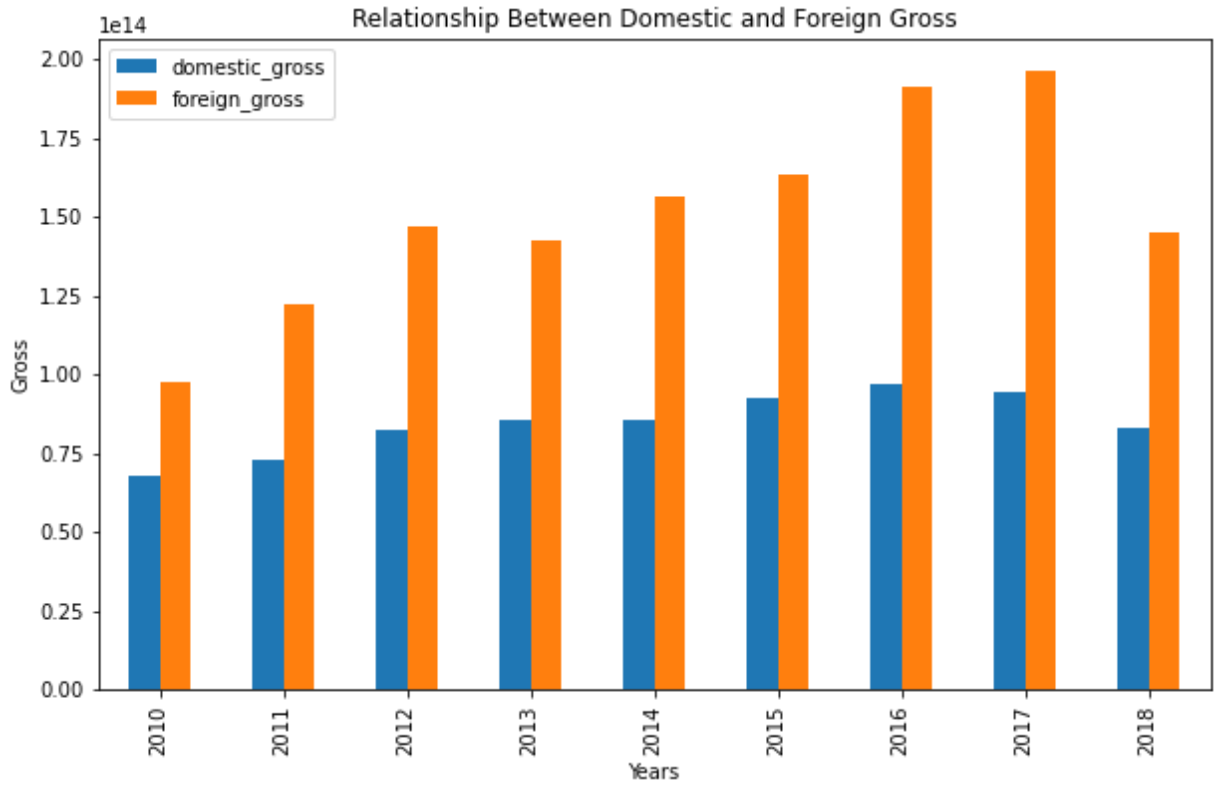
year	domestic_gross	foreign_gross
2010	6.806063e+13	9.775215e+13
2011	7.319210e+13	1.225105e+14
2012	8.267151e+13	1.468275e+14
2013	8.526550e+13	1.426771e+14
2014	8.542614e+13	1.563084e+14
2015	9.276127e+13	1.633392e+14
2016	9.681405e+13	1.910681e+14
2017	9.440813e+13	1.964745e+14
2018	8.277638e+13	1.451599e+14

Q2: Is Foreign gross different from Domestic grosss

In [48]:
```
gross= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Gross')
plt.set_title('Relationship Between Domestic and Foreign Gross')
```

```
<ipython-input-48-e83751207236>:1: FutureWarning: Indexing with multiple keys
(implicitly converted to a tuple of keys) will be deprecated, use a list instea
d.
  gross= merged_df.groupby('year')['domestic_gross', 'foreign_gross'].sum()
```

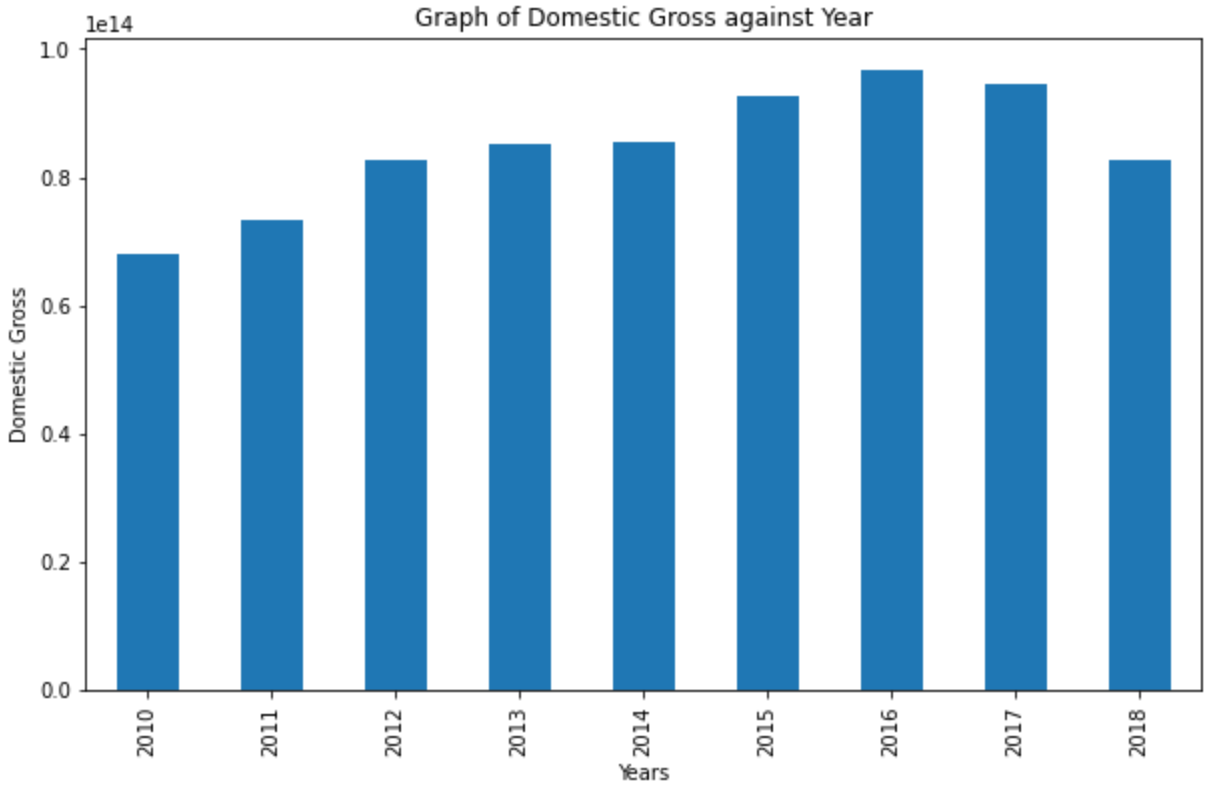
Out[48]: Text(0.5, 1.0, 'Relationship Between Domestic and Foreign Gross')



Q3: Do years have a difference on domestic gross

In [49]:
```
gross= merged_df.groupby('year')['domestic_gross'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Domestic Gross')
plt.set_title('Graph of Domestic Gross against Year')
```

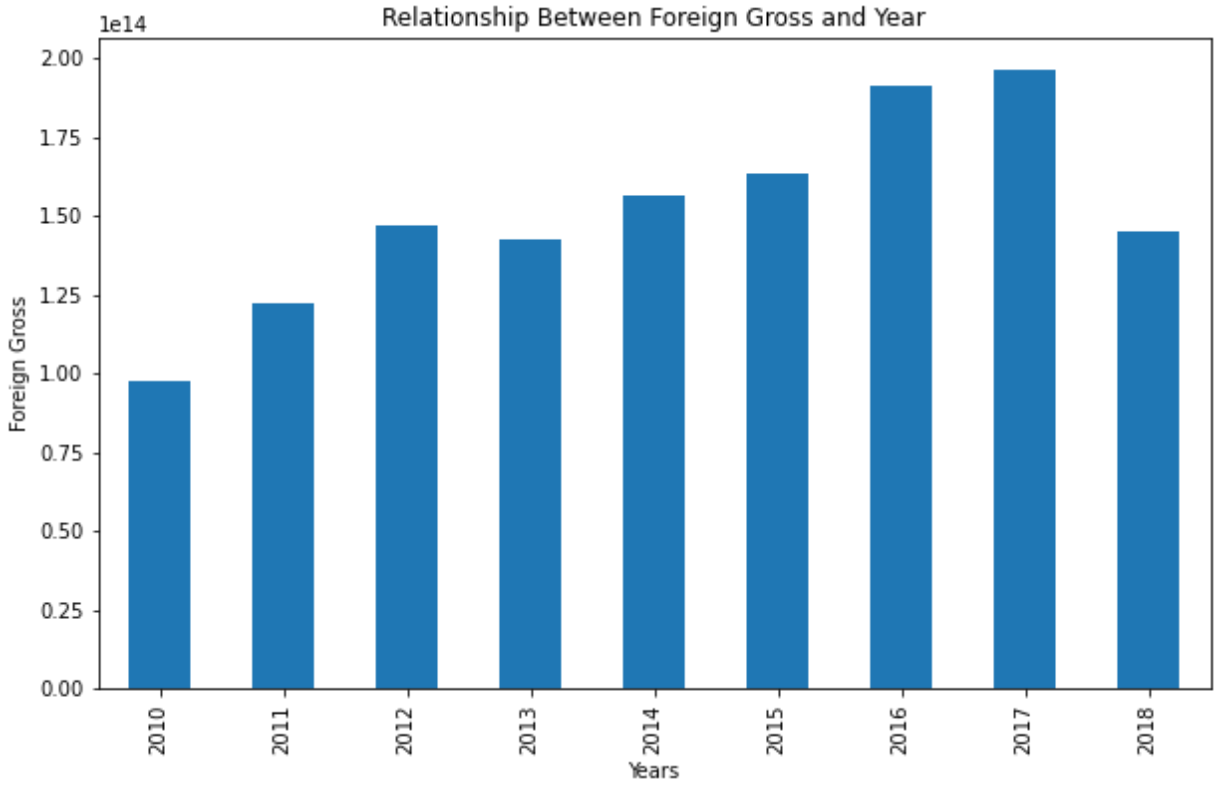
Out[49]:
```
Text(0.5, 1.0, 'Graph of Domestic Gross against Year')
```



Q4: Do years have a difference on foreign gross

In [50]:
```
gross= merged_df.groupby('year')['foreign_gross'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Foreign Gross')
plt.set_title('Relationship Between Foreign Gross and Year')
```

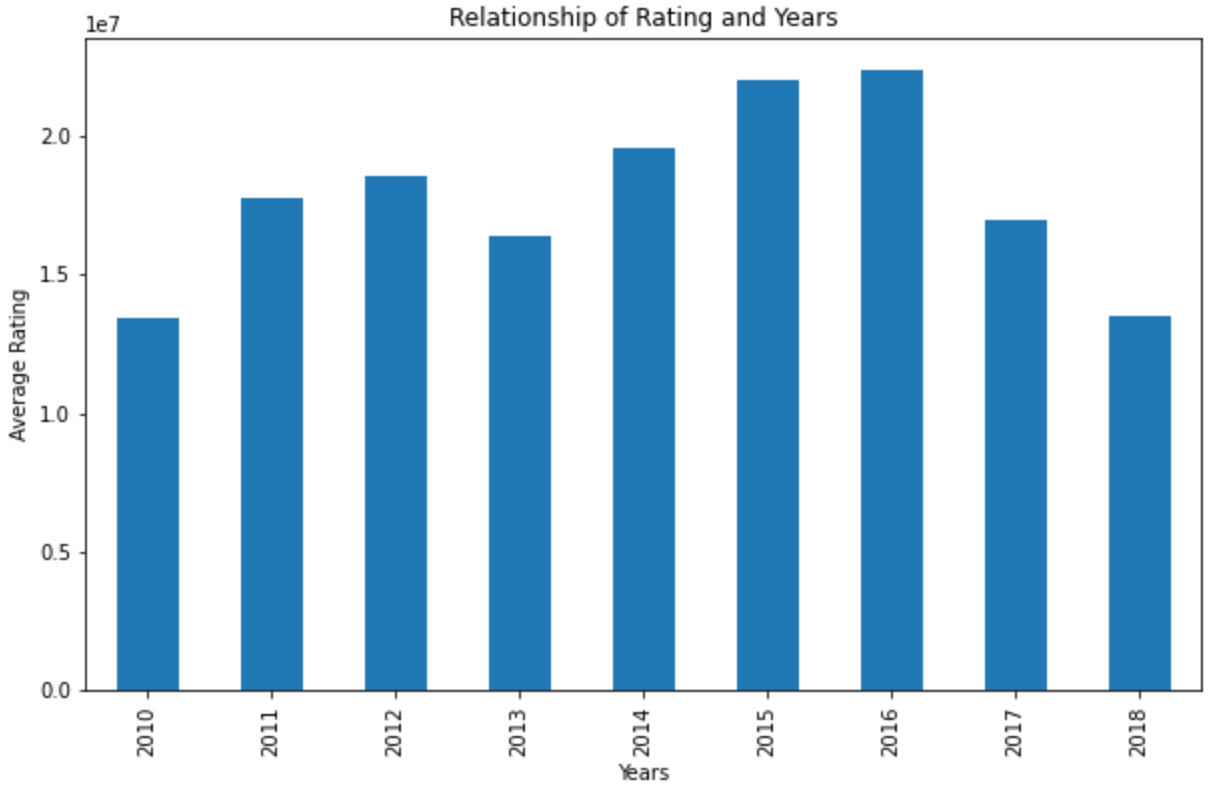
Out[50]:
```
Text(0.5, 1.0, 'Relationship Between Foreign Gross and Year')
```



Q5:Do rating change over the years

In [51]:
```
gross= merged_df.groupby('year')['averagerating'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Average Rating')
plt.set_title('Relationship of Rating and Years')
```

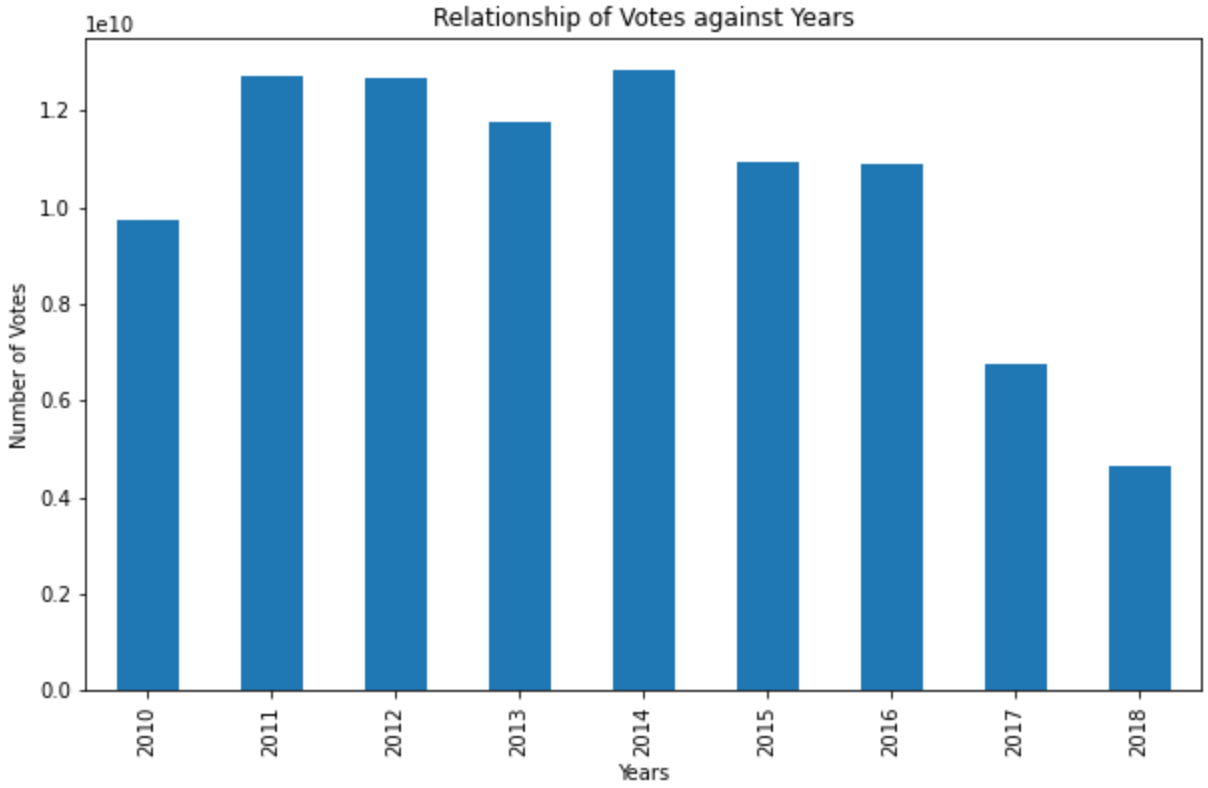
Out[51]:
```
Text(0.5, 1.0, 'Relationship of Rating and Years')
```



Q6: Is number of votes affected by year

In [52]:
```
gross= merged_df.groupby('year')['numvotes'].sum()
plt = gross.plot(kind = 'bar', figsize = (10,6))
plt.set_xlabel('Years')
plt.set_ylabel('Number of Votes')
plt.set_title('Relationship of Votes against Years')
```

Out[52]: Text(0.5, 1.0, 'Relationship of Votes against Years')



#Checking for Correlation In our dataset

In [53]:
```
merged_df.corr()
```

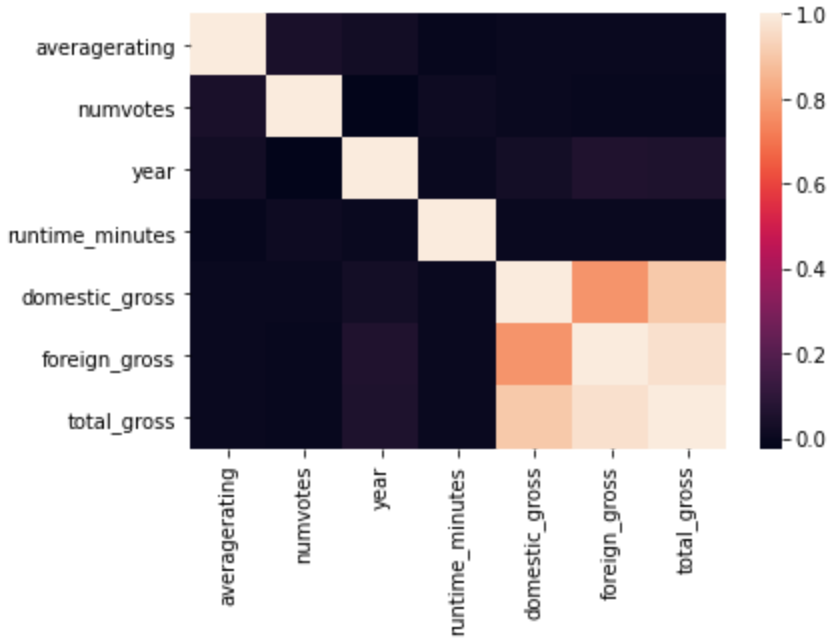
Out[53]:

	averagerating	numvotes	year	runtime_minutes	domestic_gross	foreign_gro
averagerating	1.000000	0.046568	0.026679	-0.007250	0.001328	0.0024
numvotes	0.046568	1.000000	-0.025084	0.012136	-0.000751	-0.0015
year	0.026679	-0.025084	1.000000	0.000627	0.027366	0.0629
runtime_minutes	-0.007250	0.012136	0.000627	1.000000	0.000092	0.0001
domestic_gross	0.001328	-0.000751	0.027366	0.000092	1.000000	0.7736
foreign_gross	0.002467	-0.001594	0.062996	0.000170	0.773684	1.0000
total_gross	0.002246	-0.001420	0.055059	0.000156	0.905418	0.9680

#Use of Heatmap to show Correlated columns

In [54]:
```
sns.heatmap(merged_df.corr())
```

Out[54]: <AxesSubplot:>



#Groupimg of Data

In [55]:
```
merged_df.groupby('studio').mean()
```

Out[55]:

studio	averagerating	numvotes	year	runtime_minutes	domestic_gross	foreign_gross	
3D	6.256245	4549.161021	2010.000000	92.345471	6.100000e+06	9.900000e+06	1
A24	6.331197	3304.752504	2015.400840	93.609765	6.595252e+06	1.074238e+07	2
AF	6.297276	4079.213893	2013.377591	94.714330	3.531439e+05	1.750000e+06	2
AGF	6.287352	4500.906379	2011.000000	93.901292	1.580000e+04	1.610000e+05	1
AR	6.344421	3090.628817	2016.000000	93.517241	3.500000e+05	5.770000e+07	5
...	...	...	...	...	...	...	
WOW	6.256245	4549.161021	2010.000000	92.345471	3.080000e+04	1.860000e+04	4
Wein.	6.302461	3969.304512	2013.394827	94.057699	1.980137e+07	3.509394e+07	5
Yash	6.319882	3507.242583	2014.298152	94.836094	2.511849e+06	6.506352e+07	7
Zee	6.344421	3090.628817	2016.000000	93.517241	1.100000e+06	5.710000e+05	1
Zeit.	6.301605	4066.520760	2013.045564	93.757012	3.435228e+05	3.771485e+06	4

172 rows × 7 columns

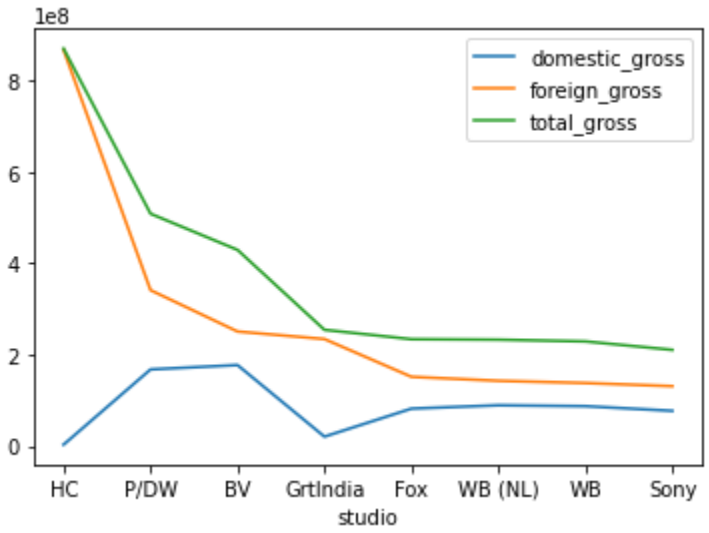
In [56]:
```
studio_df = merged_df.groupby('studio')[['domestic_gross', 'foreign_gross', 'tota
studio_df
```

Out[56]:

studio	domestic_gross	foreign_gross	total_gross
HC	2.700000e+06	8.676000e+08	8.703000e+08
P/DW	1.674887e+08	3.411118e+08	5.086005e+08
BV	1.770272e+08	2.505467e+08	4.293699e+08
GrtIndia	2.020000e+07	2.340000e+08	2.542000e+08
Fox	8.160329e+07	1.512828e+08	2.336867e+08
WB (NL)	8.896499e+07	1.425370e+08	2.326244e+08
WB	8.698103e+07	1.376677e+08	2.288317e+08
Sony	7.674292e+07	1.307795e+08	2.103304e+08

In [57]:
```
studio_df.plot()
```

Out[57]: <AxesSubplot:xlabel='studio'>



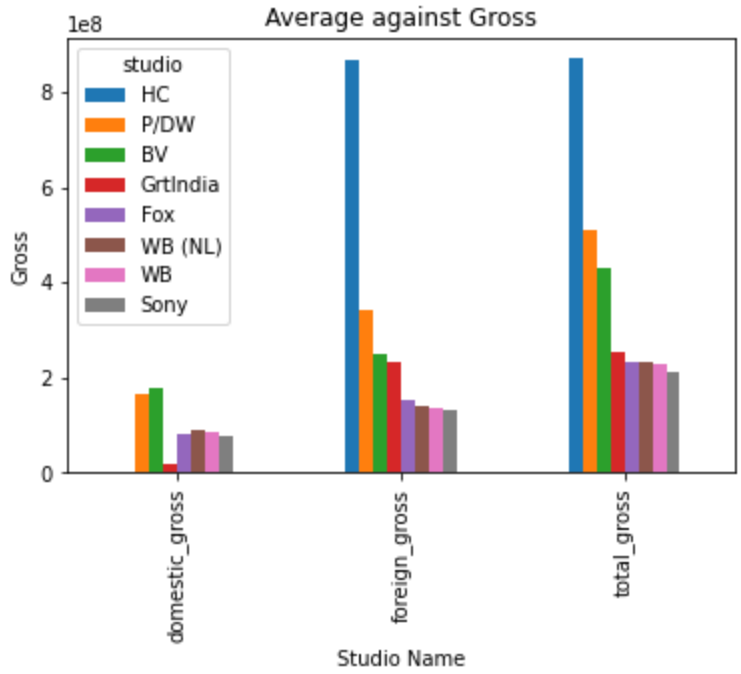
In [58]:
```
studio_df_2 = studio_df.transpose()
studio_df_2
```

Out[58]:

studio	HC	P/DW	BV	GrtIndia	Fox	WB (NL)
domestic_gross	2700000.0	1.674887e+08	1.770272e+08	20200000.0	8.160329e+07	8.896499e+07
foreign_gross	867600000.0	3.411118e+08	2.505467e+08	234000000.0	1.512828e+08	1.425370e+08
total_gross	870300000.0	5.086005e+08	4.293699e+08	254200000.0	2.336867e+08	2.326244e+08

In [59]:
```
ax = studio_df_2.plot(kind = 'bar')
ax.set_xlabel('Studio Name')
ax.set_ylabel('Gross')
ax.set_title('Average against Gross')
```

Out[59]:
```
Text(0.5, 1.0, 'Average against Gross')
```



In [60]:
```
merged_df.groupby('first_word').mean()
```

Out[60]:

first_word	averagerating	numvotes	year	runtime_minutes	domestic_gross	foreign_g
Action	5.796700	14768.313538	2014.064614	102.361185	2.994405e+07	5.358514
Adult	2.000000	128.000000	2015.000000	120.000000	2.640296e+07	4.649182
Adventure	6.378338	10720.949640	2014.034012	90.400316	2.981947e+07	5.333743
Animation	6.263256	2137.228849	2014.065828	83.123348	2.993126e+07	5.350744
Biography	7.178217	5143.546256	2013.814918	89.864761	2.945055e+07	5.229083
Comedy	5.978380	2860.671755	2014.007050	96.336034	2.991528e+07	5.340343
Crime	6.144144	5369.936148	2014.160955	96.417991	2.979998e+07	5.355338
Documentary	7.307931	213.604882	2014.072282	87.560697	2.991831e+07	5.365006
Drama	6.345706	2284.806061	2014.037470	97.256038	2.998921e+07	5.356550
Family	5.980186	476.908061	2014.383745	90.676625	3.039924e+07	5.490103
Fantasy	5.673360	1418.646097	2014.052578	91.299754	2.981097e+07	5.352715
Game-Show	9.000000	7.000000	2013.000000	130.000000	3.268568e+07	5.469383
History	6.453183	94.396665	2014.236212	99.020418	3.026259e+07	5.413001
Horror	4.827821	2378.021205	2014.143879	87.514415	3.005734e+07	5.399054
Music	7.484954	208.432509	2013.437892	98.695413	2.930717e+07	5.161153
Musical	6.602964	148.090143	2014.109500	103.505467	3.028784e+07	5.395067
Mystery	6.069136	5614.129497	2014.259182	96.818780	3.033322e+07	5.454966
News	5.463817	10.832719	2013.334562	93.319823	3.136632e+07	5.577417
Reality-TV	5.921154	24.524887	2014.200226	121.020362	3.059289e+07	5.142568
Romance	5.997018	631.036093	2014.350097	104.941837	3.030121e+07	5.445705
Sci-Fi	5.312885	571.005044	2014.418277	88.492750	3.036278e+07	5.531815
Sport	6.906397	57.832358	2013.875572	89.817644	2.983083e+07	5.370081
Thriller	5.677116	305.118764	2014.359669	93.850718	3.041121e+07	5.470171
War	6.240573	119.870770	2014.127311	92.877359	3.035378e+07	5.288620
Western	4.950109	211.674252	2014.407302	89.807824	3.031112e+07	5.449654

In [61]:
```
merged_df_2 = merged_df.groupby('first_word')[['domestic_gross', 'foreign_gross',
merged_df_2
```

Out[61]:

first_word	domestic_gross	foreign_gross	total_gross
Game-Show	3.268568e+07	5.469383e+07	8.834008e+07
News	3.136632e+07	5.577417e+07	8.829434e+07
Sci-Fi	3.036278e+07	5.531815e+07	8.699295e+07
Family	3.039924e+07	5.490103e+07	8.659028e+07
Thriller	3.041121e+07	5.470171e+07	8.639736e+07
Mystery	3.033322e+07	5.454966e+07	8.615581e+07
Western	3.031112e+07	5.449654e+07	8.611017e+07
Romance	3.030121e+07	5.445705e+07	8.603038e+07
History	3.026259e+07	5.413001e+07	8.566303e+07
Musical	3.028784e+07	5.395067e+07	8.548621e+07
Horror	3.005734e+07	5.399054e+07	8.528849e+07
Documentary	2.991831e+07	5.365006e+07	8.479587e+07
Drama	2.998921e+07	5.356550e+07	8.477608e+07
Action	2.994405e+07	5.358514e+07	8.475347e+07
Sport	2.983083e+07	5.370081e+07	8.474760e+07
Animation	2.993126e+07	5.350744e+07	8.466578e+07
Crime	2.979998e+07	5.355338e+07	8.458208e+07
Fantasy	2.981097e+07	5.352715e+07	8.456343e+07
Comedy	2.991528e+07	5.340343e+07	8.453150e+07
War	3.035378e+07	5.288620e+07	8.447838e+07
Adventure	2.981947e+07	5.333743e+07	8.436333e+07
Reality-TV	3.059289e+07	5.142568e+07	8.315820e+07
Biography	2.945055e+07	5.229083e+07	8.288797e+07
Music	2.930717e+07	5.161153e+07	8.201517e+07
Adult	2.640296e+07	4.649182e+07	7.395031e+07

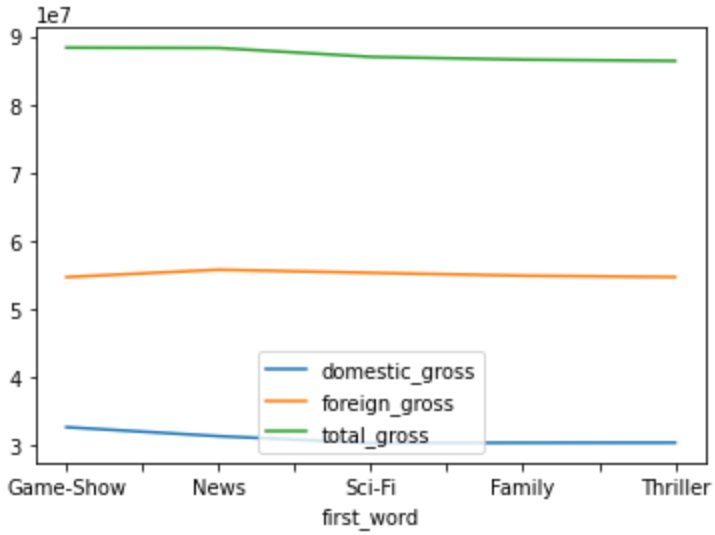
In [62]:
```
merged_df_2 = merged_df.groupby('first_word')[['domestic_gross', 'foreign_gross',
merged_df_2
```

Out[62]:

first_word	domestic_gross	foreign_gross	total_gross
Game-Show	3.268568e+07	5.469383e+07	8.834008e+07
News	3.136632e+07	5.577417e+07	8.829434e+07
Sci-Fi	3.036278e+07	5.531815e+07	8.699295e+07
Family	3.039924e+07	5.490103e+07	8.659028e+07
Thriller	3.041121e+07	5.470171e+07	8.639736e+07

In [63]:
```
merged_df_2.plot()
```

Out[63]:
```
<AxesSubplot:xlabel='first_word'>
```



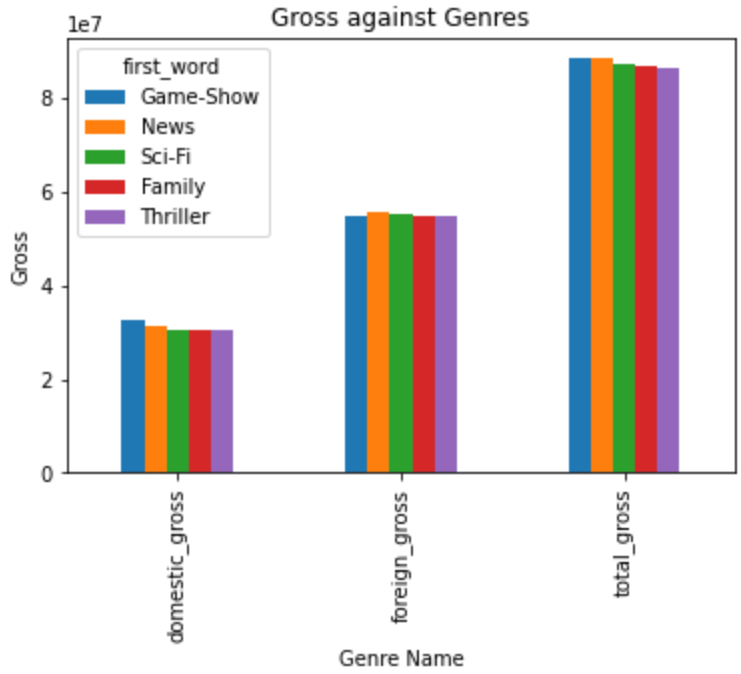
In [64]:
```
merged_df_3 = merged_df_2.transpose()
merged_df_3
```

Out[64]:

first_word	Game-Show	News	Sci-Fi	Family	Thriller
domestic_gross	3.268568e+07	3.136632e+07	3.036278e+07	3.039924e+07	3.041121e+07
foreign_gross	5.469383e+07	5.577417e+07	5.531815e+07	5.490103e+07	5.470171e+07
total_gross	8.834008e+07	8.829434e+07	8.699295e+07	8.659028e+07	8.639736e+07

In [65]:
```
ax = merged_df_3.plot(kind = 'bar')
ax.set_xlabel('Genre Name')
ax.set_ylabel('Gross')
ax.set_title('Gross against Genres')
```

Out[65]:
```
Text(0.5, 1.0, 'Gross against Genres')
```



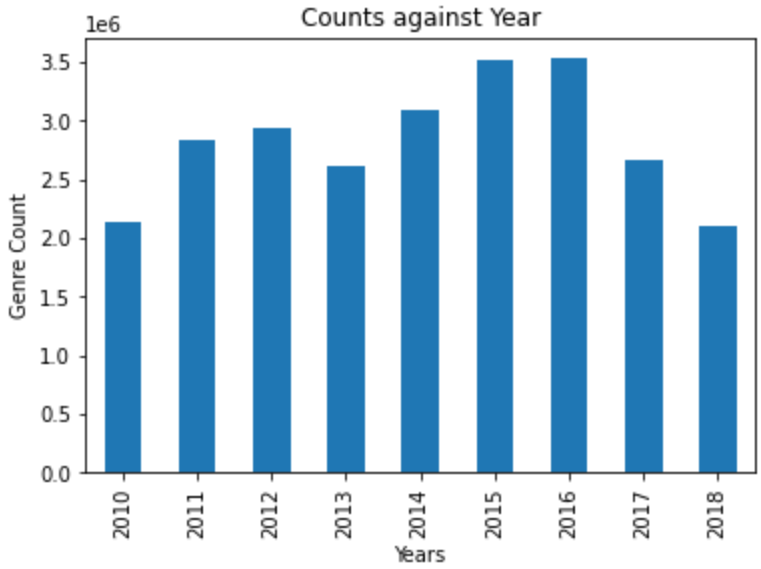
#Count of movies created

In [66]:
```
merged_df_4 = merged_df.groupby('year')['first_word'].count()
merged_df_4
```

Out[66]:
```
year
2010    2144320
2011    2829586
2012    2941974
2013    2608650
2014    3092606
2015    3513290
2016    3531330
2017    2660504
2018    2108232
Name: first_word, dtype: int64
```

In [67]:
```
ax = merged_df_4.plot(kind = 'bar')
ax.set_xlabel('Years')
ax.set_ylabel('Genre Count')
ax.set_title('Counts against Year')
```

Out[67]:
```
Text(0.5, 1.0, 'Counts against Year')
```



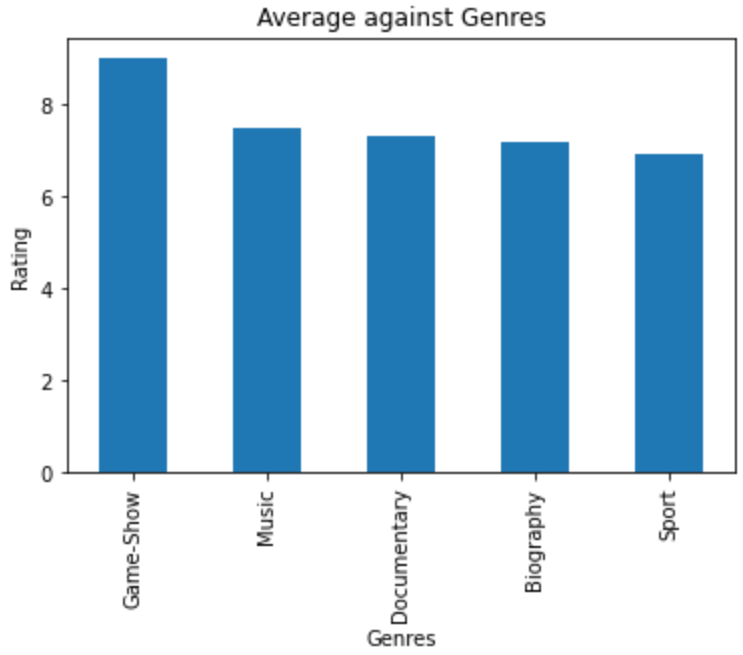
#Relationship Between Averagerating and Genres

In [68]:
```
genre_ratings = merged_df.groupby('first_word')['averagerating'].mean().sort_valu
genre_ratings
```

Out[68]:
```
first_word
Game-Show      9.000000
Music          7.484954
Documentary    7.307931
Biography      7.178217
Sport          6.906397
Name: averagerating, dtype: float64
```

In [69]:
```
ax = genre_ratings.plot(kind = 'bar')
ax.set_xlabel('Genres')
ax.set_ylabel('Rating')
ax.set_title('Average against Genres')
```

Out[69]:
```
Text(0.5, 1.0, 'Average against Genres')
```

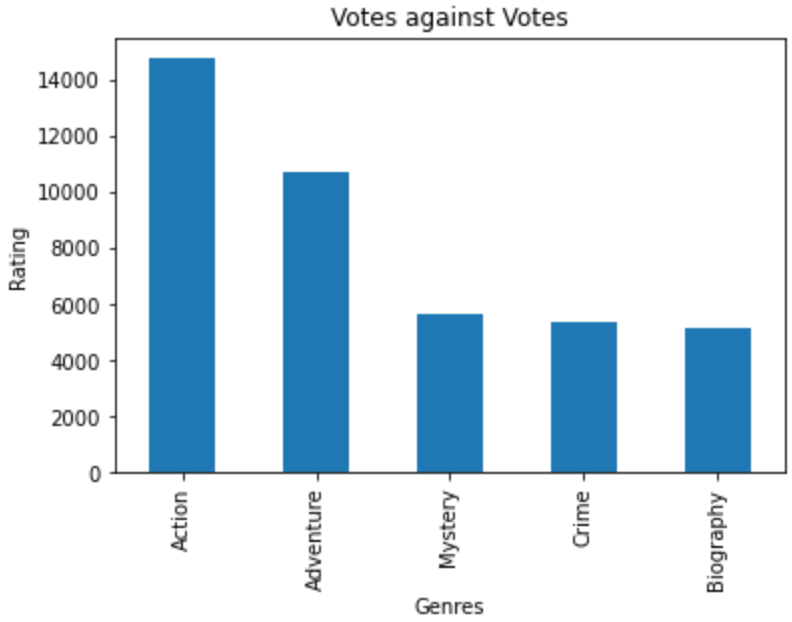


In [70]:
```
genre_votes = merged_df.groupby('first_word')['numvotes'].mean().sort_values(asce
genre_votes
```

Out[70]:
```
first_word
Action       14768.313538
Adventure    10720.949640
Mystery       5614.129497
Crime         5369.936148
Biography     5143.546256
Name: numvotes, dtype: float64
```

In [71]:
```
ax = genre_votes.plot(kind = 'bar')
ax.set_xlabel('Genres')
ax.set_ylabel('Rating')
ax.set_title('Votes against Votes')
```

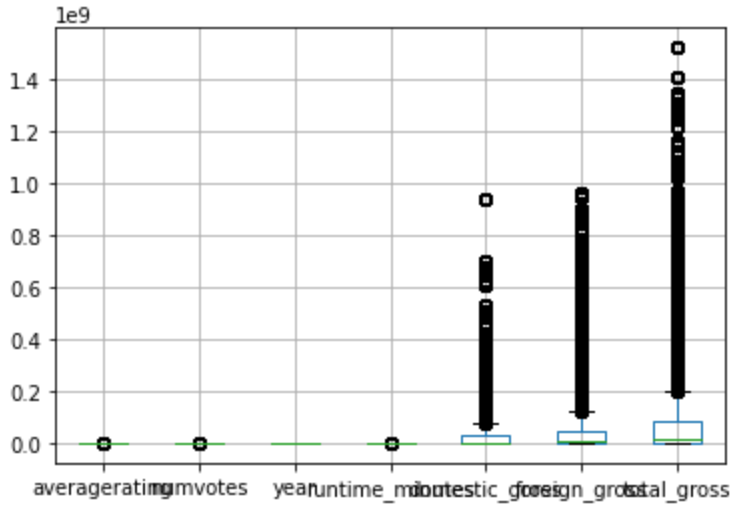
Out[71]: Text(0.5, 1.0, 'Votes against Votes')



#Checking For outliers

In [72]:
```
merged_df.boxplot()
```

Out[72]: <AxesSubplot:>



#How to remove outliers

CONCLUSION

In coclusion, data analysis has provided valuable insights from the datasets providing information to help in business starategies. Throghoutthe project I have been able read, prepare, clean and

RECOMMENDATIONS

This are my recommendations; 1. Microsoft to focus more on foreign sales than domestic sales. Foreign sales are more with highest sales recorded in a year at 1.964745e+14 while domestic gross was at 9.681405e+13.
2. For the copmany to make more money year on year, more movies need to be produced in subsequent years. In the year 2017 and 2018, 2660504 and 2108232 a drop from previous years. This lead to a deacrease in gross sales in 2018 which had a gross sales of 8.277638e+13 in foreign sales and 1.451599e+14 in domestic sales. 3. Focus to be more in producing movie genre of Game-Show. They have the highest foreign gross of 5.469383e+07 and also have the highest ratings and are rated 9.000000 out of 10.

In [ ]:

In [ ]:

In [ ]: