```
# Importing Required libraries

import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (12,8)

# Reading the data and while reading the reading data faced issues
with encoding so had to add these parameters

df = pd.read_csv('movies.csv',encoding = 'unicode_escape', engine
='python')


# Now let's take a look at the data

df.head()

       budget                                  company  ...
writer  year
0   8000000.0           Columbia Pictures Corporation  ...   Stephen
King  1986
1   6000000.0                      Paramount Pictures  ...    John
Hughes  1986
2  15000000.0                      Paramount Pictures  ...       Jim
Cash  1986
3  18500000.0  Twentieth Century Fox Film Corporation  ...  James
Cameron  1986
4   9000000.0                    Walt Disney Pictures  ...  Mark H.
Baker  1986

[5 rows x 15 columns]

df.columns

Index(['budget', 'company', 'country', 'director', 'genre', 'gross',
'name',
       'rating', 'released', 'runtime', 'score', 'star', 'votes',
'writer',
       'year'],
      dtype='object')

#Checking the central tendancy
df.describe()
```

```
             budget         gross  ...         votes         year
count  6.820000e+03  6.820000e+03  ...  6.820000e+03  6820.000000
mean   2.458113e+07  3.349783e+07  ...  7.121952e+04  2001.000293
std    3.702254e+07  5.819760e+07  ...  1.305176e+05     8.944501
min    0.000000e+00  7.000000e+01  ...  2.700000e+01  1986.000000
25%    0.000000e+00  1.515839e+06  ...  7.665250e+03  1993.000000
50%    1.100000e+07  1.213568e+07  ...  2.589250e+04  2001.000000
75%    3.200000e+07  4.006534e+07  ...  7.581225e+04  2009.000000
max    3.000000e+08  9.366622e+08  ...  1.861666e+06  2016.000000

[8 rows x 6 columns]
```

```python
#checking for the missing values

df.isnull().mean()*100
```

```
budget      0.0
company     0.0
country     0.0
director    0.0
genre       0.0
gross       0.0
name        0.0
rating      0.0
released    0.0
runtime     0.0
score       0.0
star        0.0
votes       0.0
writer      0.0
year        0.0
dtype: float64
```

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6820 entries, 0 to 6819
Data columns (total 15 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   budget    6820 non-null   float64
 1   company   6820 non-null   object 
 2   country   6820 non-null   object 
 3   director  6820 non-null   object 
 4   genre     6820 non-null   object 
 5   gross     6820 non-null   float64
 6   name      6820 non-null   object 
 7   rating    6820 non-null   object 
 8   released  6820 non-null   object 
 9   runtime   6820 non-null   int64  
 10  score     6820 non-null   float64
```

```
 11  star      6820 non-null   object
 12  votes     6820 non-null   int64
 13  writer    6820 non-null   object
 14  year      6820 non-null   int64
dtypes: float64(3), int64(3), object(9)
memory usage: 799.3+ KB
```

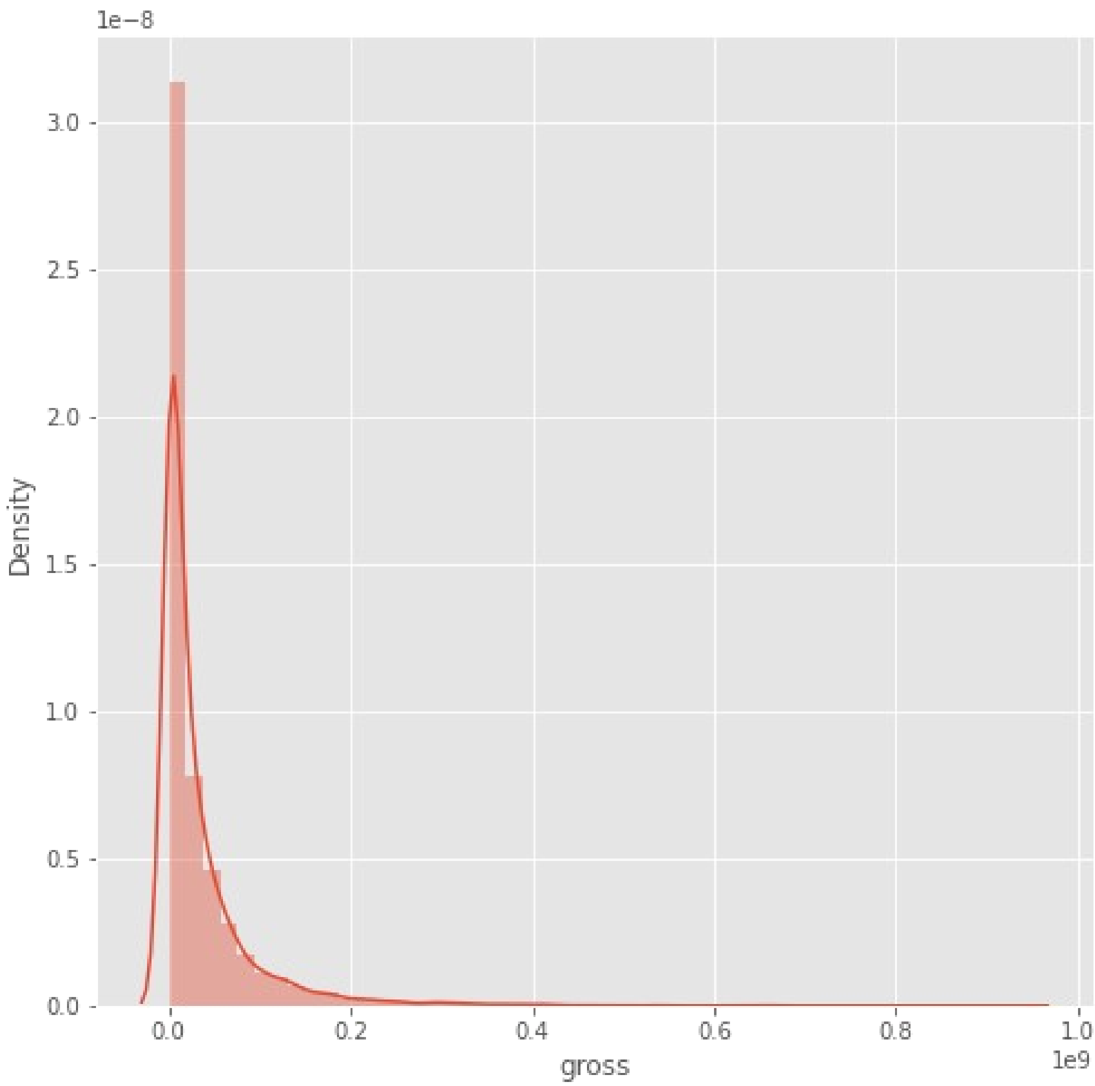
```
# Data Types for our columns

print(df.dtypes)
```

```
budget      float64
company      object
country      object
director     object
genre        object
gross       float64
name         object
rating       object
released     object
runtime       int64
score       float64
star         object
votes         int64
writer       object
year          int64
dtype: object
```

```
#Checking the gross using dist plot
fig, ax = plt.subplots(figsize=(8,8))
sns.distplot(df.gross)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed
in a future version. Please adapt your code to use either `displot` (a
figure-level function with similar flexibility) or `histplot` (an
axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f66a582cd50>
```




```
# displaying  the whole array

pd.set_option('display.max_rows', None)
df
```

Output hidden; open in https://colab.research.google.com to view.

```
# found year (which is the release year of the movie ) is not getting synced with the released column , so creating the new column

df['Correct_year']=df['released'].astype(str).str[:4]

df
```

Output hidden; open in https://colab.research.google.com to view.

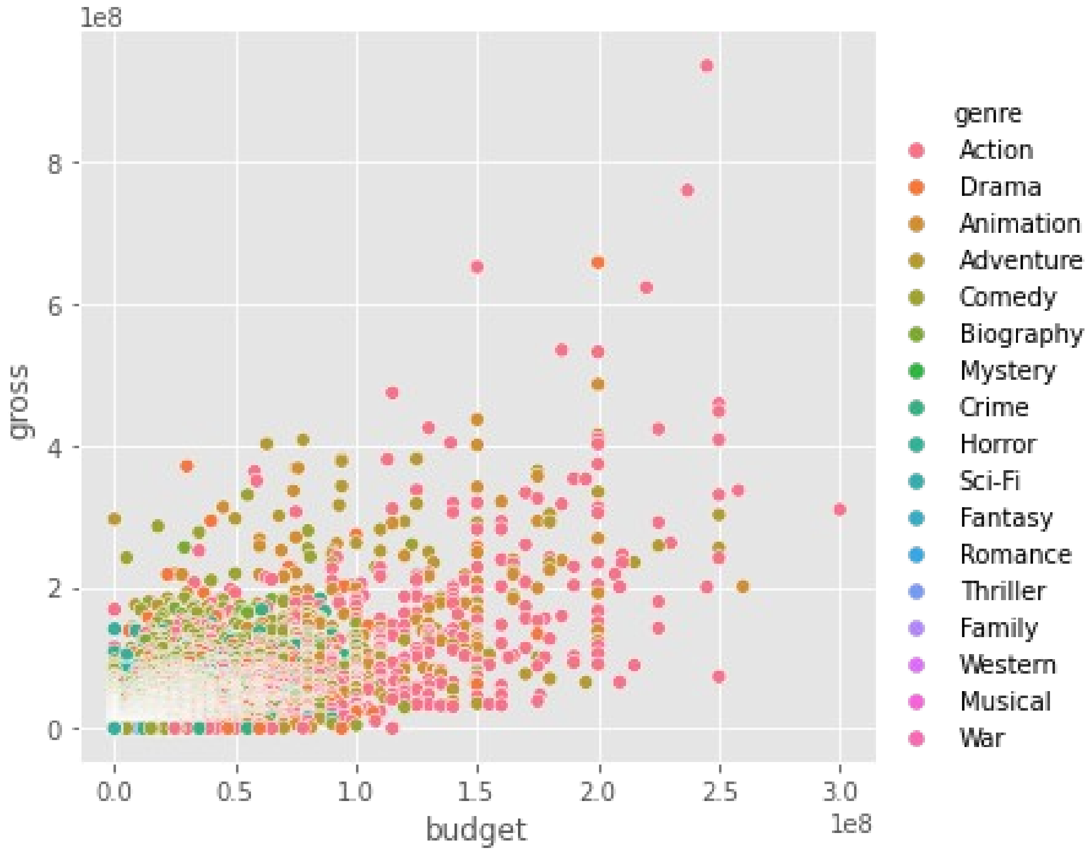
```
#sorting the data set based on the gross
df.sort_values(by=['gross'],inplace=True,ascending=False)
```

```
df
```

Output hidden; open in https://colab.research.google.com to view.

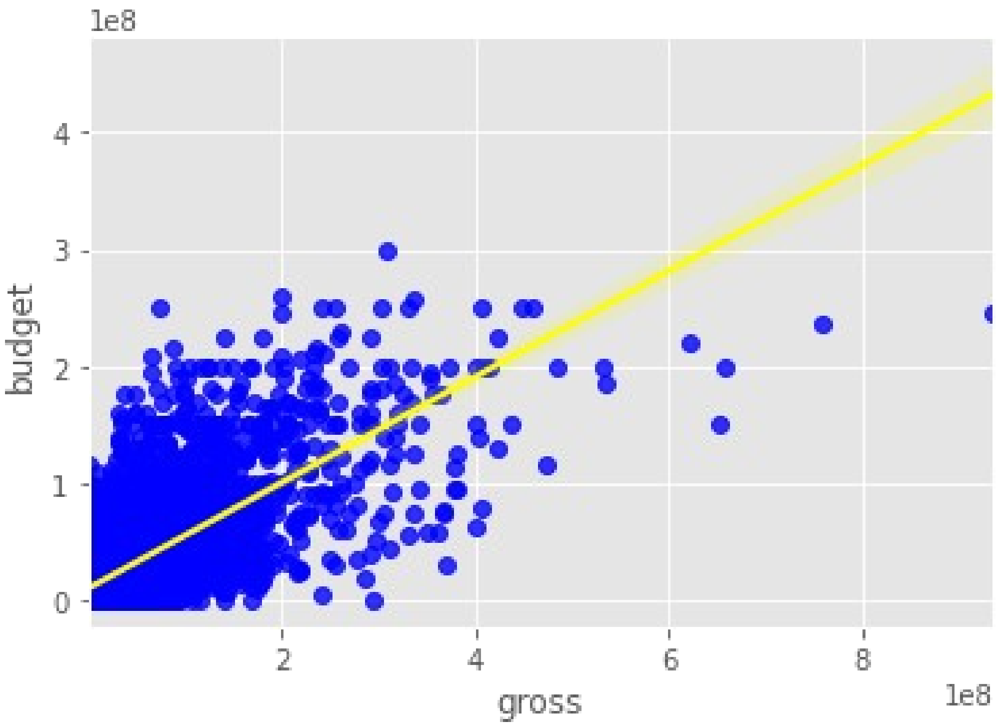
```
# checking which genres grossed high using a rlplot

sns.relplot(x="budget", y="gross", data=df,hue='genre')
```

<seaborn.axisgrid.FacetGrid at 0x7f6692deb5d0>




```
sns.regplot(x="gross", y="budget", data=df,
scatter_kws={"color":"blue"},line_kws = {"color":"Yellow"})
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f66931e7a90>




```
# Looking at the top 15 compaies by gross revenue

CompanyGrossSum = df.groupby('company')[["gross"]].sum()

CompanyGrossSumSorted = CompanyGrossSum.sort_values('gross', ascending
= False)[:15]

CompanyGrossSumSorted = CompanyGrossSumSorted['gross'].astype('int64')


CompanyGrossSumSorted

company
Warner Bros.                                 21322318408
Universal Pictures                           19430051320
Paramount Pictures                           17115702495
Twentieth Century Fox Film Corporation       14788570587
Walt Disney Pictures                         10455507123
Columbia Pictures                             8824216545
New Line Cinema                               8540112287
Columbia Pictures Corporation                 7720114061
Touchstone Pictures                           6688156475
DreamWorks                                    5458121021
DreamWorks Animation                          4143974397
Metro-Goldwyn-Mayer (MGM)                     3384812932
Pixar Animation Studios                       3242024778
Fox 2000 Pictures                             3113861473
```

```
TriStar Pictures                              2967117827
Name: gross, dtype: int64
```

```
#Lets start looking at correlation

df.corr()
```

	budget	gross	runtime	score	votes	year
budget	1.000000	0.712196	0.268226	0.042145	0.503924	0.291009
gross	0.712196	1.000000	0.224579	0.165693	0.662457	0.191548
runtime	0.268226	0.224579	1.000000	0.395343	0.317399	0.087639
score	0.042145	0.165693	0.395343	1.000000	0.393607	0.105276
votes	0.503924	0.662457	0.317399	0.393607	1.000000	0.229304
year	0.291009	0.191548	0.087639	0.105276	0.229304	1.000000

Correlation in Python

Correlation values range between -1 and 1.

There are two key components of a correlation value:

magnitude – The larger the magnitude (closer to 1 or -1), the stronger the correlation sign – If negative, there is an inverse correlation. If positive, there is a regular correlation.

Types of correlation

Pearson Correlation - measures the strength (0 to 1) and direction (negative or positive) of the linear relationship between two (or in this case more) variables. Key assumptions

1. the data is interval or ratio (https://www.usablestats.com/lessons/noir),
2. the relationship is linear (due to two variables), so the correlation is only approximate,
3. outliers affect the correlation and
4. the data is normally distributed (which is a common statistics assumption- Assumption of Normality => that bell curve shape.).

Spearman Correlation Same as Pearson, but

1. The model does not rely on normality (Assumption 4), and
2. The data can be ordinal (Assumption 1).

Kendall Correlation

Same as Spearman, but

1. The data can be continuous (Assumption 1),

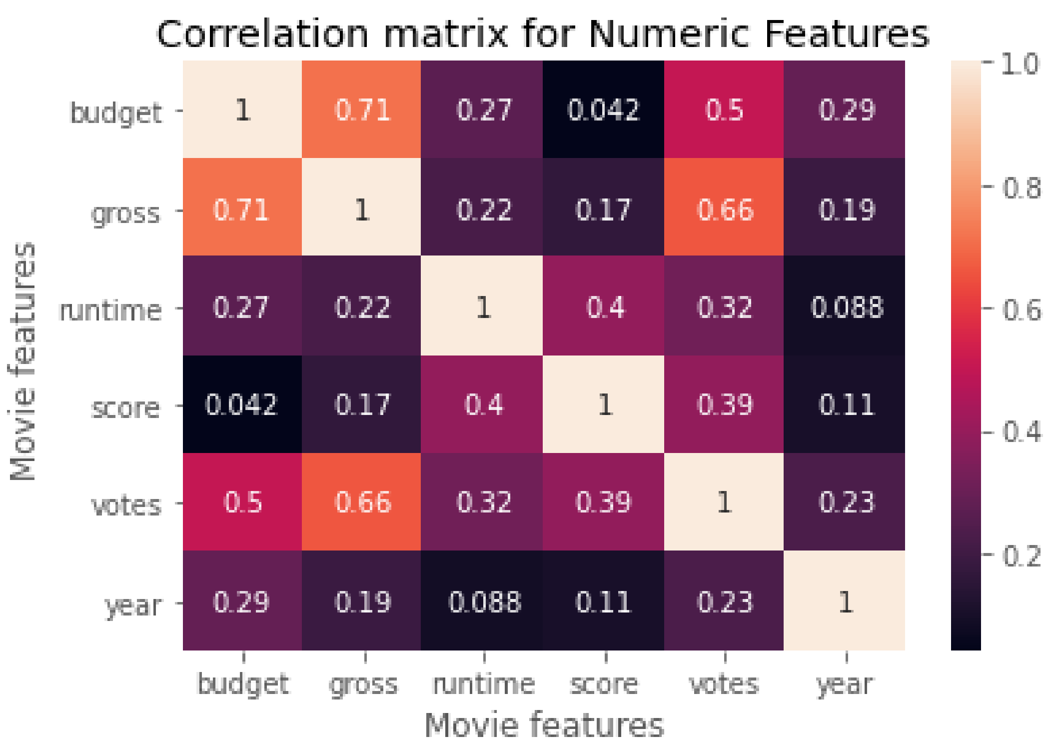
Source: https://ademos.people.uic.edu/Chapter22.html

Pearson appears to be the flagship of correlation analysis, whereas Spearman and Kendall correlations can be more robust when considering nonlinear relationships.

Correlation Matrix

If we're using pandas we can create a correlation matrix to view the correlations between different variables in a dataframe

```
# Analysing correlation matrix using pearson

correlation_matrix = df.corr()

sns.heatmap(correlation_matrix, annot = True)

plt.title("Correlation matrix for Numeric Features")

plt.xlabel("Movie features")

plt.ylabel("Movie features")

plt.show()
```




```
df_numerized = df
```

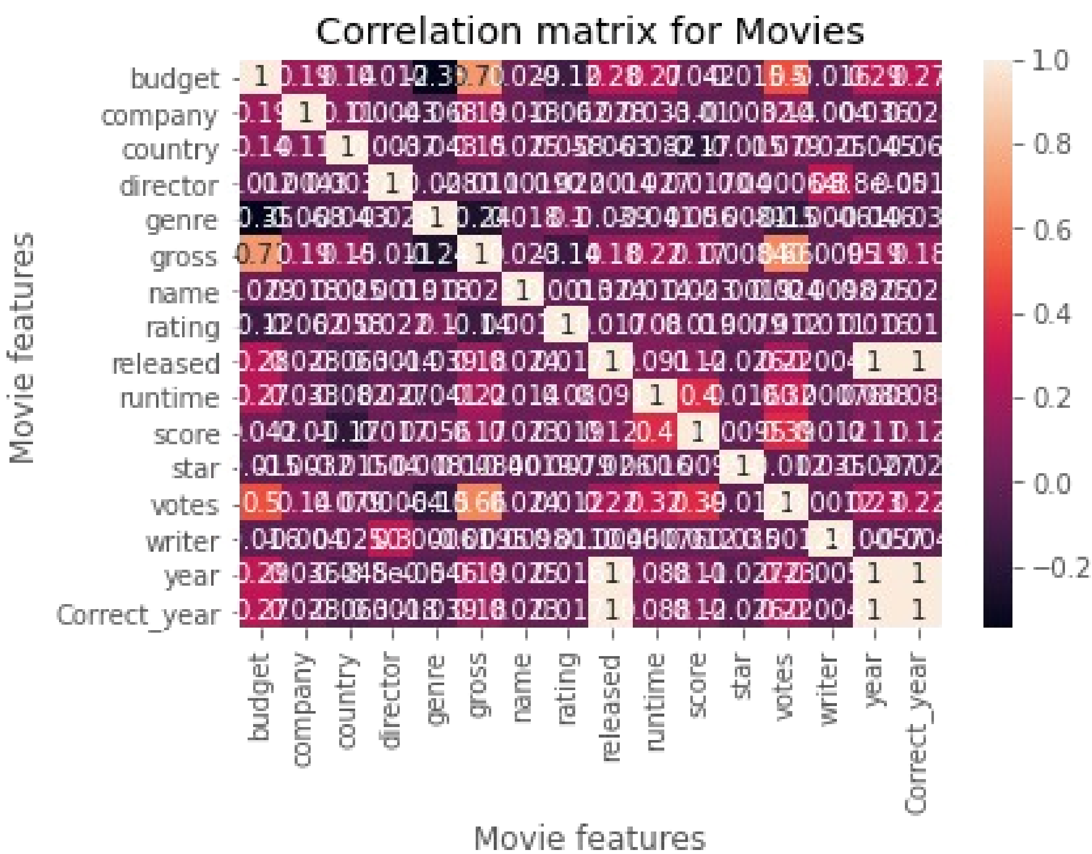
```
for col_name in df_numerized.columns:
    if(df_numerized[col_name].dtype == 'object'):
        df_numerized[col_name]=
df_numerized[col_name].astype('category')
        df_numerized[col_name] = df_numerized[col_name].cat.codes

df_numerized
```

Output hidden; open in https://colab.research.google.com to view.

```
df_numerized.corr(method='pearson')
```

```
                budget   company   country  ...    writer      year  
Correct_year
budget        1.000000  0.187205  0.137635  ... -0.015611  0.291009  
0.274820
company       0.187205  1.000000  0.107950  ... -0.004032  0.036272  
0.028012
country       0.137635  0.107950  1.000000  ...  0.024981 -0.045204 
-0.062707
director      0.011602  0.004320  0.003698  ...  0.298997 -0.000088  
0.001822
genre        -0.346794 -0.068330 -0.042793  ... -0.000608 -0.046259 
-0.039014
gross         0.712196  0.187220  0.149988  ... -0.009455  0.191548  
0.176879
name          0.028712  0.018098  0.025020  ...  0.009821  0.024624  
0.023411
rating       -0.119660 -0.062250  0.057979  ...  0.010740  0.016221  
0.017438
released      0.276635  0.027898 -0.062609  ... -0.004635  0.996187  
0.999389
runtime       0.268226  0.033058 -0.081796  ...  0.000759  0.087639  
0.088342
score         0.042145 -0.010426 -0.174414  ...  0.012223  0.105276  
0.117679
star         -0.015061 -0.003160 -0.014566  ...  0.035378 -0.026680 
-0.026050
votes         0.503924  0.138662  0.078657  ...  0.001154  0.229304  
0.220797
writer       -0.015611 -0.004032  0.024981  ...  1.000000 -0.005665 
-0.004546
year          0.291009  0.036272 -0.045204  ... -0.005665  1.000000  
0.996229
Correct_year  0.274820  0.028012 -0.062707  ... -0.004546  0.996229  
1.000000

[16 rows x 16 columns]
```

```
correlation_matrix = df_numerized.corr(method='pearson')

sns.heatmap(correlation_matrix, annot = True)

plt.title("Correlation matrix for Movies")

plt.xlabel("Movie features")

plt.ylabel("Movie features")

plt.show()
```




```
correlation_mat = df_numerized.corr()

corr_pairs = correlation_mat.unstack()

(corr_pairs)

budget         budget           1.000000
               company          0.187205
               country          0.137635
               director         0.011602
               genre           -0.346794
```

	gross	0.712196
	name	0.028712
	rating	-0.119660
	released	0.276635
	runtime	0.268226
	score	0.042145
	star	-0.015061
	votes	0.503924
	writer	-0.015611
	year	0.291009
	Correct_year	0.274820
company	budget	0.187205
	company	1.000000
	country	0.107950
	director	0.004320
	genre	-0.068330
	gross	0.187220
	name	0.018098
	rating	-0.062250
	released	0.027898
	runtime	0.033058
	score	-0.010426
	star	-0.003160
	votes	0.138662
	writer	-0.004032
	year	0.036272
	Correct_year	0.028012
country	budget	0.137635
	company	0.107950
	country	1.000000
	director	0.003698
	genre	-0.042793
	gross	0.149988
	name	0.025020
	rating	0.057979
	released	-0.062609
	runtime	-0.081796
	score	-0.174414
	star	-0.014566
	votes	0.078657
	writer	0.024981
	year	-0.045204
	Correct_year	-0.062707
director	budget	0.011602
	company	0.004320
	country	0.003698
	director	1.000000
	genre	-0.027668
	gross	-0.011429
	name	0.001905

	rating	0.021926
	released	0.001440
	runtime	0.026779
	score	0.017130
	star	0.039813
	votes	0.000639
	writer	0.298997
	year	-0.000088
	Correct_year	0.001822
genre	budget	-0.346794
	company	-0.068330
	country	-0.042793
	director	-0.027668
	genre	1.000000
	gross	-0.242676
	name	0.018062
	rating	0.100960
	released	-0.039179
	runtime	-0.041357
	score	0.056234
	star	0.008140
	votes	-0.150519
	writer	-0.000608
	year	-0.046259
	Correct_year	-0.039014
gross	budget	0.712196
	company	0.187220
	country	0.149988
	director	-0.011429
	genre	-0.242676
	gross	1.000000
	name	0.022768
	rating	-0.135538
	released	0.178564
	runtime	0.224579
	score	0.165693
	star	0.008382
	votes	0.662457
	writer	-0.009455
	year	0.191548
	Correct_year	0.176879
name	budget	0.028712
	company	0.018098
	country	0.025020
	director	0.001905
	genre	0.018062
	gross	0.022768
	name	1.000000
	rating	0.001288
	released	0.024120

```
              runtime          0.013942
              score            0.023342
              star            -0.001910
              votes            0.023665
              writer           0.009821
              year             0.024624
              Correct_year     0.023411
rating        budget          -0.119660
              company         -0.062250
              country          0.057979
              director         0.021926
              genre            0.100960
              gross           -0.135538
              name             0.001288
              rating           1.000000
              released         0.016696
              runtime          0.079542
              score            0.019271
              star             0.007893
              votes            0.011678
              writer           0.010740
              year             0.016221
              Correct_year     0.017438
released      budget           0.276635
              company          0.027898
              country         -0.062609
              director         0.001440
              genre           -0.039179
              gross            0.178564
              name             0.024120
              rating           0.016696
              released         1.000000
              runtime          0.091102
              score            0.119577
              star            -0.025504
              votes            0.221736
              writer          -0.004635
              year             0.996187
              Correct_year     0.999389
runtime       budget           0.268226
              company          0.033058
              country         -0.081796
              director         0.026779
              genre           -0.041357
              gross            0.224579
              name             0.013942
              rating           0.079542
              released         0.091102
              runtime          1.000000
              score            0.395343
```

	star	0.016019
	votes	0.317399
	writer	0.000759
	year	0.087639
	Correct_year	0.088342
score	budget	0.042145
	company	-0.010426
	country	-0.174414
	director	0.017130
	genre	0.056234
	gross	0.165693
	name	0.023342
	rating	0.019271
	released	0.119577
	runtime	0.395343
	score	1.000000
	star	0.009482
	votes	0.393607
	writer	0.012223
	year	0.105276
	Correct_year	0.117679
star	budget	-0.015061
	company	-0.003160
	country	-0.014566
	director	0.039813
	genre	0.008140
	gross	0.008382
	name	-0.001910
	rating	0.007893
	released	-0.025504
	runtime	0.016019
	score	0.009482
	star	1.000000
	votes	-0.011919
	writer	0.035378
	year	-0.026680
	Correct_year	-0.026050
votes	budget	0.503924
	company	0.138662
	country	0.078657
	director	0.000639
	genre	-0.150519
	gross	0.662457
	name	0.023665
	rating	0.011678
	released	0.221736
	runtime	0.317399
	score	0.393607
	star	-0.011919
	votes	1.000000

	writer	0.001154
	year	0.229304
	Correct_year	0.220797
writer	budget	-0.015611
	company	-0.004032
	country	0.024981
	director	0.298997
	genre	-0.000608
	gross	-0.009455
	name	0.009821
	rating	0.010740
	released	-0.004635
	runtime	0.000759
	score	0.012223
	star	0.035378
	votes	0.001154
	writer	1.000000
	year	-0.005665
	Correct_year	-0.004546
year	budget	0.291009
	company	0.036272
	country	-0.045204
	director	-0.000088
	genre	-0.046259
	gross	0.191548
	name	0.024624
	rating	0.016221
	released	0.996187
	runtime	0.087639
	score	0.105276
	star	-0.026680
	votes	0.229304
	writer	-0.005665
	year	1.000000
	Correct_year	0.996229
Correct_year	budget	0.274820
	company	0.028012
	country	-0.062707
	director	0.001822
	genre	-0.039014
	gross	0.176879
	name	0.023411
	rating	0.017438
	released	0.999389
	runtime	0.088342
	score	0.117679
	star	-0.026050
	votes	0.220797
	writer	-0.004546
	year	0.996229

```
               Correct_year    1.000000
dtype: float64
```

Conclusions

1) IMDB Rating(Rating) is Negatively correlated to Gross (Revenue) 2) Gross is positvely Correlated to Budget 3) Name (companies which produced the movies) were also negatively correlated to budget