movies_pro

September 7, 2024

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib
plt.style.use('ggplot')
from matplotlib.pyplot import figure

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

pd.options.mode.chained_assignment = None
```

[2]:
```
df= pd.read_csv(r"C:\Users\admin\Desktop\Data analytics\projects␣
 ↪complted\Python\movies.csv")
df
```

[2]:

	name	rating	genre	year	released	score	votes	director
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis
…	…	…	…	…	…	…	…	…
7663	More to Life	NaN	Drama	2020				
7664	Dream Round	NaN	Comedy	2020				
7665	Saving Mbango	NaN	Drama	2020				
7666	It's Just Us	NaN	Drama	2020				
7667	Tee em el	NaN	Horror	2020				

```
7663  October 23, 2020 (United States)    3.1       18.0    Joseph Ebanks
7664  February 7, 2020 (United States)    4.7       36.0     Dusty Dukatz
7665         April 27, 2020 (Cameroon)    5.7       29.0     Nkanya Nkwai
7666   October 1, 2020 (United States)    NaN        NaN    James Randall
7667   August 19, 2020 (United States)    5.7        7.0     Pereko Mosia

                      writer              star        country    budget  \
0               Stephen King    Jack Nicholson  United States  19000000
1     Henry De Vere Stacpoole    Brooke Shields  United States   4500000
2               Leigh Brackett       Mark Hamill  United States  18000000
3                 Jim Abrahams       Robert Hays  United States   3500000
4           Brian Doyle-Murray       Chevy Chase  United States   6000000
…                          …                 …              …       …
7663            Joseph Ebanks      Shannon Bond  United States      7000
7664              Lisa Huston  Michael Saquella  United States         0
7665             Lynno Lovert      Onyama Laura       Cameroon     58750
7666            James Randall     Christina Roz  United States     15000
7667             Pereko Mosia  Siyabonga Mabaso  United States         0

          gross                    company  runtime             date
0      46998772              Warner Bros.    146.0     June 13, 1980
1      58853106         Columbia Pictures    104.0      July 2, 1980
2     538375067                 Lucasfilm    124.0     June 20, 1980
3      83453539        Paramount Pictures     88.0      July 2, 1980
4      39846344            Orion Pictures     98.0     July 25, 1980
…          …                         …      …              …
7663          0                       NaN     90.0  October 23, 2020
7664          0  Cactus Blue Entertainment     90.0  February 7, 2020
7665          0          Embi Productions      NaN    April 27, 2020
7666          0                       NaN    120.0   October 1, 2020
7667          0             PK 65 Films    102.0   August 19, 2020

[7668 rows x 16 columns]
```

[3]:
```
for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    print('{} - {}%'.format(col, round(pct_missing*100)))
```

```
name - 0%
rating - 1%
genre - 0%
year - 0%
released - 0%
score - 0%
votes - 0%
director - 0%
writer - 0%
star - 0%
```

```
country - 1%
budget - 0%
gross - 0%
company - 0%
runtime - 0%
date - 1%
```

[4]:
```
df.dtypes
```

[4]:
```
name         object
rating       object
genre        object
year          int64
released     object
score       float64
votes       float64
director     object
writer       object
star         object
country      object
budget        int64
gross         int64
company      object
runtime     float64
date         object
dtype: object
```

[5]:
```
df['date'] = pd.to_datetime(df['date'], format='%B %d, %Y')
df['corr_year'] = df['date'].astype(str).str[:4]
```

[6]:
```
df.head()
```

[6]:

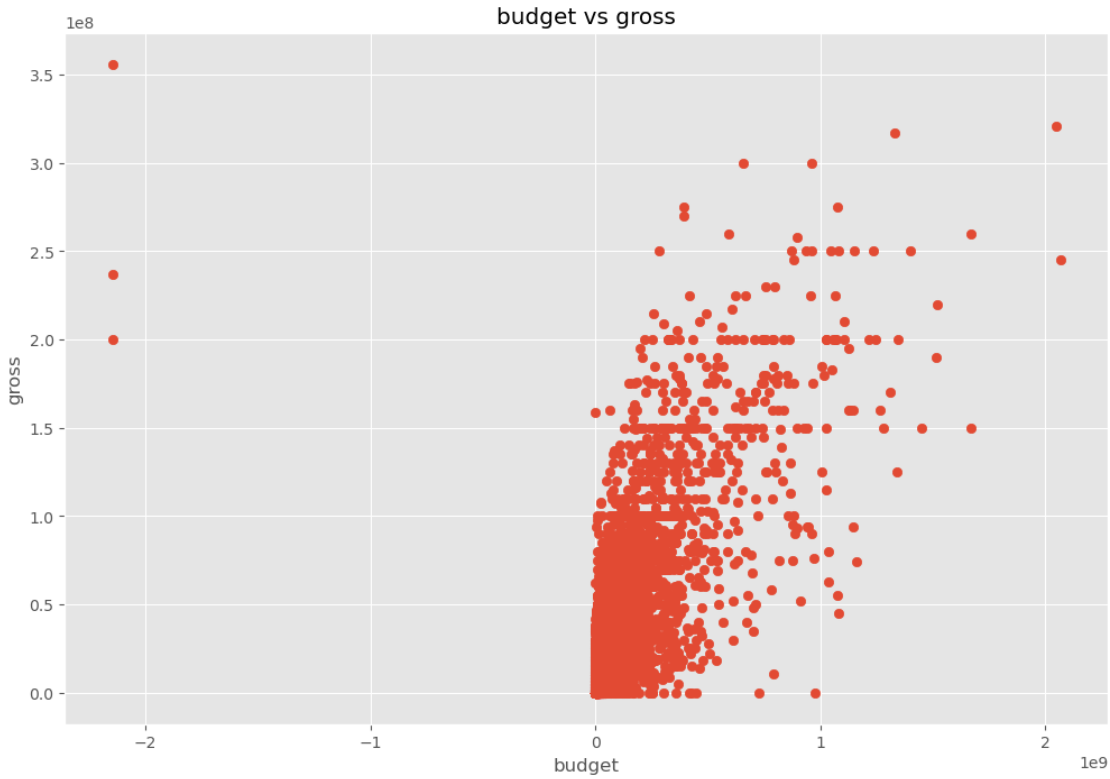
	name	rating	genre	year	released	score	votes	director	writer	star	country	budget
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Jack Nicholson	United States	19000000
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser				
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner				
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams				
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis				

```
1  Henry De Vere Stacpoole  Brooke Shields  United States   4500000
2           Leigh Brackett     Mark Hamill  United States  18000000
3             Jim Abrahams     Robert Hays  United States   3500000
4       Brian Doyle-Murray     Chevy Chase  United States   6000000

       gross             company  runtime       date corr_year
0   46998772        Warner Bros.    146.0 1980-06-13      1980
1   58853106   Columbia Pictures    104.0 1980-07-02      1980
2  538375067           Lucasfilm    124.0 1980-06-20      1980
3   83453539  Paramount Pictures     88.0 1980-07-02      1980
4   39846344      Orion Pictures     98.0 1980-07-25      1980
```

[7]:
```
df['company'].drop_duplicates
```

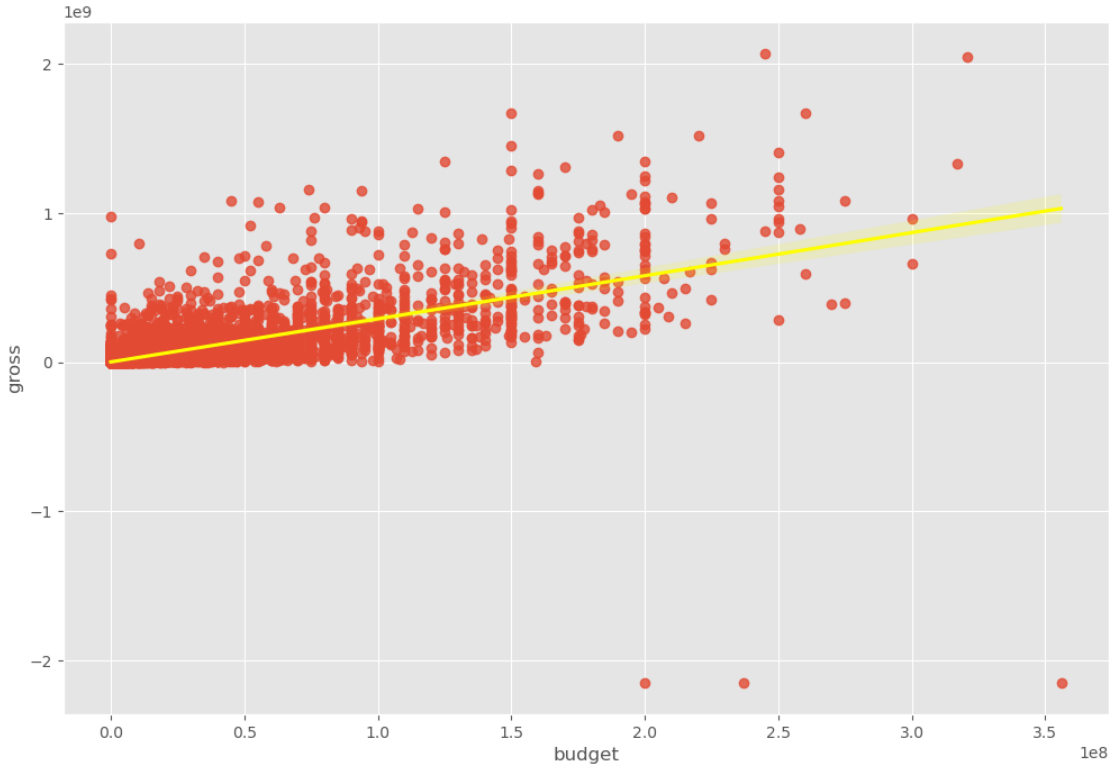
[7]:
```
<bound method Series.drop_duplicates of 0                    Warner Bros.
1                  Columbia Pictures
2                          Lucasfilm
3                 Paramount Pictures
4                     Orion Pictures
                     …
7663                             NaN
7664    Cactus Blue Entertainment
7665               Embi Productions
7666                             NaN
7667                    PK 65 Films
Name: company, Length: 7668, dtype: object>
```

[9]:
```
plt.scatter(x=df['gross'],y=df['budget'])
plt.title('budget vs gross')
plt.xlabel('budget')
plt.ylabel('gross')
plt.show()
```



[12]:
```
sns.regplot(x='budget',y='gross',data=df,line_kws={"color":"yellow"})
```

[12]: <Axes: xlabel='budget', ylabel='gross'>



[15]:
```
df.select_dtypes(include=[np.number])
```

[15]:

	year	score	votes	budget	gross	runtime
0	1980	8.4	927000.0	19000000	46998772	146.0
1	1980	5.8	65000.0	4500000	58853106	104.0
2	1980	8.7	1200000.0	18000000	538375067	124.0
3	1980	7.7	221000.0	3500000	83453539	88.0
4	1980	7.3	108000.0	6000000	39846344	98.0
…	…	…	…	…	…	…
7663	2020	3.1	18.0	7000	0	90.0
7664	2020	4.7	36.0	0	0	90.0
7665	2020	5.7	29.0	58750	0	NaN
7666	2020	NaN	NaN	15000	0	120.0
7667	2020	5.7	7.0	0	0	102.0

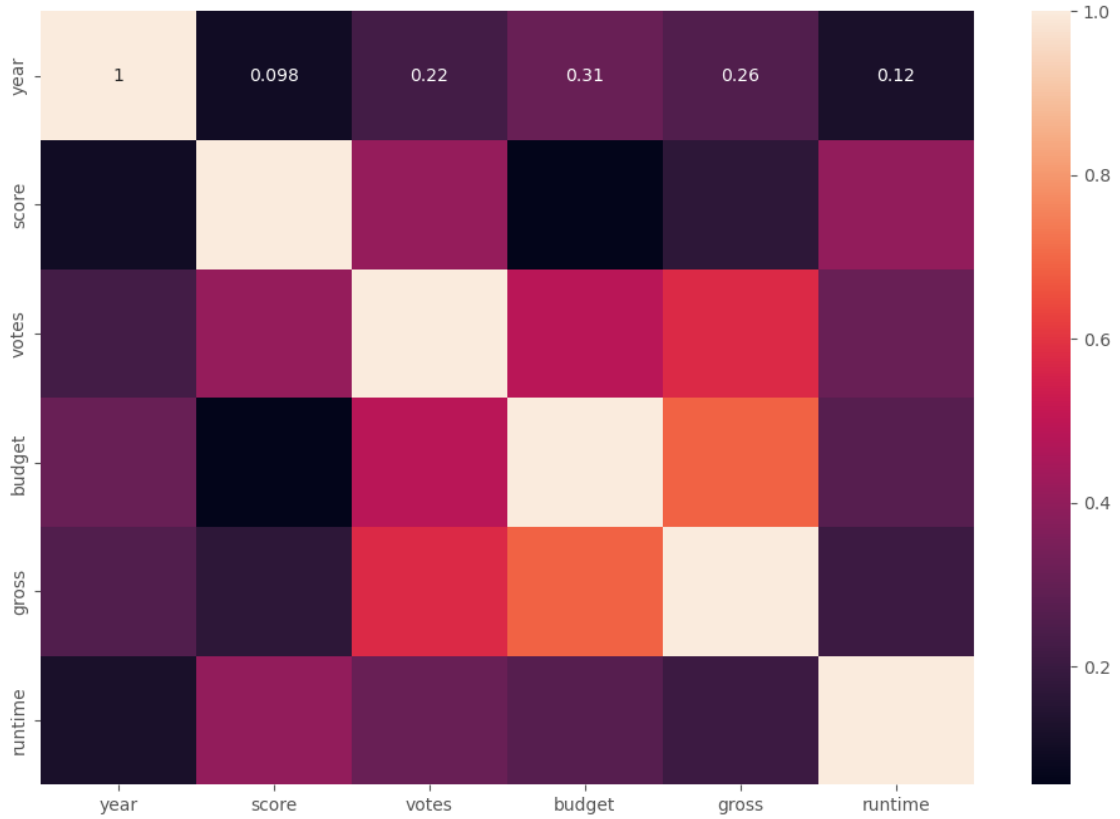
[7668 rows x 6 columns]

[16]:
```
numeric_df = df.select_dtypes(include=[np.number])
correlation_matrix = numeric_df.corr(method='pearson')

print(correlation_matrix)
```

```
             year     score     votes    budget     gross   runtime
```

```
year      1.000000  0.097995  0.222945  0.309212  0.256331  0.120811
score     0.097995  1.000000  0.409182  0.055665  0.169406  0.399451
votes     0.222945  0.409182  1.000000  0.486862  0.573889  0.309212
budget    0.309212  0.055665  0.486862  1.000000  0.687124  0.269510
gross     0.256331  0.169406  0.573889  0.687124  1.000000  0.204177
runtime   0.120811  0.399451  0.309212  0.269510  0.204177  1.000000
```

[23]:
```
sns.heatmap(correlation_matrix, annot=True)
plt.show()
```



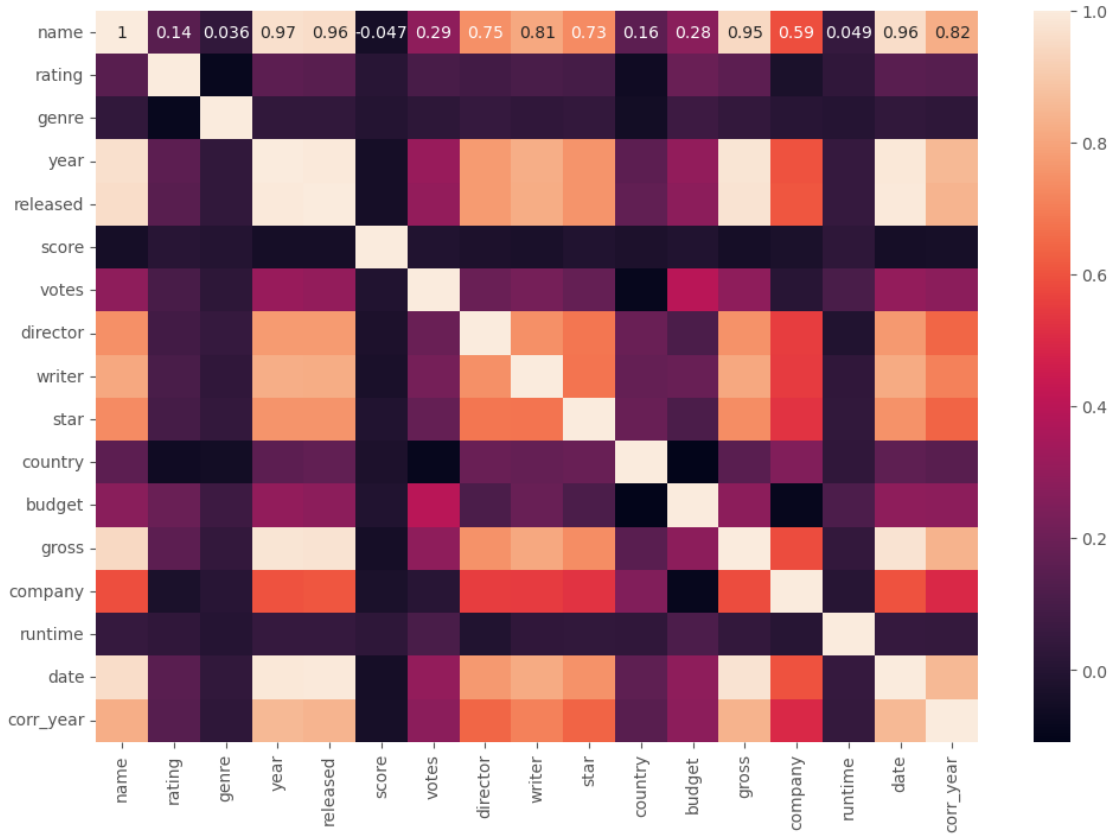
[24]:
```
df.apply(lambda x: x.factorize()[0]).corr(method='pearson')
```

[24]:
```
                name    rating     genre      year  released     score  \
name        1.000000  0.143938  0.036367  0.965761  0.959015 -0.046733
rating      0.143938  1.000000 -0.086723  0.156713  0.146606  0.012595
genre       0.036367 -0.086723  1.000000  0.037184  0.035940 -0.002437
year        0.965761  0.156713  0.037184  1.000000  0.993190 -0.044981
released    0.959015  0.146606  0.035940  0.993190  1.000000 -0.045761
score      -0.046733  0.012595 -0.002437 -0.044981 -0.045761  1.000000
votes       0.287776  0.099972  0.023285  0.312401  0.299905 -0.009749
director    0.745905  0.085520  0.047288  0.770497  0.770876 -0.022687
```

```
writer     0.805211  0.103623  0.033688  0.824770  0.819617 -0.034685
star       0.731565  0.093116  0.038649  0.756400  0.754468 -0.009896
country    0.156957 -0.064928 -0.054400  0.155758  0.168516 -0.022466
budget     0.275691  0.193229  0.069445  0.298022  0.284017 -0.011749
gross      0.947192  0.158007  0.038781  0.980741  0.976912 -0.046835
company    0.591667 -0.028035  0.009566  0.601571  0.607954 -0.028432
runtime    0.048955  0.032741  0.001462  0.050647  0.048235  0.026436
date       0.956695  0.148669  0.036158  0.990614  0.995500 -0.044743
corr_year  0.824034  0.138457  0.028434  0.852843  0.843590 -0.040821

               votes  director    writer      star   country    budget  \
name        0.287776  0.745905  0.805211  0.731565  0.156957  0.275691
rating      0.099972  0.085520  0.103623  0.093116 -0.064928  0.193229
genre       0.023285  0.047288  0.033688  0.038649 -0.054400  0.069445
year        0.312401  0.770497  0.824770  0.756400  0.155758  0.298022
released    0.299905  0.770876  0.819617  0.754468  0.168516  0.284017
score      -0.009749 -0.022687 -0.034685 -0.009896 -0.022466 -0.011749
votes       1.000000  0.192220  0.224122  0.179601 -0.089770  0.395374
director    0.192220  1.000000  0.748340  0.682385  0.191433  0.108033
writer      0.224122  0.748340  1.000000  0.675685  0.180745  0.186680
star        0.179601  0.682385  0.675685  1.000000  0.185871  0.110870
country    -0.089770  0.191433  0.180745  0.185871  1.000000 -0.108666
budget      0.395374  0.108033  0.186680  0.110870 -0.108666  1.000000
gross       0.285529  0.751155  0.805812  0.735987  0.149360  0.281209
company     0.008900  0.552258  0.546151  0.527116  0.250812 -0.087285
runtime     0.106024 -0.011070  0.032264  0.035392  0.032763  0.115359
date        0.302078  0.766265  0.817198  0.749509  0.163976  0.289375
corr_year   0.277711  0.643187  0.708881  0.639445  0.146749  0.284085

               gross   company   runtime      date  corr_year
name        0.947192  0.591667  0.048955  0.956695   0.824034
rating      0.158007 -0.028035  0.032741  0.148669   0.138457
genre       0.038781  0.009566  0.001462  0.036158   0.028434
year        0.980741  0.601571  0.050647  0.990614   0.852843
released    0.976912  0.607954  0.048235  0.995500   0.843590
score      -0.046835 -0.028432  0.026436 -0.044743  -0.040821
votes       0.285529  0.008900  0.106024  0.302078   0.277711
director    0.751155  0.552258 -0.011070  0.766265   0.643187
writer      0.805812  0.546151  0.032264  0.817198   0.708881
star        0.735987  0.527116  0.035392  0.749509   0.639445
country     0.149360  0.250812  0.032763  0.163976   0.146749
budget      0.281209 -0.087285  0.115359  0.289375   0.284085
gross       1.000000  0.588216  0.041957  0.974297   0.835787
company     0.588216  1.000000  0.005137  0.600295   0.493998
runtime     0.041957  0.005137  1.000000  0.048948   0.046451
date        0.974297  0.600295  0.048948  1.000000   0.853960
corr_year   0.835787  0.493998  0.046451  0.853960   1.000000
```

[25]:
```
correlation_matrix = df.apply(lambda x: x.factorize()[0]).corr(method='pearson')

sns.heatmap(correlation_matrix, annot = True)
```

[25]: <Axes: >



[26]:
```
correlation_mat = df.apply(lambda x: x.factorize()[0]).corr()

corr_pairs = correlation_mat.unstack()

print(corr_pairs)
```

```
name        name          1.000000
            rating        0.143938
            genre         0.036367
            year          0.965761
            released      0.959015
                              …
corr_year   gross         0.835787
            company       0.493998
            runtime       0.046451
```

```
           date          0.853960
           corr_year     1.000000
Length: 289, dtype: float64
```

[27]:
```
sorted_pairs = corr_pairs.sort_values(kind="quicksort")

print(sorted_pairs)
```

```
country     budget      -0.108666
budget      country     -0.108666
country     votes       -0.089770
votes       country     -0.089770
budget      company     -0.087285
                           …
year        year         1.000000
genre       genre        1.000000
rating      rating       1.000000
date        date         1.000000
corr_year   corr_year    1.000000
Length: 289, dtype: float64
```

[28]:
```
strong_pairs = sorted_pairs[abs(sorted_pairs) > 0.5]

print(strong_pairs)
```

```
company     star         0.527116
star        company      0.527116
company     writer       0.546151
writer      company      0.546151
director    company      0.552258
                           …
year        year         1.000000
genre       genre        1.000000
rating      rating       1.000000
date        date         1.000000
corr_year   corr_year    1.000000
Length: 105, dtype: float64
```

[29]:
```
CompanyGrossSum = df.groupby('company')[["gross"]].sum()

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

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

CompanyGrossSumSorted
```

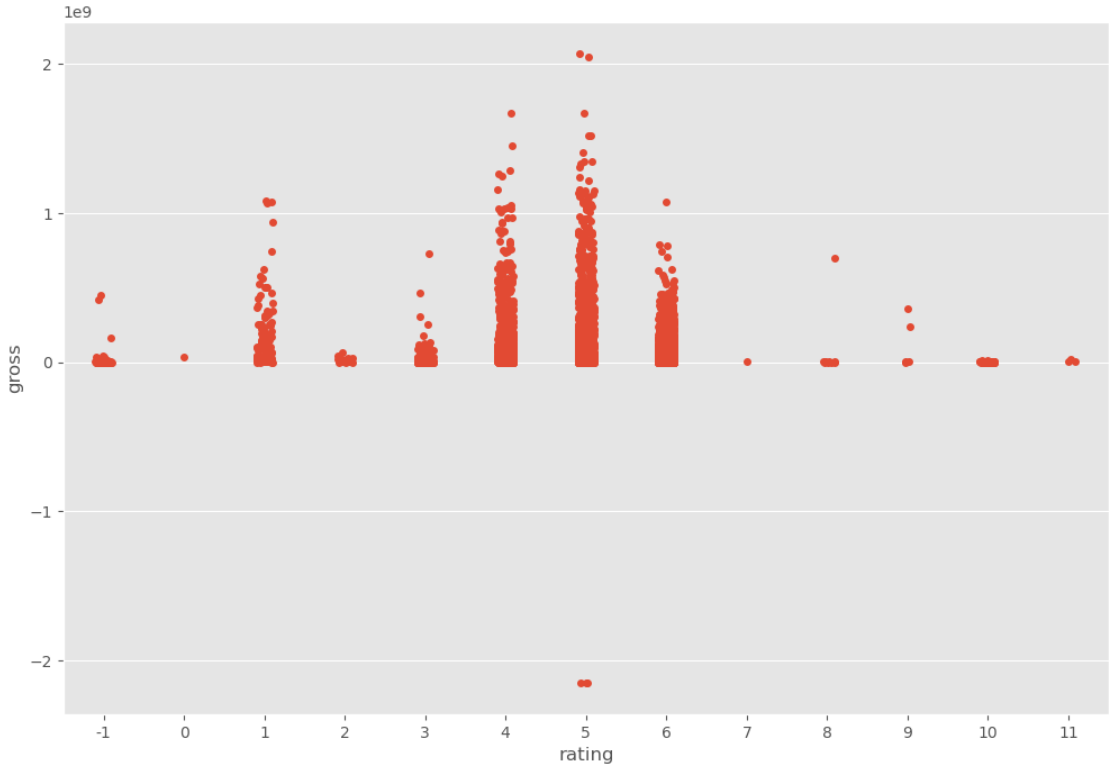
[29]:
```
company
Warner Bros.                 56491421806
Universal Pictures           52514188890
Columbia Pictures            43008941346
Paramount Pictures           40493607415
Walt Disney Pictures         36327887792
Twentieth Century Fox        30913193094
New Line Cinema              19883797684
DreamWorks Animation         11873612858
Touchstone Pictures          11795832638
Dreamworks Pictures          11635441081
Marvel Studios               10120607435
Metro-Goldwyn-Mayer (MGM)     9230230105
Summit Entertainment          8373718838
Pixar Animation Studios       7886344526
Fox 2000 Pictures             7443502667
Name: gross, dtype: int64
```

[30]:
```
for col_name in df.columns:
    if(df[col_name].dtype == 'object'):
        df[col_name]= df[col_name].astype('category')
        df[col_name] = df[col_name].cat.codes
```

[32]:
```
sns.stripplot(x="rating", y="gross", data=df)
```

```
D:\python\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
D:\python\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[32]:
```
<Axes: xlabel='rating', ylabel='gross'>
```


1e9
2
1
0
−1
−2
gross
-1
0
1
2
3
4
5
6
7
8
9
10
11
rating

[ ]:

[ ]: