Movie Correlation Project

Using python to clean data, test for correlation between variables, and visualize the results.

In [2]:
```
# First let's import the packages we will use in this project
# You can do this all now or as you need them
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



# Now we need to read in the data
df = pd.read_csv(r'C:\Users\Megan\Documents\Portfolio\Movie Project\movies.csv')
```

In [3]:
```
# Now let's take a look at the data

df
```

Out[3]:

	name	rating	genre	year	released	score	votes	director	writer	st
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ja Nicholsc
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brool Shiel
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Ma Ham
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robe Ha
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Che Cha
...	...	...	...	...	...	...	...	...	...	

	name	rating	genre	year	released	score	votes	director	writer	st
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Shannc Bor
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	Micha Saque
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	Onyan Lau
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Christi R
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siyabon Maba

7668 rows × 15 columns

In [ ]:

In [ ]:

In [ ]:

In [4]:
```
# We need to see if we have any missing data
# Let's loop through the data and see if there is anything missing

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 - 0%
budget - 28%
gross - 2%
```

```
company - 0%
runtime - 0%
```

In [ ]:

In [ ]:

In [5]:
```
# Data Types for our columns

print(df.dtypes)
```

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

In [ ]:

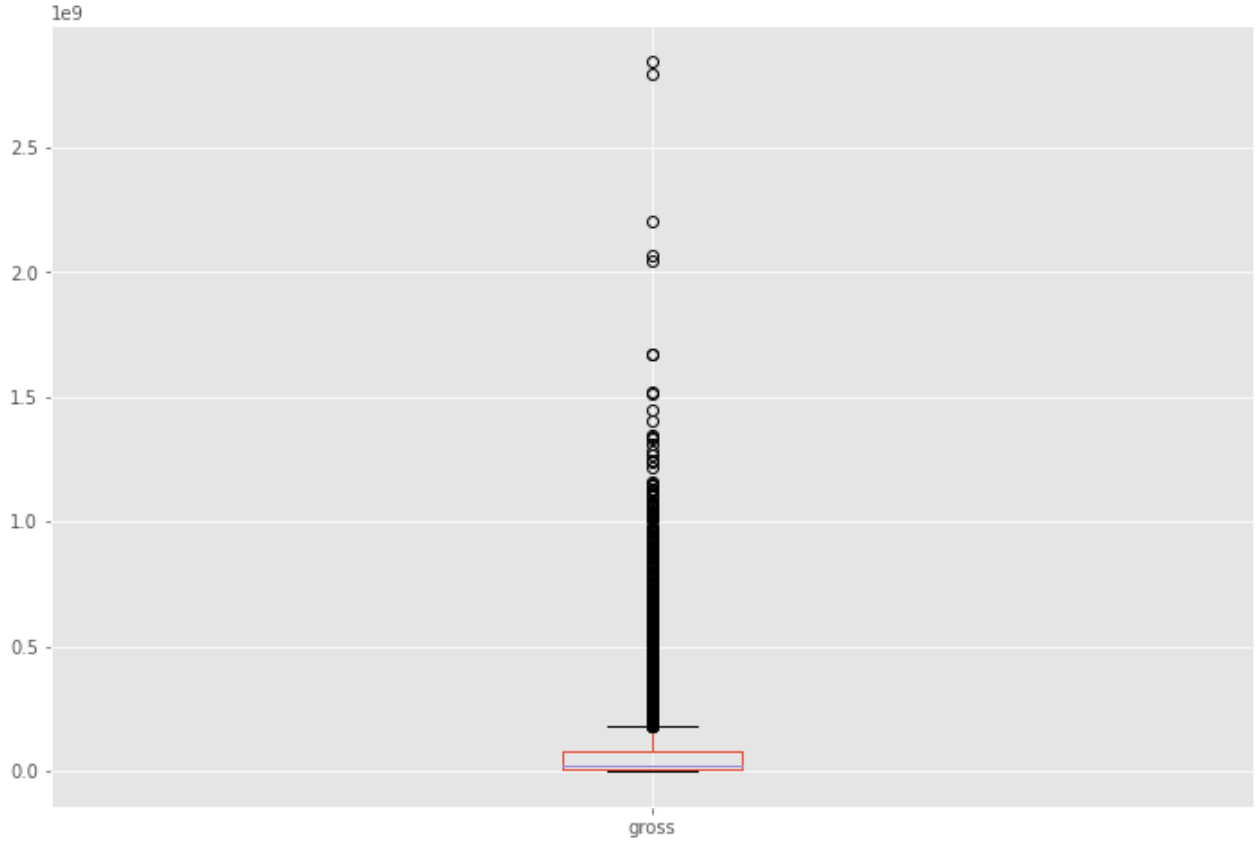
In [ ]:

In [ ]:

In [ ]:

In [6]:
```
# Are there any Outliers?

df.boxplot(column=['gross'])
```

Out[6]:
```
<AxesSubplot:>
```



In [7]:
```
df.drop_duplicates()
```

Out[7]:

	name	rating	genre	year	released	score	votes	director	writer	st
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ja Nicholsc
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brool Shiel
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Ma Ham
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robe Ha
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Che Cha
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Shannc Bor

	name	rating	genre	year	released	score	votes	director	writer	st
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	Micha Saque
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	Onyan Lau
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Christi R
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siyabon Maba

7668 rows × 15 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [8]:
```
# Order our Data a little bit to see

df.sort_values(by=['gross'], inplace=False, ascending=False)
```

Out[8]:

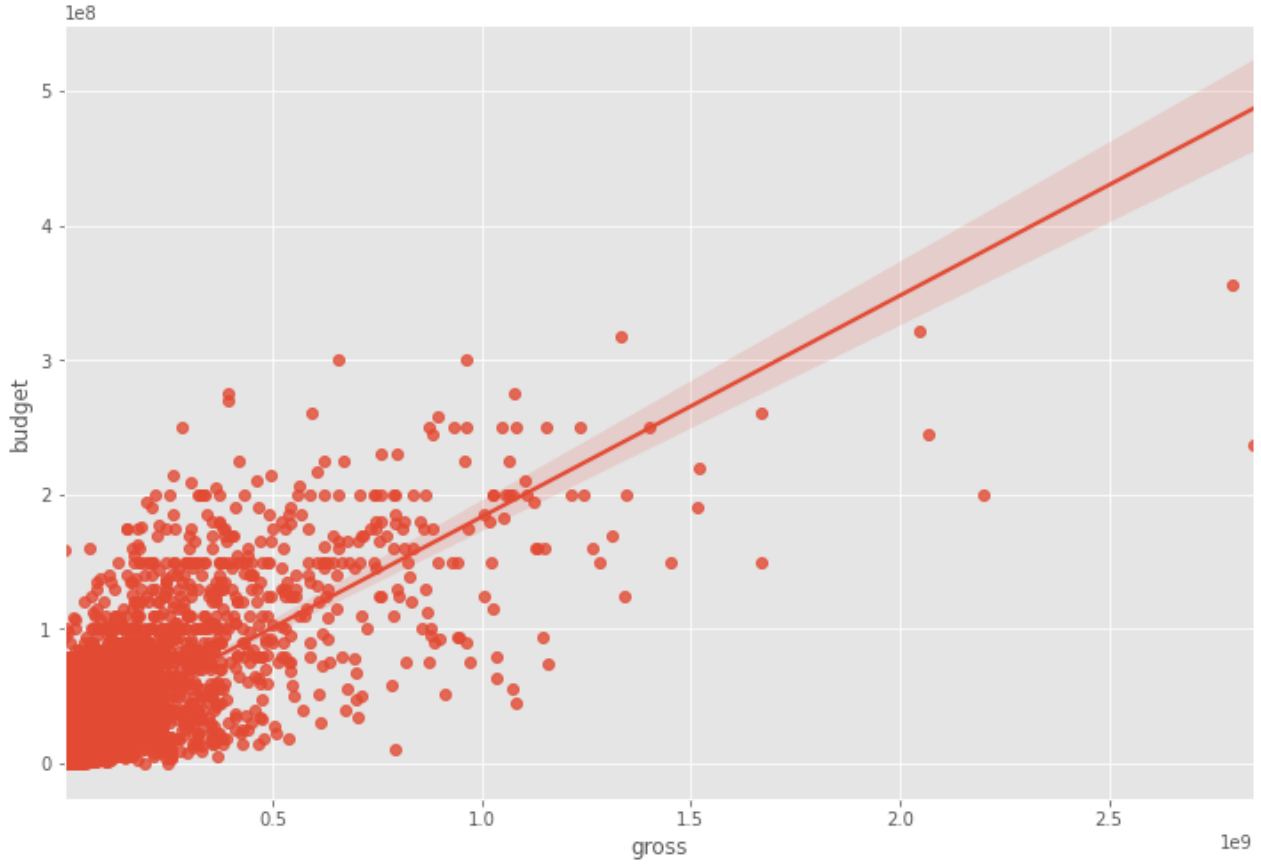
	name	rating	genre	year	released	score	votes	director	writer	sta
5445	Avatar	PG-13	Action	2009	December 18, 2009 (United States)	7.8	1100000.0	James Cameron	James Cameron	Sam Worthingtor
7445	Avengers: Endgame	PG-13	Action	2019	April 26, 2019 (United States)	8.4	903000.0	Anthony Russo	Christopher Markus	Rober Downey Jr

	name	rating	genre	year	released	score	votes	director	writer	sta
3045	Titanic	PG-13	Drama	1997	December 19, 1997 (United States)	7.8	1100000.0	James Cameron	James Cameron	Leonardo DiCapric
6663	Star Wars: Episode VII - The Force Awakens	PG-13	Action	2015	December 18, 2015 (United States)	7.8	876000.0	J.J. Abrams	Lawrence Kasdan	Daisy Ridley
7244	Avengers: Infinity War	PG-13	Action	2018	April 27, 2018 (United States)	8.4	897000.0	Anthony Russo	Christopher Markus	Rober Downey Jr
...	...	...	...	...	...	...	...	...	...	..
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Shannor Bonc
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	Michae Saquella
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	Onyama Laura
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Christina Roz
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siyabonga Mabasc

7668 rows × 15 columns

In [9]:
```
sns.regplot(x="gross", y="budget", data=df)
```

Out[9]:
```
<AxesSubplot:xlabel='gross', ylabel='budget'>
```



In [ ]:

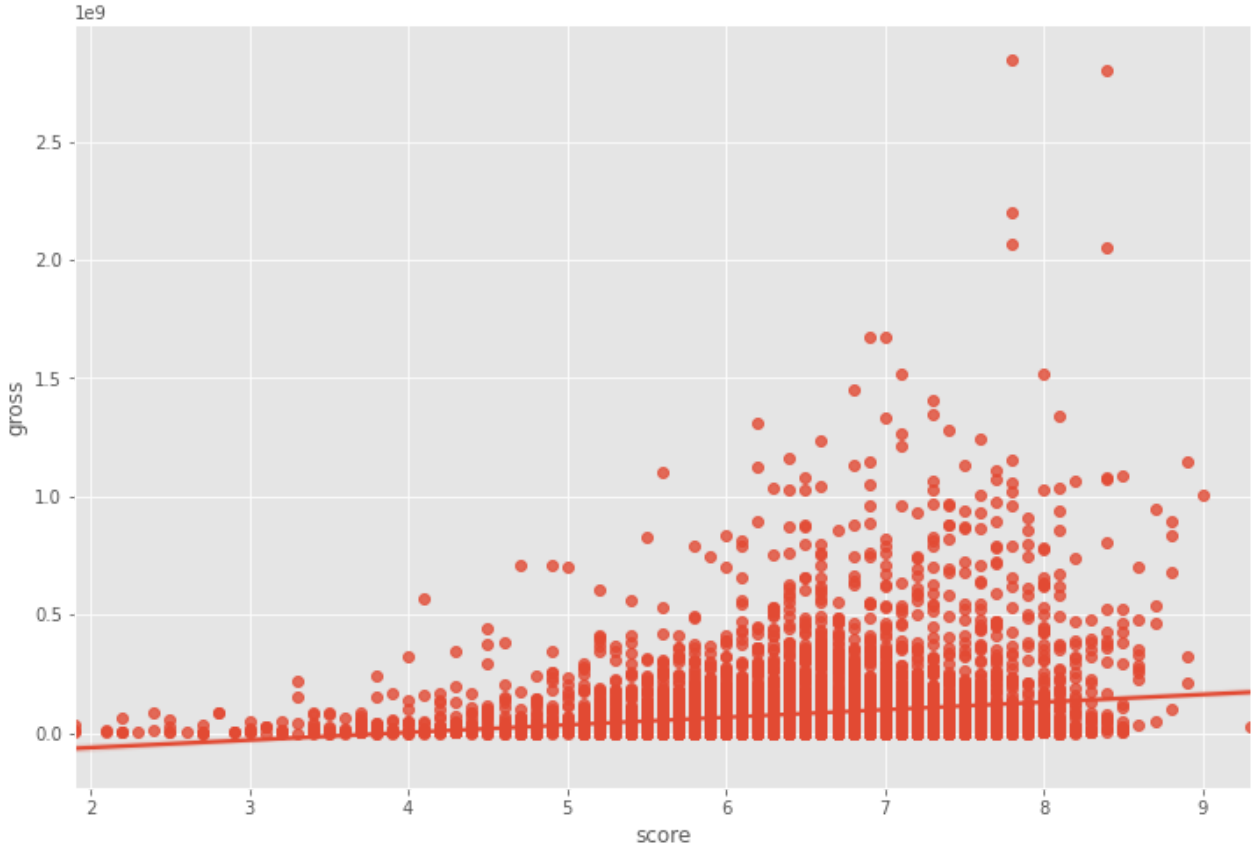
In [ ]:

In [ ]:

In [ ]:

In [10]:
```
sns.regplot(x="score", y="gross", data=df)
```

Out[10]:
```
<AxesSubplot:xlabel='score', ylabel='gross'>
```



In [11]:
```
# Correlation Matrix between all numeric columns

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

Out[11]:

	year	score	votes	budget	gross	runtime
year	1.000000	0.097995	0.222945	0.329321	0.257486	0.120811
score	0.097995	1.000000	0.409182	0.076254	0.186258	0.399451
votes	0.222945	0.409182	1.000000	0.442429	0.630757	0.309212
budget	0.329321	0.076254	0.442429	1.000000	0.740395	0.320447
gross	0.257486	0.186258	0.630757	0.740395	1.000000	0.245216
runtime	0.120811	0.399451	0.309212	0.320447	0.245216	1.000000

In [12]:
```
df.corr(method ='kendall')
```

Out[12]:

	year	score	votes	budget	gross	runtime
year	1.000000	0.067652	0.331465	0.224120	0.200618	0.097184
score	0.067652	1.000000	0.300115	-0.000566	0.086046	0.283611
votes	0.331465	0.300115	1.000000	0.353702	0.548899	0.198240
budget	0.224120	-0.000566	0.353702	1.000000	0.512637	0.235483
gross	0.200618	0.086046	0.548899	0.512637	1.000000	0.168933

	year	score	votes	budget	gross	runtime
runtime	0.097184	0.283611	0.198240	0.235483	0.168933	1.000000

In [13]:
```
df.corr(method ='spearman')
```

Out[13]:

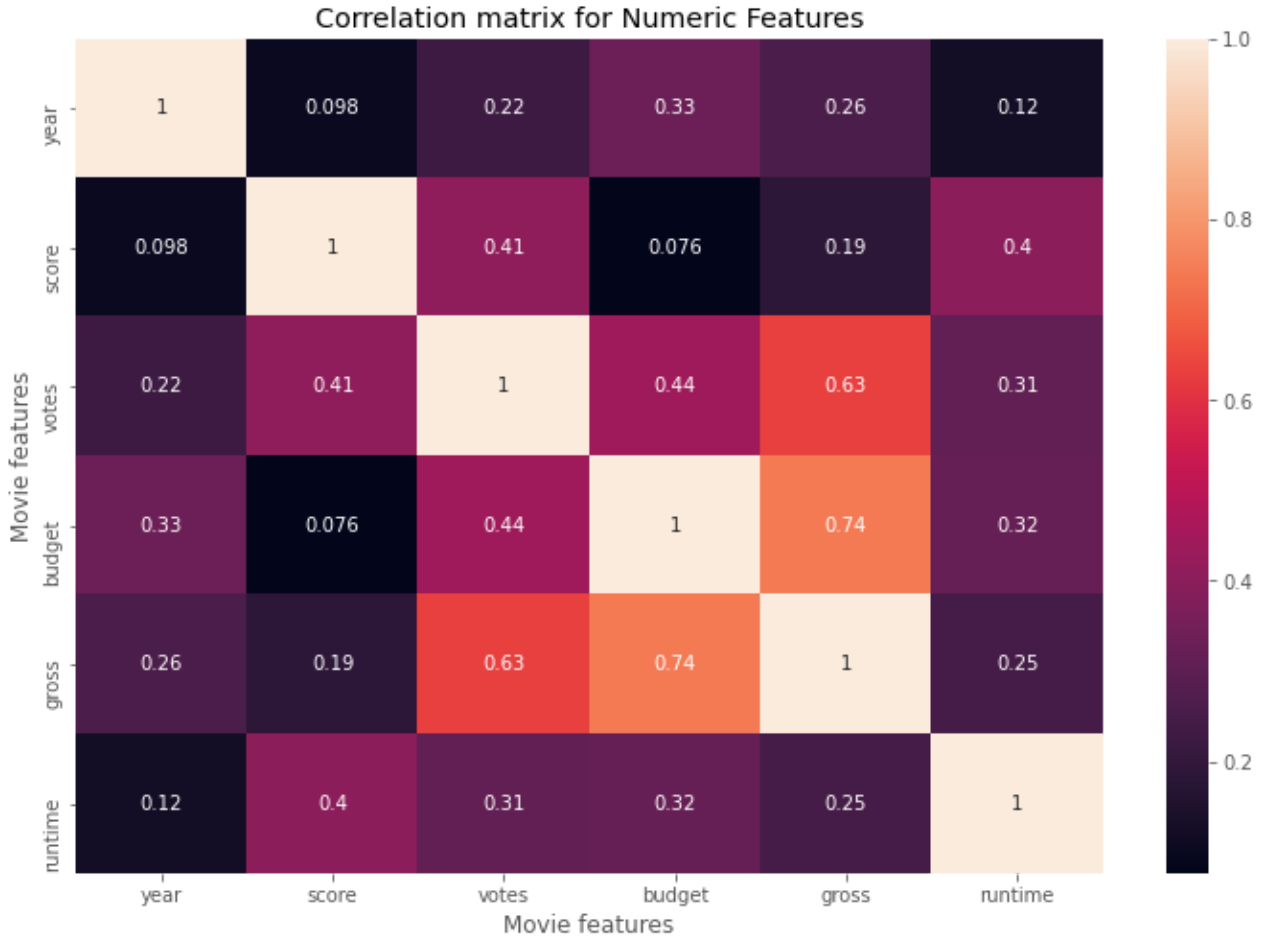
	year	score	votes	budget	gross	runtime
year	1.000000	0.099045	0.469829	0.317336	0.293084	0.142977
score	0.099045	1.000000	0.428138	-0.001403	0.126116	0.399857
votes	0.469829	0.428138	1.000000	0.502466	0.742050	0.290159
budget	0.317336	-0.001403	0.502466	1.000000	0.693670	0.336370
gross	0.293084	0.126116	0.742050	0.693670	1.000000	0.246243
runtime	0.142977	0.399857	0.290159	0.336370	0.246243	1.000000

In [ ]:
```

```

In [ ]:
```

```

In [ ]:
```

```

In [14]:
```
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()
```



In [ ]:

In [15]:
```
# Using factorize - this assigns a random numeric value for each unique categorical val

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

Out[15]:

	name	rating	genre	year	released	score	votes	director	writ
name	1.000000	0.143938	0.036367	0.965761	0.959015	-0.046733	0.287776	0.745905	0.80521
rating	0.143938	1.000000	-0.086723	0.156713	0.146606	0.012595	0.099972	0.085520	0.10362
genre	0.036367	-0.086723	1.000000	0.037184	0.035940	-0.002437	0.023285	0.047288	0.03368
year	0.965761	0.156713	0.037184	1.000000	0.993190	-0.044981	0.312401	0.770497	0.82477
released	0.959015	0.146606	0.035940	0.993190	1.000000	-0.045761	0.299905	0.770876	0.81961
score	-0.046733	0.012595	-0.002437	-0.044981	-0.045761	1.000000	-0.009749	-0.022687	-0.03468
votes	0.287776	0.099972	0.023285	0.312401	0.299905	-0.009749	1.000000	0.192220	0.22412
director	0.745905	0.085520	0.047288	0.770497	0.770876	-0.022687	0.192220	1.000000	0.74834
writer	0.805211	0.103623	0.033688	0.824770	0.819617	-0.034685	0.224122	0.748340	1.00000
star	0.731565	0.093116	0.038649	0.756400	0.754468	-0.009896	0.179601	0.682385	0.67568
country	0.142828	0.000494	-0.015795	0.140216	0.148468	0.023097	-0.045914	0.155471	0.1572C

	name	rating	genre	year	released	score	votes	director	write
budget	0.277488	0.193353	0.073008	0.300621	0.285691	-0.012642	0.398519	0.106617	0.18723
gross	0.947324	0.158582	0.038616	0.980873	0.976423	-0.047041	0.286180	0.750911	0.80557
company	0.591667	-0.028035	0.009566	0.601571	0.607954	-0.028432	0.008900	0.552258	0.54615
runtime	0.048955	0.032741	0.001462	0.050647	0.048235	0.026436	0.106024	-0.011070	0.03226

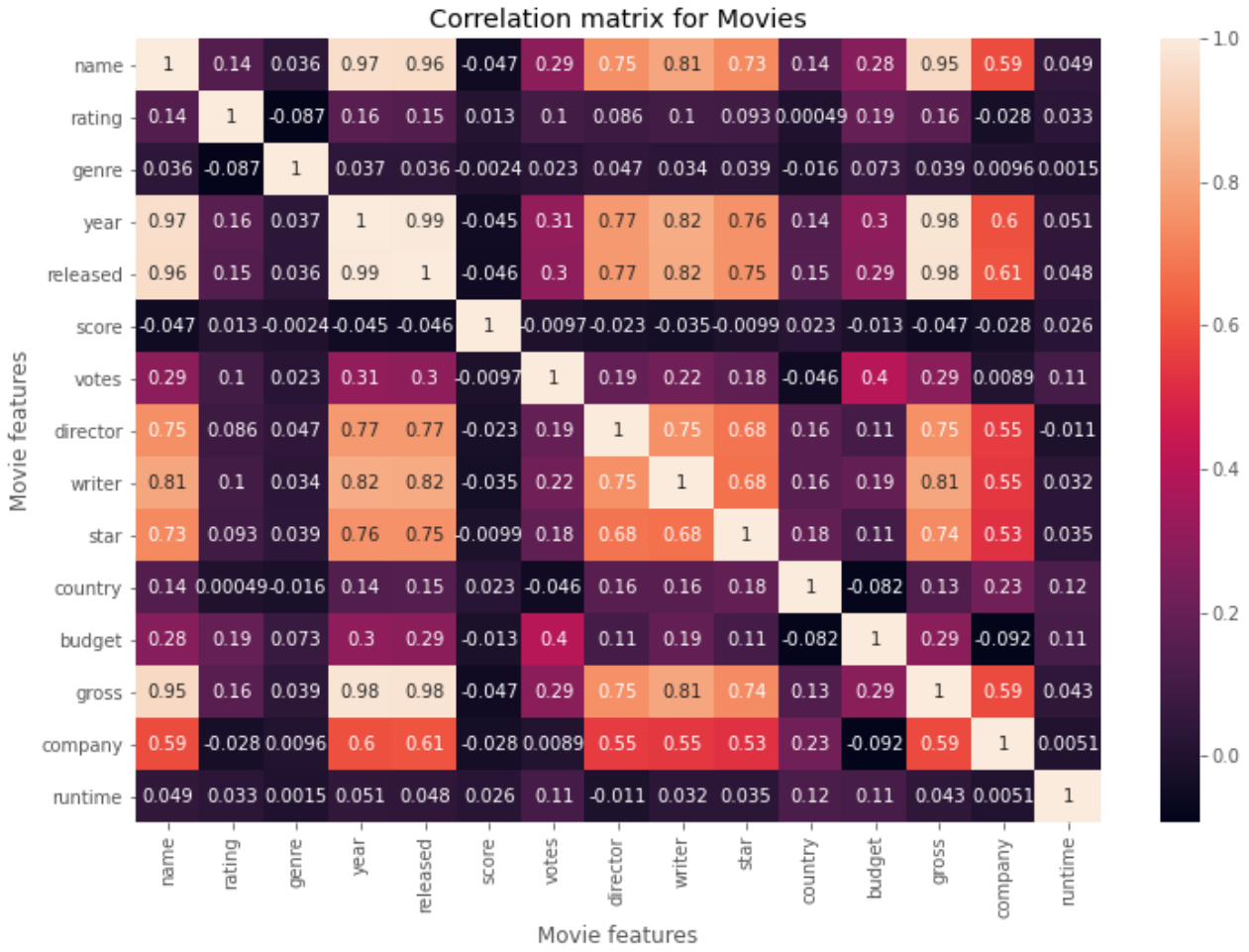
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [16]:
```
correlation_matrix = df.apply(lambda x: x.factorize()[0]).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()
```



In [ ]:

In [ ]:

In [ ]:
```
2
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [19]:
```
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
                       ...
runtime  country     0.124154
         budget      0.112097
         gross       0.042978
         company     0.005137
         runtime     1.000000
Length: 225, dtype: float64
```

In [20]:

```
sorted_pairs = corr_pairs.sort_values(kind="quicksort")

print(sorted_pairs)
```

```
budget   company   -0.092249
company  budget    -0.092249
genre    rating    -0.086723
rating   genre     -0.086723
budget   country   -0.082082
                      ...
year     year       1.000000
genre    genre      1.000000
rating   rating     1.000000
company  company    1.000000
runtime  runtime    1.000000
Length: 225, dtype: float64
```

In [21]:

```
# We can now take a look at the ones that have a high correlation (> 0.5)

strong_pairs = sorted_pairs[abs(sorted_pairs) > 0.5]

print(strong_pairs)
```

```
star      company    0.527116
company   star       0.527116
          writer     0.546151
writer    company    0.546151
director  company    0.552258
                       ...
year      year       1.000000
genre     genre      1.000000
rating    rating     1.000000
company   company    1.000000
runtime   runtime    1.000000
Length: 71, dtype: float64
```

In [22]:

```
# 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
```

Out[22]:
```
company
Warner Bros.                 56491421806
Universal Pictures           52514188890
Columbia Pictures            43008941346
Paramount Pictures           40493607415
Twentieth Century Fox        40257053857
Walt Disney Pictures         36327887792
New Line Cinema              19883797684
Marvel Studios               15065592411
DreamWorks Animation         11873612858
Touchstone Pictures          11795832638
Dreamworks Pictures          11635441081
Metro-Goldwyn-Mayer (MGM)     9230230105
Summit Entertainment          8373718838
Pixar Animation Studios       7886344526
Fox 2000 Pictures             7443502667
Name: gross, dtype: int64
```

In [23]:
```
df['Year'] = df['released'].astype(str).str[:4]
df
```

Out[23]:

	name	rating	genre	year	released	score	votes	director	writer	st
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ja Nicholsc
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brool Shiel
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Ma Ham
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robe Ha
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Che Cha
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Shanno Bor
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	Micha Saque

	name	rating	genre	year	released	score	votes	director	writer	st
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	Onyan Lau
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Christi R
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siyabon Maba

7668 rows × 16 columns

In [24]:
```
df.groupby(['company', 'year'])[["gross"]].sum()
```

Out[24]:

		gross
company	**year**	
"DIA" Productions GmbH & Co. KG	**2003**	44350926.0
"Weathering With You" Film Partners	**2019**	193457467.0
.406 Production	**1996**	10580.0
1+2 Seisaku Iinkai	**2000**	1196218.0
10 West Studios	**2010**	814906.0
...	**...**	...
i am OTHER	**2015**	17986781.0
i5 Films	**2001**	10031529.0
iDeal Partners Film Fund	**2013**	506303.0
micro_scope	**2010**	7099598.0
thefyzz	**2017**	62198461.0

4536 rows × 1 columns

In [25]:
```
CompanyGrossSum = df.groupby(['company', 'year'])[["gross"]].sum()

CompanyGrossSumSorted = CompanyGrossSum.sort_values(['gross','company','year'], ascendi

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

CompanyGrossSumSorted
```

Out[25]:
```
company                year
Walt Disney Pictures   2019    5773131804
```

```
Marvel Studios          2018    4018631866
Universal Pictures      2015    3834354888
Twentieth Century Fox   2009    3793491246
Walt Disney Pictures    2017    3789382071
Paramount Pictures      2011    3565705182
Warner Bros.            2010    3300479986
                        2011    3223799224
Walt Disney Pictures    2010    3104474158
Paramount Pictures      2014    3071298586
Columbia Pictures       2006    2934631933
                        2019    2932757449
Marvel Studios          2019    2797501328
Warner Bros.            2018    2774168962
Columbia Pictures       2011    2738363306
Name: gross, dtype: int64
```

In [26]:

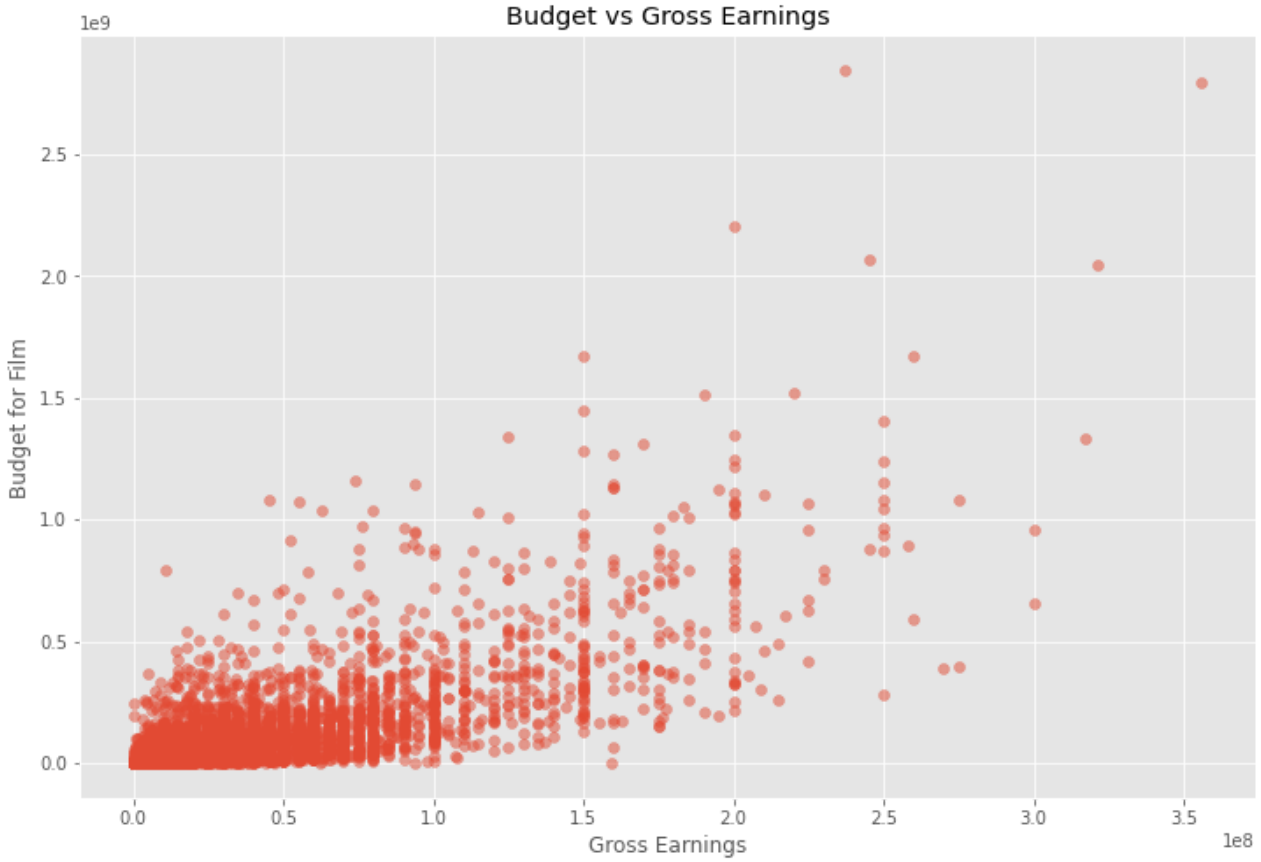
```
CompanyGrossSum = df.groupby(['company'])[["gross"]].sum()

CompanyGrossSumSorted = CompanyGrossSum.sort_values(['gross','company'], ascending = Fa

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

CompanyGrossSumSorted
```

Out[26]:

```
company
Warner Bros.                 56491421806
Universal Pictures           52514188890
Columbia Pictures            43008941346
Paramount Pictures           40493607415
Twentieth Century Fox        40257053857
Walt Disney Pictures         36327887792
New Line Cinema              19883797684
Marvel Studios               15065592411
DreamWorks Animation         11873612858
Touchstone Pictures          11795832638
Dreamworks Pictures          11635441081
Metro-Goldwyn-Mayer (MGM)     9230230105
Summit Entertainment          8373718838
Pixar Animation Studios       7886344526
Fox 2000 Pictures             7443502667
Name: gross, dtype: int64
```

In [27]:

```
plt.scatter(x=df['budget'], y=df['gross'], alpha=0.5)
plt.title('Budget vs Gross Earnings')
plt.xlabel('Gross Earnings')
plt.ylabel('Budget for Film')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [28]:
```
df
```

Out[28]:

	name	rating	genre	year	released	score	votes	director	writer	st
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ja Nicholsc
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brool Shiel
2	Star Wars: Episode V - The Empire Strikes Back	PG	Action	1980	June 20, 1980 (United States)	8.7	1200000.0	Irvin Kershner	Leigh Brackett	Ma Ham
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robe Ha
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Che Cha
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Shanno Bor
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	Micha Saque
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	Onyan Lau
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Christi R

	name	rating	genre	year	released	score	votes	director	writer	st
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siyabon Maba

7668 rows × 16 columns

In [29]:
```
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
```

Out[29]:

	name	rating	genre	year	released	score	votes	director	writer	star	country	budg
0	6587	6	6	1980	1705	8.4	927000.0	2589	4014	1047	54	1900000
1	5573	6	1	1980	1492	5.8	65000.0	2269	1632	327	55	450000
2	5142	4	0	1980	1771	8.7	1200000.0	1111	2567	1745	55	1800000
3	286	4	4	1980	1492	7.7	221000.0	1301	2000	2246	55	350000
4	1027	6	4	1980	1543	7.3	108000.0	1054	521	410	55	600000
...	...	...	...	...	...	...	...	...	...	...	...	
7663	3705	-1	6	2020	2964	3.1	18.0	1500	2289	2421	55	700
7664	1678	-1	4	2020	1107	4.7	36.0	774	2614	1886	55	Na
7665	4717	-1	6	2020	193	5.7	29.0	2061	2683	2040	55	5875
7666	2843	-1	6	2020	2817	NaN	NaN	1184	1824	450	55	1500
7667	5394	-1	10	2020	391	5.7	7.0	2165	3344	2463	44	Na

7668 rows × 16 columns

In [30]:
```
df_numerized.corr(method='pearson')
```

Out[30]:

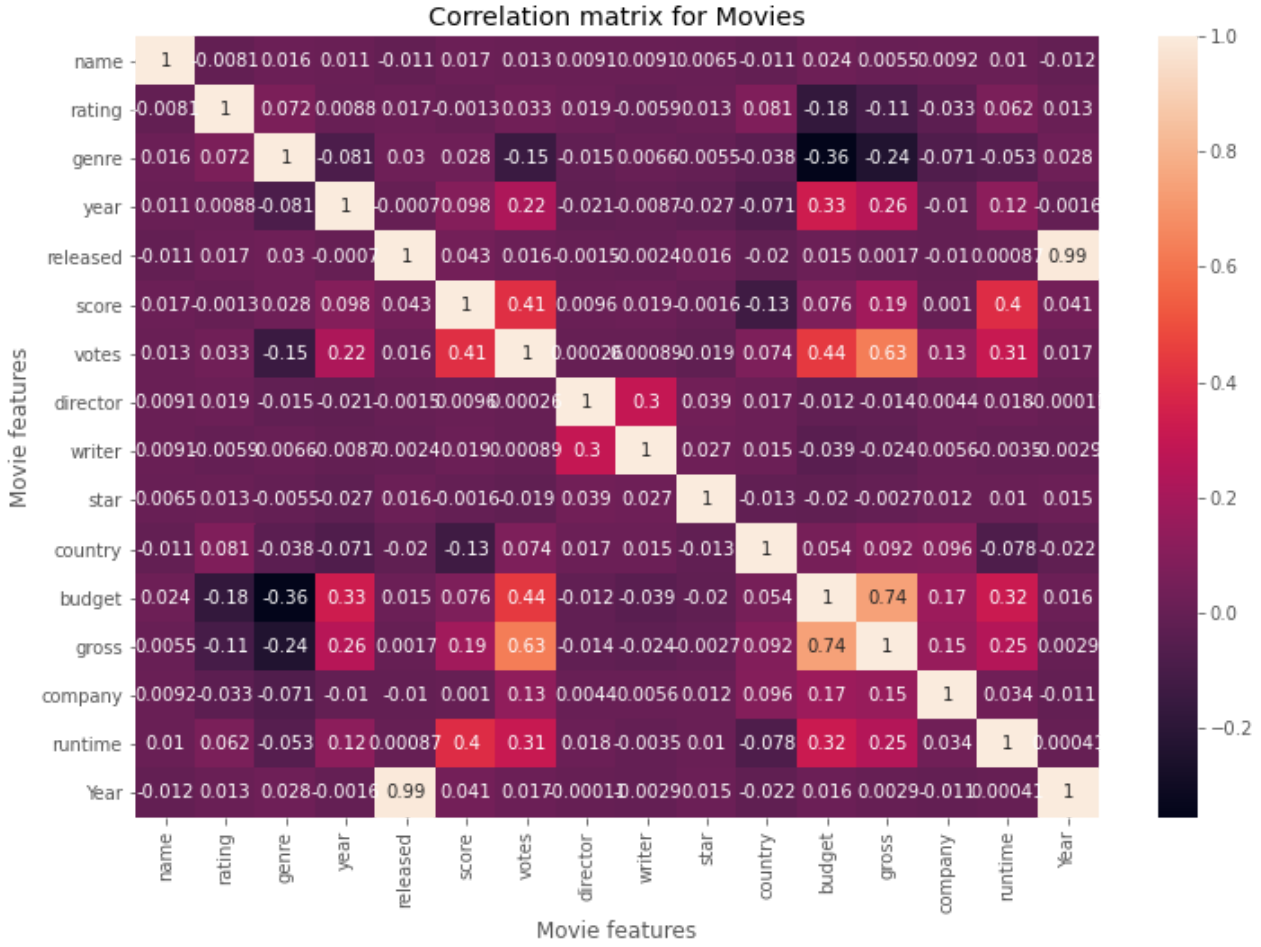
	name	rating	genre	year	released	score	votes	director	writ
name	1.000000	-0.008069	0.016355	0.011453	-0.011311	0.017097	0.013088	0.009079	0.0090
rating	-0.008069	1.000000	0.072423	0.008779	0.016613	-0.001314	0.033225	0.019483	-0.0059
genre	0.016355	0.072423	1.000000	-0.081261	0.029822	0.027965	-0.145307	-0.015258	0.0065
year	0.011453	0.008779	-0.081261	1.000000	-0.000695	0.097995	0.222945	-0.020795	-0.0086

	name	rating	genre	year	released	score	votes	director	writ
released	-0.011311	0.016613	0.029822	-0.000695	1.000000	0.042788	0.016097	-0.001478	-0.0024(
score	0.017097	-0.001314	0.027965	0.097995	0.042788	1.000000	0.409182	0.009559	0.01941
votes	0.013088	0.033225	-0.145307	0.222945	0.016097	0.409182	1.000000	0.000260	0.0008(
director	0.009079	0.019483	-0.015258	-0.020795	-0.001478	0.009559	0.000260	1.000000	0.2990(
writer	0.009081	-0.005921	0.006567	-0.008656	-0.002404	0.019416	0.000892	0.299067	1.0000(
star	0.006472	0.013405	-0.005477	-0.027242	0.015777	-0.001609	-0.019282	0.039234	0.0272(
country	-0.010737	0.081244	-0.037615	-0.070938	-0.020427	-0.133348	0.073625	0.017490	0.0153(
budget	0.023970	-0.176002	-0.356564	0.329321	0.014683	0.076254	0.442429	-0.012272	-0.0394(
gross	0.005533	-0.107339	-0.235650	0.257486	0.001659	0.186258	0.630757	-0.014441	-0.0235(
company	0.009211	-0.032943	-0.071067	-0.010431	-0.010474	0.001030	0.133204	0.004404	0.0056(
runtime	0.010392	0.062145	-0.052711	0.120811	0.000868	0.399451	0.309212	0.017624	-0.0035(
Year	-0.011725	0.013475	0.028397	-0.001562	0.993694	0.040993	0.017337	-0.000105	-0.0028(

In [ ]:

In [ ]:

In [31]:
```
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()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [38]:
```
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
```

In [ ]:

In [ ]:

In [ ]:

In [39]:
```
df[cat_columns] = df[cat_columns].apply(lambda x: x.cat.codes)

df
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-39-70c45a254ab5> in <module>
----> 1 df[cat_columns] = df[cat_columns].apply(lambda x: x.cat.codes)
      2
      3 df

NameError: name 'cat_columns' is not defined
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

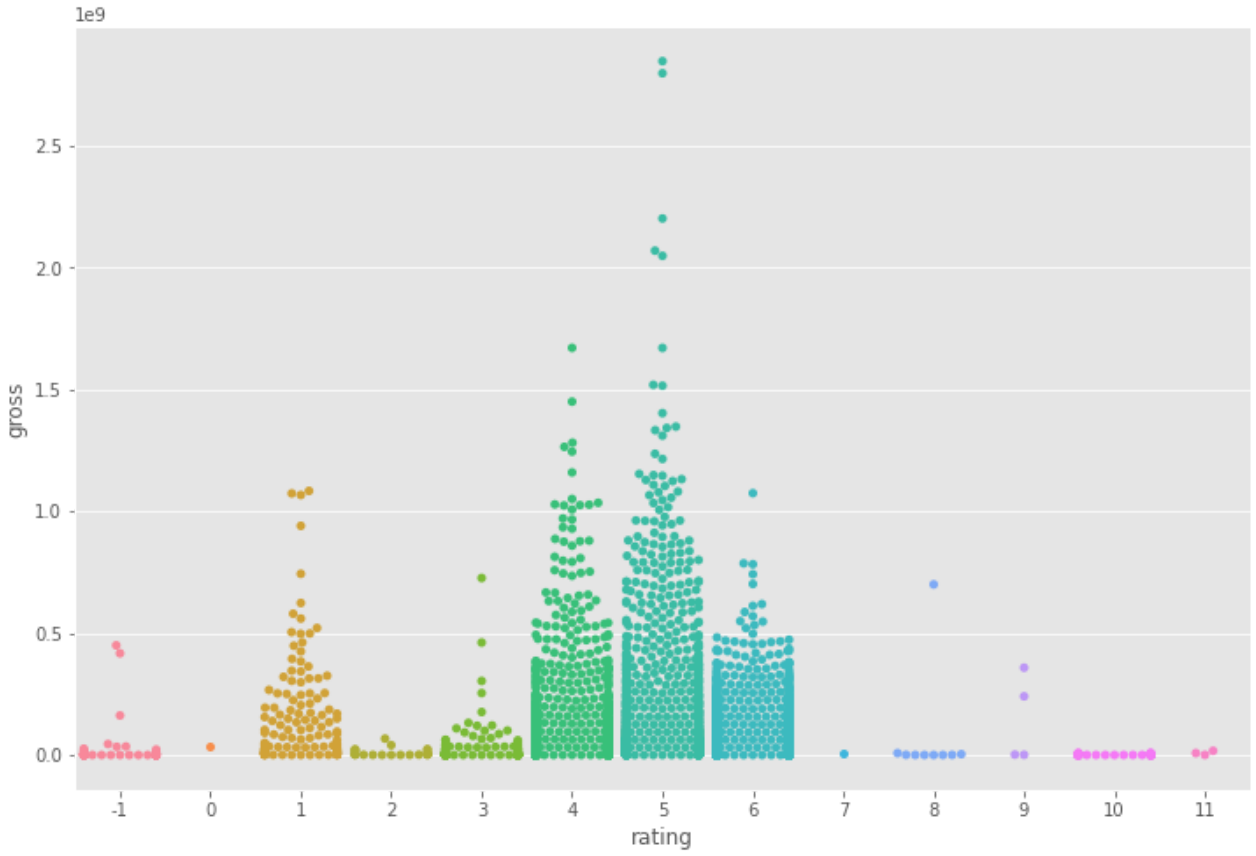
In [40]:
```
sns.swarmplot(x="rating", y="gross", data=df)
```

```
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 53.
2% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 48.
4% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
```

```
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 60.
9% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 80.
6% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 84.
4% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 88.
2% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 94.
4% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 11.
1% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Megan\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 76.
9% of the points cannot be placed; you may want to decrease the size of the markers or u
se stripplot.
  warnings.warn(msg, UserWarning)
```

Out[40]:

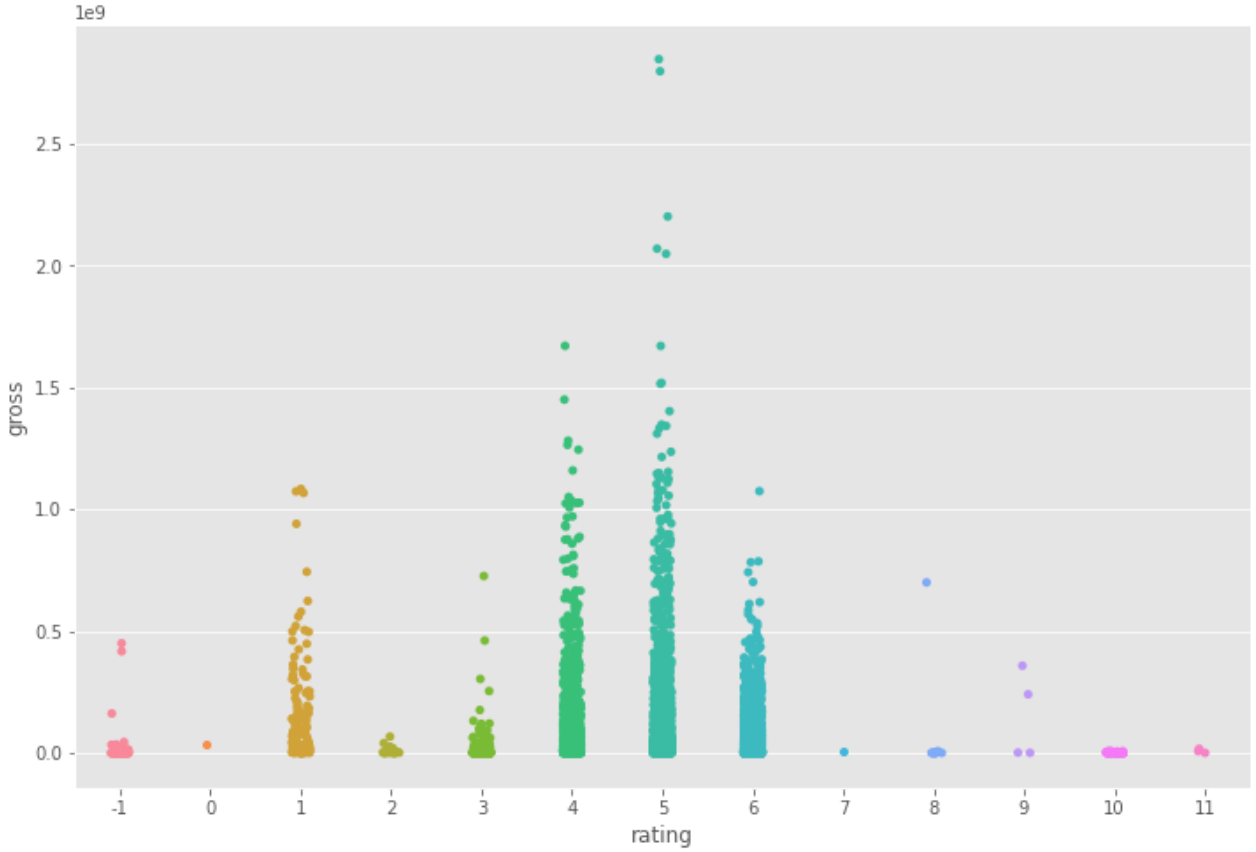
```
<AxesSubplot:xlabel='rating', ylabel='gross'>
```



In [ ]:

In [42]:
```
sns.stripplot(x="rating", y="gross", data=df)
```

Out[42]: <AxesSubplot:xlabel='rating', ylabel='gross'>



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: