Importing packages

In [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
```

Loading the data

In [2]:
```
df = pd.read_csv(r'B:\MY COMPUTER (HOME)\2 IT\data science courses\projects\Data ana
df
```

Out[2]:

	name	rating	genre	year	released	score	votes	director	writer	
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ni
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	
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	
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	S
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Si

7668 rows × 15 columns

In [3]:
```
# We need to see if we have any missing data

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 [4]:
```
# 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 [ ]:
```
# change the datatype of column
df['budget']=df['budget'].astype('int64')
df['gross']=df['gross'].astype('int64')
```

In [ ]:
```
# create correct year column
# df['yearcorrect']=df['released'].astype(str).str[:4]
# df
```

In [6]:
```
# Order our Data a little bit to see

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

Out[6]:

	name	rating	genre	year	released	score	votes	director	writer	
5445	Avatar	PG-13	Action	2009	December 18, 2009 (United States)	7.8	1100000.0	James Cameron	James Cameron	Worth
7445	Avengers: Endgame	PG-13	Action	2019	April 26, 2019 (United States)	8.4	903000.0	Anthony Russo	Christopher Markus	Dow
3045	Titanic	PG-13	Drama	1997	December 19, 1997 (United States)	7.8	1100000.0	James Cameron	James Cameron	Le Di
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
7244	Avengers: Infinity War	PG-13	Action	2018	April 27, 2018 (United States)	8.4	897000.0	Anthony Russo	Christopher Markus	Dow
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	Sh
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	M Sa
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	O
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	Ch
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Siya M

7668 rows × 15 columns

In [7]:
```
# drop any duplicates
df.drop_duplicates()
```

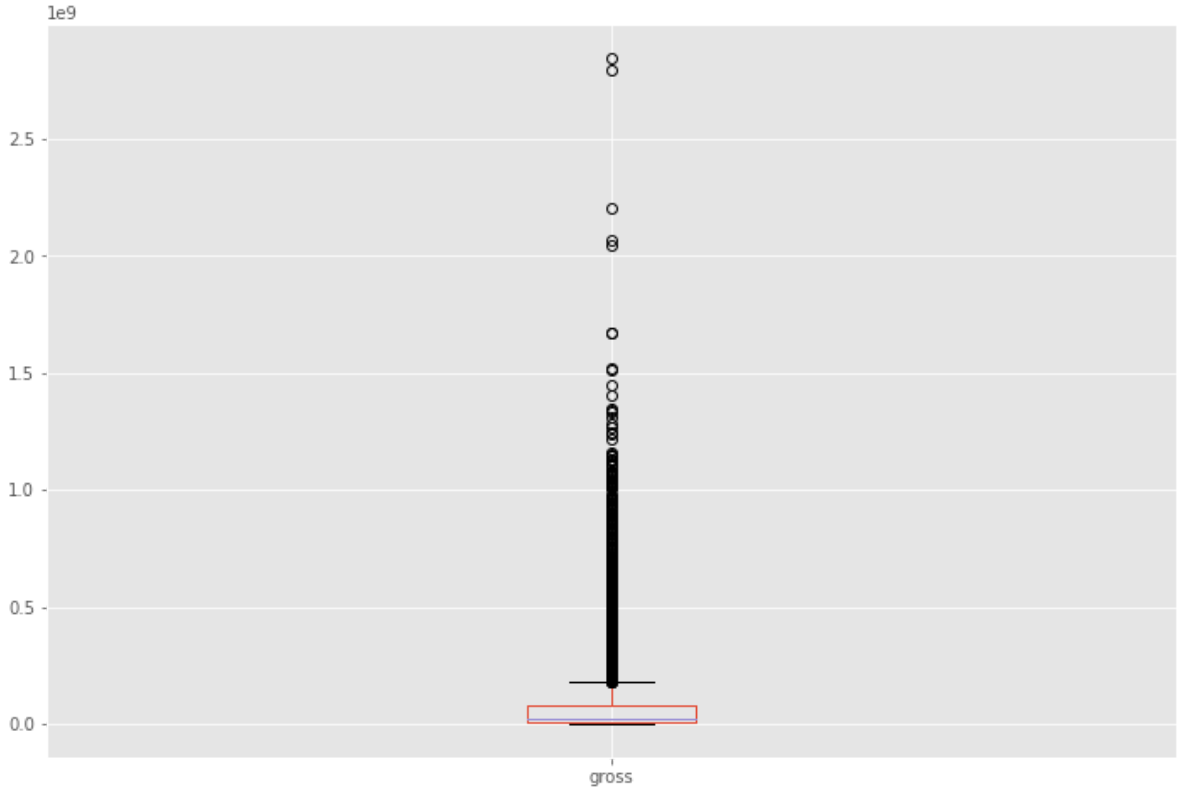
Out[7]:

	name	rating	genre	year	released	score	votes	director	writer	
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Ni
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	
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	
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	
...	...	...	...	...	...	...	...	...	...	
7663	More to Life	NaN	Drama	2020	October 23, 2020 (United States)	3.1	18.0	Joseph Ebanks	Joseph Ebanks	S
7664	Dream Round	NaN	Comedy	2020	February 7, 2020 (United States)	4.7	36.0	Dusty Dukatz	Lisa Huston	
7665	Saving Mbango	NaN	Drama	2020	April 27, 2020 (Cameroon)	5.7	29.0	Nkanya Nkwai	Lynno Lovert	
7666	It's Just Us	NaN	Drama	2020	October 1, 2020 (United States)	NaN	NaN	James Randall	James Randall	
7667	Tee em el	NaN	Horror	2020	August 19, 2020 (United States)	5.7	7.0	Pereko Mosia	Pereko Mosia	Si

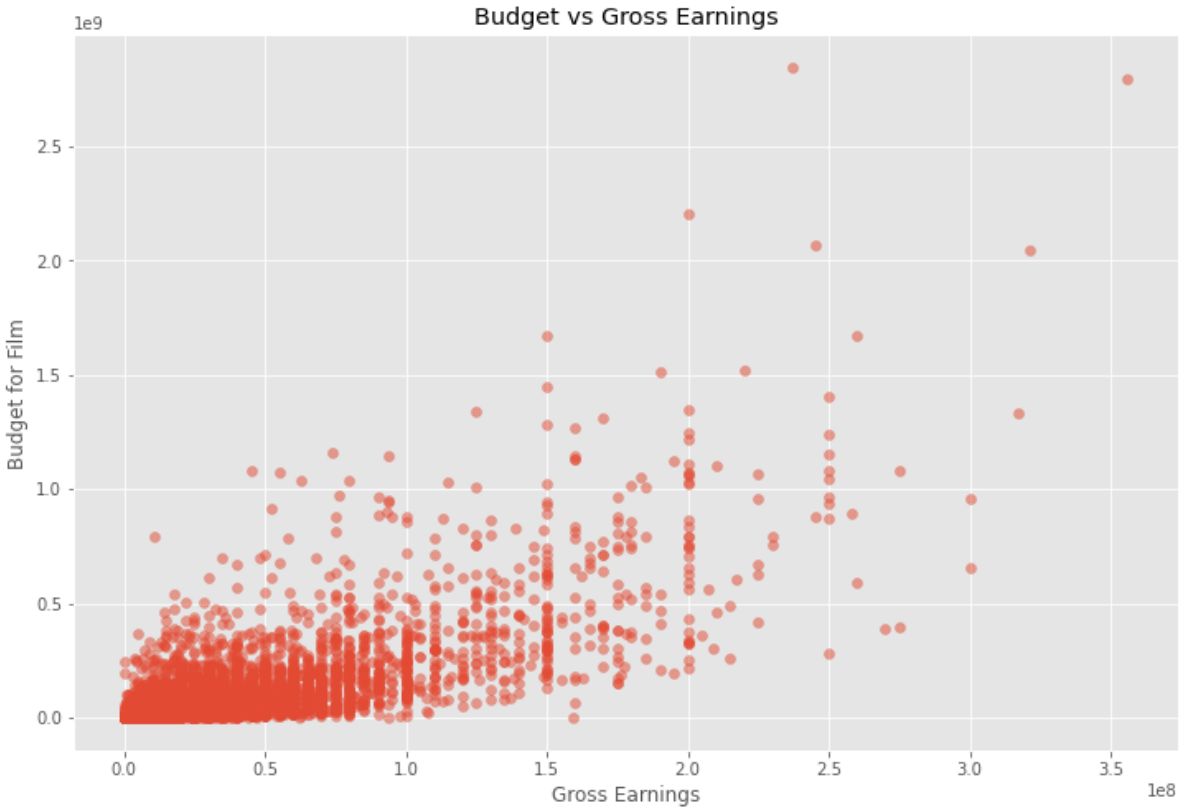
7668 rows × 15 columns

In [8]:
```
# Are there any Outliers?

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

Out[8]:
```
<AxesSubplot:>
```

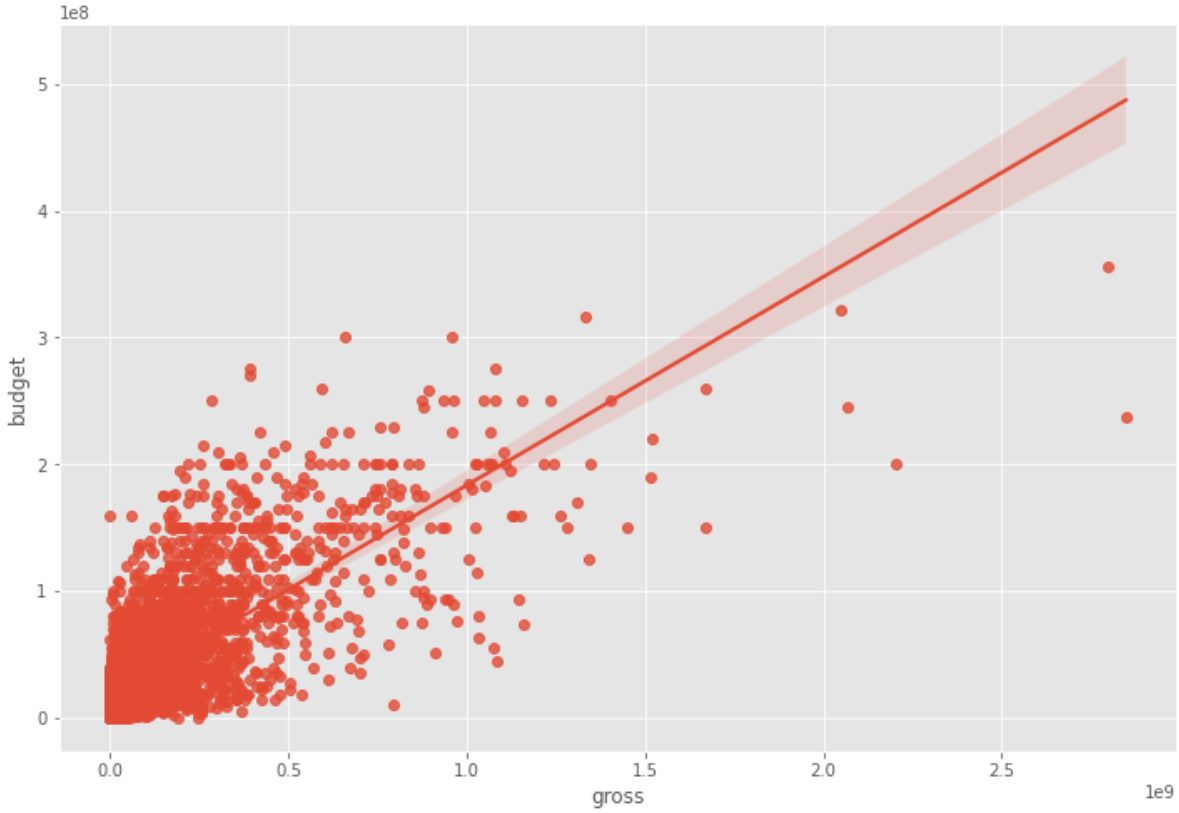


In [9]:
```
# scatter plot with budget vs gross
plt.scatter(x=df['budget'], y=df['gross'], alpha=0.5)
plt.title('Budget vs Gross Earnings')
plt.xlabel('Gross Earnings')
plt.ylabel('Budget for Film')
plt.show()
```



In [10]:
```
# plot budget vs gross using seaborn
sns.regplot(x="gross", y="budget", data=df)
```

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



Correlation

In [11]:
```
# Correlation Matrix between all numeric columns
# default method is pearson
# high corr bet gross and budget
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
runtime	0.097184	0.283611	0.198240	0.235483	0.168933	1.000000

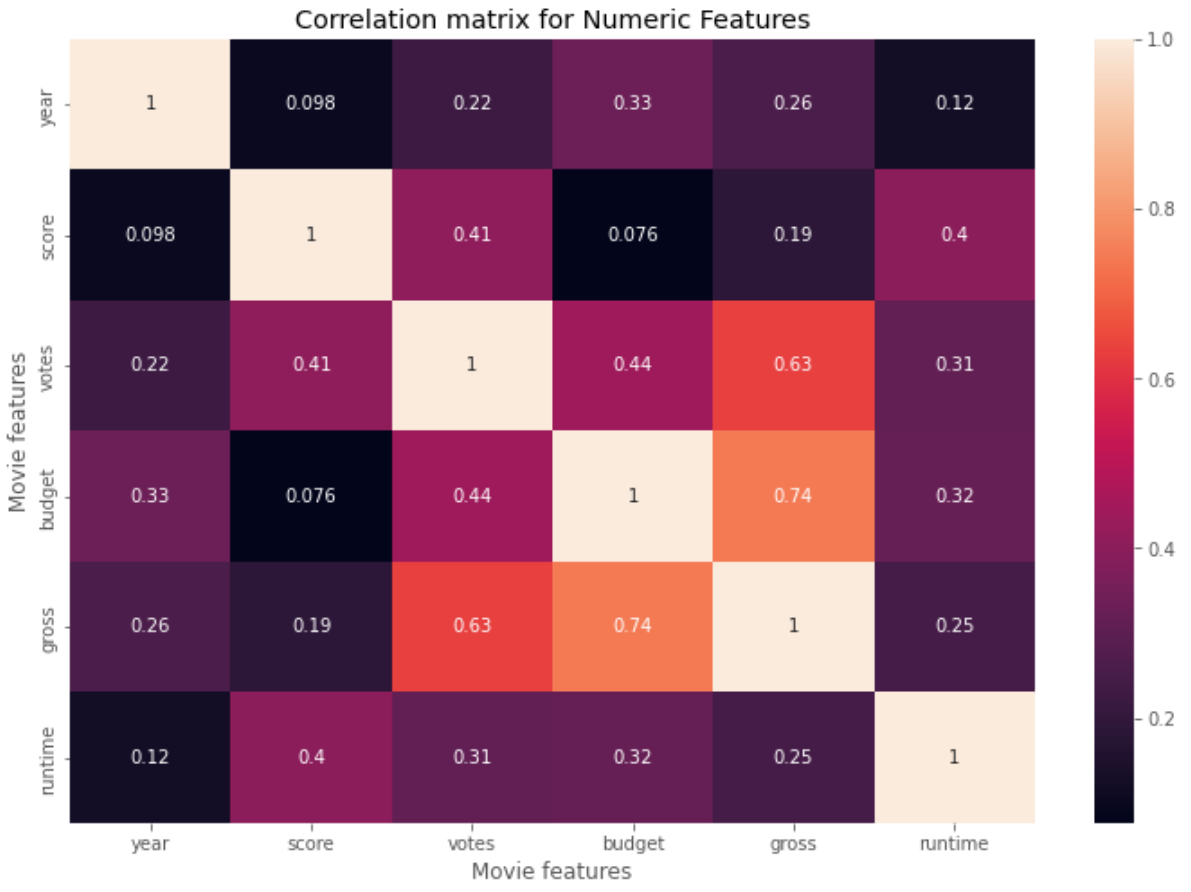
In [13]:
```
df.corr(method ='spearman')
```

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 [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 [15]:

```
# numerization- giving a number to objects column for correlation
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
```

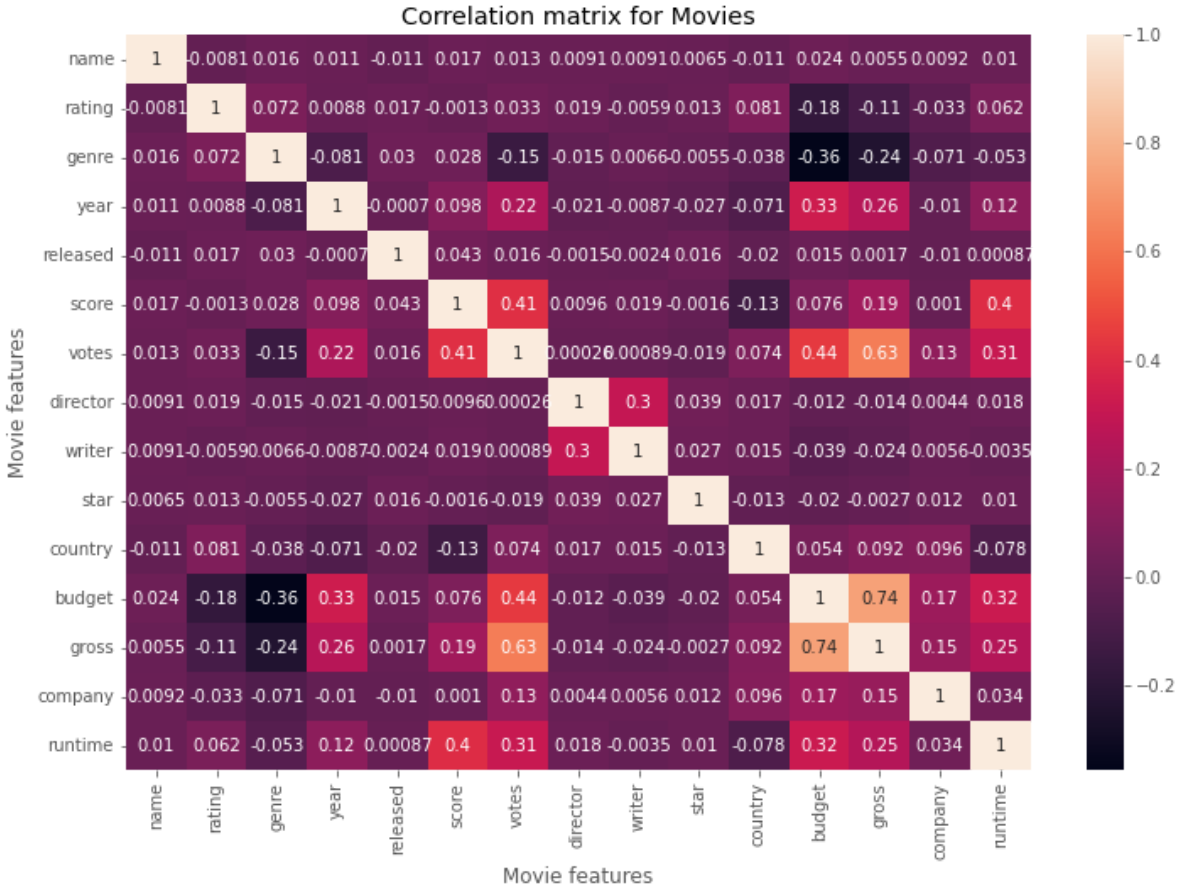
Out[15]:

	name	rating	genre	year	released	score	votes	director	writer	star	country	b
0	6587	6	6	1980	1705	8.4	927000.0	2589	4014	1047	54	1900
1	5573	6	1	1980	1492	5.8	65000.0	2269	1632	327	55	450
2	5142	4	0	1980	1771	8.7	1200000.0	1111	2567	1745	55	1800
3	286	4	4	1980	1492	7.7	221000.0	1301	2000	2246	55	350
4	1027	6	4	1980	1543	7.3	108000.0	1054	521	410	55	600
...	...	...	...	...	...	...	...	...	...	...	...	
7663	3705	-1	6	2020	2964	3.1	18.0	1500	2289	2421	55	
7664	1678	-1	4	2020	1107	4.7	36.0	774	2614	1886	55	
7665	4717	-1	6	2020	193	5.7	29.0	2061	2683	2040	55	5
7666	2843	-1	6	2020	2817	NaN	NaN	1184	1824	450	55	1
7667	5394	-1	10	2020	391	5.7	7.0	2165	3344	2463	44	

7668 rows × 15 columns

In [16]:

```
# correlation of all the 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()
```



In [17]:
```
correlation_mat = df_numerized.corr()

corr_pairs = correlation_mat.unstack()

print(corr_pairs)
```

```
name     name         1.000000
         rating      -0.008069
         genre        0.016355
         year         0.011453
         released    -0.011311
                          ...
runtime  country     -0.078412
         budget       0.320447
         gross        0.245216
         company      0.034402
         runtime      1.000000
Length: 225, dtype: float64
```

In [18]:
```
sorted_pairs = corr_pairs.sort_values(kind="quicksort")

sorted_pairs
```

Out[18]:
```
budget   genre      -0.356564
genre    budget     -0.356564
         gross      -0.235650
gross    genre      -0.235650
rating   budget     -0.176002
                        ...
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 [19]:
```
# 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)
```

```
gross     votes       0.630757
votes     gross       0.630757
budget    gross       0.740395
gross     budget      0.740395
name      name        1.000000
director  director    1.000000
gross     gross       1.000000
budget    budget      1.000000
country   country     1.000000
star      star        1.000000
writer    writer      1.000000
votes     votes       1.000000
score     score       1.000000
released  released    1.000000
year      year        1.000000
genre     genre       1.000000
rating    rating      1.000000
company   company     1.000000
runtime   runtime     1.000000
dtype: float64
```

In [20]:
```
# 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[20]:
```
company
2319    56491421806
2281    52514188890
731     43008941346
1812    40493607415
2253    40257053857
2316    36327887792
1713    19883797684
1606    15065592411
887     11873612858
2232    11795832638
889     11635441081
1637     9230230105
2147     8373718838
1856     7886344526
1109     7443502667
Name: gross, dtype: int64
```

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

```
C:\Users\Admin\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 marke
rs or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Admin\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 marke
rs or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Admin\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 marke
rs or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Admin\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 marke
rs or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\Admin\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 marke
rs or use stripplot.
  warnings.warn(msg, UserWarning)
```

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

In [ ]:
```

```