In [1]:
```
# import libraries
import pandas as pd
import seaborn as sns
import numpy as np

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

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (12,8) # adjusts configuration of plots
```

In [2]:
```
# Read Data csv
df = pd.read_csv(r'C:\Users\haoea\Desktop\Portfolio Projects\movies\movies.csv')

df.head()
```

Out[2]:

	name	rating	genre	year	released	score	votes	director	writer	star
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Jack Nicholson
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brooke Shields
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	Mark Hamil
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robert Hays
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Chevy Chase

In [3]:
```
# Check for missing data

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

```
name - 0.0%
rating - 0.010041731872717789%
genre - 0.0%
year - 0.0%
released - 0.00026082420448617 63%
score - 0.00039123630672926 45%
votes - 0.00039123630672926 45%
director - 0.0%
writer - 0.00039123630672926 45%
star - 0.00013041210224308815%
country - 0.00039123630672926 45%
budget - 0.2831246739697444%
gross - 0.02464788732394366%
company - 0.002217005738132499%
runtime - 0.0005216484089723526%
```

In [4]:
```
# Dealing with missing data
# In this case, drop data
df = df.dropna()
```

In [5]:
```
# Data Types for columns
df.dtypes
```

Out[5]:
```
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 [6]:
```
# Changing Data Types Budget,Gross,
df['budget'] = df['budget'].astype('int64')

df['gross'] = df['gross'].astype('int64')
```

In [7]:
```
# Extracting Year from released
df['yearcorrect'] = df['released'].str.extract(pat = '([0-9]{4})').astype('Int64')
```

In [8]:
```
# Checking year
df
```

Out[8]:

	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	Nicho
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Bro Shi
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	M Ha
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Ro I
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Cl Cl
...	...	...	...	...	...	...	...	...	...	
7648	Bad Boys for Life	R	Action	2020	January 17, 2020 (United States)	6.6	140000.0	Adil El Arbi	Peter Craig	S
7649	Sonic the Hedgehog	PG	Action	2020	February 14, 2020 (United States)	6.5	102000.0	Jeff Fowler	Pat Casey	Schw
7650	Dolittle	PG	Adventure	2020	January 17, 2020 (United States)	5.6	53000.0	Stephen Gaghan	Stephen Gaghan	Ro Dov
7651	The Call of the Wild	PG	Adventure	2020	February 21, 2020 (United States)	6.8	42000.0	Chris Sanders	Michael Green	Harr
7652	The Eight Hundred	Not Rated	Action	2020	August 28, 2020 (United States)	6.8	3700.0	Hu Guan	Hu Guan	zh Hu

5421 rows × 16 columns

In [9]:
```
# Check the highest grossing movies , show all rows
# pd.set_option('display.max_rows', None)

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

Out[9]:

	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	Wor
7445	Avengers: Endgame	PG-13	Action	2019	April 26, 2019 (United States)	8.4	903000.0	Anthony Russo	Christopher Markus	Dc
3045	Titanic	PG-13	Drama	1997	December 19, 1997 (United States)	7.8	1100000.0	James Cameron	James Cameron	L I
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	Dais
7244	Avengers: Infinity War	PG-13	Action	2018	April 27, 2018 (United States)	8.4	897000.0	Anthony Russo	Christopher Markus	Dc
...	...	...	...	...	...	...	...	...	...	
5640	Tanner Hall	R	Drama	2009	January 15, 2015 (Sweden)	5.8	3500.0	Francesca Gregorini	Tatiana von Fürstenberg	
2434	Philadelphia Experiment II	PG-13	Action	1993	June 4, 1994 (South Korea)	4.5	1900.0	Stephen Cornwell	Wallace C. Bennett	
3681	Ginger Snaps	Not Rated	Drama	2000	May 11, 2001 (Canada)	6.8	43000.0	John Fawcett	Karen Walton	
272	Parasite	R	Horror	1982	March 12, 1982 (United States)	3.9	2300.0	Charles Band	Alan J. Adler	
3203	Trojan War	PG-13	Comedy	1997	October 1, 1997 (Brazil)	5.7	5800.0	George Huang	Andy Burg	Wi

5421 rows × 16 columns

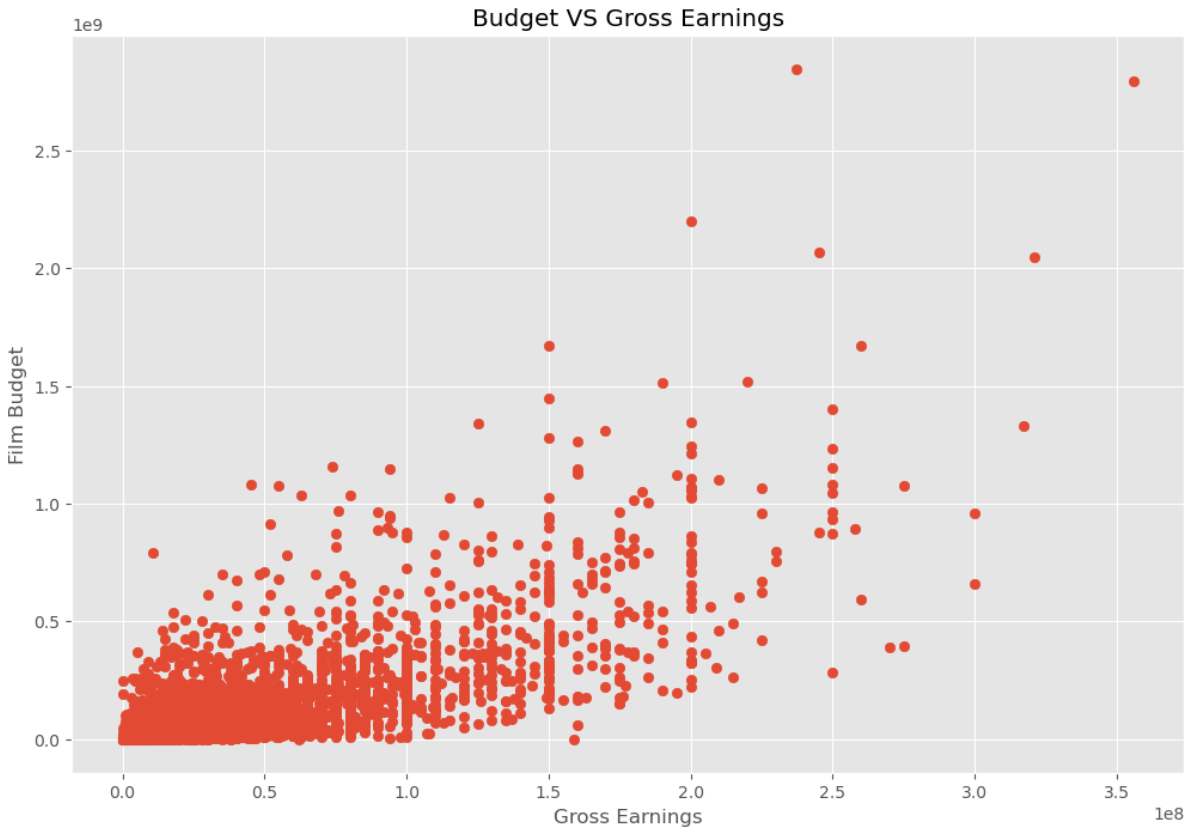
In [10]:
```
# Drop Duplicates

df.drop_duplicates()
```

Out[10]:

	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	Nicho
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Bro Shi
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	M Ha
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Ro I
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Cl Cl
...	...	...	...	...	...	...	...	...	...	
7648	Bad Boys for Life	R	Action	2020	January 17, 2020 (United States)	6.6	140000.0	Adil El Arbi	Peter Craig	S
7649	Sonic the Hedgehog	PG	Action	2020	February 14, 2020 (United States)	6.5	102000.0	Jeff Fowler	Pat Casey	Schw
7650	Dolittle	PG	Adventure	2020	January 17, 2020 (United States)	5.6	53000.0	Stephen Gaghan	Stephen Gaghan	Ro Dov
7651	The Call of the Wild	PG	Adventure	2020	February 21, 2020 (United States)	6.8	42000.0	Chris Sanders	Michael Green	Harr
7652	The Eight Hundred	Not Rated	Action	2020	August 28, 2020 (United States)	6.8	3700.0	Hu Guan	Hu Guan	zh Hu

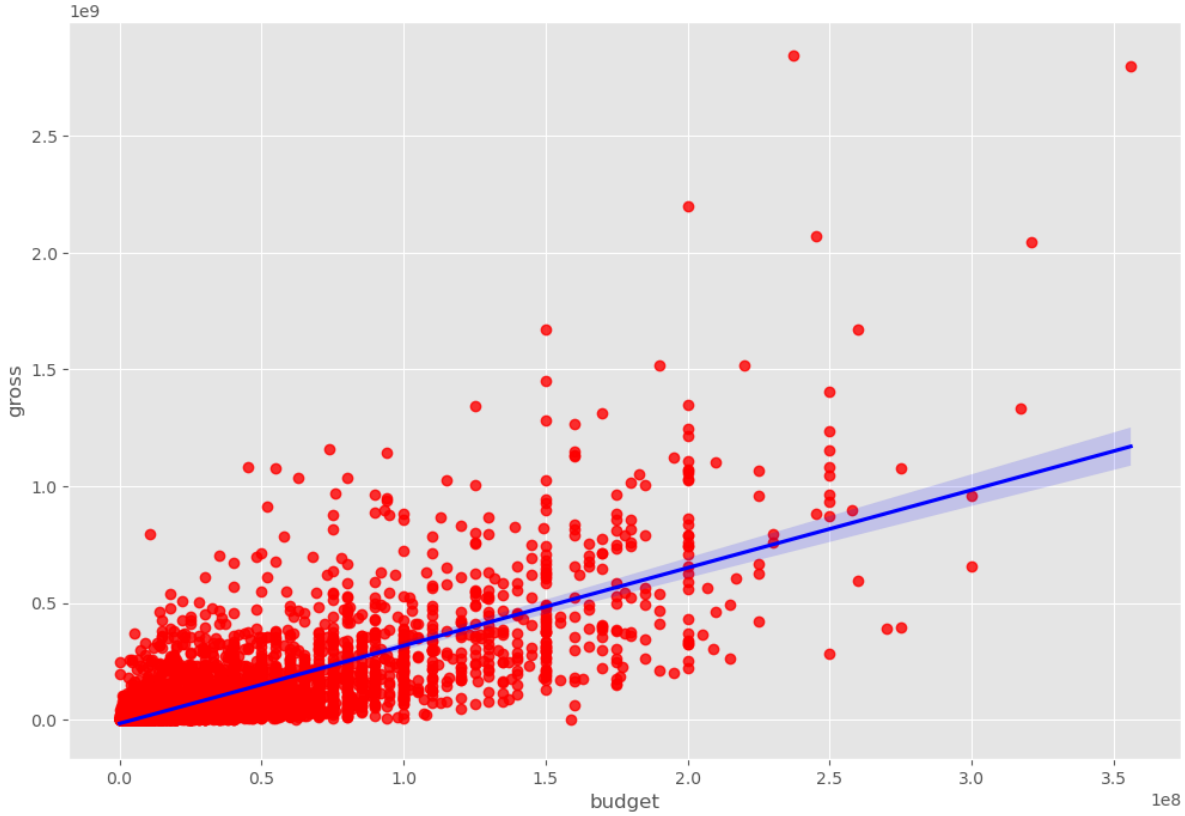
5421 rows × 16 columns

In [11]:
```
# What has the highest correlation with gross (money made from the movie)?

# 1a. Scatter plot of budget vs gross using matplotlib
plt.scatter(x=df['budget'], y=df['gross'])

plt.title("Budget VS Gross Earnings")

plt.xlabel("Gross Earnings")
plt.ylabel("Film Budget")

plt.show()
```



In [12]:
```
# 1b. Plot Budget vs Gross using Seaborn

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

Out[12]:
```
<AxesSubplot:xlabel='budget', ylabel='gross'>
```



In [13]:
```
# Looking at Correlation, Pearsons by default

df.corr()
```

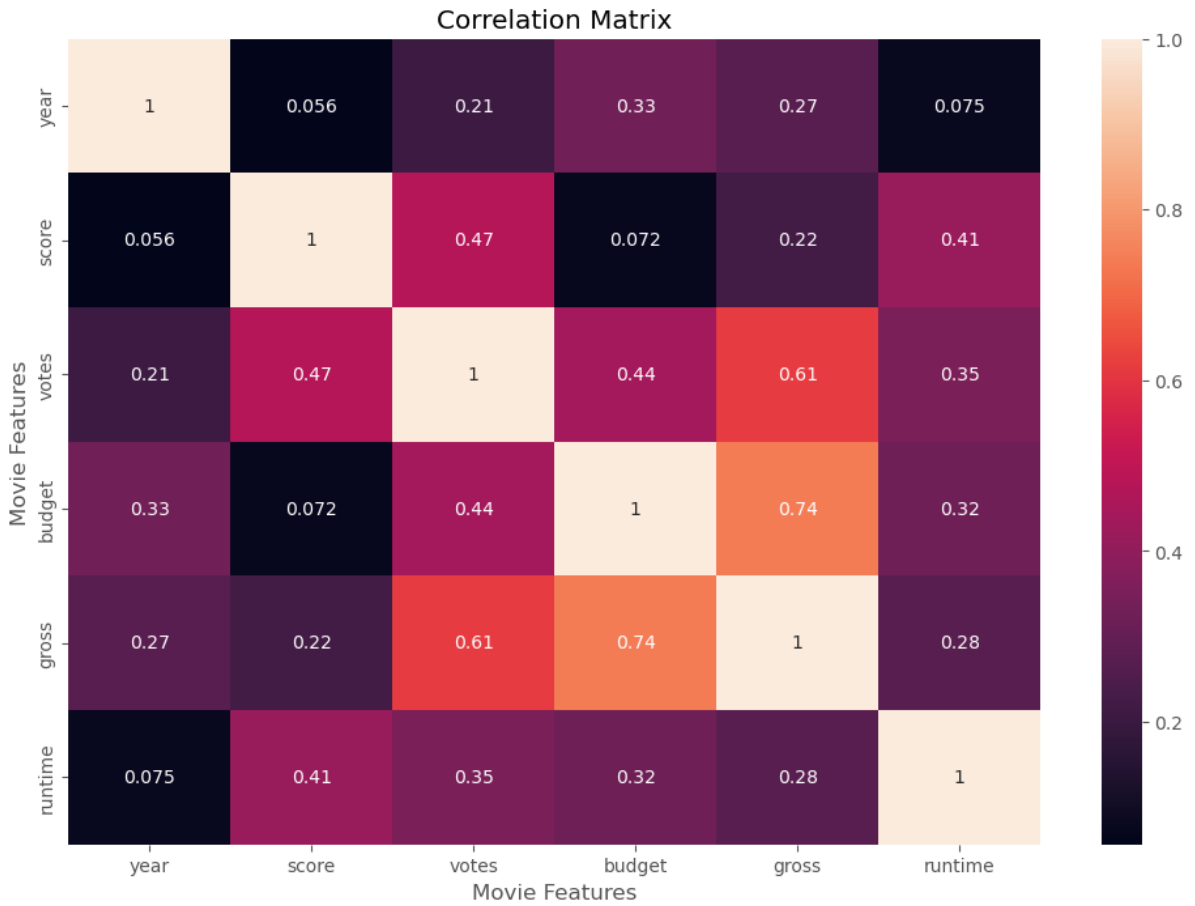
Out[13]:

	year	score	votes	budget	gross	runtime
year	1.000000	0.056386	0.206021	0.327722	0.274321	0.075077
score	0.056386	1.000000	0.474256	0.072001	0.222556	0.414068
votes	0.206021	0.474256	1.000000	0.439675	0.614751	0.352303
budget	0.327722	0.072001	0.439675	1.000000	0.740247	0.318695
gross	0.274321	0.222556	0.614751	0.740247	1.000000	0.275796
runtime	0.075077	0.414068	0.352303	0.318695	0.275796	1.000000

In [14]:
```
# High Correlation between Budget & Gross
```

In [15]:
```
# Heatmap for Correlation
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix,annot=True)

plt.title("Correlation Matrix")

plt.xlabel("Movie Features")
plt.ylabel("Movie Features")
plt.show
```

Out[15]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [16]:
```
# Budget-Gross and Votes-Gross have high correlations
# One possible col to look at would be Company.
# Would movies from certain companies be more likely to have higher gross earnings?
df.head()
```

Out[16]:

	name	rating	genre	year	released	score	votes	director	writer	star
0	The Shining	R	Drama	1980	June 13, 1980 (United States)	8.4	927000.0	Stanley Kubrick	Stephen King	Jack Nicholson
1	The Blue Lagoon	R	Adventure	1980	July 2, 1980 (United States)	5.8	65000.0	Randal Kleiser	Henry De Vere Stacpoole	Brooke Shields
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	Mark Hamil
3	Airplane!	PG	Comedy	1980	July 2, 1980 (United States)	7.7	221000.0	Jim Abrahams	Jim Abrahams	Robert Hays
4	Caddyshack	R	Comedy	1980	July 25, 1980 (United States)	7.3	108000.0	Harold Ramis	Brian Doyle-Murray	Chevy Chase

In [17]:
```
# Since company is not a numerical col, numerizing cols would help make comparisons
df_numerized = df.copy()

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
```

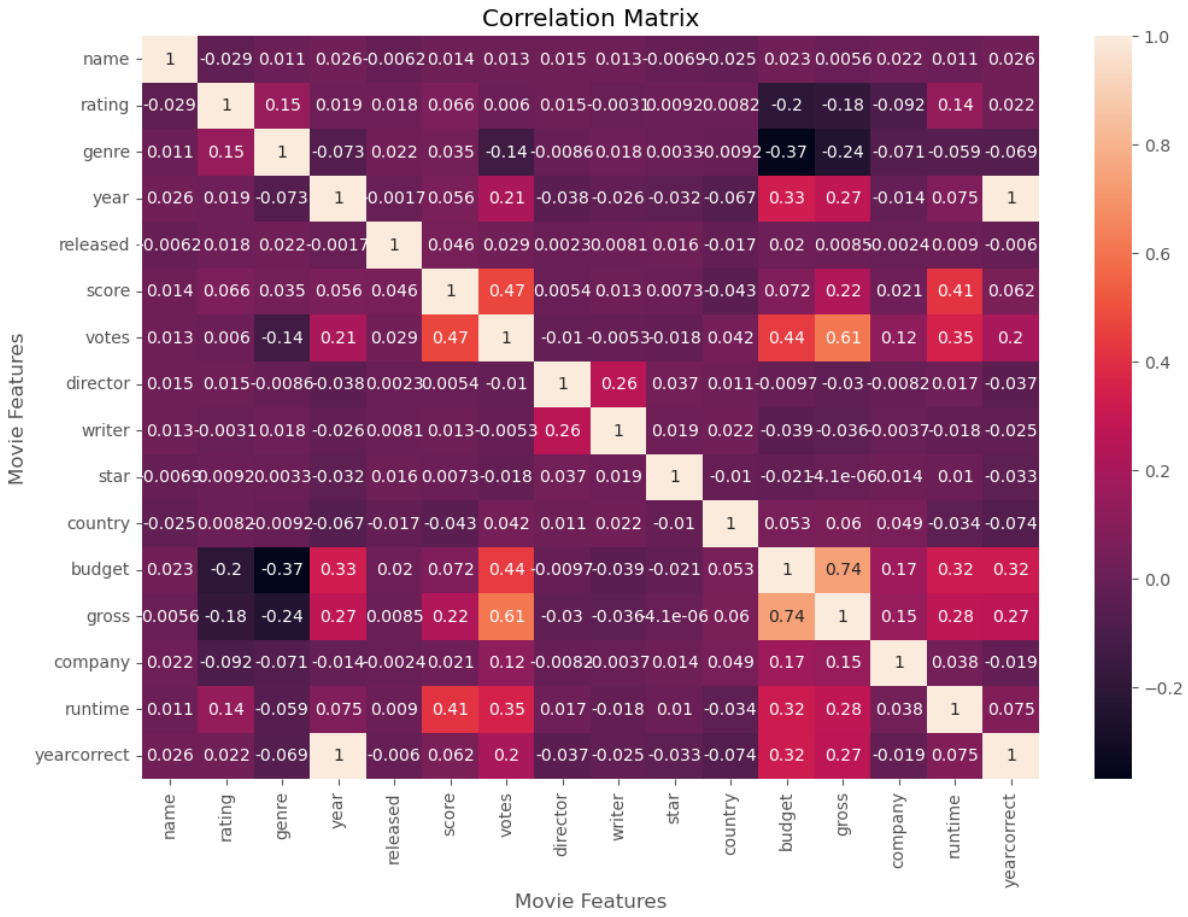
Out[17]:

	name	rating	genre	year	released	score	votes	director	writer	star	country	
0	4692	6	6	1980	1304	8.4	927000.0	1795	2832	699	46	19
1	3929	6	1	1980	1127	5.8	65000.0	1578	1158	214	47	4
2	3641	4	0	1980	1359	8.7	1200000.0	757	1818	1157	47	18
3	204	4	4	1980	1127	7.7	221000.0	889	1413	1474	47	3
4	732	6	4	1980	1170	7.3	108000.0	719	351	271	47	6
...	...	...	...	...	...	...	...	...	...	...	...	
7648	415	6	0	2020	904	6.6	140000.0	16	2390	1812	47	90
7649	3556	4	0	2020	713	6.5	102000.0	852	2309	147	47	85
7650	1153	4	1	2020	904	5.6	53000.0	1809	2827	1470	47	175
7651	3978	4	1	2020	758	6.8	42000.0	294	2091	640	6	135
7652	4090	3	0	2020	370	6.8	3700.0	746	1184	1839	8	80

5421 rows × 16 columns

In [18]:
```
#Numerizing all the cols for bigger correlation matrix
correlation_matrix = df_numerized.corr()
sns.heatmap(correlation_matrix,annot=True)

plt.title("Correlation Matrix")

plt.xlabel("Movie Features")
plt.ylabel("Movie Features")
plt.show
```

Out[18]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [19]:
```
df_numerized.corr()
```

Out[19]:

	name	rating	genre	year	released	score	votes	director
name	1.000000	-0.029234	0.010996	0.025542	-0.006152	0.014450	0.012615	0.015246
rating	-0.029234	1.000000	0.147796	0.019499	0.018083	0.065983	0.006031	0.014656
genre	0.010996	0.147796	1.000000	-0.073167	0.022142	0.035106	-0.135990	-0.008553
year	0.025542	0.019499	-0.073167	1.000000	-0.001740	0.056386	0.206021	-0.038354
released	-0.006152	0.018083	0.022142	-0.001740	1.000000	0.045874	0.028833	0.002308
score	0.014450	0.065983	0.035106	0.056386	0.045874	1.000000	0.474256	0.005413
votes	0.012615	0.006031	-0.135990	0.206021	0.028833	0.474256	1.000000	-0.010376
director	0.015246	0.014656	-0.008553	-0.038354	0.002308	0.005413	-0.010376	1.000000
writer	0.012880	-0.003149	0.017578	-0.025908	0.008072	0.012843	-0.005316	0.261735
star	-0.006882	0.009196	0.003341	-0.032157	0.015706	0.007296	-0.017638	0.036593
country	-0.025490	0.008230	-0.009164	-0.066748	-0.017228	-0.043051	0.041551	0.011133
budget	0.023392	-0.203946	-0.368523	0.327722	0.019952	0.072001	0.439675	-0.009662
gross	0.005639	-0.181906	-0.244101	0.274321	0.008501	0.222556	0.614751	-0.029560
company	0.021697	-0.092357	-0.071334	-0.014333	-0.002407	0.020656	0.118470	-0.008223
runtime	0.010850	0.140792	-0.059237	0.075077	0.008975	0.414068	0.352303	0.017433
yearcorrect	0.025542	0.022021	-0.069147	0.998726	-0.005989	0.061923	0.203098	-0.037371

In [20]:
```
# Correlations for numerized df
correlation_numerized_mat = df_numerized.corr()
corr_pairs = correlation_numerized_mat.unstack()

corr_pairs
```

Out[20]:
```
name         name           1.000000
             rating        -0.029234
             genre          0.010996
             year           0.025542
             released      -0.006152
                             ...
yearcorrect  budget         0.320312
             gross          0.268721
             company       -0.018806
             runtime        0.075294
             yearcorrect    1.000000
Length: 256, dtype: float64
```

In [21]:
```
# Arranging pairs by value
sorted_pairs = corr_pairs.sort_values()

sorted_pairs
```

Out[21]:
```
genre        budget        -0.368523
budget       genre         -0.368523
gross        genre         -0.244101
genre        gross         -0.244101
rating       budget        -0.203946
                              ...
year         year           1.000000
genre        genre          1.000000
rating       rating         1.000000
runtime      runtime        1.000000
yearcorrect  yearcorrect     1.000000
Length: 256, dtype: float64
```

In [22]:
```
# Filtering for pairs with high correlation (say >0.5)
high_corr = sorted_pairs[(sorted_pairs) > 0.5]
high_corr
```

Out[22]:
```
gross        votes          0.614751
votes        gross          0.614751
gross        budget         0.740247
budget       gross          0.740247
year         yearcorrect    0.998726
yearcorrect  year           0.998726
name         name           1.000000
company      company        1.000000
gross        gross          1.000000
budget       budget         1.000000
country      country        1.000000
star         star           1.000000
writer       writer         1.000000
director     director       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
runtime      runtime        1.000000
yearcorrect  yearcorrect     1.000000
dtype: float64
```

In [ ]:
```
# Gross-Votes, Gross-Budgets have high correlations
# Companies ultimately did not have a high correlation with Gross.
```