# Movies - Correlation in Python

## (1) Import Python Libraries

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns

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

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

## (2) Read in Data using Pandas

In [2]:

```
df = pd.read_csv(r'movies.csv')
df.head()
```

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 | |

In [3]:

```
df.dtypes
```

Out[3]:

```
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 [4]:

```
df.shape
```

Out[4]:

```
(7668, 15)
```

## (3) Deal with Missing Data

In [5]:

```
for col in df.columns:
    pct_missing = np.mean(df[col].isnull())
    print('{} - {}%'.format(col, pct_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 [6]:

```
df = df.dropna()
```

In [7]:

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

```
name - 0.0%
rating - 0.0%
genre - 0.0%
year - 0.0%
released - 0.0%
score - 0.0%
votes - 0.0%
director - 0.0%
writer - 0.0%
star - 0.0%
country - 0.0%
budget - 0.0%
gross - 0.0%
company - 0.0%
runtime - 0.0%
```

In [8]:

```
df.shape
```

Out[8]:

```
(5421, 15)
```

## (4) Data Cleaning

In [9]:

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

Out[9]:

| | 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 | |

In [10]:

```
df['yearcorrect'] = df['released'].str.extract(pat='([0-9]{4})').astype(int)
df.head()
```

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 | 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 | |

In [11]:

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

In [12]:

```
pd.set_option('display.max_rows', None)
df.sort_values(by=['gross'], inplace=False, ascending=False)
```

Out[12]:

| | name | rating | genre | year | released | score | votes | director | |
|---|---|---|---|---|---|---|---|---|---|
| **5445** | Avatar | PG-13 | Action | 2009 | December 18, 2009 (United States) | 7.8 | 1100000.0 | James Cameron | C |
| **7445** | Avengers: Endgame | PG-13 | Action | 2019 | April 26, 2019 (United States) | 8.4 | 903000.0 | Anthony Russo | Chri |
| **3045** | Titanic | PG-13 | Drama | 1997 | December 19, 1997 (United States) | 7.8 | 1100000.0 | James Cameron | C |
| **6663** | Star Wars: Episode VII - The Force | PG-13 | Action | 2015 | December 18, 2015 (United | 7.8 | 876000.0 | J.J. Abrams | La |

In [13]:

```
df.shape
```

Out[13]:

```
(5421, 16)
```

In [14]:

```
df.drop_duplicates()
df.shape
```

Out[14]:

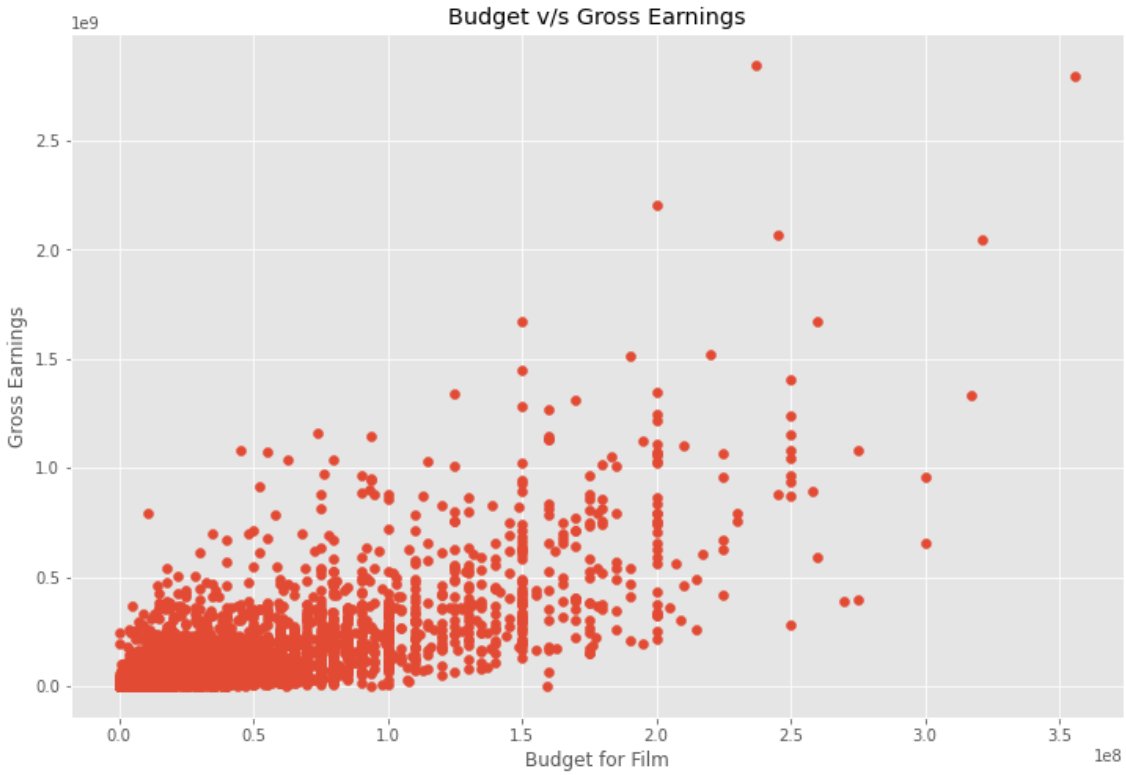
```
(5421, 16)
```

## (5) Finding Correlations in the Data

In [15]:

```
plt.scatter(x=df['budget'], y=df['gross'])
plt.title('Budget v/s Gross Earnings')
plt.ylabel('Gross Earnings')
plt.xlabel('Budget for Film')

plt.show()
```

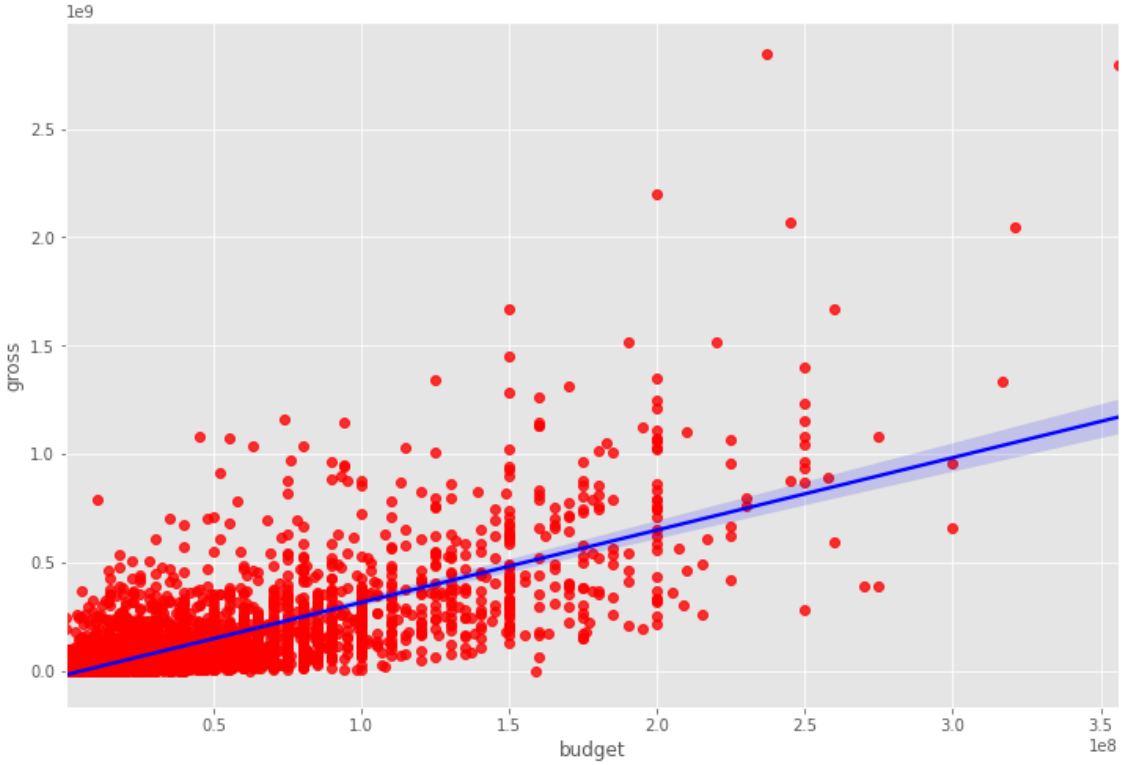


In [16]:

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

Out[16]:

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




In [17]:

```
df.corr()
```

Out[17]:

| | year | score | votes | budget | gross | runtime | yearcorrect |
|---|---|---|---|---|---|---|---|
| **year** | 1.000000 | 0.056386 | 0.206021 | 0.327722 | 0.274321 | 0.075077 | 0.998726 |
| **score** | 0.056386 | 1.000000 | 0.474256 | 0.072001 | 0.222556 | 0.414068 | 0.061923 |
| **votes** | 0.206021 | 0.474256 | 1.000000 | 0.439675 | 0.614751 | 0.352303 | 0.203098 |
| **budget** | 0.327722 | 0.072001 | 0.439675 | 1.000000 | 0.740247 | 0.318695 | 0.320312 |
| **gross** | 0.274321 | 0.222556 | 0.614751 | 0.740247 | 1.000000 | 0.275796 | 0.268721 |
| **runtime** | 0.075077 | 0.414068 | 0.352303 | 0.318695 | 0.275796 | 1.000000 | 0.075294 |
| **yearcorrect** | 0.998726 | 0.061923 | 0.203098 | 0.320312 | 0.268721 | 0.075294 | 1.000000 |

In [18]:

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

Out[18]:

| | year | score | votes | budget | gross | runtime | yearcorrect |
|---|---|---|---|---|---|---|---|
| **year** | 1.000000 | 0.056386 | 0.206021 | 0.327722 | 0.274321 | 0.075077 | 0.998726 |
| **score** | 0.056386 | 1.000000 | 0.474256 | 0.072001 | 0.222556 | 0.414068 | 0.061923 |
| **votes** | 0.206021 | 0.474256 | 1.000000 | 0.439675 | 0.614751 | 0.352303 | 0.203098 |
| **budget** | 0.327722 | 0.072001 | 0.439675 | 1.000000 | 0.740247 | 0.318695 | 0.320312 |
| **gross** | 0.274321 | 0.222556 | 0.614751 | 0.740247 | 1.000000 | 0.275796 | 0.268721 |
| **runtime** | 0.075077 | 0.414068 | 0.352303 | 0.318695 | 0.275796 | 1.000000 | 0.075294 |
| **yearcorrect** | 0.998726 | 0.061923 | 0.203098 | 0.320312 | 0.268721 | 0.075294 | 1.000000 |

In [19]:

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

Out[19]:

| | year | score | votes | budget | gross | runtime | yearcorrect |
|---|---|---|---|---|---|---|---|
| **year** | 1.000000 | 0.039389 | 0.296512 | 0.220833 | 0.239539 | 0.064824 | 0.991304 |
| **score** | 0.039389 | 1.000000 | 0.350185 | -0.006406 | 0.124943 | 0.292254 | 0.043400 |
| **votes** | 0.296512 | 0.350185 | 1.000000 | 0.346274 | 0.553625 | 0.205344 | 0.293044 |
| **budget** | 0.220833 | -0.006406 | 0.346274 | 1.000000 | 0.512057 | 0.231278 | 0.213719 |
| **gross** | 0.239539 | 0.124943 | 0.553625 | 0.512057 | 1.000000 | 0.176979 | 0.232372 |
| **runtime** | 0.064824 | 0.292254 | 0.205344 | 0.231278 | 0.176979 | 1.000000 | 0.064793 |
| **yearcorrect** | 0.991304 | 0.043400 | 0.293044 | 0.213719 | 0.232372 | 0.064793 | 1.000000 |

In [20]:

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

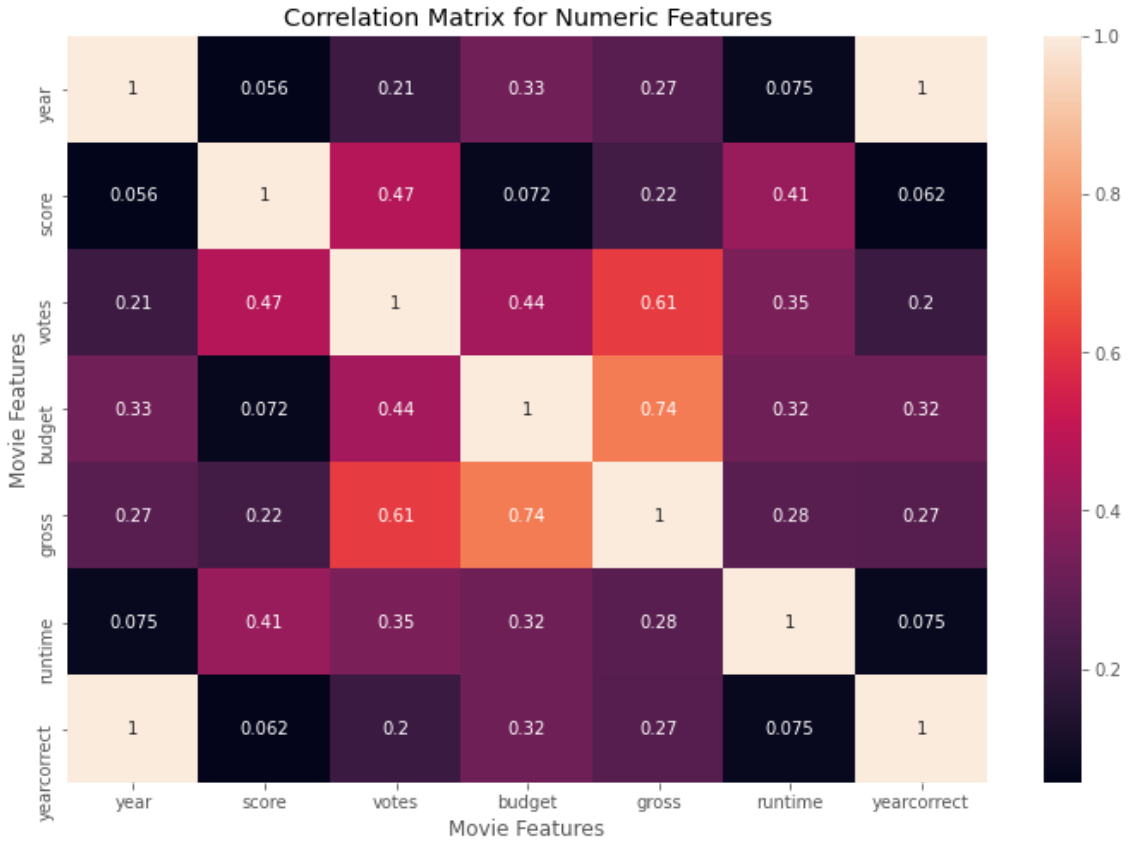
Out[20]:

| | year | score | votes | budget | gross | runtime | yearcorrect |
|---|---|---|---|---|---|---|---|
| **year** | 1.000000 | 0.057741 | 0.427623 | 0.312886 | 0.351045 | 0.095444 | 0.998694 |
| **score** | 0.057741 | 1.000000 | 0.495409 | -0.009971 | 0.183192 | 0.412155 | 0.063674 |
| **votes** | 0.427623 | 0.495409 | 1.000000 | 0.493461 | 0.745793 | 0.300621 | 0.422988 |
| **budget** | 0.312886 | -0.009971 | 0.493461 | 1.000000 | 0.692958 | 0.330794 | 0.302535 |
| **gross** | 0.351045 | 0.183192 | 0.745793 | 0.692958 | 1.000000 | 0.257400 | 0.340529 |
| **runtime** | 0.095444 | 0.412155 | 0.300621 | 0.330794 | 0.257400 | 1.000000 | 0.095507 |
| **yearcorrect** | 0.998694 | 0.063674 | 0.422988 | 0.302535 | 0.340529 | 0.095507 | 1.000000 |

In [21]:

```
correlation_matrix = df.corr(method='pearson')
sns.heatmap(correlation_matrix, annot=True)
plt.title('Correlation Matrix for Numeric Features')
plt.xlabel('Movie Features')
plt.ylabel('Movie Features')

plt.show()
```



In [22]:

```
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.head()
```

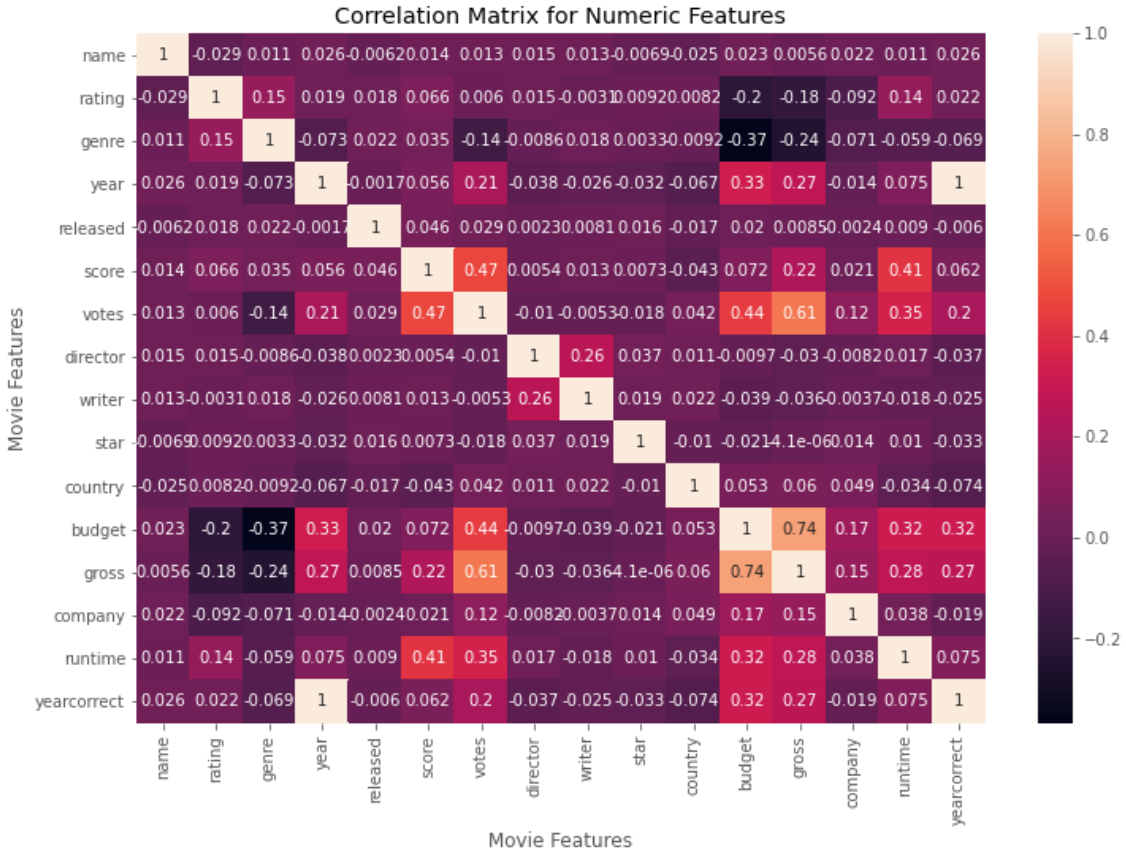
Out[22]:

| | name | rating | genre | year | released | score | votes | director | writer | star | country |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **5445** | 386 | 5 | 0 | 2009 | 527 | 7.8 | 1100000.0 | 785 | 1263 | 1534 | 47 |
| **7445** | 388 | 5 | 0 | 2019 | 137 | 8.4 | 903000.0 | 105 | 513 | 1470 | 47 |
| **3045** | 4909 | 5 | 6 | 1997 | 534 | 7.8 | 1100000.0 | 785 | 1263 | 1073 | 47 |
| **6663** | 3643 | 5 | 0 | 2015 | 529 | 7.8 | 876000.0 | 768 | 1806 | 356 | 47 |
| **7244** | 389 | 5 | 0 | 2018 | 145 | 8.4 | 897000.0 | 105 | 513 | 1470 | 47 |

In [23]:

```
correlation_matrix = df_numerized.corr(method='pearson')
sns.heatmap(correlation_matrix, annot=True)
plt.title('Correlation Matrix for Numeric Features')
plt.xlabel('Movie Features')
plt.ylabel('Movie Features')

plt.show()
```



In [24]:

```
df_numerized.corr()
```

Out[24]:

| | name | rating | genre | year | released | score | votes | dir |
|---|---|---|---|---|---|---|---|---|
| **name** | 1.000000 | -0.029234 | 0.010996 | 0.025542 | -0.006152 | 0.014450 | 0.012615 | 0.01 |
| **rating** | -0.029234 | 1.000000 | 0.147796 | 0.019499 | 0.018083 | 0.065983 | 0.006031 | 0.01 |
| **genre** | 0.010996 | 0.147796 | 1.000000 | -0.073167 | 0.022142 | 0.035106 | -0.135990 | -0.00 |
| **year** | 0.025542 | 0.019499 | -0.073167 | 1.000000 | -0.001740 | 0.056386 | 0.206021 | -0.03 |
| **released** | -0.006152 | 0.018083 | 0.022142 | -0.001740 | 1.000000 | 0.045874 | 0.028833 | 0.00 |
| **score** | 0.014450 | 0.065983 | 0.035106 | 0.056386 | 0.045874 | 1.000000 | 0.474256 | 0.00 |
| **votes** | 0.012615 | 0.006031 | -0.135990 | 0.206021 | 0.028833 | 0.474256 | 1.000000 | -0.01 |
| **director** | 0.015246 | 0.014656 | -0.008553 | -0.038354 | 0.002308 | 0.005413 | -0.010376 | 1.00 |
| **writer** | 0.012880 | -0.003149 | 0.017578 | -0.025908 | 0.008072 | 0.012843 | -0.005316 | 0.26 |
| **star** | -0.006882 | 0.009196 | 0.003341 | -0.032157 | 0.015706 | 0.007296 | -0.017638 | 0.03 |
| **country** | -0.025490 | 0.008230 | -0.009164 | -0.066748 | -0.017228 | -0.043051 | 0.041551 | 0.01 |
| **budget** | 0.023392 | -0.203946 | -0.368523 | 0.327722 | 0.019952 | 0.072001 | 0.439675 | -0.00 |
| **gross** | 0.005639 | -0.181906 | -0.244101 | 0.274321 | 0.008501 | 0.222556 | 0.614751 | -0.02 |
| **company** | 0.021697 | -0.092357 | -0.071334 | -0.014333 | -0.002407 | 0.020656 | 0.118470 | -0.00 |
| **runtime** | 0.010850 | 0.140792 | -0.059237 | 0.075077 | 0.008975 | 0.414068 | 0.352303 | 0.01 |
| **yearcorrect** | 0.025542 | 0.022021 | -0.069147 | 0.998726 | -0.005989 | 0.061923 | 0.203098 | -0.03 |

In [25]:

```
correlation_mat = df_numerized.corr()
corr_pairs = correlation_mat.unstack()

corr_pairs
```

Out[25]:

```
name         name           1.000000
             rating        -0.029234
             genre          0.010996
             year           0.025542
             released      -0.006152
             score          0.014450
             votes          0.012615
             director       0.015246
             writer         0.012880
             star          -0.006882
             country       -0.025490
             budget         0.023392
             gross          0.005639
             company        0.021697
             runtime        0.010850
             yearcorrect    0.025542
rating       name          -0.029234
             rating         1.000000
```

In [26]:

```
sorted_pairs = corr_pairs.sort_values()
sorted_pairs
```

Out[26]:

```
genre        budget        -0.368523
budget       genre         -0.368523
gross        genre         -0.244101
genre        gross         -0.244101
rating       budget        -0.203946
budget       rating        -0.203946
rating       gross         -0.181906
gross        rating        -0.181906
votes        genre         -0.135990
genre        votes         -0.135990
company      rating        -0.092357
rating       company       -0.092357
country      yearcorrect   -0.073569
yearcorrect  country       -0.073569
year         genre         -0.073167
genre        year          -0.073167
             company       -0.071334
company      genre         -0.071334
```

In [27]:

```
high_corr = sorted_pairs[(sorted_pairs) > 0.5]
high_corr
```

Out[27]:

```
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
```