Project: Investigate a Dataset (Tmdb Movies Data set)

Table of Contents

Introduction

The movie database(tmdb) is an records that contins about 10865 movies which contain information such as the id, imdb_id,popularity,budget,revenue,original title, cast, release_date,vote_count,vote_average,release_year,runtime,genres,director, The homepage,tangling,overview,production_companies are some of the additionanal information. some column contains multiple values separated with pipe(|), columns such as genres, cast,production_companies At the last two column it contain the budget_adj,revenue_adj which represent the budget and revenue of the associated movie in US Dollar base on inflation as at 2010

Motives

From the data set we can answer some question such as :

Top Ten Years which Movie are Produced.

Top Genres according to the Revenue Genreated.

Average year which Movies are produced by year.

The relatinship between popularity and vote_count.

The relationship with Budget and Revenue.

Top 10 Directors according the Revenue Generated.

Top 10 genres according to the Revenue generated.

The range of movies released by Month.

Testing correlaton between the numerical Values.

In [15]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [16]:
```
movies = pd.read_csv(r'C:/Users/ADMIN/Desktop/MY DATABASE/tmdb-movies.csv')
movies.head(5)
```

Out[16]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overview	runtime	genres	production_companies	release_date
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.	...	Twenty-two years after the events of Jurassic ...	124	Action|Adventure|Science Fiction|Thriller	Universal Studios|Amblin Entertainment|Legenda...	2015-06-09
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.	...	An apocalyptic story set in the furthest reach...	120	Action|Adventure|Science Fiction|Thriller	Village Roadshow Pictures|Kennedy Miller Produ...	2015-05-13
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You	...	Beatrice Prior must confront her inner demons ...	119	Adventure|Science Fiction|Thriller	Summit Entertainment|Mandeville Films|Red Wago...	2015-03-18
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http://www.starwars.com/films/star-wars-episod...	J.J. Abrams	Every generation has a story.	...	Thirty years after defeating the Galactic Empi...	136	Action|Adventure|Science Fiction|Fantasy	Lucasfilm|Truenorth Productions|Bad Robot	2015-12-15
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	http://www.furious7.com/	James Wan	Vengeance Hits Home	...	Deckard Shaw seeks revenge against Dominic Tor...	137	Action|Crime|Thriller	Universal Pictures|Original Film|Media Rights ...	2015-04-01

5 rows × 21 columns

In [17]:
```
movies.isna().sum()
```

Out[17]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7930
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj                 0
revenue_adj                0
dtype: int64
```

In [18]:
```
movies.shape
```

Out[18]:
```
(10866, 21)
```

In [19]:
```
movies.drop(columns = {'imdb_id','original_title','cast', 'homepage','tagline','keywords','overview','budget_adj','revenue_adj'}, inplace = True)
```

In [20]:
```
movies.isna().sum()
```

Out[20]:
```
id                         0
popularity                 0
budget                     0
revenue                    0
director                  44
runtime                    0
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
dtype: int64
```

In [21]:
```
movies.nunique()
```

Out[21]:
```
id                      10865
popularity              10814
budget                    557
revenue                  4702
director                 5067
runtime                   247
genres                   2039
production_companies     7445
release_date             5909
vote_count               1289
vote_average               72
release_year               56
dtype: int64
```

In [22]:
```
movies.columns
```

Out[22]:
```
Index(['id', 'popularity', 'budget', 'revenue', 'director', 'runtime',
       'genres', 'production_companies', 'release_date', 'vote_count',
       'vote_average', 'release_year'],
      dtype='object')
```

In [23]:
```
movies.describe()
```

Out[23]:

	id	popularity	budget	revenue	runtime	vote_count	vote_average	release_year
count	10866.000000	10866.000000	1.086600e+04	1.086600e+04	10866.000000	10866.000000	10866.000000	10866.000000
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	5.974922	2001.322658
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	0.935142	12.812941
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	1.500000	1960.000000
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	5.400000	1995.000000
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	6.000000	2006.000000
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	6.600000	2011.000000
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	9.200000	2015.000000

In [24]:
```
movies[movies['revenue']==0].head(5)
```

Out[24]:

	id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year
48	265208	2.932340	30000000	0	Simon West	92	Thriller|Crime|Drama	Current Entertainment|Lionsgate|Sierra / Affin...	2015-01-14	481	5.3	2015
67	334074	2.331636	20000000	0	James McTeigue	96	Crime|Thriller|Action	Nu Image Films|Winkler Films|Millennium Films|...	2015-05-21	280	5.4	2015
74	347096	2.165433	0	0	Anne K. Black	108	Action|Adventure|Fantasy	Arrowstorm Entertainment	2015-06-24	27	5.1	2015
75	308369	2.141506	0	0	Alfonso Gomez-Rejon	105	Comedy|Drama	Indian Paintbrush	2015-06-12	569	7.7	2015
92	370687	1.876037	0	0	A. Todd Smith	0	Fantasy|Action|Adventure	Arrowstorm Entertainment|Camera 40 Productions...	2015-12-19	11	5.4	2015

In [25]:
```
movies[movies['budget'] == 0].head(5)
```

Out[25]:

	id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year
30	280996	3.927333	0	29355203	Bill Condon	103	Mystery|Drama	BBC Films|See-Saw Films|FilmNation Entertainme...	2015-06-19	425	6.4	2015
36	339527	3.358321	0	22354572	Afonso Poyart	101	Crime|Drama|Mystery	Eden Rock Media|FilmNation Entertainment|Flynn...	2015-09-03	474	6.2	2015
72	284289	2.272044	0	45895	Jean-Baptiste LÃ©onetti	95	Thriller	Furthur Films	2015-04-17	81	5.5	2015
74	347096	2.165433	0	0	Anne K. Black	108	Action|Adventure|Fantasy	Arrowstorm Entertainment	2015-06-24	27	5.1	2015
75	308369	2.141506	0	0	Alfonso Gomez-Rejon	105	Comedy|Drama	Indian Paintbrush	2015-06-12	569	7.7	2015

In [26]:
```
movies[movies['budget'] == 0].head(5)
```

Out[26]:

	id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year
30	280996	3.927333	0	29355203	Bill Condon	103	Mystery|Drama	BBC Films|See-Saw Films|FilmNation Entertainme...	2015-06-19	425	6.4	2015
36	339527	3.358321	0	22354572	Afonso Poyart	101	Crime|Drama|Mystery	Eden Rock Media|FilmNation Entertainment|Flynn...	2015-09-03	474	6.2	2015
72	284289	2.272044	0	45895	Jean-Baptiste LÃ©onetti	95	Thriller	Furthur Films	2015-04-17	81	5.5	2015
74	347096	2.165433	0	0	Anne K. Black	108	Action|Adventure|Fantasy	Arrowstorm Entertainment	2015-06-24	27	5.1	2015
75	308369	2.141506	0	0	Alfonso Gomez-Rejon	105	Comedy|Drama	Indian Paintbrush	2015-06-12	569	7.7	2015

Errors Detected in the Dataset

The data contains missing values

Some of the columns are not needed for analysis

The data set contain one duplicate row

Two or more genre in genre column separated with pipe(|)

Zero value at runtime, budget,revenue columns this could propably be a data entry errors

Data Cleaning

In [27]:
```
movies.columns
```

Out[27]:
```
Index(['id', 'popularity', 'budget', 'revenue', 'director', 'runtime',
       'genres', 'production_companies', 'release_date', 'vote_count',
       'vote_average', 'release_year'],
      dtype='object')
```

In [28]:
```
movie_copy = movies.copy()
```

Replacing Zeros in Runtime,Revenue,Budget with the Mean Values Respectively

In [29]:
```
def rep_zero(col_name:list, data: pd.DataFrame):
    for name in col_name:
        for i in range(data.shape[0]):
            if(data[name].iloc[i]==0):
                data[name].iloc[i]= data[name].mean()
rep_zero(['revenue','budget','runtime'], movie_copy)
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_6132\1811297964.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  data[name].iloc[i]= data[name].mean()
```

Dropping the Duplicated Rows

In [30]:
```
movies.drop_duplicates(inplace = True)
```

Dropping the Rows with null values

In [31]:
```
movies.dropna(inplace =True)
```

Remove Pipe the Genres Columns

In [32]:
```
df_genres = movies.copy()['genres']
movies['genres'] = df_genres.apply(lambda x : x.split('|')[0])
```

In [33]:
```
movies['genres'].value_counts()
```

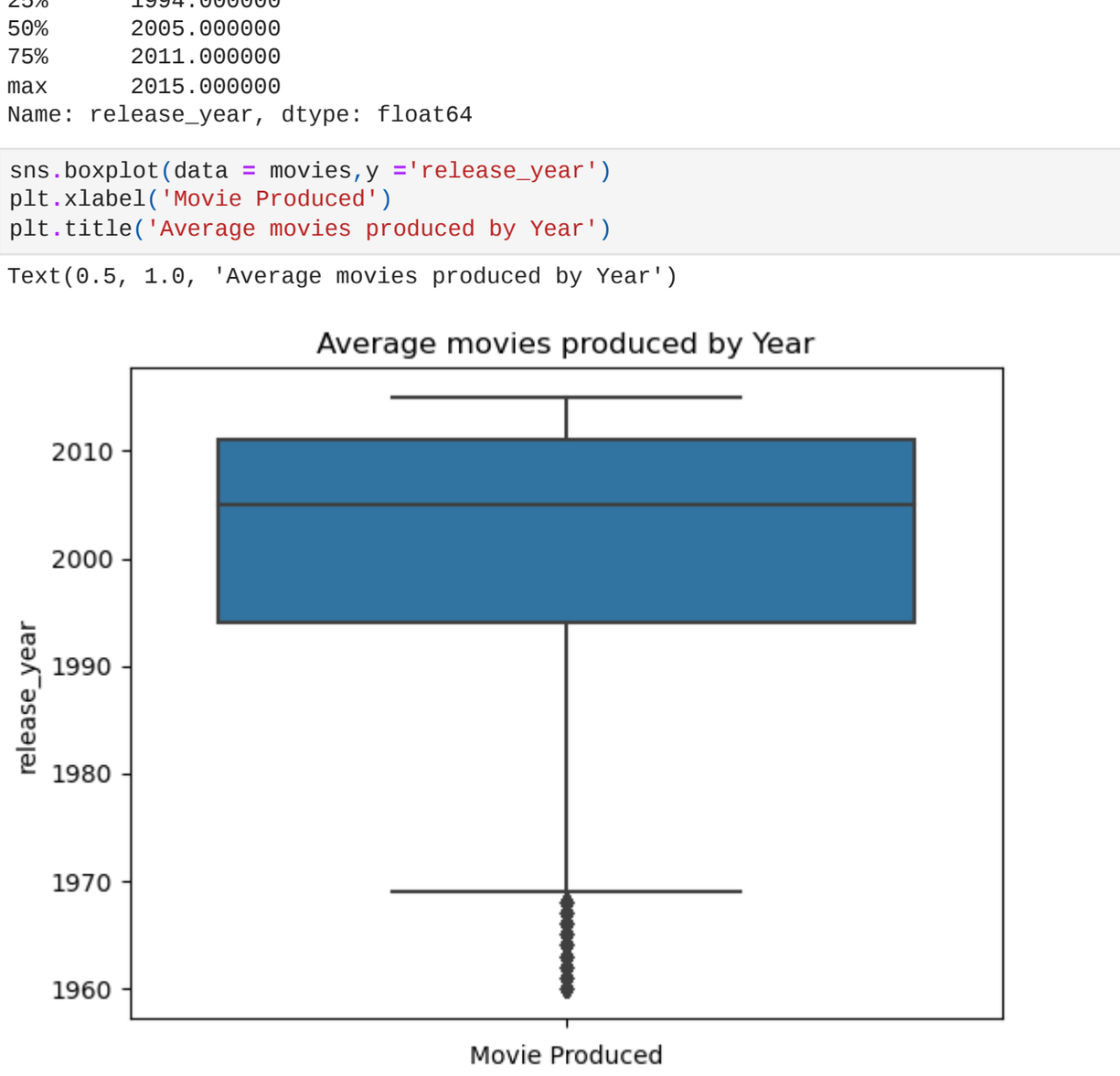
Out[33]:
```
Drama              2225
Comedy             2084
Action             1479
Horror              853
Adventure           551
Thriller            474
Animation           362
Crime               361
Documentary         278
Fantasy             254
Science Fiction     196
Romance             161
Family              128
Mystery             118
Music                79
TV Movie             59
War                  57
Western              42
History              39
Foreign               6
Name: genres, dtype: int64
```

In [34]:
```
TopMoviesGenres = movies['genres'].value_counts()[:10]
TopMoviesGenres_order=TopMoviesGenres.index
TopMoviesGenres.index
```

Out[34]:
```
Index(['Drama', 'Comedy', 'Action', 'Horror', 'Adventure', 'Thriller',
       'Animation', 'Crime', 'Documentary', 'Fantasy'],
      dtype='object')
```

In [35]:
```
base_color = sns.color_palette()[0]
sns.countplot(data = movies,y= 'genres',color=base_color, order=TopMoviesGenres_order)
```

Out[35]:
```
<Axes: xlabel='count', ylabel='genres'>
```



The Drama is the highest Genres Release within the Years in the Dataset

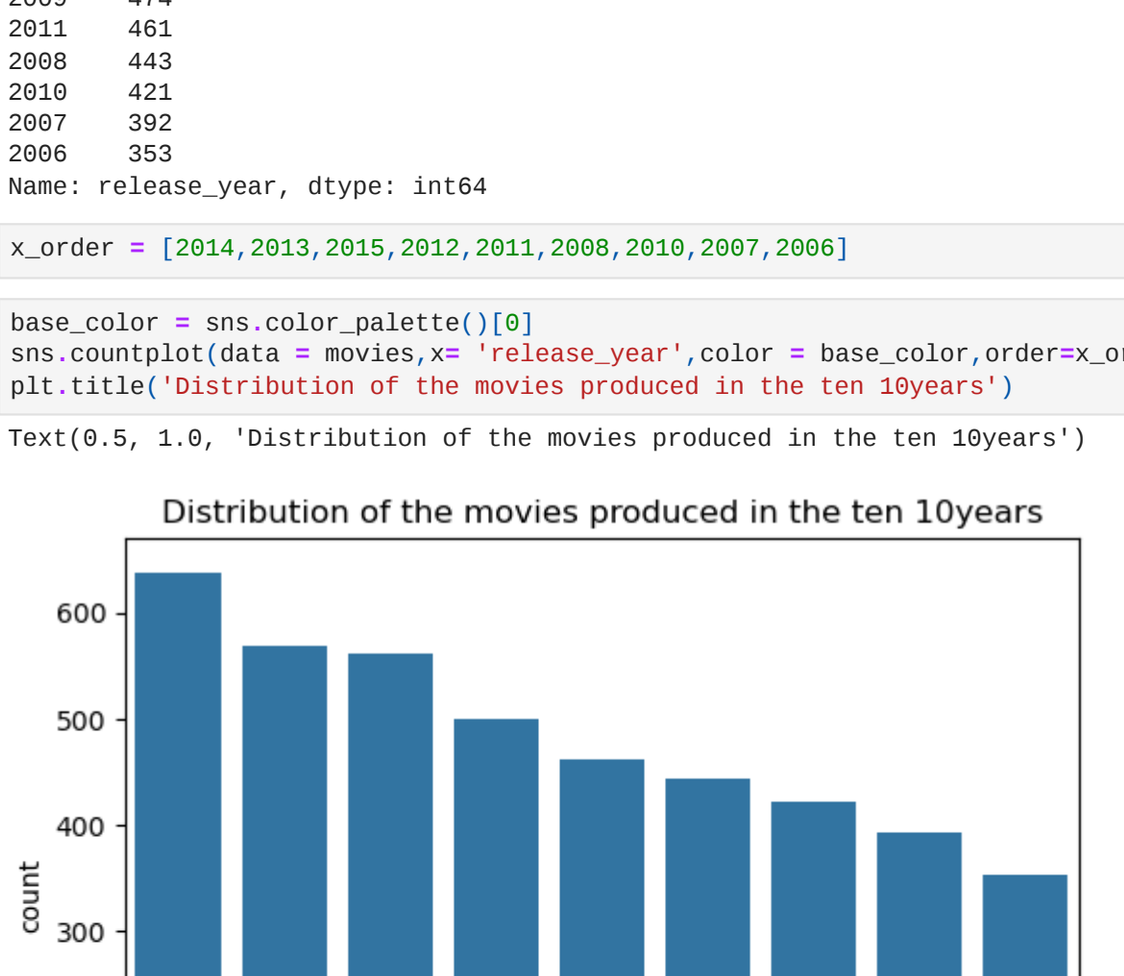
Average movies produced per Year in the dataset

In [36]:
```
movies['release_year'].describe()
```

Out[36]:
```
count    9806.000000
mean     2000.890271
std        13.032372
min      1960.000000
25%      1994.000000
50%      2005.000000
75%      2011.000000
max      2015.000000
Name: release_year, dtype: float64
```

In [37]:
```
sns.boxplot(data = movies,y ='release_year')
plt.xlabel('Movie Produced')
plt.title('Average movies produced by Year')
```

Out[37]:
```
Text(0.5, 1.0, 'Average movies produced by Year')
```



The average year of movies produced is around 2008, and maximum year of movies produce is around 2015

Top 10 Movies which Movies are produced By Year

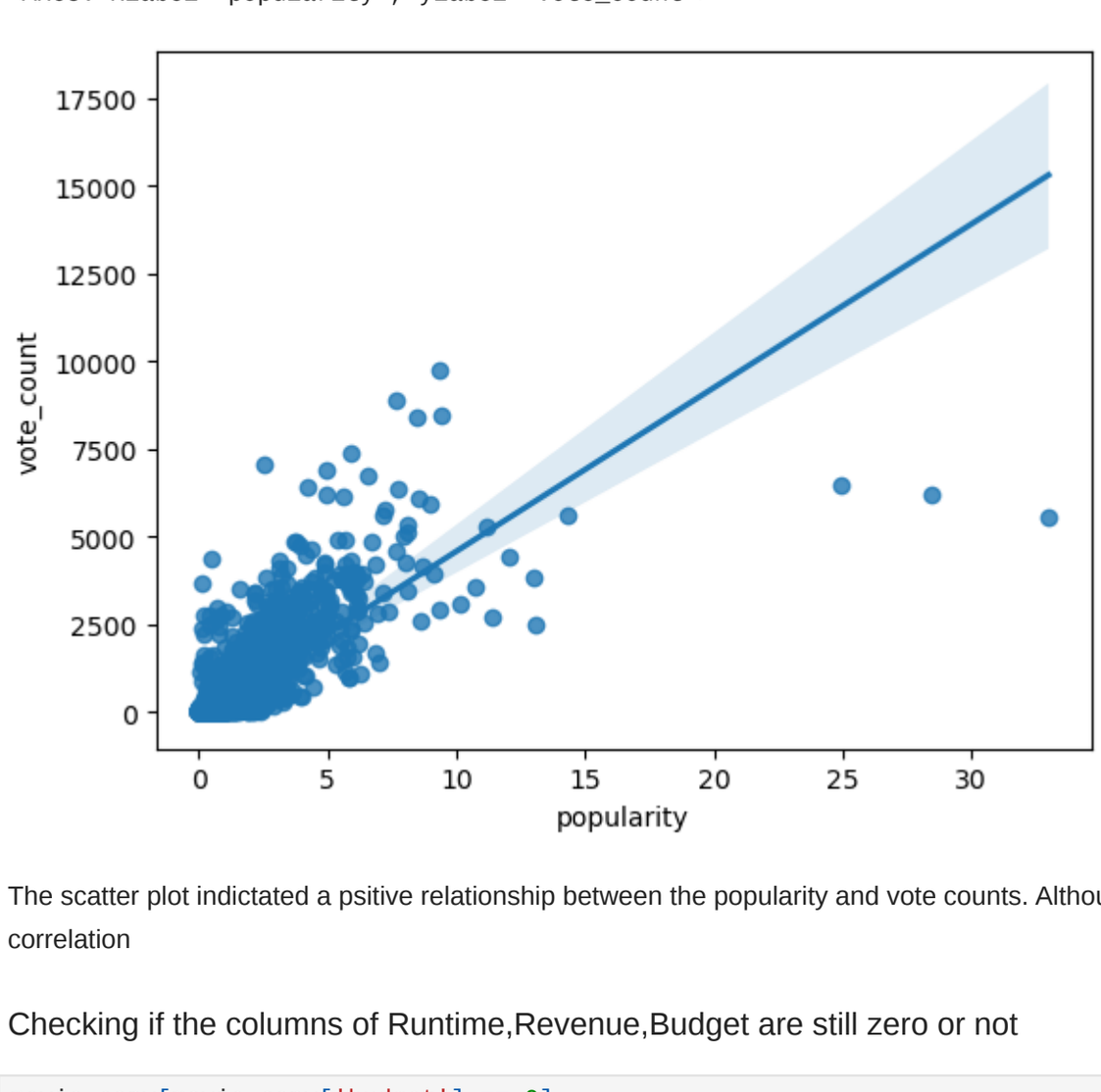
In [38]:
```
movies['release_year'].value_counts()[:10]
```

Out[38]:
```
2014    638
2013    568
2015    561
2012    500
2009    474
2011    461
2008    443
2010    421
2007    392
2006    353
Name: release_year, dtype: int64
```

In [39]:
```
x_order = [2014,2013,2015,2012,2011,2008,2010,2007,2006]
```

In [40]:
```
base_color = sns.color_palette()[0]
sns.countplot(data = movies,x= 'release_year',color = base_color,order=x_order)
plt.title('Distribution of the movies produced in the ten 10years')
```

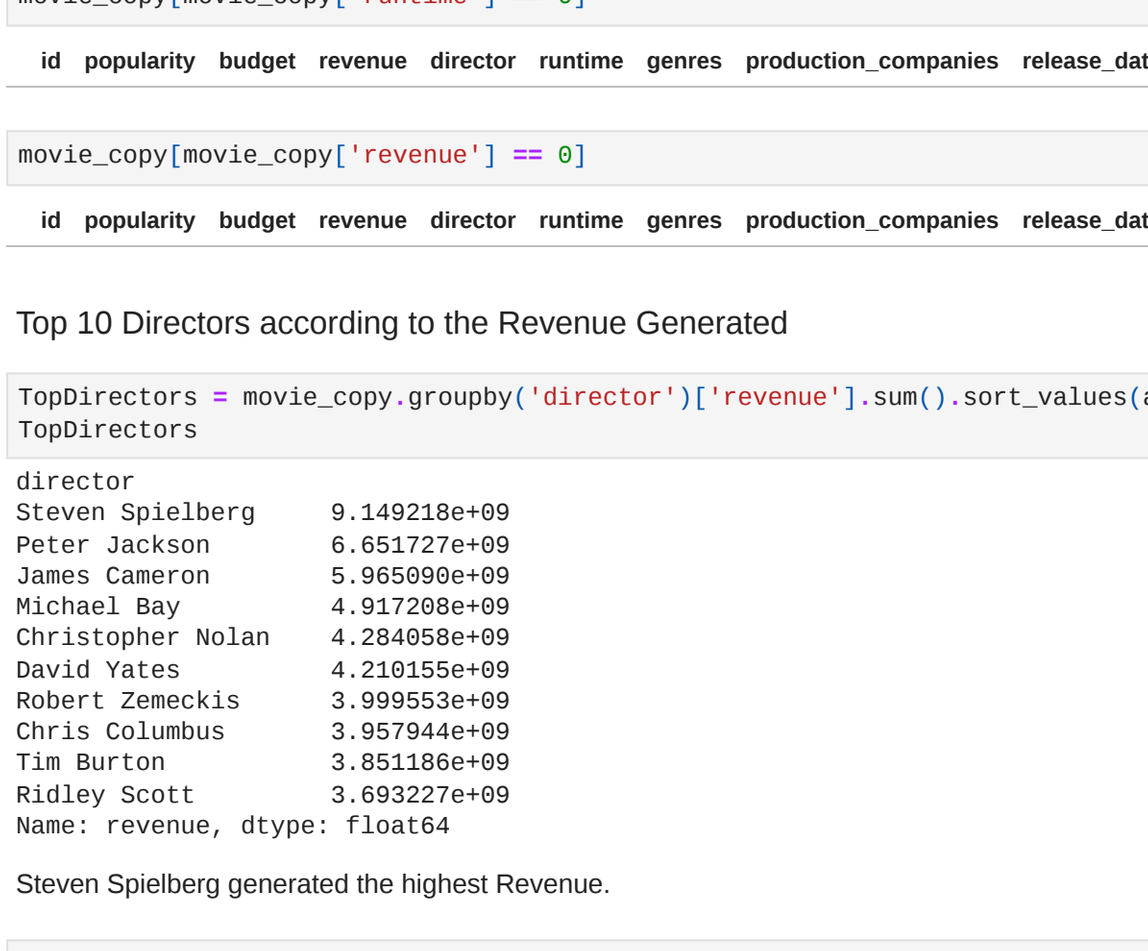
Out[40]:
```
Text(0.5, 1.0, 'Distribution of the movies produced in the ten 10years')
```



The Relationship bwtween popularity and vote_counts

In [41]:
```
sns.regplot(data = movies, x ='popularity', y = 'vote_count')
```

Out[41]:
```
<Axes: xlabel='popularity', ylabel='vote_count'>
```



The scatter plot indictated a psitive relationship between the popularity and vote counts. Although there is high frequency value in movie popularity between 0-5 and 0-2500 vote counts but generally the two variable exhibit a positive correlation

Checking if the columns of Runtime,Revenue,Budget are still zero or not

In [42]:
```
movie_copy[movie_copy['budget'] == 0]
```

Out[42]:

id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year

In [43]:
```
movie_copy[movie_copy['runtime'] == 0]
```

Out[43]:

id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year

In [44]:
```
movie_copy[movie_copy['revenue'] == 0]
```

Out[44]:

id	popularity	budget	revenue	director	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year

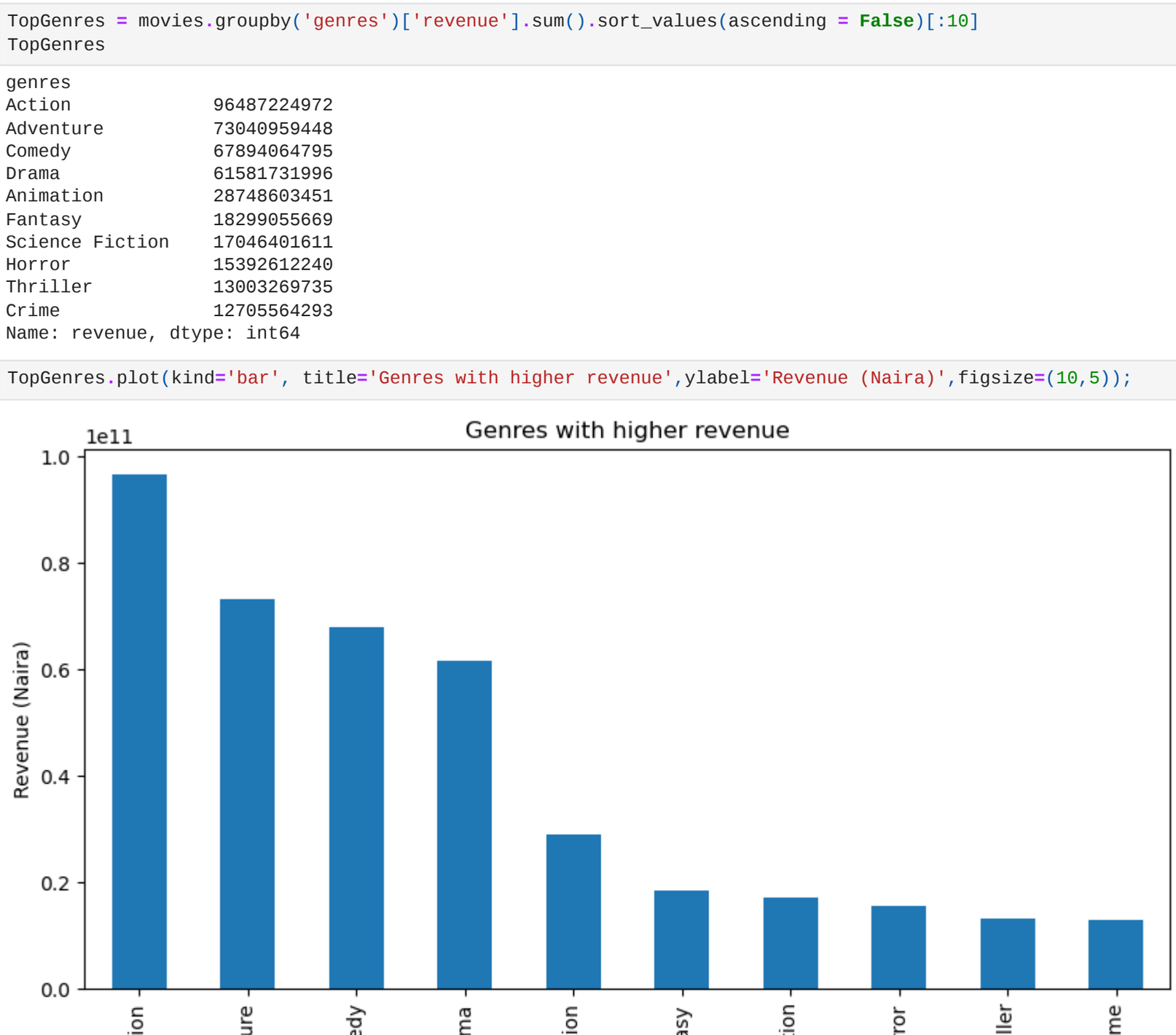
Top 10 Directors according to the Revenue Generated

In [45]:
```
TopDirectors = movie_copy.groupby('director')['revenue'].sum().sort_values(ascending = False)[:10]
TopDirectors
```

Out[45]:
```
director
Steven Spielberg     9.149218e+09
Peter Jackson        6.651727e+09
James Cameron        5.965090e+09
Michael Bay          4.917208e+09
Christopher Nolan    4.284058e+09
David Yates          4.210155e+09
Robert Zemeckis      3.999553e+09
Chris Columbus       3.957944e+09
Tim Burton           3.851186e+09
Ridley Scott         3.693227e+09
Name: revenue, dtype: float64
```

Steven Spielberg generated the highest Revenue.

In [46]:
```
TopDirectors.plot(kind='bar', title='Directors with higher revenue',ylabel='Revenue (Naira)',figsize=(10,5));
```

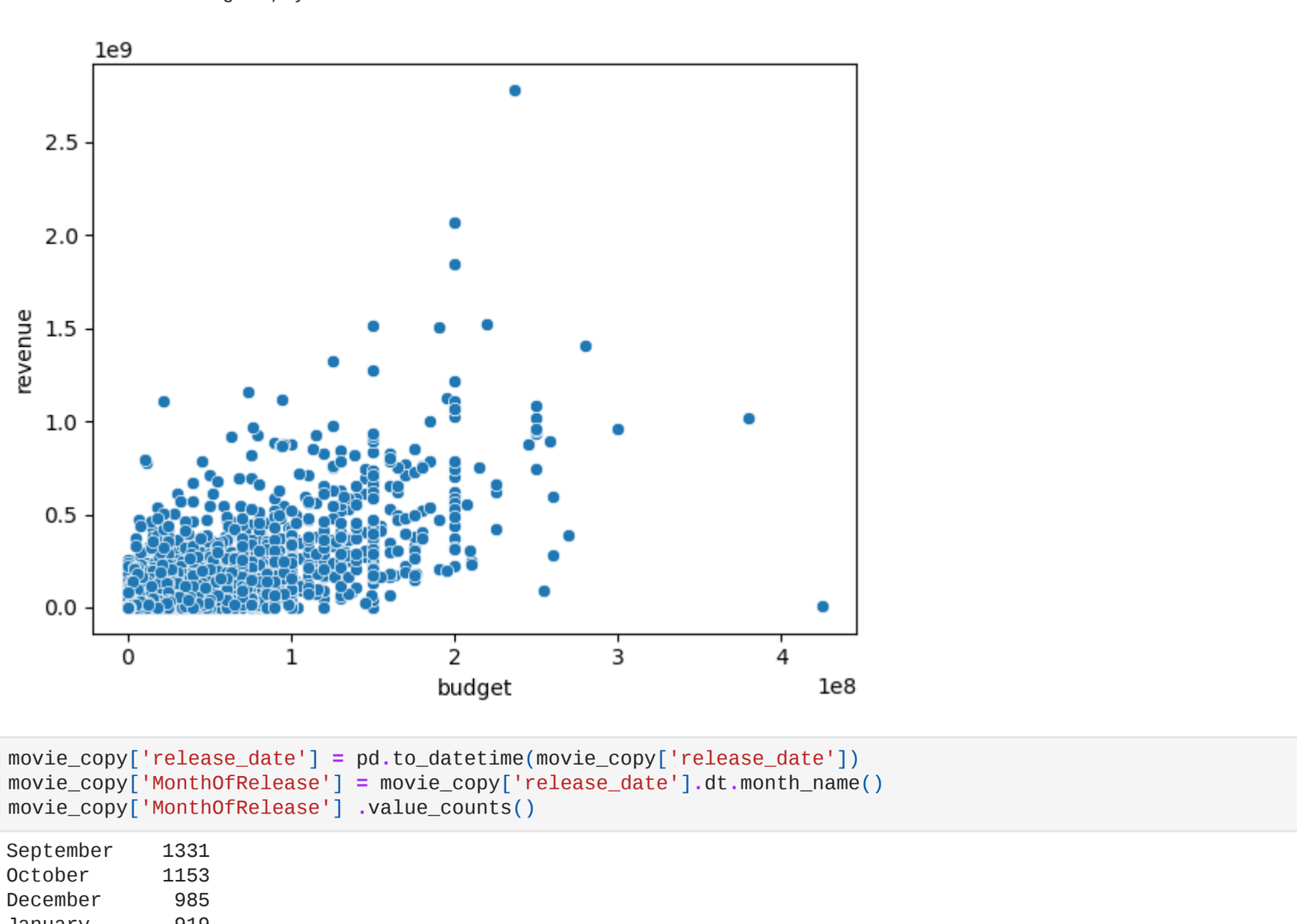


Top 10 Genres according to the Revenue Generated

In [56]:
```
TopGenres = movies.groupby('genres')['revenue'].sum().sort_values(ascending = False)[:10]
TopGenres
```

Out[56]:
```
genres
Action             96487224972
Adventure          73040959448
Comedy             67894064795
Drama              61581731996
Animation          28748603451
Fantasy            18299055669
Science Fiction    17046401611
Horror             15392612240
Thriller           13003269735
Crime              12705564293
Name: revenue, dtype: int64
```

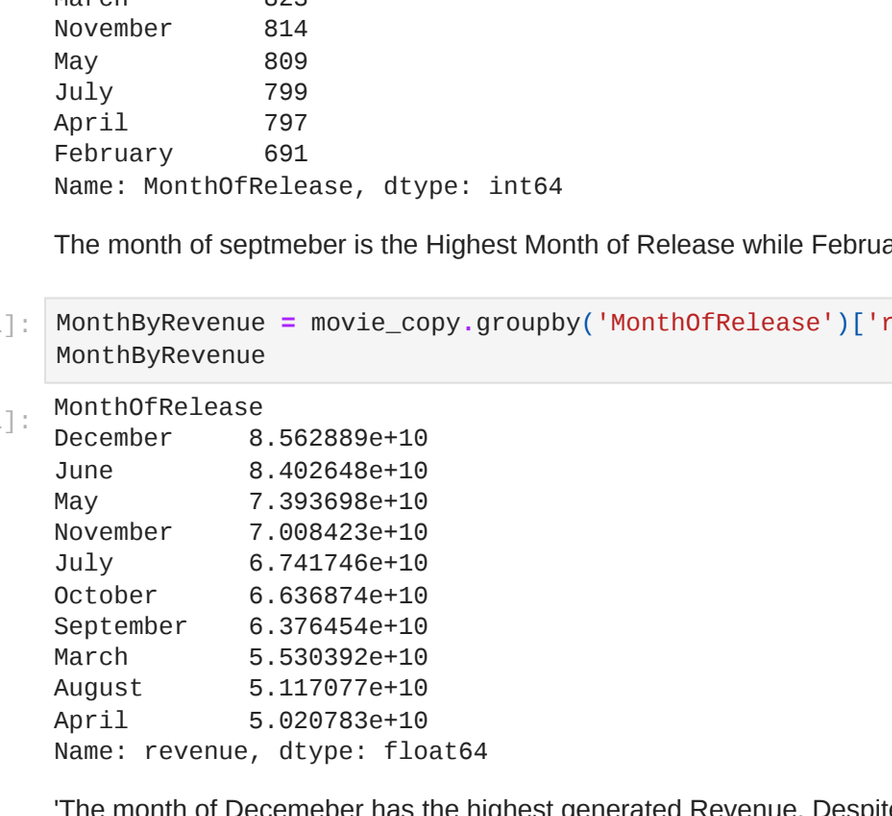
In [57]:
```
TopGenres.plot(kind='bar', title='Genres with higher revenue',ylabel='Revenue (Naira)',figsize=(10,5));
```



Action genre generate the highest revenue among the genres.

In [49]:
```
sns.scatterplot(data = movies, x ='budget', y = 'revenue')
```

Out[49]:
```
<Axes: xlabel='budget', ylabel='revenue'>
```



In [50]:
```
movie_copy['release_date'] = pd.to_datetime(movie_copy['release_date'])
movie_copy['MonthOfRelease'] = movie_copy['release_date'].dt.month_name()
movie_copy['MonthOfRelease'] .value_counts()
```

Out[50]:
```
September    1331
October      1153
December      985
January       919
August        918
June          827
March         823
November      814
May           809
July          799
April         797
February      691
Name: MonthOfRelease, dtype: int64
```

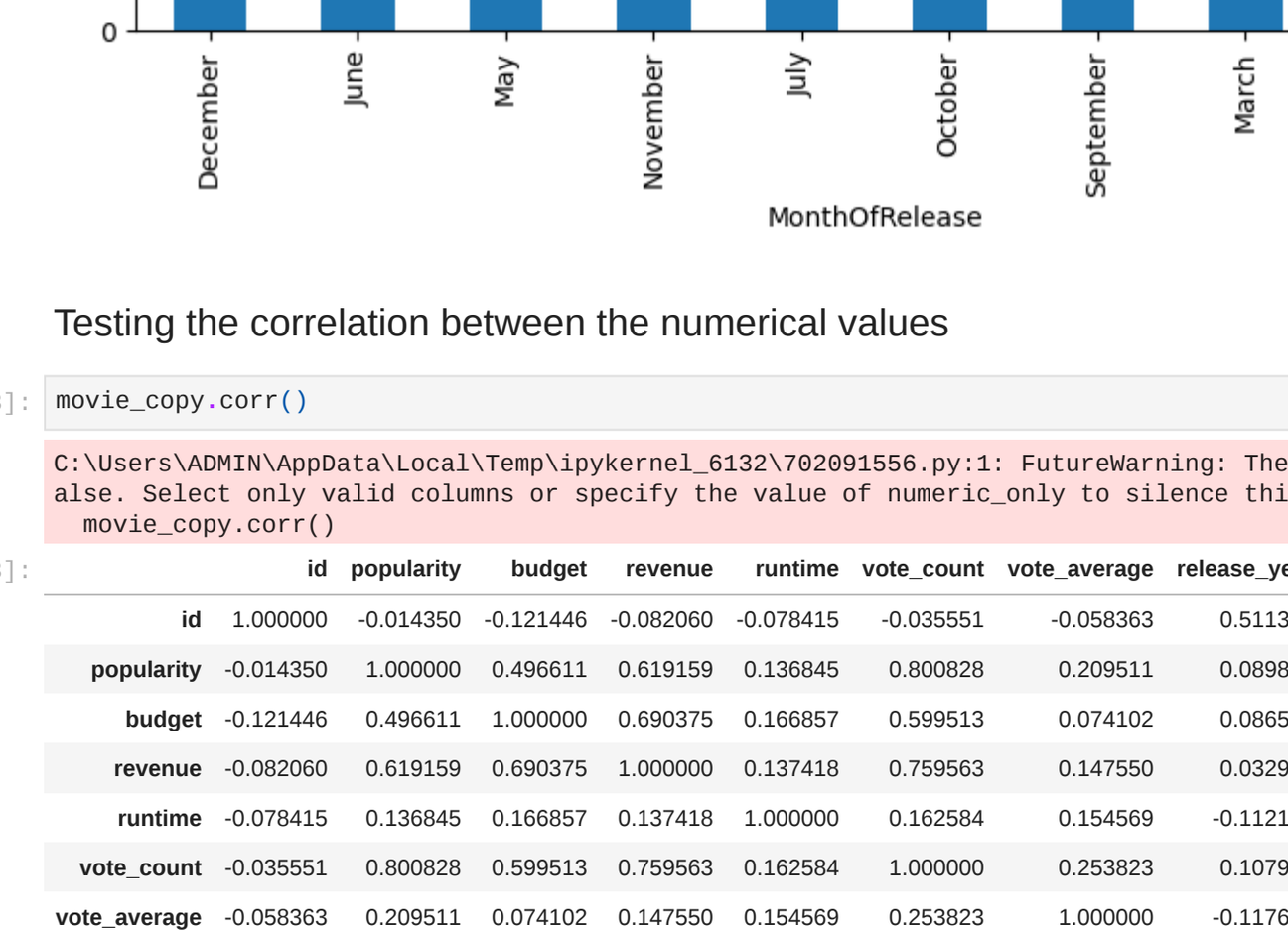
The month of septmeber is the Highest Month of Release while February has the lowest amount of movie released

In [51]:
```
MonthByRevenue = movie_copy.groupby('MonthOfRelease')['revenue'].sum().sort_values(ascending = False)[:10]
MonthByRevenue
```

Out[51]:
```
MonthOfRelease
December     8.562889e+10
June         8.402648e+10
May          7.393698e+10
November     7.008423e+10
July         6.741746e+10
October      6.636874e+10
September    6.376454e+10
March        5.530392e+10
August       5.117077e+10
April        5.020783e+10
Name: revenue, dtype: float64
```

'The month of Decemeber has the highest generated Revenue, Despite Septemeer being the Highest month of Release

In [52]:
```
MonthByRevenue.plot(kind='bar', title='Month Movie Released with higher revenue',ylabel='Revenue (Naira)',figsize=(10,5));
```



Testing the correlation between the numerical values

In [53]:
```
movie_copy.corr()
```

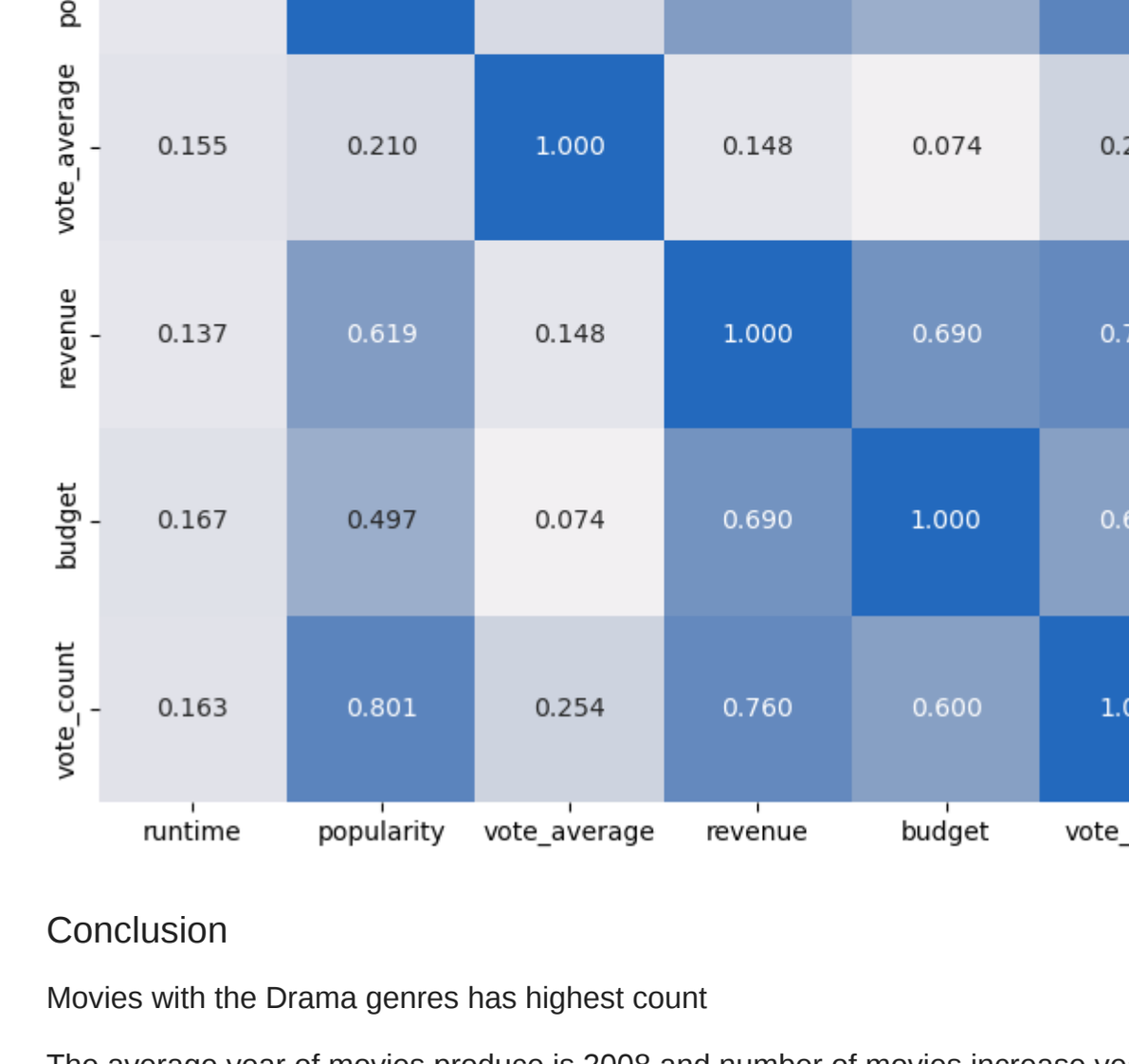
```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_6132\702091556.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to F
alse. Select only valid columns or specify the value of numeric_only to silence this warning.
  movie_copy.corr()
```

Out[53]:

	id	popularity	budget	revenue	runtime	vote_count	vote_average	release_year
id	1.000000	-0.014350	-0.121446	-0.082060	-0.078415	-0.035551	-0.058363	0.511364
popularity	-0.014350	1.000000	0.496611	0.619159	0.136845	0.800828	0.209511	0.089801
budget	-0.121446	0.496611	1.000000	0.690375	0.166857	0.599513	0.074102	0.086502
revenue	-0.082060	0.619159	0.690375	1.000000	0.137418	0.759563	0.147550	0.032937
runtime	-0.078415	0.136845	0.166857	0.137418	1.000000	0.162584	0.154569	-0.112178
vote_count	-0.035551	0.800828	0.599513	0.759563	0.162584	1.000000	0.253823	0.107948
vote_average	-0.058363	0.209511	0.074102	0.147550	0.154569	0.253823	1.000000	-0.117632
release_year	0.511364	0.089801	0.086502	0.032937	-0.112178	0.107948	-0.117632	1.000000

In [54]:
```
num_var=['runtime','popularity','vote_average','revenue','budget','vote_count']
```

In [55]:
```
plt.figure(figsize=(10,8))
sns.heatmap(movie_copy[num_var].corr(), annot=True, fmt='.3f', cmap='vlag_r',center=0)
plt.title('correlation heatmap of the numerical variables');
```



Conclusion

Movies with the Drama genres has highest count

The average year of movies produce is 2008 and number of movies increase year from 1960-2015, although 2014 tend to have high number of movies produce

The correlation between popularity and vote count is positive

All the numerical variable between the data set are positively correlleted ie an increase in one affect another

Diretor with the name Steven Spielberg yield highest revenue to company

Action movies generate highest revenue

LIMITATION

The data has a lot of columns that are not useful and nedded to be remove for better analysis

The data has so many null and zero values

The data set has a great number of unique column

some columns values are separated with pipe e.g genres which later splited by explode method and this increase the number of rows available