Project: Investigate TMDb Movies Dataset

Table of Contents

1.Introduction

1.1 Dataset Description

I choose the TMDb movie dataset which include user ratings and revenue and budget of movies and etc ...

1.2 Question(s) for Analysis

1.Which year has the highest release of movies?

2.Which length movies most liked by the audiences according to their popularity?

3.What is the relationship between runtime and vote average?

4.Which Movie Has The Highest Or Lowest Profit?

5.What month is considered "best" for releasing a film/show?

6.Which Genre Has The Highest Release Of Movies?

7.What Kind Of Properties Are Associated With Movies With High Revenue?

In [42]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

In [43]:
```
df = pd.read_csv('tmdb-movies.csv')
```

2.Data Wrangling

2.1General Properties

In [44]:
```
df.head(1)
```

Out[44]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tag
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	pa o

1 rows × 21 columns

In [45]:
```
df.tail(1)
```

Out[45]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...	overv
10865	22293	tt0060666	0.035919	19000	0	Manos: The Hands of Fate	Harold P. Warren|Tom Neyman|John Reynolds|Dian...	NaN	Harold P. Warren	It's Shocking! It's Beyond Your Imagination!	...	A fa gets or road stum

1 rows × 21 columns

In [46]:
```
print('The data contain (rows,columns) :',df.shape)
```

```
The data contain (rows,columns) : (10866, 21)
```

In [47]:
```
df.describe()
```

Out[47]:

	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	1.0
mean	66064.177434	0.646441	1.462570e+07	3.982332e+07	102.070863	217.389748	5.974922	2001.322658	1.7
std	92130.136561	1.000185	3.091321e+07	1.170035e+08	31.381405	575.619058	0.935142	12.812941	3.4
min	5.000000	0.000065	0.000000e+00	0.000000e+00	0.000000	10.000000	1.500000	1960.000000	0.0
25%	10596.250000	0.207583	0.000000e+00	0.000000e+00	90.000000	17.000000	5.400000	1995.000000	0.0
50%	20669.000000	0.383856	0.000000e+00	0.000000e+00	99.000000	38.000000	6.000000	2006.000000	0.0
75%	75610.000000	0.713817	1.500000e+07	2.400000e+07	111.000000	145.750000	6.600000	2011.000000	2.0
max	417859.000000	32.985763	4.250000e+08	2.781506e+09	900.000000	9767.000000	9.200000	2015.000000	4.2

In [48]:
```
df.runtime.mean()
```

Out[48]: 102.07086324314375

In [49]:
```
df.budget.mean()
```

Out[49]: 14625701.094146879

In [50]:
```
df.revenue.mean()
```

Out[50]: 39823319.793392234

In [51]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
id                      10866 non-null int64
imdb_id                 10856 non-null object
popularity              10866 non-null float64
budget                  10866 non-null int64
revenue                 10866 non-null int64
original_title          10866 non-null object
cast                    10790 non-null object
homepage                2936 non-null object
director                10822 non-null object
tagline                 8042 non-null object
keywords                9373 non-null object
overview                10862 non-null object
runtime                 10866 non-null int64
genres                  10843 non-null object
production_companies    9836 non-null object
release_date            10866 non-null object
vote_count              10866 non-null int64
vote_average            10866 non-null float64
release_year            10866 non-null int64
budget_adj              10866 non-null float64
revenue_adj             10866 non-null float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

In [52]:
```
print ('The data contain' ,df.duplicated().sum() ,'duplicated row(s)')
```

```
The data contain 1 duplicated row(s)
```

In [53]:
```
df.isnull().sum(axis=0)
```

Out[53]:
```
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
```

3. Data Cleaning

3.1 Remove Duplicate Rows

In [54]:
```
df.drop_duplicates(inplace = True)
```

3.2 Changing Format Of Release Date Into Datetime Format

In [55]:
```
df['release_date']=pd.to_datetime(df['release_date'])
df['months']=df['release_date'].dt.month_name()
```

3.3 Fill the null values with zero

In [56]:
```
df.fillna(0,inplace=True)
print()
```

3.4 Remove the unused columns

In [57]:
```
df.drop([ 'imdb_id','homepage', 'tagline','overview'] , axis = 1 , inplace = True)
```

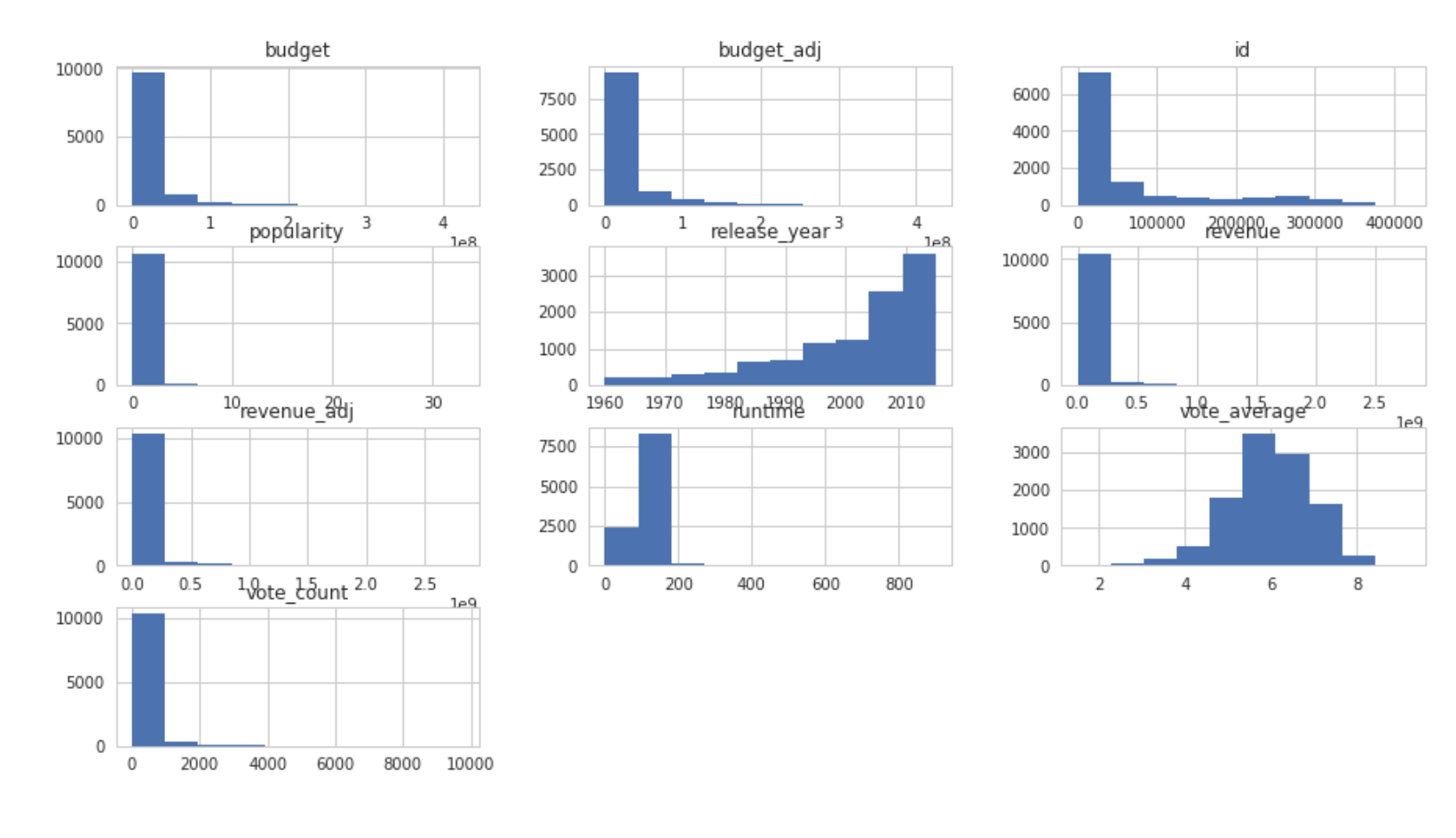
In [58]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 18 columns):
id                      10865 non-null int64
popularity              10865 non-null float64
budget                  10865 non-null int64
revenue                 10865 non-null int64
original_title          10865 non-null object
cast                    10865 non-null object
director                10865 non-null object
keywords                10865 non-null object
runtime                 10865 non-null int64
genres                  10865 non-null object
production_companies    10865 non-null object
release_date            10865 non-null datetime64[ns]
vote_count              10865 non-null int64
vote_average            10865 non-null float64
release_year            10865 non-null int64
budget_adj              10865 non-null float64
revenue_adj             10865 non-null float64
months                  10865 non-null object
dtypes: datetime64[ns](1), float64(4), int64(6), object(7)
memory usage: 1.6+ MB
```

Now we have the columns, rows in right way and ready to explore data .

4. Exploratory Data Analysis

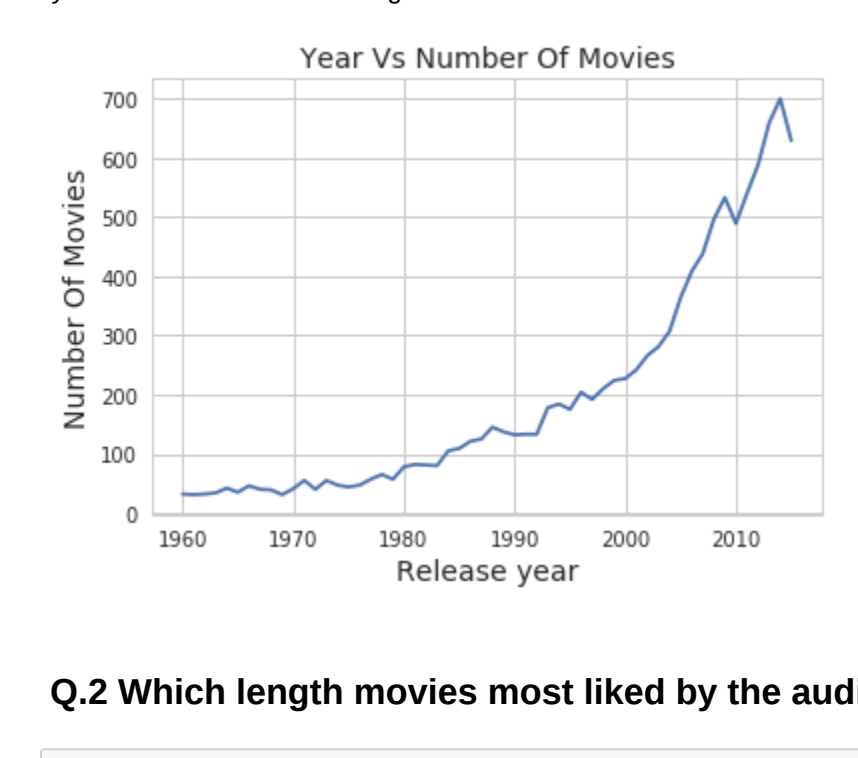
In [59]:
```
df.hist(figsize=(15,8))
print()
```



Q.1 Which year has the highest release of movies?

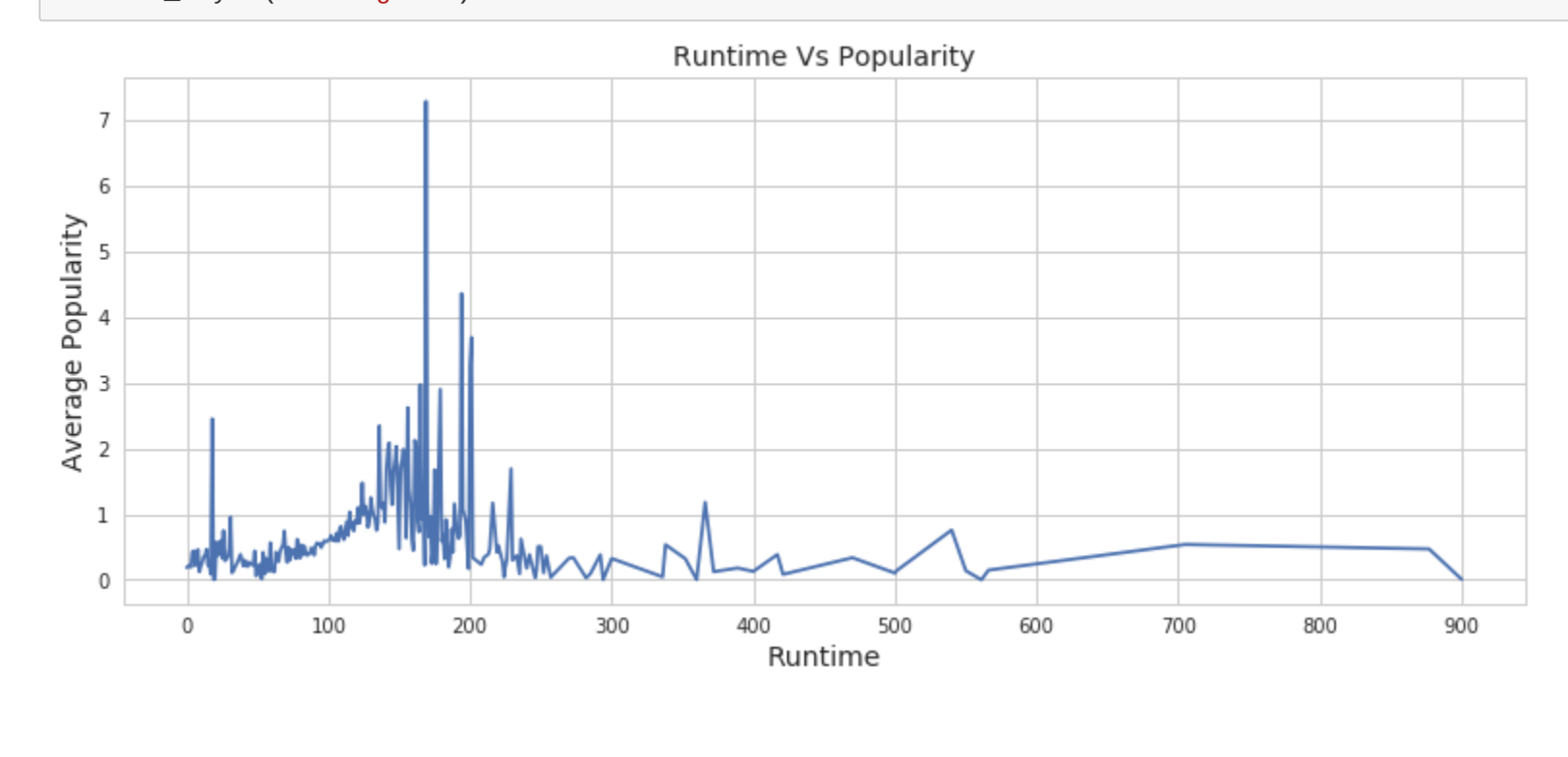
In [60]:
```
#create series (values=number of movies ,index=years)
years= list(df['release_year'].values)
s=(pd.value_counts(years)).sort_index()

#plot the figure and setup the title and labels
plt.plot(s)
plt.title("Year Vs Number Of Movies",fontsize = 14)
plt.xlabel('Release year',fontsize = 14)
plt.ylabel('Number Of Movies',fontsize = 14)
print ('year',s[s==s.max()].index[0],'has the highest release of movies',s.max())
```

```
year 2014 has the highest release of movies 700
```



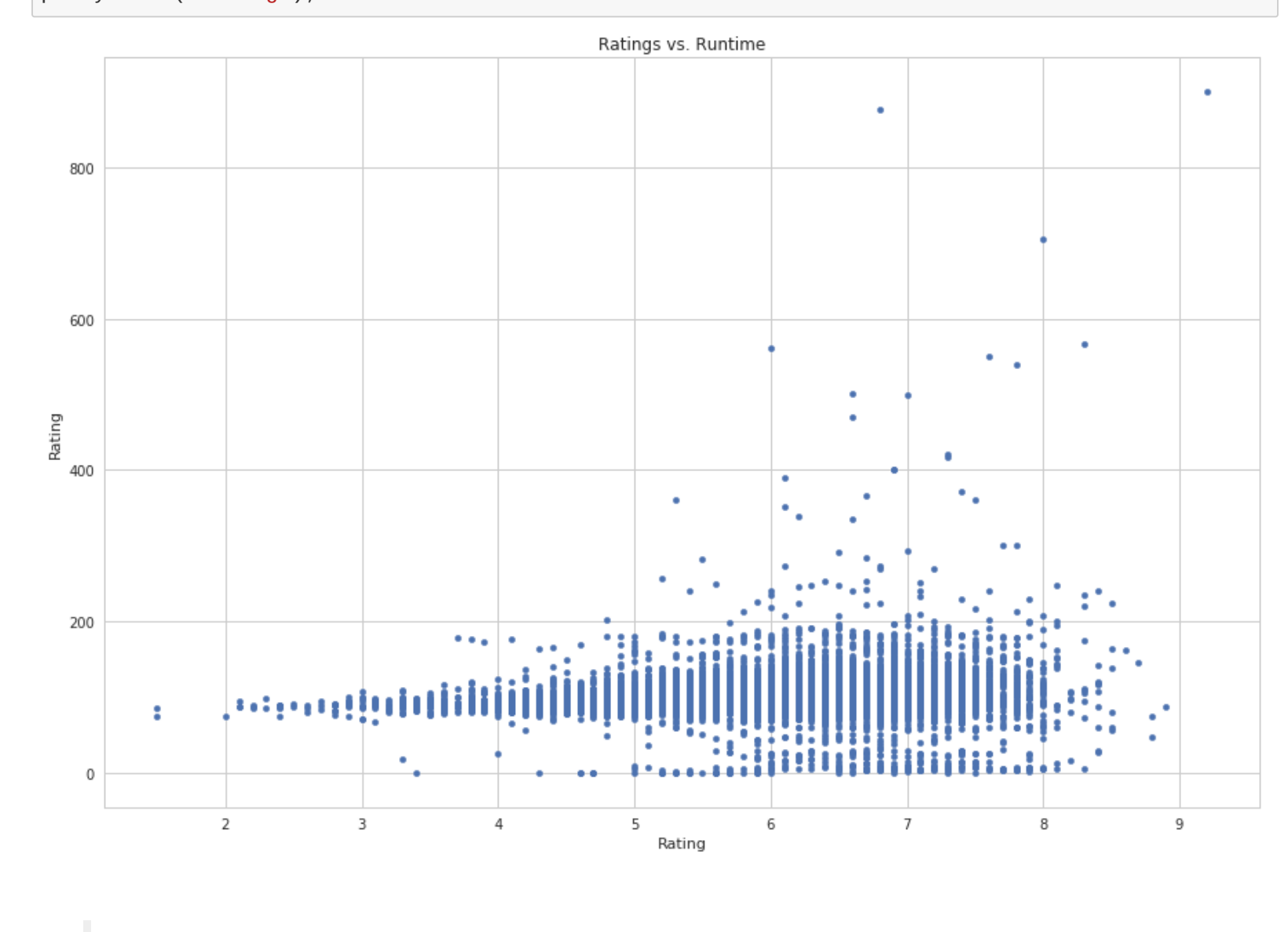
Q.2 Which length movies most liked by the audiences according to their popularity?

In [61]:
```
#make group with runtime & popularity
group=df.groupby('runtime')['popularity'].mean()

#plot the figure and setup the title and labels
group.plot(figsize = (13,5),xticks=np.arange(0,1000,100))
plt.title("Runtime Vs Popularity",fontsize = 14)
plt.xlabel('Runtime',fontsize = 14)
plt.ylabel('Average Popularity',fontsize = 14)
sns.set(rc={'figure.figsize':(10,5)})
sns.set_style("whitegrid")
```



Q.3 What is the relationship between runtime and vote average?

In [62]:
```
# Plot scatter plot of these two columns
df.plot(x='vote_average', y='runtime', kind='scatter', figsize=(15,10))
plt.title('Ratings vs. Runtime')
plt.xlabel('Rating')
plt.ylabel('Rating');
```



> From this scatter plot, we can draw several conclusions:

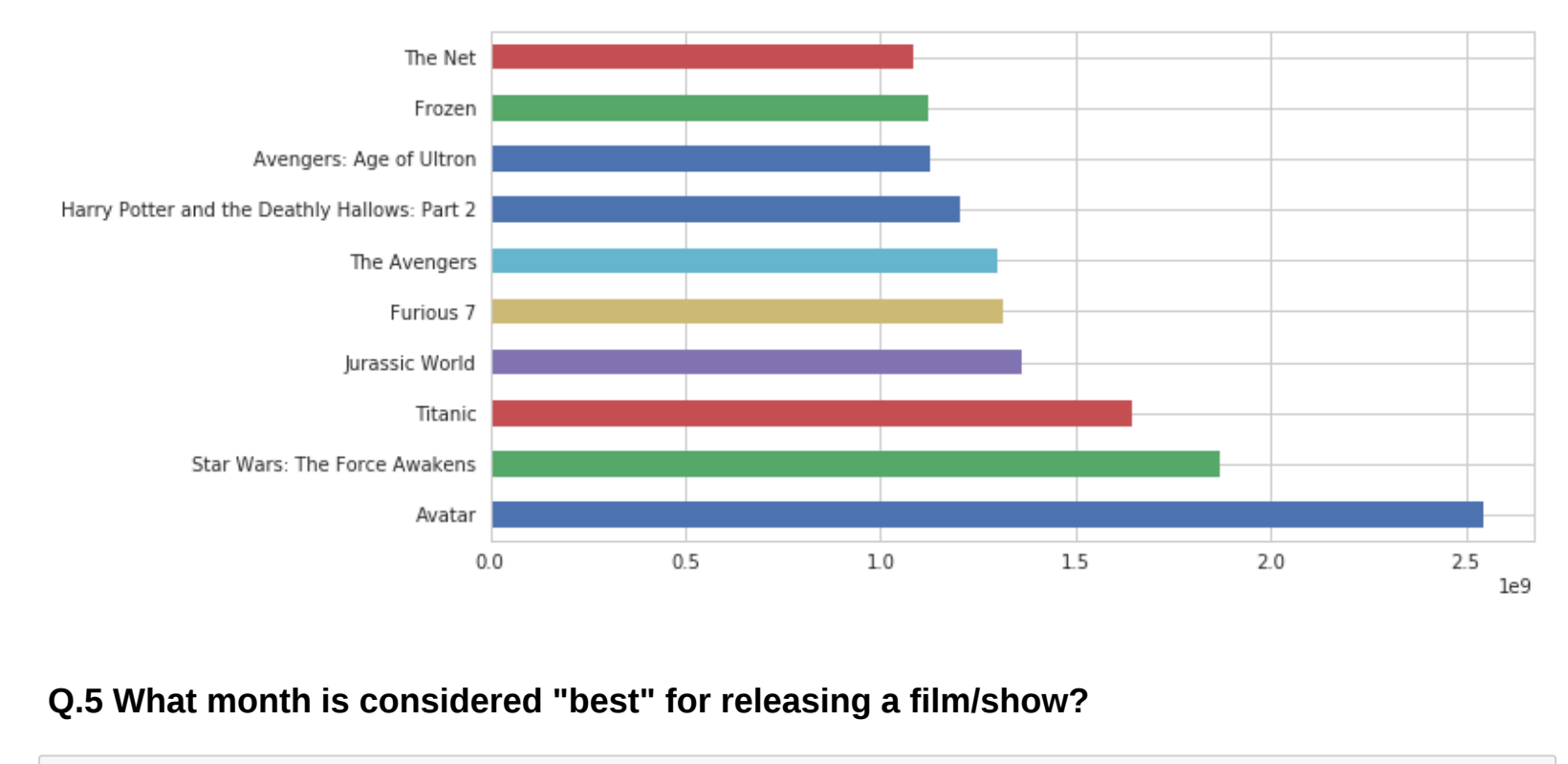
1.If it's a short film, it's likely to have a mid-to-high rating.

2.Films/shows with a runtime above or below 100 minues tend to have mid-to-high ratings.

Q.4 Which Movie Has The Highest Or Lowest Profit?

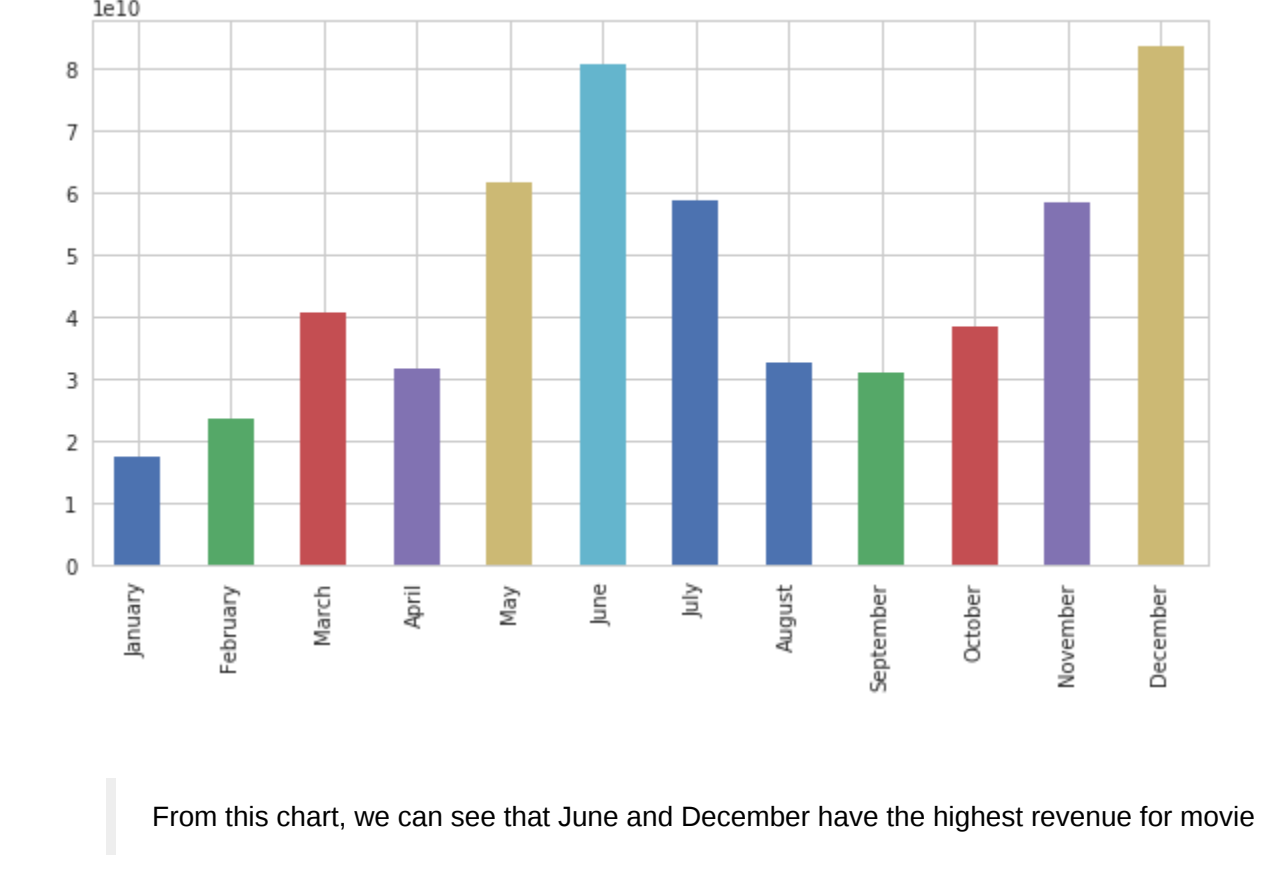
In [63]:
```
#create new column for profit
df['profit']=df['revenue']-df['budget']

#extract the top 10 profit
t= df.nlargest(10, 'profit')

#create series (value=profit ,index=original_title)
tob_10 = pd.Series(t.profit.values,t.original_title.values)

#plot the figure and setup the title and labels
tob_10.plot(kind='barh')

#ectract Movie Which Has Highest Profit
highest_Profit=df[df['profit']==df.profit.max()].original_title.values[0]

#extract Movie Which Has Lowest Profit
lowest_Profit=df[df['profit']==df.profit.min()].original_title.values[0]

print('Movie Which Has Highest Profit :',highest_Profit)
print('Movie Which Has Lowest Profit :',lowest_Profit)
```

```
Movie Which Has Highest Profit : Avatar
Movie Which Has Lowest Profit : The Warrior's Way
```



Q.5 What month is considered "best" for releasing a film/show?

In [64]:
```
#create series(index = months ,values = revenue_adj for every month)
month_revenue=df.groupby(['months']).sum().revenue_adj

#order month
sort_order = ["January", "February", "March", "April", "May", "June", "July", "August","Sept
ember", "October", "November", "December"]
month_revenue.index = pd.Categorical(month_revenue.index, categories = sort_order, ordered =
True)
month_revenue=month_revenue.sort_index()

#plot the figure and setup the title and labels
month_revenue.plot(kind='bar')
print()
```

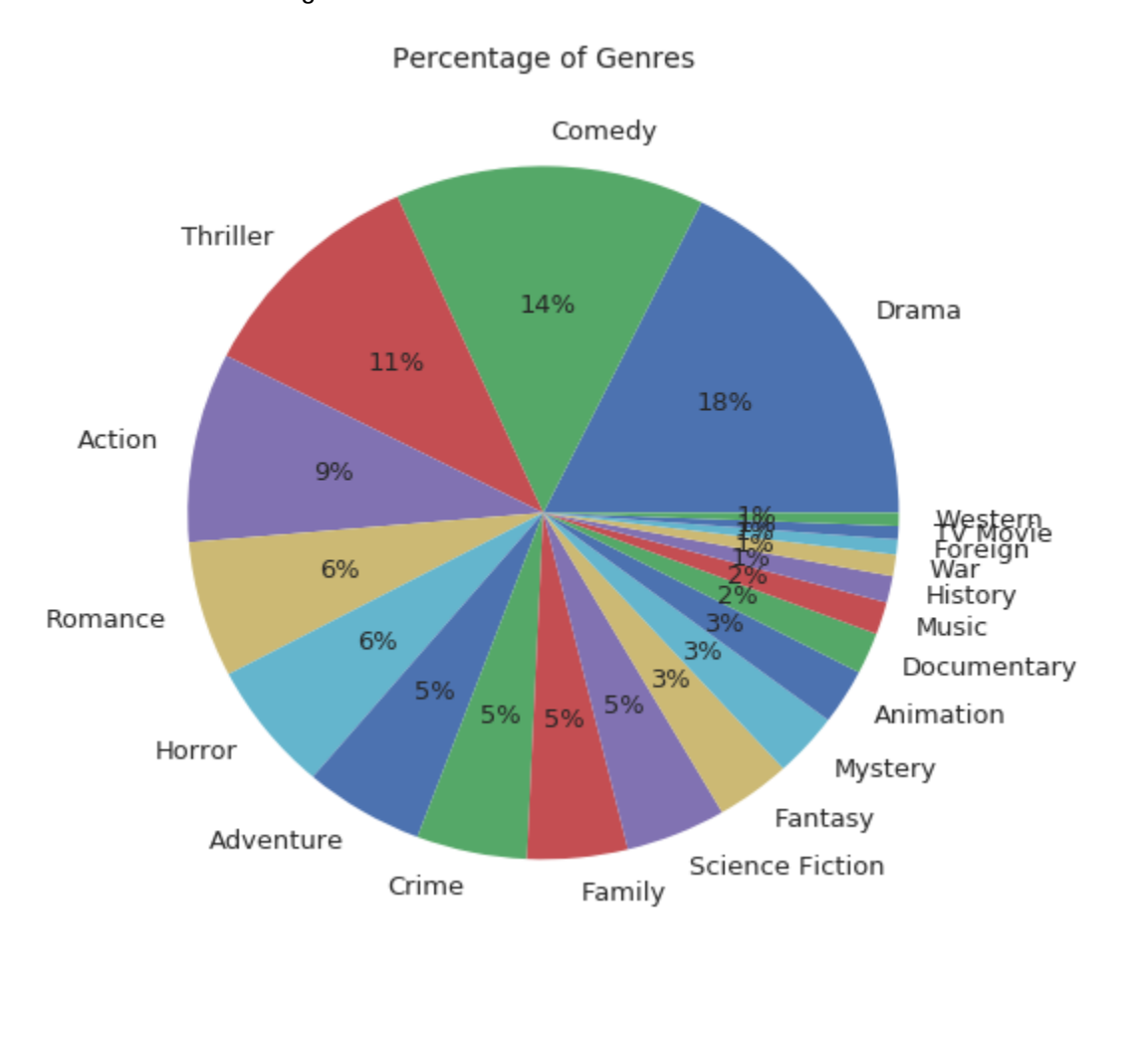


> From this chart, we can see that June and December have the highest revenue for movie releases

Q.6 Which Genre Has The Highest Release Of Movies?

In [65]:
```
#drop nan value in genres
d_f=df.copy()
d_f['genres'] = d_f['genres'].replace(0, np.nan)
d_f=d_f.dropna()

#extract different type of movies
genre= list(d_f['genres'].values)
genre="|".join(str(v)for v in genre)
genre=genre.split('|')
m=pd.value_counts(genre)

#plot the figure and setup the title and labels
m.plot(kind='pie',figsize=(8,8), fontsize=13, autopct='%1.0f%%')
plt.title('Percentage of Genres',fontsize = 14)
plt.ylabel('');

print (m[m==m.max()].index[0],'has the highest release of movies',m.max())
```

```
Drama has the highest release of movies 4760
```

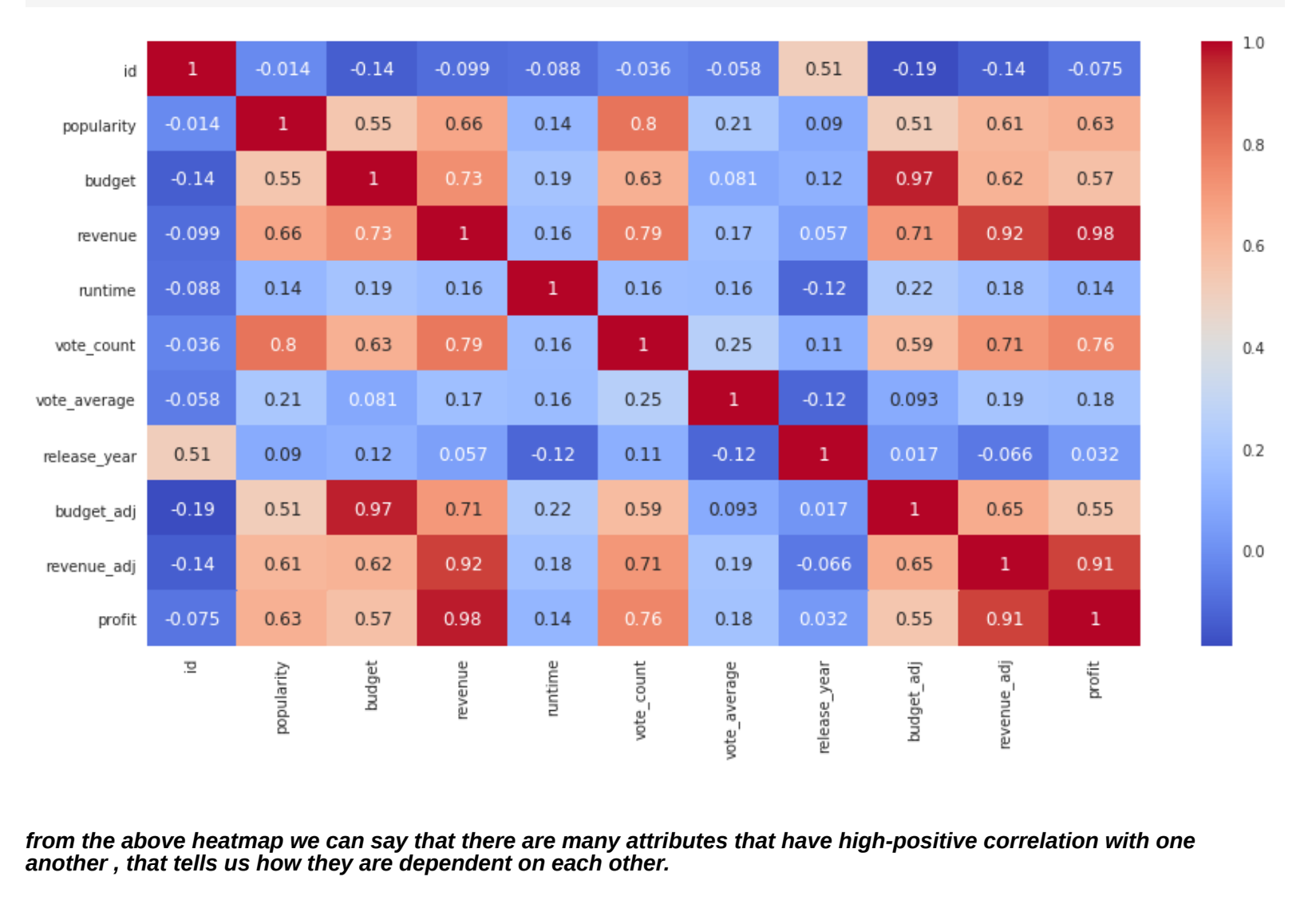


Q.7 What Kind Of Properties Are Associated With Movies With High Revenue?

In [66]:
```
plt.figure(figsize=(15,7.5))
c=df.corr()
sns.heatmap(c,cmap='coolwarm',annot=True)
c
```

Out[66]:

	id	popularity	budget	revenue	runtime	vote_count	vote_average	release_year	budget_adj	rever
id	1.000000	-0.014351	-0.141341	-0.099235	-0.088368	-0.035555	-0.058391	0.511393	-0.189008	-0
popularity	-0.014351	1.000000	0.545481	0.663360	0.139032	0.800828	0.209517	0.089806	0.513555	0
budget	-0.141341	0.545481	1.000000	0.734928	0.191300	0.632719	0.081067	0.115904	0.968963	0
revenue	-0.099235	0.663360	0.734928	1.000000	0.162830	0.791174	0.172541	0.057070	0.706446	0
runtime	-0.088368	0.139032	0.191300	0.162830	1.000000	0.163273	0.156813	-0.117187	0.221127	0
vote_count	-0.035555	0.800828	0.632719	0.791174	0.163273	1.000000	0.253818	0.107962	0.587062	0
vote_average	-0.058391	0.209517	0.081067	0.172541	0.156813	0.253818	1.000000	-0.117576	0.093079	0
release_year	0.511393	0.089806	0.115904	0.057070	-0.117187	0.107962	-0.117576	1.000000	0.016771	-0
budget_adj	-0.189008	0.513555	0.968963	0.706446	0.221127	0.587062	0.093079	0.016771	1.000000	0
revenue_adj	-0.138487	0.609085	0.622531	0.919109	0.175668	0.707941	0.193062	-0.066236	0.646627	1
profit	-0.074975	0.628997	0.570222	0.976162	0.136022	0.755908	0.183067	0.032038	0.545654	0



from the above heatmap we can say that there are many attributes that have high-positive correlation with one another , that tells us how they are dependent on each other.

Budget has the highest correlation with revenue , that implies when a movie's budget is high that tends to generate more revenue .

profit has a high correlation with vote counts , which means if a movie receives a high number of votes from people or from critics , it results in the movie making more profit .

5. Conclusions

1.year 2014 has the highest release of movies 700

2.that movies in the range of 100-200 runtime are more popular than other runtime movies .

3.Movie Which Has Highest Profit :' Avatar' , 'Star Wars' and 'Titanic'

4.Movie Which Has Lowest Profit : The Warrior's Way

5.June and December have the highest revenue for movie releases.

6.Drama is the most popular genre, following by action, comedy and thriller.

7.Budget has the positive correlation with revenue

8.profit has a positive correlation with vote counts

In [67]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])
```

Out[67]: 0