In [ ]:
```
#Do the higher budget movies achieve higher ratings ?
```

In [ ]:
```
#Which features or properties can be associated with high ratings ?
# how many rows and columns do we have ?
# do we have a duplicated rows ?
# what are the data types ?
# do we have null values ?
# what are the best year in relising movies ?
```

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#import the pakages that we will use
```

In [5]:
```
#how many rows and columns
dfmain=pd.read_csv("movies.csv")
dfmain.shape
```

Out[5]: (10866, 21)

In [6]:
```
#we need to inquery about the duplicated rows
sum(dfmain.duplicated())
```

Out[6]: 1

In [7]:
```
#drop the duplicated rows
dfmain.drop_duplicates(inplace = True)
```

In [8]:
```
dfmain['runtime'] = dfmain['runtime'].replace(0, np.NaN)
dfmain['budget_adj'] = dfmain['budget_adj'].replace(0, np.NaN)
dfmain['revenue_adj'] = dfmain['revenue_adj'].replace(0, np.NaN)
    #a great methods to replace the '0' with 'NULLS
```

In [9]:

```

```

In [10]:

```
dfmain.dtypes
```

Out[10]:

```
id                        int64
imdb_id                  object
popularity              float64
budget                    int64
revenue                   int64
original_title           object
cast                     object
homepage                 object
director                 object
tagline                  object
keywords                 object
overview                 object
runtime                 float64
genres                   object
production_companies     object
release_date             object
vote_count                int64
vote_average            float64
release_year              int64
budget_adj              float64
revenue_adj             float64
dtype: object
```

In [11]:

```
## descrption of the dataset
dfmain['release_year'].describe()
#the movies release date is between 1960 and 2015 and 2011 takes the mo
st relaising movies date
```

Out[11]:

```
count    10865.000000
mean      2001.321859
std         12.813260
min       1960.000000
25%       1995.000000
50%       2006.000000
75%       2011.000000
```

```
max      2015.000000
Name: release_year, dtype: float64
```

In [12]:
```
#here is the amount of null values in the columns
dfmain.isnull().sum()
```

Out[12]:
```
id                         0
imdb_id                   10
popularity                 0
budget                     0
revenue                    0
original_title             0
cast                      76
homepage                7929
director                  44
tagline                 2824
keywords                1493
overview                   4
runtime                   31
genres                    23
production_companies    1030
release_date               0
vote_count                 0
vote_average               0
release_year               0
budget_adj              5696
revenue_adj             6016
dtype: int64
```

In [13]:
```
# bin edges to cut the data into groups
bin_edges = [1.5, 5.4, 6.0, 6.6, 9.2]

# labels for the rating categories
bin_names = ['low', 'mediocre', 'high', 'very_high']
```

In [14]:
```
# create rating categories
dfmain['rating_catagory'] = pd.cut(dfmain['vote_average'], bin_edges, l
abels=bin_names)
```

```python
# confirm the creation
dfmain.head()
```

Out[14]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	

5 rows × 22 columns

In [15]:

```python
# Description of vote_average
dfmain['vote_average'].describe()
# the vote average range from 1.5 to 9.2
```

Out[15]:

```
count    10865.000000
mean         5.975012
std          0.935138
min          1.500000
```

```
25%          5.400000
50%          6.000000
75%          6.600000
max          9.200000
Name: vote_average, dtype: float64
```

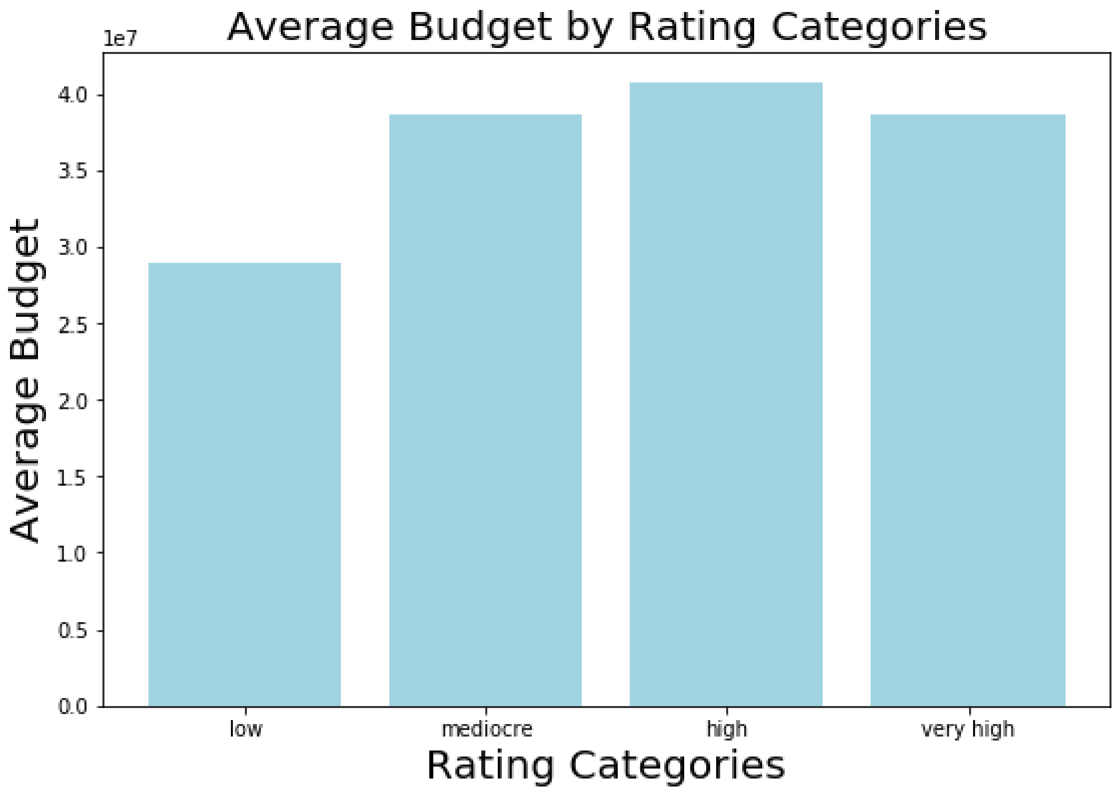
In [16]:
```
# i create averages grouped by the rating categories
means = dfmain.groupby('rating_catagory')['budget_adj'].mean()
```

In [24]:
```
# :Do the higher budget movies achieve higher ratings ?
```

In [25]:
```
# Create a bar chart with proper labels
locations = [1,2,3,4]

heights= [means['low'],means ['mediocre'],means['high'],means['very_hig
h']]
labels= ['low','mediocre','high','very high']
plt.figure(figsize=(9, 6))
plt.bar(locations, heights, color='#88cada', alpha=.8, tick_label=label
s)
plt.title('Average Budget by Rating Categories', fontdict={'fontsize':
20})
plt.xlabel('Rating Categories', fontdict={'fontsize': 20})
plt.ylabel('Average Budget', fontdict={'fontsize': 20})
```

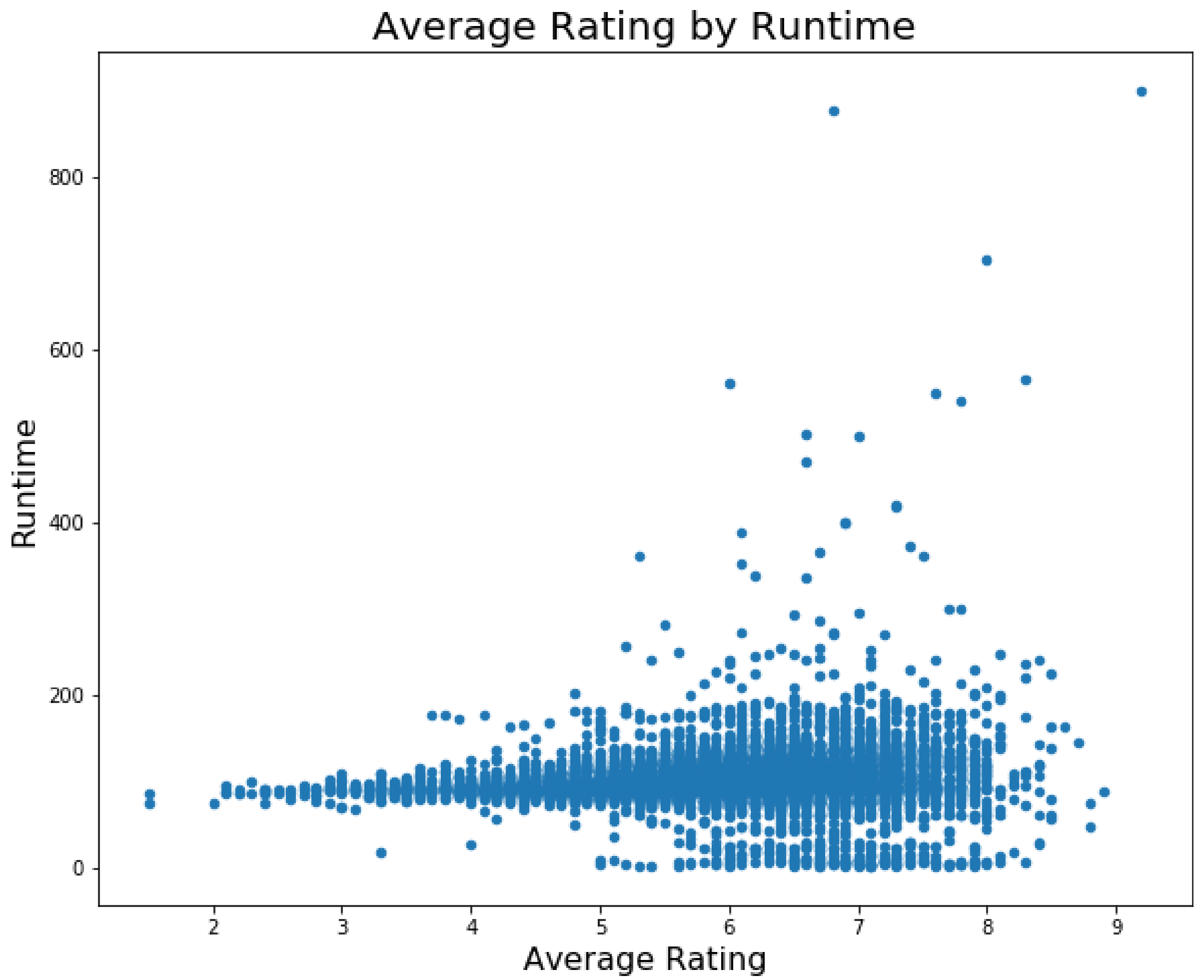
Out[25]:
```
Text(0, 0.5, 'Average Budget')
```



In [26]:
```
# in thid visulization it shows that the movies with less budget will t
ake a lower rating
#in the other  hand it is not necessary if you have a Huge budget you w
ill garante the very high ratings
# the most spending budget takes the high rate by catagory , even thogh
 the very high is less budget
```

In [27]:
```
##I created scatterplot for runtime, popularity in combination with the
 average rating to identify possible correlations
dfmain.plot(y='runtime', x='vote_average', kind='scatter', figsize=(10,
 8))
```

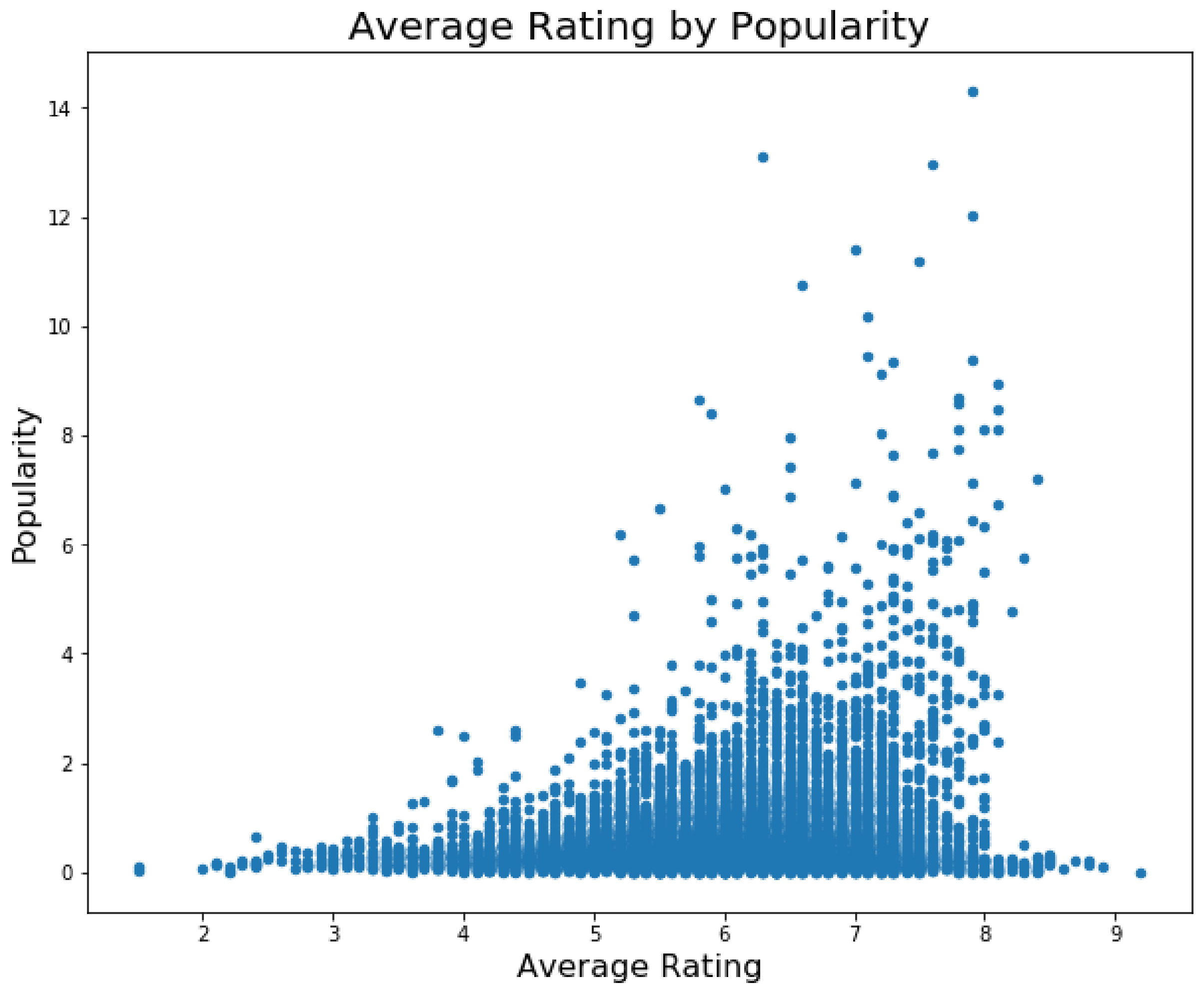
```
plt.title('Average Rating by Runtime', fontdict={'fontsize': 20})
plt.xlabel('Average Rating', fontdict={'fontsize': 16})
plt.ylabel('Runtime', fontdict={'fontsize': 16});
```



In [ ]:
```
# in this scatter plot shows  theat most movies less than 200 minutes w
hitch make sinse to rate it
# the rlation is strong when the movie is less thann 200 min
```

In [39]:
```
dfmain_pop = dfmain.query('popularity <= 15') # excluding the outliers
```

```
dfmain_pop.plot(y='popularity', x='vote_average', kind='scatter', figsize=(10, 8))
plt.title('Average Rating by Popularity', fontdict={'fontsize': 20})
plt.xlabel('Average Rating', fontdict={'fontsize': 16})
plt.ylabel('Popularity', fontdict={'fontsize': 16});
```



In [ ]:

In [ ]:
```
# the same idea from the previos scatter plot but here the relatinship
 between popularity and writings
```

In [ ]:
```
# Conclusions
#The chart indicates minor differences in the average budgets, while me
diocre and high rating categories contribute for higher average budgets
 than low and very high ratings.
#From the visualization, a positive correlation between popularity and
 rating
```

In [35]:
```
dfmain.head()
```

Out[35]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	http
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	

5 rows × 22 columns

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```