Investigate_a_Dataset

October 20, 2018

Tip: Welcome to the Investigate a Dataset project! You will find tips in quoted sections like this to help organize your approach to your investigation. Before submitting your project, it will be a good idea to go back through your report and remove these sections to make the presentation of your work as tidy as possible. First things first, you might want to double-click this Markdown cell and change the title so that it reflects your dataset and investigation.

1 Project: Investigate a Dataset (TMDb_Movies Dataset)

1.1 Table of Contents

Introduction
Data Wrangling
Exploratory Data Analysis
Conclusions
## Introduction

For third Data Analysis project: I have selected TMDb movies dataset. This data set contains information about 10,000 movies collected from The Movie Database (TMDb), including user ratings and revenue. It consists of 21 columns such as imdb_id, revenue, budget, vote_count etc.

The questions that I am interested to analyze from this dataset: * What the successful movie genres? * What the relationship between the popularity and the runtime of the movie? * What the relationship between the budget and the runtime of the movie? * what's the highest budget and profit in the TMDb_Movies dataset? * What's the lowest budget and profit in the TMDb_Movies dataset?

First: In this section of the report, Import the necessary package and use pd.read_csv to load the movie dataset, then print the first rows.

```
In [26]: # Use this cell to set up import statements for all of the packages that you
         #   plan to use.

         # Remember to include a 'magic word' so that your visualizations are plotted
         #   inline with the notebook. See this page for more:
         #   http://ipython.readthedocs.io/en/stable/interactive/magics.html
         import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
% matplotlib inline
```

## Data Wrangling

First: In this section of the report, Import the necessary package and use pd.read_csv to load the movie dataset, then print the first rows.

1.1.1 General Properties

```
In [27]: # Load your data and print out a few lines. Perform operations to inspect data
         #   types and look for instances of missing or possibly errant data.
         df= pd.read_csv('tmdb-movies.csv')
         df.head()
```

```
Out[27]:        id    imdb_id  popularity     budget     revenue  \
         0  135397  tt0369610   32.985763  150000000  1513528810
         1   76341  tt1392190   28.419936  150000000   378436354
         2  262500  tt2908446   13.112507  110000000   295238201
         3  140607  tt2488496   11.173104  200000000  2068178225
         4  168259  tt2820852    9.335014  190000000  1506249360

                         original_title  \
         0                 Jurassic World
         1             Mad Max: Fury Road
         2                      Insurgent
         3  Star Wars: The Force Awakens
         4                      Furious 7

                                                          cast  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...
         1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...
         2  Shailene Woodley|Theo James|Kate Winslet|Ansel...
         3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...
         4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...

                                                     homepage           director  \
         0                      http://www.jurassicworld.com/    Colin Trevorrow
         1                        http://www.madmaxmovie.com/      George Miller
         2     http://www.thedivergentseries.movie/#insurgent  Robert Schwentke
         3  http://www.starwars.com/films/star-wars-episod...        J.J. Abrams
         4                           http://www.furious7.com/          James Wan

                               tagline      ...       \
         0           The park is open.      ...
         1          What a Lovely Day.      ...
         2      One Choice Can Destroy You      ...
```

```
3  Every generation has a story.      ...
4            Vengeance Hits Home      ...

                                             overview runtime  \
0  Twenty-two years after the events of Jurassic ...     124
1  An apocalyptic story set in the furthest reach...     120
2  Beatrice Prior must confront her inner demons ...     119
3  Thirty years after defeating the Galactic Empi...     136
4  Deckard Shaw seeks revenge against Dominic Tor...     137

                                      genres  \
0  Action|Adventure|Science Fiction|Thriller
1  Action|Adventure|Science Fiction|Thriller
2         Adventure|Science Fiction|Thriller
3   Action|Adventure|Science Fiction|Fantasy
4                      Action|Crime|Thriller

                                production_companies release_date vote_count  \
0  Universal Studios|Amblin Entertainment|Legenda...       6/9/15       5562
1  Village Roadshow Pictures|Kennedy Miller Produ...      5/13/15       6185
2  Summit Entertainment|Mandeville Films|Red Wago...      3/18/15       2480
3          Lucasfilm|Truenorth Productions|Bad Robot     12/15/15       5292
4  Universal Pictures|Original Film|Media Rights ...       4/1/15       2947

   vote_average  release_year    budget_adj   revenue_adj
0           6.5          2015  1.379999e+08  1.392446e+09
1           7.1          2015  1.379999e+08  3.481613e+08
2           6.3          2015  1.012000e+08  2.716190e+08
3           7.5          2015  1.839999e+08  1.902723e+09
4           7.3          2015  1.747999e+08  1.385749e+09

[5 rows x 21 columns]
```

```
In [28]: df.shape
```

```
Out[28]: (10866, 21)
```

It has 10866 columns and 21 rows.

```
In [29]: 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 [30]: df.describe()
```

```
Out[30]:                   id    popularity        budget       revenue       runtime  \
         count   10866.000000  10866.000000  1.086600e+04  1.086600e+04  10866.000000
         mean    66064.177434      0.646441  1.462570e+07  3.982332e+07    102.070863
         std     92130.136561      1.000185  3.091321e+07  1.170035e+08     31.381405
         min         5.000000      0.000065  0.000000e+00  0.000000e+00      0.000000
         25%     10596.250000      0.207583  0.000000e+00  0.000000e+00     90.000000
         50%     20669.000000      0.383856  0.000000e+00  0.000000e+00     99.000000
         75%     75610.000000      0.713817  1.500000e+07  2.400000e+07    111.000000
         max    417859.000000     32.985763  4.250000e+08  2.781506e+09    900.000000

                  vote_count  vote_average  release_year    budget_adj   revenue_adj
         count  10866.000000  10866.000000  10866.000000  1.086600e+04  1.086600e+04
         mean     217.389748      5.974922   2001.322658  1.755104e+07  5.136436e+07
         std      575.619058      0.935142     12.812941  3.430616e+07  1.446325e+08
         min       10.000000      1.500000   1960.000000  0.000000e+00  0.000000e+00
         25%       17.000000      5.400000   1995.000000  0.000000e+00  0.000000e+00
         50%       38.000000      6.000000   2006.000000  0.000000e+00  0.000000e+00
         75%      145.750000      6.600000   2011.000000  2.085325e+07  3.369710e+07
         max     9767.000000      9.200000   2015.000000  4.250000e+08  2.827124e+09
```

From The Movie Database (TMDb): We can see that: Certain columns, like ‘cast’ and ‘genres’, * contain multiple values separated by pipe (|) characters. There are some odd characters in the ‘cast’ column. Don’t worry about cleaning them. You can leave them as is. * The final two columns ending with “_adj” show the budget and revenue of the associated movie in terms of 2010 dollars, accounting for inflation over time.

1.1.2 Data Cleaning (Removing the unused columns from The Movie Database (TMDb))

In this step: I will do * Removing the duplicate record(if any). * Removing the rows with null values (if any). * Removing the unused columns from The Movie Database. * Removing the zeros from budget and the revenue columns.

First: I will remove the unused columns which are id, imdb_id,original_title, homepage, tagline, overview, production_companies,vote_count,vote_average, budget_adj, and revenue_adj.

```
In [31]: #After discussing the structure of the data and any problems that need to be
         #cleaned, perform those cleaning steps in the second part of this section.
         df.drop(['id', 'imdb_id','original_title', 'homepage', 'tagline', 'overview', 'producti
         df.head()
```

```
Out[31]:    popularity     budget     revenue  \
         0   32.985763  150000000  1513528810
         1   28.419936  150000000   378436354
         2   13.112507  110000000   295238201
         3   11.173104  200000000  2068178225
         4    9.335014  190000000  1506249360

                                                          cast          director  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   Colin Trevorrow
         1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...     George Miller
         2  Shailene Woodley|Theo James|Kate Winslet|Ansel...  Robert Schwentke
         3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...       J.J. Abrams
         4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...         James Wan

                                                  keywords  runtime  \
         0  monster|dna|tyrannosaurus rex|velociraptor|island      124
         1   future|chase|post-apocalyptic|dystopia|australia      120
         2  based on novel|revolution|dystopia|sequel|dyst...      119
         3              android|spaceship|jedi|space opera|3d      136
         4                car race|speed|revenge|suspense|car      137

                                           genres release_date  release_year
         0  Action|Adventure|Science Fiction|Thriller       6/9/15          2015
         1  Action|Adventure|Science Fiction|Thriller      5/13/15          2015
         2         Adventure|Science Fiction|Thriller      3/18/15          2015
         3   Action|Adventure|Science Fiction|Fantasy     12/15/15          2015
         4                      Action|Crime|Thriller       4/1/15          2015
```

Removing the duplicate record.

```
In [32]: # I have removed the duplicate record.
         df.drop_duplicates(inplace=True)
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 10 columns):
popularity      10865 non-null float64
budget          10865 non-null int64
revenue         10865 non-null int64
cast            10789 non-null object
director        10821 non-null object
keywords        9372 non-null object
runtime         10865 non-null int64
genres          10842 non-null object
release_date    10865 non-null object
release_year    10865 non-null int64
dtypes: float64(1), int64(4), object(5)
memory usage: 933.7+ KB
```

Removing the rows with null values

```
In [33]: # I have removed rows with null values
         df.dropna(inplace=True)
         df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 9306 entries, 0 to 10865
Data columns (total 10 columns):
popularity      9306 non-null float64
budget          9306 non-null int64
revenue         9306 non-null int64
cast            9306 non-null object
director        9306 non-null object
keywords        9306 non-null object
runtime         9306 non-null int64
genres          9306 non-null object
release_date    9306 non-null object
release_year    9306 non-null int64
dtypes: float64(1), int64(4), object(5)
memory usage: 799.7+ KB
```

Removing zeros from budget and revenue columns.

```
In [34]: value=['budget', 'revenue']


         df[value] = df[value].replace(0, np.NAN)


         df.dropna(subset = value, inplace = True)
```

Create a column called "profit'

```
In [35]: df.loc[:,'Profit'] = (df['revenue'] - df['budget'])
         df.head()
```

```
Out[35]:    popularity       budget       revenue  \
         0   32.985763  150000000.0  1.513529e+09
         1   28.419936  150000000.0  3.784364e+08
         2   13.112507  110000000.0  2.952382e+08
         3   11.173104  200000000.0  2.068178e+09
         4    9.335014  190000000.0  1.506249e+09

                                                        cast          director  \
         0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...   Colin Trevorrow
         1  Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...     George Miller
         2  Shailene Woodley|Theo James|Kate Winslet|Ansel...  Robert Schwentke
         3  Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...       J.J. Abrams
         4  Vin Diesel|Paul Walker|Jason Statham|Michelle ...         James Wan

                                                    keywords  runtime  \
         0  monster|dna|tyrannosaurus rex|velociraptor|island      124
         1   future|chase|post-apocalyptic|dystopia|australia      120
         2  based on novel|revolution|dystopia|sequel|dyst...      119
         3              android|spaceship|jedi|space opera|3d      136
         4                car race|speed|revenge|suspense|car      137

                                               genres release_date  release_year  \
         0  Action|Adventure|Science Fiction|Thriller       6/9/15          2015
         1  Action|Adventure|Science Fiction|Thriller      5/13/15          2015
         2         Adventure|Science Fiction|Thriller      3/18/15          2015
         3   Action|Adventure|Science Fiction|Fantasy     12/15/15          2015
         4                      Action|Crime|Thriller       4/1/15          2015

                  Profit
         0  1.363529e+09
         1  2.284364e+08
         2  1.852382e+08
         3  1.868178e+09
         4  1.316249e+09
```

Create function to return the highest value

```
In [36]: def highest(column):
             high= df[column].idxmax()
             value=pd.DataFrame(df.loc[high])


             return value
```

```
In [37]: highest('budget')

Out[37]:                                                    2244
         popularity                                      0.25054
         budget                                         4.25e+08
         revenue                                     1.10876e+07
         cast           Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
         director                                      Sngmoo Lee
         keywords       assassin|small town|revenge|deception|super speed
         runtime                                             100
         genres                  Adventure|Fantasy|Action|Western|Thriller
         release_date                                    12/2/10
         release_year                                       2010
         Profit                                     -4.13912e+08
```

```
In [38]: highest('Profit')

Out[38]:                                                    1386
         popularity                                      9.43277
         budget                                         2.37e+08
         revenue                                     2.78151e+09
         cast           Sam Worthington|Zoe Saldana|Sigourney Weaver|S...
         director                                   James Cameron
         keywords       culture clash|future|space war|space colony|so...
         runtime                                             162
         genres                   Action|Adventure|Fantasy|Science Fiction
         release_date                                   12/10/09
         release_year                                       2009
         Profit                                      2.54451e+09
```

The highest value in the budget of movies was 4.25e+08. The highest value in the profit of movies was 2.54451e+09.

Create function to return the lowest value

```
In [39]: def lowest(column):
             low= df[column].idxmin()
             value=pd.DataFrame(df.loc[low])



             return value
```

```
In [40]: lowest('budget')

Out[40]:                                                    2618
         popularity                                     0.090186
         budget                                                1
         revenue                                             100
         cast           David Spade|Sophie Marceau|Ever Carradine|Step...
         director                                    Jeff Pollack
```

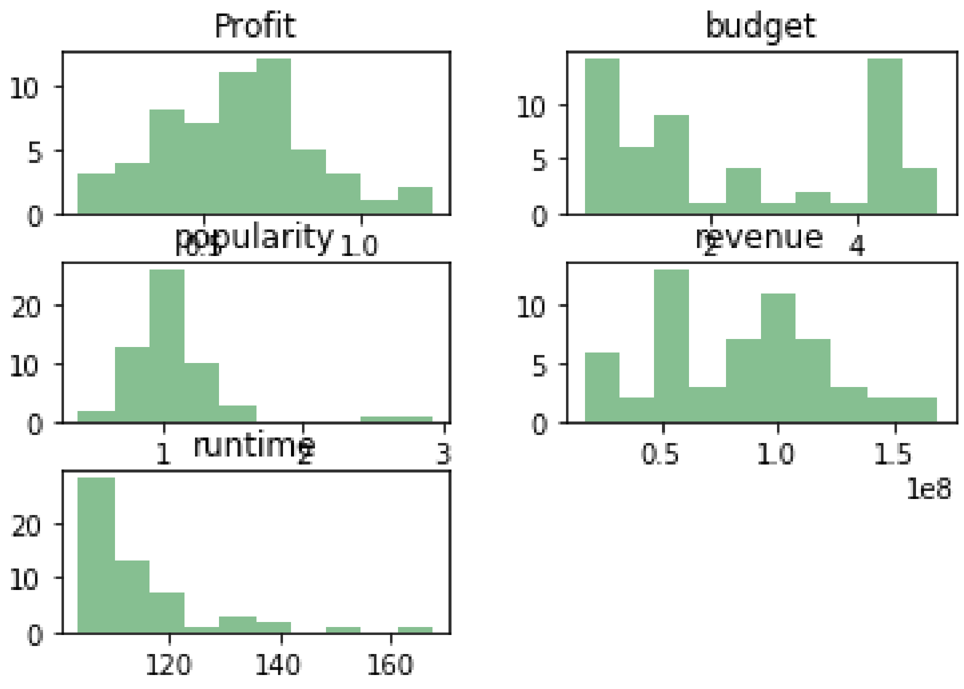
```
keywords                                    restaurant|neighbor|dog|ring
runtime                                                               95
genres                                                    Comedy|Romance
release_date                                                     4/23/99
release_year                                                        1999
Profit                                                                99
```

```
In [41]: lowest('Profit')
```

```
Out[41]:                                                             2244
         popularity                                               0.25054
         budget                                                  4.25e+08
         revenue                                              1.10876e+07
         cast            Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
         director                                               Sngmoo Lee
         keywords        assassin|small town|revenge|deception|super speed
         runtime                                                       100
         genres                    Adventure|Fantasy|Action|Western|Thriller
         release_date                                              12/2/10
         release_year                                                 2010
         Profit                                               -4.13912e+08
```

The lowest value in budget was 1. The lowest value in profit was -4.13912e+08.

```
In [42]: data = df.groupby('release_year').mean()
         data.hist(color='#86bf91',grid=False,);
```



Here, I just want to know the mean of the values of the release year.
## Exploratory Data Analysis

Before answering the first question: let's create a function to split the values separated by pipe (|) characters.

```
In [43]: def count(column):
             count = df[column].str.cat(sep = '|')

             count = pd.Series(count.split('|'))

             x = count.value_counts(ascending = False)

             return x
```

1.1.3 Research Question 1 (Replace this header name!)

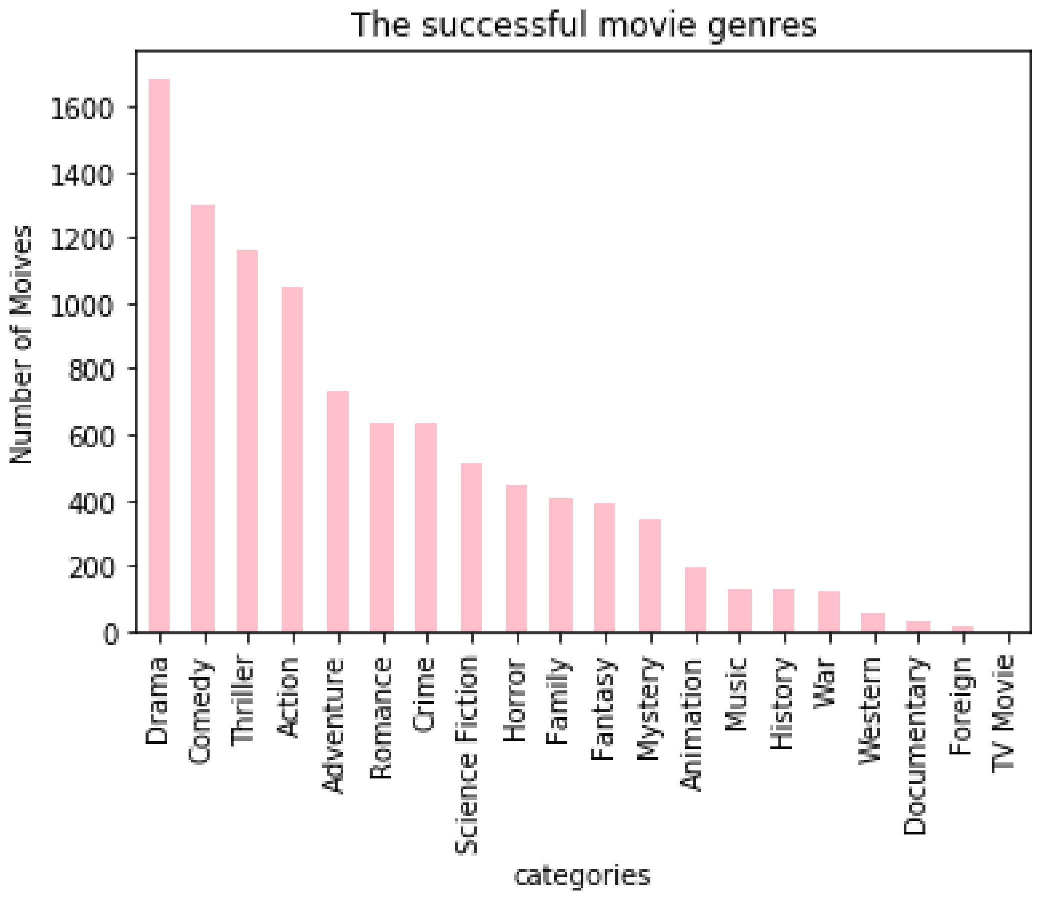
- What the successful movie genres?

```
In [44]: # Use this, and more code cells, to explore your data. Don't forget to add
         #   Markdown cells to document your observations and findings.
         x = count('genres')
         x.head()
```

```
Out[44]: Drama        1686
         Comedy       1301
         Thriller     1164
         Action       1050
         Adventure     729
         dtype: int64
```

```
In [52]: #Create bars and choose color
         #Add title and axis names
         x.plot(kind = "bar", color='pink')
         plt.title('The successful movie genres')
         plt.xlabel('categories')
         plt.ylabel('Number of Moives')
```

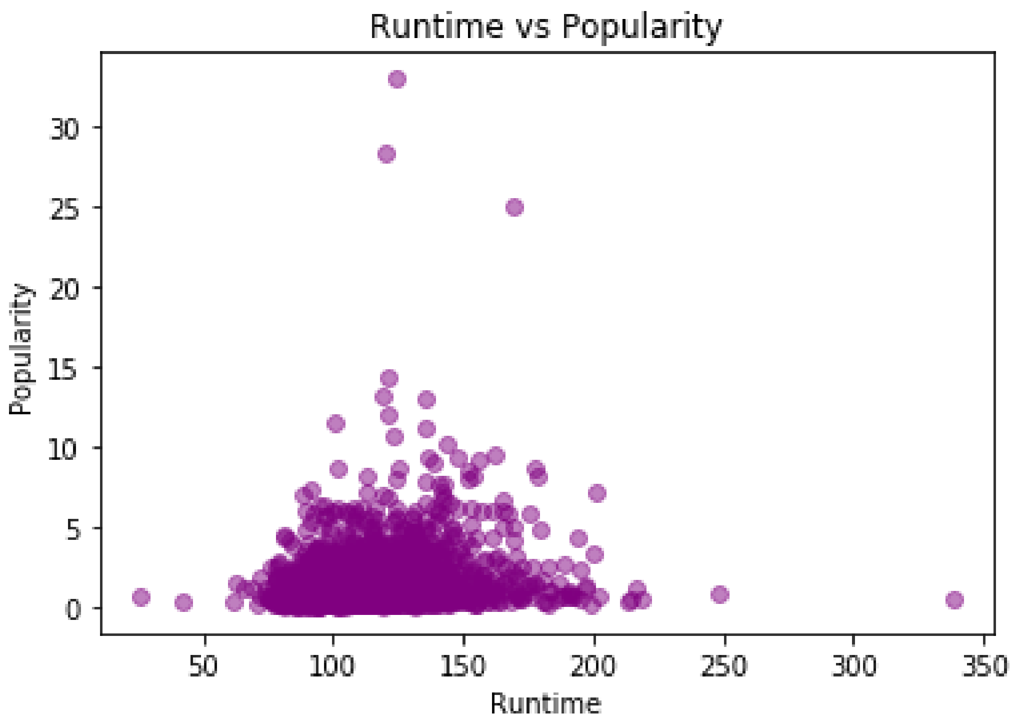
```
Out[52]: Text(0,0.5,'Number of Moives')
```



As we see from the bar chart the Drama and Comedy movies have the most popularity.

1.1.4 Research Question 2 (What the relationship between the runtime and the popularity of the movie?)

```
In [46]: # Create scatter and choose color
         # Add title and axis names
         plt.scatter(x=df['runtime'], y=df['popularity'],color='purple', alpha=0.5)
         plt.title('Runtime vs Popularity')
         plt.xlabel('Runtime')
         plt.ylabel('Popularity')
         plt.show()
```



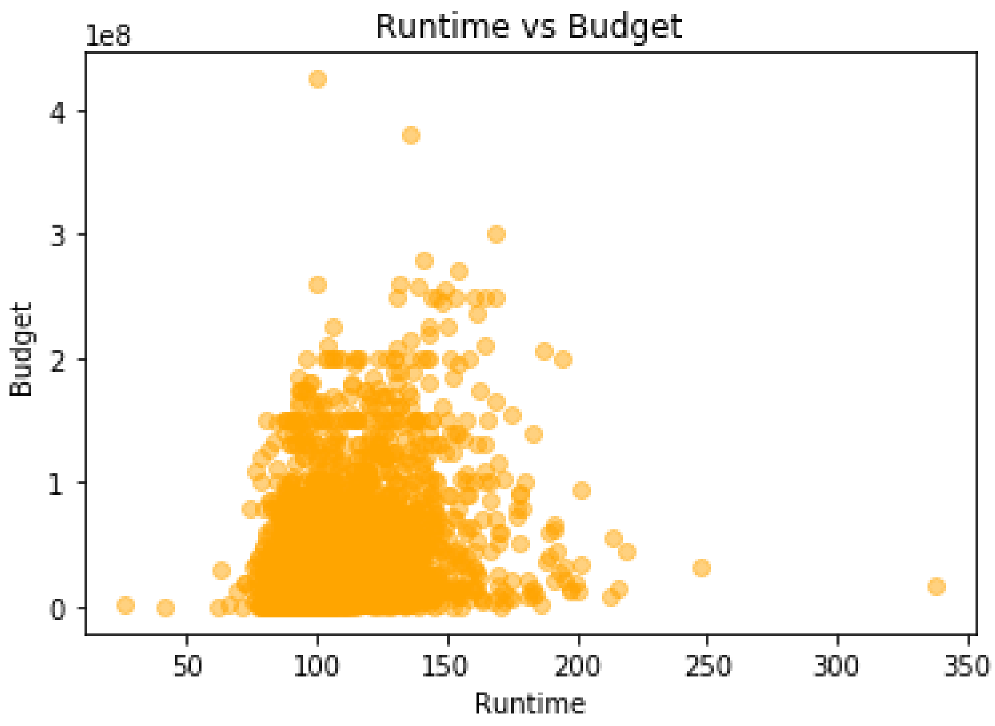
We can see that the correlation coefficient is negative. That shows the relationship between runtime and popularity weak.

```
In [47]: df['runtime'].corr(df['popularity'])

Out[47]: 0.21342246569698353
```

This correlation is 0.213, a negative correlation between runtime and popularity.
Now, I want to know What the relationship between the runtime and the budget of the movie?

```
In [48]: # Create scatter and choose color
         # Add title and axis names
         plt.scatter(x=df['runtime'], y=df['budget'],color='orange', alpha=0.5)
         plt.title('Runtime vs Budget')
         plt.xlabel('Runtime')
         plt.ylabel('Budget')
         plt.show()
```



As we can see that the sign of the correlation coefficient is negative. That means the runtime decreases as budget increases.

```
In [49]: df['runtime'].corr(df['budget'])

Out[49]: 0.26079081142563248
```

Also, This correlation between runtime and budget is a negative correlation.
## Conclusions

Finally: After I answered my questions. I came out with some facts about movies. After this analysis we can conclude the following: * The highest value of budget was 425000000 and in the profit was 2544505847. * The lowest value of the budget was zero and in profit was -413912431. * Drama movies and Comedy movies have the most popularity. * The correlation between runtime and budget is a negative correlation. * Also, the correlation between runtime and popularity is a negative correlation.

1.2 Limitations:

- In this dataset there something has hindered my analysis. At the budget and revenue column have some zero values in it. So that is erroneous and will adversely affect my overall analysis. I have removed the zero and null or missing values from the budget and revenue column, that makes the analysis more accurate.

1.3 Submitting your Project

Before you submit your project, you need to create a .html or .pdf version of this notebook in the workspace here. To do that, run the code cell below. If it worked correctly, you should get a return code of 0, and you should see the generated .html file in the workspace directory (click on the orange Jupyter icon in the upper left).

Alternatively, you can download this report as .html via the **File** > **Download as** submenu, and then manually upload it into the workspace directory by clicking on the orange Jupyter icon in the upper left, then using the Upload button.

Once you've done this, you can submit your project by clicking on the "Submit Project" button in the lower right here. This will create and submit a zip file with this .ipynb doc and the .html or .pdf version you created. Congratulations!

```
In [50]: from subprocess import call
         call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])

Out[50]: 0

In [ ]:
```