Investigate_a_Dataset

1 Project: Investigate TMDb Movie DataSet

1.1 Table of Contents

1.2 Introduction

TMDb Movie dataset from kaggle to investigate.It contains 10,000 movie data having information including user rating, budget, revenue, date of release, genres and much more information. ## The potential problem that can be observe from the dataset.

1)In which year the most number of movies release.

2)In which movie had largest and lowest budget.
3)In Which movie had most profit and loss.
4)Number of movie release every year.
5)Which movie had largest and shortest runtime.
6)Average budget of the movie.
7)Average revenue earned by the movie. 8)Average runtime of the movie using Box-Model.
9)Average duration of the movie.

[4]:
```
 # 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 pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

## Data Wrangling

To analyse the dataset and find coloumn which is neccessary to answer the proposed question and delete the unused data for easy calculation and understandable.

1.2.1 General Properties

[5]:
```
# 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')
```

[6]:
```
 df.tail(1)
```

[6]:
```
          id   imdb_id popularity budget  revenue  \
10865 22293 tt0060666   0.035919  19000        0
                original_title  \

10865 Manos: The Hands of Fate
                                                           cast homepage  \

10865 Harold P. Warren|Tom Neyman|John Reynolds|Dian…       NaN
               director                                         tagline   \

10865 Harold P. Warren It's Shocking! It's Beyond Your Imagination!
         …                                                  overview runtime  \

10865    …    A family gets lost on the road and stumbles up…      74
      genres production_companies release_date vote_count  vote_average  \

10865 Horror            Norm-Iris    11/15/66        15           1.5
      release_year    budget_adj  revenue_adj

10865         1966 127642.279154         0.0

[1 rows x 21 columns]
```

[6]:
```
 #To find number of rows and column
df.shape
```

[6]:
```
(10866, 21)
```

1.2.2 Find the basic information about the dataset

[7]:
```
 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
```

After observation of the data set , we know that total number of coloumn is 21 and total 10866 entries in the dataset.There are many null value present in the column cast,genres, production_companies, tagline,director,homepage. So removed the unused coloumn.

```
[8]: #To know overview of the dataset
     df.describe()
```

```
[8]:                 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
```

1.2.3 Data Cleaning

1.2.4 1.Removed unused column

[7]:
```
delete_column=[ 'id', 'imdb_id', 'popularity', 'budget_adj',
 'revenue_adj',␣ ↪'homepage','keywords','overview',
        'production_companies', 'vote_count', 'vote_average']
#delete the coulmn
df= df.drop(delete_column,1)
#preview after removing the column
df.head(1)
```

[7]:
```
      budget    revenue original_title  \
 0 150000000 1513528810 Jurassic World
                                                 cast        director  \

0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow

            tagline runtime                                genres \
 0 The park is open.     124  Action|Adventure|Science Fiction|Thriller
   release_date release_year

 0      6/9/15         2015
```

1.2.5 2. Removing the duplicacy in the rows(if any).

[10]:
```
# 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. row,col = df.shape
#Now
print('{} total entries of movies and {} no.of columns in
 it.'.format(row-1,␣ ↪col))
```

```
10865 total entries of movies and 10 no.of columns in it.
```

[11]:
```
 df.drop_duplicates(inplace=True)
```

1.2.6 3. Removing 0’s from budget and the revenue column

```
[8]:  df_budget = df.query('budget == 0')

df_budget.head(1)
```

```
[8]:    budget  revenue original_title  \
30       0 29355203    Mr. Holmes   
                                                  cast    director   \

30 Ian McKellen|Milo Parker|Laura Linney|Hattie M…  Bill Condon   
                    tagline runtime       genres release_date  release_year

30 The man behind the myth     103  Mystery|Drama     6/19/15         2015
```

```
[13]:  #Number of movie having budget 0
row,col = df_budget.shape
print('Number of movie having 0 budget is {}.'.format(row-1))
```

```
Number of movie having 0 budget is 5695.
```

```
[12]:  #create separate list of column
temp_list = ['budget','revenue']

# To replace all the value from '0' to NAN in the list
df[temp_list] = df[temp_list].replace(0, np.NAN)

#Removing all the row which has NAN value in the
temp_list df.dropna(subset=temp_list, inplace=True)

row ,col = df.shape
print('After removing such entries, we have only {} no. of movies.'.
 ↪format(row-1))
```

```
After removing such entries, we have only 3854 no. of movies.
```

```
[13]:  #Change the date fromat of the release_date
df.release_date = pd.to_datetime(df['release_date'])
```

```
[14]:  #Check the new format of release_date
df.head(1)
```

```
[14]:        budget      revenue original_title  \
0 150000000.0 1.513529e+09 Jurassic World   
                                                 cast         director  \

0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi… Colin Trevorrow   
```

```
            tagline runtime                               genres \
0 The park is open.     124  Action|Adventure|Science Fiction|Thriller
  release_date release_year

0  2015-06-09         2015
```

[15]:
```
 #replace 0 with NAN of runtime column in the dataset
df['runtime']=df['runtime'].replace(0,
np.NAN) #check the current format of
column df.dtypes
```

[15]:
```
 budget                   float64
revenue                  float64
original_title            object
cast                      object
director                  object
tagline                   object
runtime                    int64
genres                    object
release_date   datetime64[ns]
release_year               int64
dtype: object
```

[16]:
```
 #change the datatype of budget and revenue
change = ['budget','revenue']
df[change]=df[change].applymap(np.int64)
#check the new format of column
df.dtypes
```

[16]:
```
 budget                      int64
revenue                     int64
original_title             object
cast                       object
director                   object
tagline                    object
runtime                     int64
genres                     object
release_date    datetime64[ns]
release_year                int64
dtype: object
```

[17]:
```
 df.head(1)
```

[17]:
```
      budget    revenue original_title  \

0  150000000  1513528810 Jurassic World
```

```
                                                    cast          director \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…   Colin Trevorrow

           tagline runtime                                       genres \
 0 The park is open.      124  Action|Adventure|Science Fiction|Thriller
  release_date release_year

 0  2015-06-09         2015
```

New dataset having date change.   ## Exploratory Data Analysis

> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. Compute statistics and create visualizations with the goal of addressing the research questions that you posed in the Introduction section. It is recommended that you be systematic with your approach. Look at one variable at a time, and then follow it up by looking at relationships between variables.

1.2.7 Profit of Each Movie

[18]:
```
 #In Which movie had most profit and loss
df.insert(2,'profit',df['revenue']-df['budget'])
df.head()
```

[18]:
```
      budget     revenue      profit             original_title  \
 0 150000000 1513528810 1363528810             Jurassic World
 1 150000000  378436354  228436354         Mad Max: Fury Road
 2 110000000  295238201  185238201                  Insurgent
 3 200000000 2068178225 1868178225 Star Wars: The Force Awakens
 4 190000000 1506249360 1316249360                  Furious 7

                                                    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
                             tagline runtime   \

 0           The park is open.      124
 1          What a Lovely Day.      120
 2  One Choice Can Destroy You      119
 3 Every generation has a story.    136
 4         Vengeance Hits Home      137
                                      genres release_date release_year

 0 Action|Adventure|Science Fiction|Thriller   2015-06-09         2015
```

```
1  Action|Adventure|Science Fiction|Thriller  2015-05-13        2015
2    Adventure|Science Fiction|Thriller  2015-03-18        2015
3  Action|Adventure|Science Fiction|Fantasy  2015-12-15        2015
4                   Action|Crime|Thriller  2015-04-01        2015
```

[19]:

```
 import pprint
# define the function to calculate each of the research
question def calculate(column):
    # High earn profit

    high = df[column].idxmax()
    high_detail = pd.DataFrame(df.loc[high])

    # Low earn profit

    low = df[column].idxmin()
    low_detail = pd.DataFrame(df.loc[low])

    #collect data at one place
    info = pd.concat([high_detail,low_detail],axis = 1)
    return info
#call the function to get result
calculate('profit')
```

[19]:

```
                                                          1386  \
budget                                                237000000
revenue                                              2781505847
profit                                               2544505847
original_title                                           Avatar
cast           Sam Worthington|Zoe Saldana|Sigourney Weaver|S…
director                                          James Cameron
tagline                             Enter the World of Pandora.
runtime                                                     162
genres                    Action|Adventure|Fantasy|Science Fiction
release_date                                2009-12-10 00:00:00
release_year                                               2009

                                                           2244
budget                                                425000000
revenue                                                11087569
profit                                               -413912431
original_title                                 The Warrior's Way
cast           Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann…
director                                             Sngmoo Lee
tagline                                Assassin. Hero. Legend.
runtime                                                     100
genres                  Adventure|Fantasy|Action|Western|Thriller
```

```
release_date                                 2010-12-02 00:00:00
release_year                                                2010
```

Avatar is the highest earned profit i.e 2544505847 .

The Warrior’s Way is the lowest earned profit i.e -413912431.

[20]:
```
 #In which movie had largest and lowest budget.
calculate('budget')
```

[20]:
```
                                                           2244 \
budget                                                425000000
revenue                                                11087569
profit                                               -413912431
original_title                               The Warrior's Way
cast           Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann…
director                                            Sngmoo Lee
tagline                               Assassin. Hero. Legend.
runtime                                                    100
genres            Adventure|Fantasy|Action|Western|Thriller
release_date                               2010-12-02 00:00:00
release_year                                              2010
                                                          2618

budget                                                       1
revenue                                                    100
profit                                                      99
original_title                                    Lost & Found
cast           David Spade|Sophie Marceau|Ever Carradine|Step…
director                                          Jeff Pollack
tagline        A comedy about a guy who would do anything to …
runtime                                                     95
genres                                         Comedy|Romance
release_date                               1999-04-23 00:00:00
release_year                                              1999
```

```
The Warrior's Way is the largest budget i.e 425000000 dollar.
Lost and Found is the smallest budget i.e 1 dollar
```

[21]:
```
#In which movie had most and least earned
revenue calculate('revenue')
```

[21]:
```
                                                           1386 \
budget                                                237000000
revenue                                              2781505847
profit                                               2544505847
original_title                                          Avatar
cast           Sam Worthington|Zoe Saldana|Sigourney Weaver|S…
```

```
director                                    James Cameron
tagline                       Enter the World of Pandora.
runtime                                               162
genres           Action|Adventure|Fantasy|Science Fiction
release_date                          2009-12-10 00:00:00
release_year                                         2009
                                                     5067

budget                                            6000000
revenue                                                 2
profit                                           -5999998
original_title                            Shattered Glass
cast           Hayden Christensen|Peter Sarsgaard|ChloÃ« Sevi…
director                                        Billy Ray
tagline                                               NaN
runtime                                                94
genres                                      Drama|History
release_date                          2003-11-14 00:00:00
release_year                                         2003
```

Avatar is the largest revenue earned i.e 2781505847 dollar.

Shattered Glass is the smallest revenue earned i.e 2 dollar

[22]:
```
# Use this, and more code cells, to explore your data. Don't forget to add
#   Markdown cells to document your observations and findings.
```

[23]:
```
#Movie which had shortest and longest
runtime calculate('runtime')
```

[23]:
```
                                                      2107 \
budget                                         18000000
revenue                                          871279
profit                                        -17128721
original_title                                   Carlos
cast           Edgar RamÃ rez|Alexander Scheer|Fadi Abi Samra…
director                                Olivier Assayas
tagline                  The man who hijacked the world
runtime                                             338
genres                     Crime|Drama|Thriller|History
release_date                        2010-05-19 00:00:00
release_year                                       2010
                                                      5162

budget                                               10
revenue                                               5
profit                                               -5
original_title                              Kid's Story
```

```
cast          Clayton Watson|Keanu Reeves|Carrie-Anne Moss|K…
director                             Shinichiro Watanabe
tagline                                              NaN
runtime                                               15
genres                         Science Fiction|Animation
release_date                         2003-06-02 00:00:00
release_year                                        2003
```

Carlos is the longest runtime i.e 338 minutes.

Kid's Story is the shortest runtime i.e 15 minutes

[24]:
```python
#Calculate average of the column
def avg_fun(column):
    return df[column].mean()
```

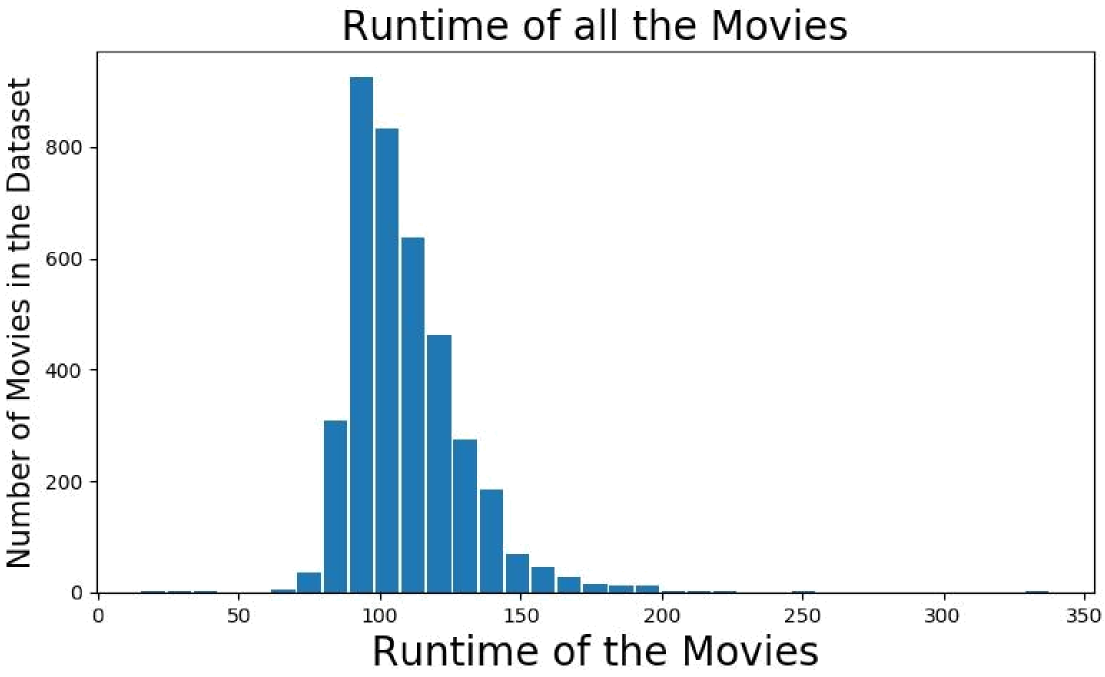
[25]:
```python
#calling average of the function
avg_fun('runtime')
```

[25]:
```
109.21582360570687
```

The average runtime a movie is 109 minutes.

1.2.8 Plotting histogram of runtime of movie

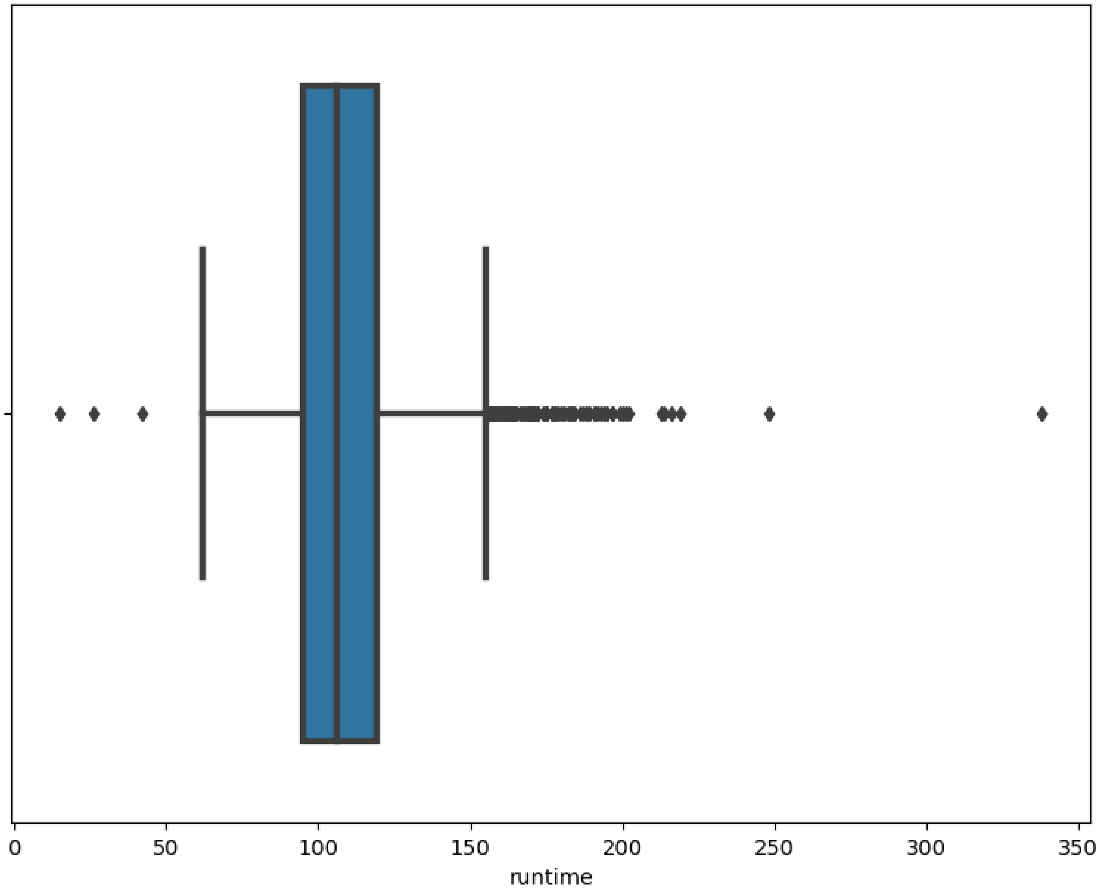
[26]:
```python
 # Continue to explore the data to address your additional research
#   questions. Add more headers as needed if you have more questions to
#   investigate.

#plotting a histogram of runtime of movies

#giving the figure size(width, height)
plt.figure(figsize=(9,5), dpi = 100)

#On x-axis
plt.xlabel('Runtime of the Movies', fontsize = 20)
#On y-axis
plt.ylabel('Number of Movies in the Dataset', fontsize=15)
#Name of the graph
plt.title('Runtime of all the Movies', fontsize=20)

#giving a histogram plot
plt.hist(df['runtime'], rwidth = 0.9, bins =35)
#displays the plot
plt.show()
```



The above formed graph is positively skewed and most of the movies have average time is between the 75 to 120

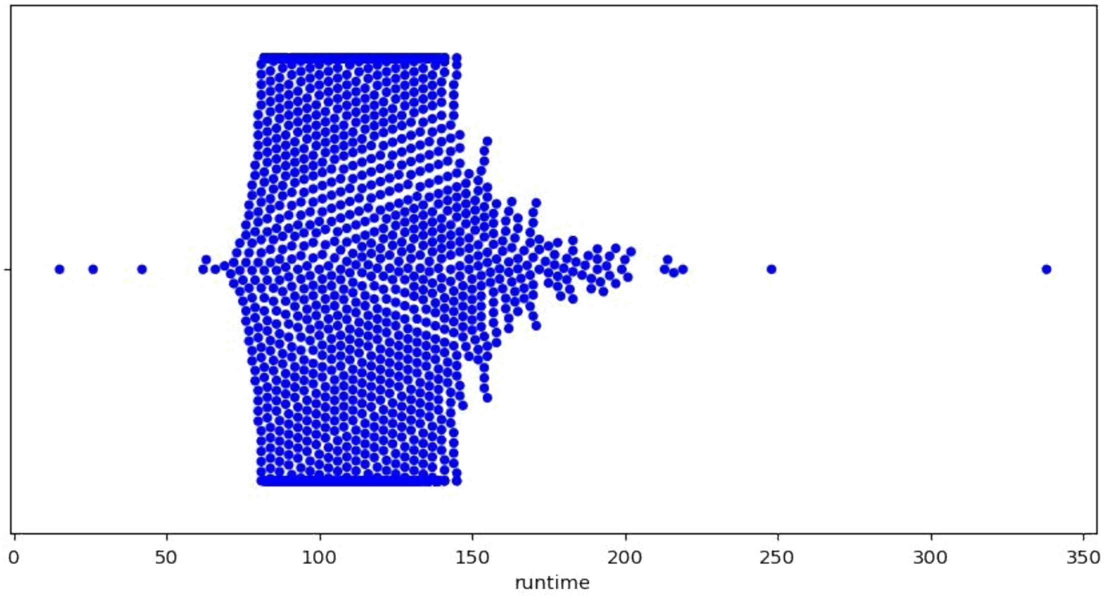
1.2.9 Lets analyse the data using box model

```
[27]:  import seaborn as sns
#The First plot is box plot of the runtime of
the movies plt.figure(figsize=(9,7), dpi = 105)

#using seaborn to generate the boxplot
sns.boxplot(df['runtime'], linewidth = 3)
#diplaying the plot
plt.show()
```



[28]:

```
#The Second plots is the data points plot of runtime of movies

plt.figure(figsize=(10,5), dpi = 105)
#using seaborn to generate the plot
sns.swarmplot(df['runtime'], color = 'Blue')
#displaying the plot
plt.show()
```

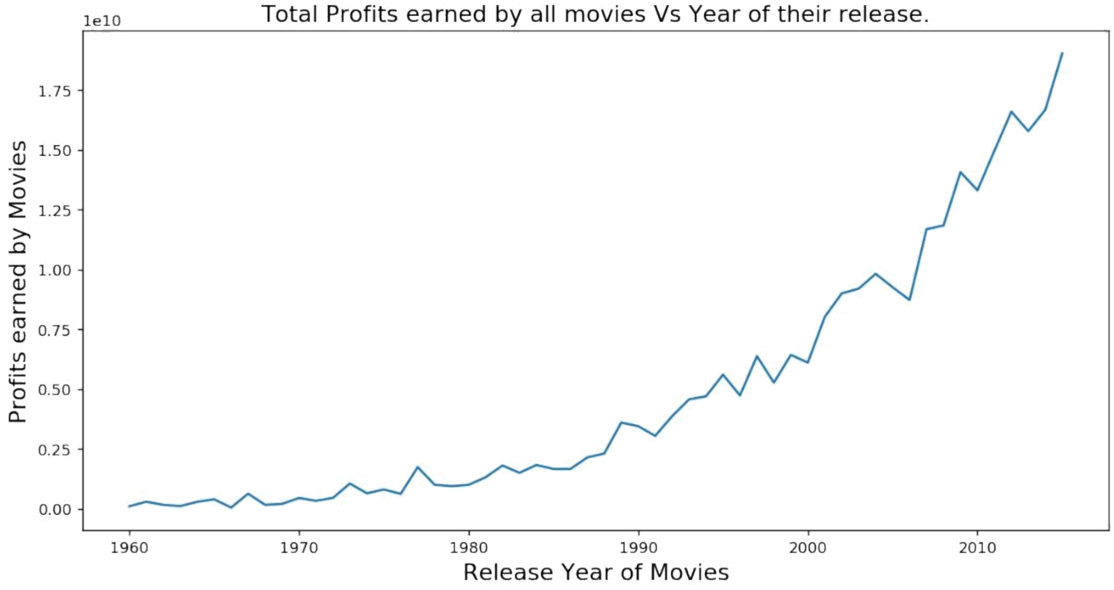


By looking at both the plot and calculations, we can conclude that..

25% of movies have a runtime of less than 95 minutes. 50% of movies have a runtime of less than 109 minutes. 75% of movies have a runtime of less than 119 minutes.

[29]:

```
 profits_year = df.groupby('release_year')['profit'].sum()

#figure size(width, height)
plt.figure(figsize=(12,6), dpi = 130)

#on x-axis
plt.xlabel('Release Year of Movies', fontsize = 15)
#on y-axis
plt.ylabel('Profits earned by Movies', fontsize = 15)
#title of the line plot
plt.title('Total Profits earned by all movies Vs Year of their
 release.',␣ ↪fontsize= 15)

#plotting the graph
plt.plot(profits_year)

#displaying the line plot
plt.show()
```



[30]:

```
 #To find that which year made the highest profit?
profits_year.idxmax()
```

[30]:

```
2015
```

So after visualisation it on the graph we get the 2015 is the year when it get most profit

[31]:

```
 #selecting the movies having profit $60M or more
profit_data = df[df['profit'] >= 60000000]

#reindexing new data
profit_data.index = range(len(profit_data))

#we will start from 1 instead of 0
profit_data.index = profit_data.index + 1

#printing the changed dataset
profit_data.head(1)
```

[31]:

```
      budget    revenue     profit original_title  \
 1 150000000 1513528810 1363528810 Jurassic World
                                                cast      director   \

1  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi…  Colin Trevorrow

          tagline runtime                                      genres   \
1  The park is open.   124  Action|Adventure|Science Fiction|Thriller
```

```
   release_date release_year
1  2015-06-09        2015
```

```
[32]: #counting the no.of rows in the new
      data base len(profit_data)
```

```
[32]: 1197
```

So number of movies havw profit greater than $60M is 1197.

1.2.10 Successful genres

```
[106]: #function which will take any column as argument from and keep
       its track def data(column):
           #will take a column, and separate the string by '|'
           data = profit_data[column].str.cat(sep = '|')

           #giving pandas series and storing the values
           separately data = pd.Series(data.split('|'))

           #arranging in descending order
           count = data.value_counts(ascending = False)

           return count
```

```
[ ]:
```

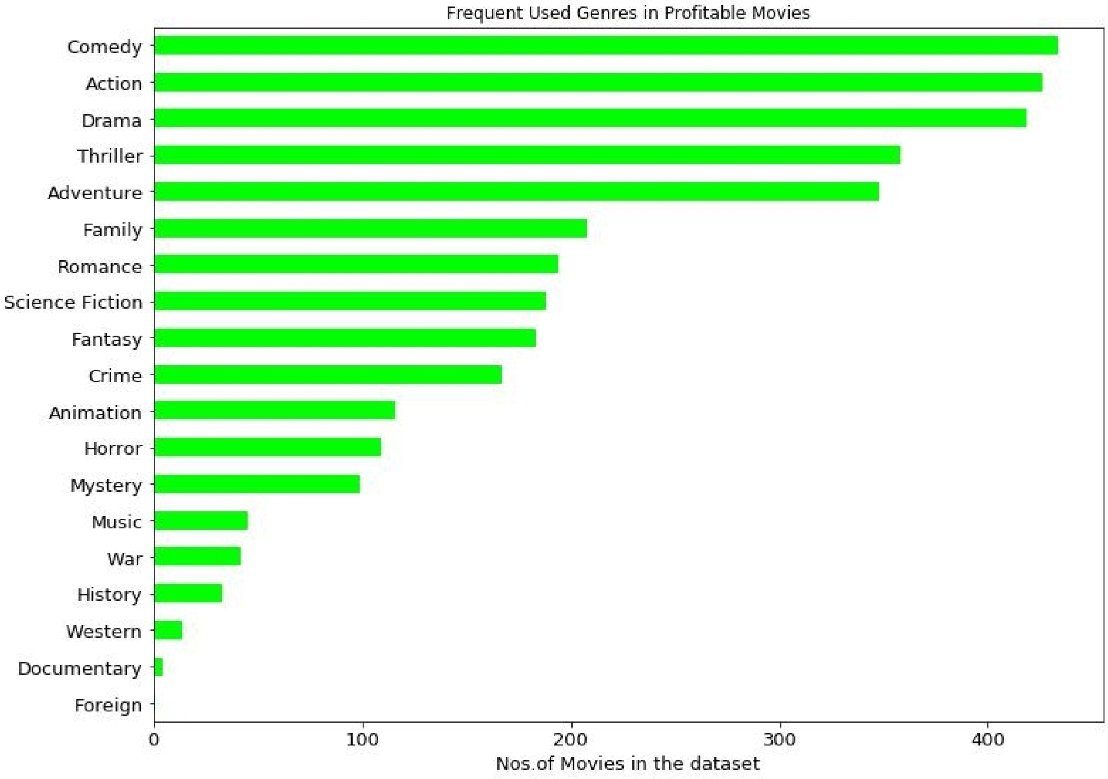
```
[107]:   #variable to store the retured value
       count = data('genres')
       #printing top 5 values
       count.head()
```

```
[107]: Comedy     434
       Action     426
       Drama      419
       Thriller   358
       Adventure  348
       dtype: int64
```

Graphical analysis of the collected data

```
[108]:  #lets plot the points in descending order top to bottom as we have
         data in same␣ ↪format.
       count.sort_values(ascending = True, inplace = True)

       #ploting
```

```
lt = count.plot.barh(color = '#00FF00', fontsize = 13)

#title
lt.set(title = 'Frequent Used Genres in Profitable Movies')

# on x axis
lt.set_xlabel('Nos.of Movies in the dataset', color = 'black', fontsize = '13')

#figure size(width, height)
lt.figure.set_size_inches(12, 9)

#ploting the graph
plt.show()
```



1.2.11 Most number of cast

```
[109]:  #variable to store the retured value
count = data('cast')
#printing top 5 values
count.head()
```

[109]:
```
 Tom Cruise          26
Brad Pitt           22
Tom Hanks           22
Sylvester Stallone  21
Cameron Diaz        20
dtype: int64
```

Now the most number of movies by Tom Cruise i.e 26 and after that Brad Pitt 22 and Tom Hanks 22.

1.2.12 Average budget of the movies

[110]:
```
  #New function to find average
def profit_avg(column):
    return profit_data[column].mean()
```

[111]:
```
# calling the above function for
budget profit_avg('budget')
```

[111]:
```
 63757867.395154551
```

The movies having profit of more than 50 million dollar have an average budget of 60 million dollar.

[112]:
```
# calling the above function for
revenue profit_avg('revenue')
```

[112]:
```
 274739298.8086884
```

The movies having profit of more than 50 million dollar have an average revenue of 255 million dollar.

[113]:
```
  # calling the above function for
profit_avg('runtime')
```

[113]:
```
 114.06850459482038
```

The movies having profit of more than 50 million dollar have an average duration of 113 minutes.

## Conclusions

In this data analysis we observe the following.

1)Average budget,revenue, profit of the movies is .
2)Average Duration of the movie is 114 min.
3)Which type of cast is most famous among the people.
4)Most genres of movie is Adventure,action, thriller,Drama, comedy.