Project: Investigate a Dataset

Table of Contents

Introduction

> This is a project to invesigate a dataset from the International Movie Database (IMDB). This will entail three phases: wrangling which involves cleaning the data, Exploratory Data Analysis to observe trends in the data and Reporting the observation. This will be done using popular python libraies like NumPy, Pandas and Matplotlib. We will answer the following research questions:
> - What movies have the highest and lowest profits
> - What movies have the highest and lowest revenues
> - What movies have the highest and lowest budgets
> - What movies have the highest and least runtimes
> - What are the top 5 actors
> - What are the top 5 movie genres

In [2]:

```
# Use this cell to set up import statements for all of the packages that you
#   plan to use.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import operator

# 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
```

Data Wrangling

> In this section of the report, data is checked for cleanliness, and then further cleaned for analysis.

In [3]:

```
# Load your data and print out a few lines. Perform operations to inspect data
df = pd.read_csv("tmdb-movies.csv")
#   types and look for instances of missing or possibly errant data.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    10866 non-null  int64  
 1   imdb_id               10856 non-null  object 
 2   popularity            10866 non-null  float64
 3   budget                10866 non-null  int64  
 4   revenue               10866 non-null  int64  
 5   original_title        10866 non-null  object 
 6   cast                  10790 non-null  object 
 7   homepage              2936 non-null   object 
 8   director              10822 non-null  object 
 9   tagline               8042 non-null   object 
 10  keywords              9373 non-null   object 
 11  overview              10862 non-null  object 
 12  runtime               10866 non-null  int64  
 13  genres                10843 non-null  object 
 14  production_companies  9836 non-null   object 
 15  release_date          10866 non-null  object 
 16  vote_count            10866 non-null  int64  
 17  vote_average          10866 non-null  float64
 18  release_year          10866 non-null  int64  
 19  budget_adj            10866 non-null  float64
 20  revenue_adj           10866 non-null  float64
dtypes: float64(4), int64(6), object(11)
memory usage: 1.7+ MB
```

Data Cleaning

Step 1: Remove undesired columns

In [4]:

```
# 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.
#During this phase, we shall be performing the following steps.

df = df.drop([ 'id', 'imdb_id', 'popularity', 'budget_adj', 'revenue_adj', 'home
page', 'keywords', 'overview', 'production_companies', 'vote_count', 'vote_avera
ge'],1)
```

In [5]:

```
df.head(2)
```

Out[5]:

	budget	revenue	original_title	cast	director	tagline	runtime	
0	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	Colin Trevorrow	The park is open.	124	Action|Adve
1	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	George Miller	What a Lovely Day.	120	Action|Adve

Step 2: Remove duplicate records

In [6]:

```
#Removing duplicate records from the dataset
df.drop_duplicates(inplace=True)
df.shape
```

Out[6]:

```
(10865, 10)
```

Step 3: Remove null values from desired columns

In [7]:

```
df = df[df['cast'].isnull() == False]
df = df[df['genres'].isnull() == False]
```

Step 4: Remove zero values from revenue and budget

In [8]:

```
df[['budget', 'revenue']] = df[['budget', 'revenue']].replace(0, np.NAN)
```

In [8]:

```
#### Step 5: Remove rows with NaN
```

In [9]:

```
df.dropna(inplace = True)
```

In [10]:

```
#Exploring the dataframe, the budget and the revenue columns contain zeros, this
is undesired so we choose
#a strategy of replacing it with the mean
#getting the mean
```

Exploratory Data Analysis

> Here we will compute statistics and create visualizations with the goal of addressing each research question. We will look at one variable at a time, and then further follow it up by looking at relationships between variables.

Summary of Movies with the highest and lowest profit

In [10]:

```
# Insert a new column to the dataframe called profit
df.insert(2,'profit',df['revenue']-df['budget'])
```

In [12]:

```
#implementing the profit calculator
def range_calculator(col):
   higehest_profit_info=pd.DataFrame(df.loc[df[col].idxmax()])
   lowest_profit_info=pd.DataFrame(df.loc[df[col].idxmin()])
   info=pd.concat([higehest_profit_info, lowest_profit_info], axis=1)
   return info
range_calculator('profit')
```

Out[12]:

	1386	2244
budget	2.37e+08	4.25e+08
revenue	2.78151e+09	1.10876e+07
profit	2.54451e+09	-4.13912e+08
original_title	Avatar	The Warrior's Way
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...
director	James Cameron	Sngmoo Lee
tagline	Enter the World of Pandora.	Assassin. Hero. Legend.
runtime	162	100
genres	Action|Adventure|Fantasy|Science Fiction	Adventure|Fantasy|Action|Western|Thriller
release_date	12/10/09	12/2/10
release_year	2009	2010

Summary of Movies with the Highest and Lowest Budget

In [13]:

```
# Calculate summary of movies with the highest and lowest budget using range_calculator
range_calculator('budget')
```

Out[13]:

	2244	2618
budget	4.25e+08	1
revenue	1.10876e+07	100
profit	-4.13912e+08	99
original_title	The Warrior's Way	Lost & Found
cast	Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Dann...	David Spade|Sophie Marceau|Ever Carradine|Step...
director	Sngmoo Lee	Jeff Pollack
tagline	Assassin. Hero. Legend.	A comedy about a guy who would do anything to ...
runtime	100	95
genres	Adventure|Fantasy|Action|Western|Thriller	Comedy|Romance
release_date	12/2/10	4/23/99
release_year	2010	1999

Summary of Movies with the Highest and Lowest Revenue

In [14]:

```
# Calculate summary of movies with the highest and lowest revenue using range_calculator
range_calculator('revenue')
```

Out[14]:

	1386	8142
budget	2.37e+08	6e+06
revenue	2.78151e+09	2
profit	2.54451e+09	-6e+06
original_title	Avatar	Mallrats
cast	Sam Worthington|Zoe Saldana|Sigourney Weaver|S...	Jason Lee|Jeremy London|Shannen Doherty|Claire...
director	James Cameron	Kevin Smith
tagline	Enter the World of Pandora.	They're not there to shop. They're not there t...
runtime	162	94
genres	Action|Adventure|Fantasy|Science Fiction	Romance|Comedy
release_date	12/10/09	10/20/95
release_year	2009	1995

Summary of Movies with the Highest and Lowest Runtimes

In [15]:

```
# Calculate summary of movies with the highest and lowest revenue using range_calculator
range_calculator('runtime')
```

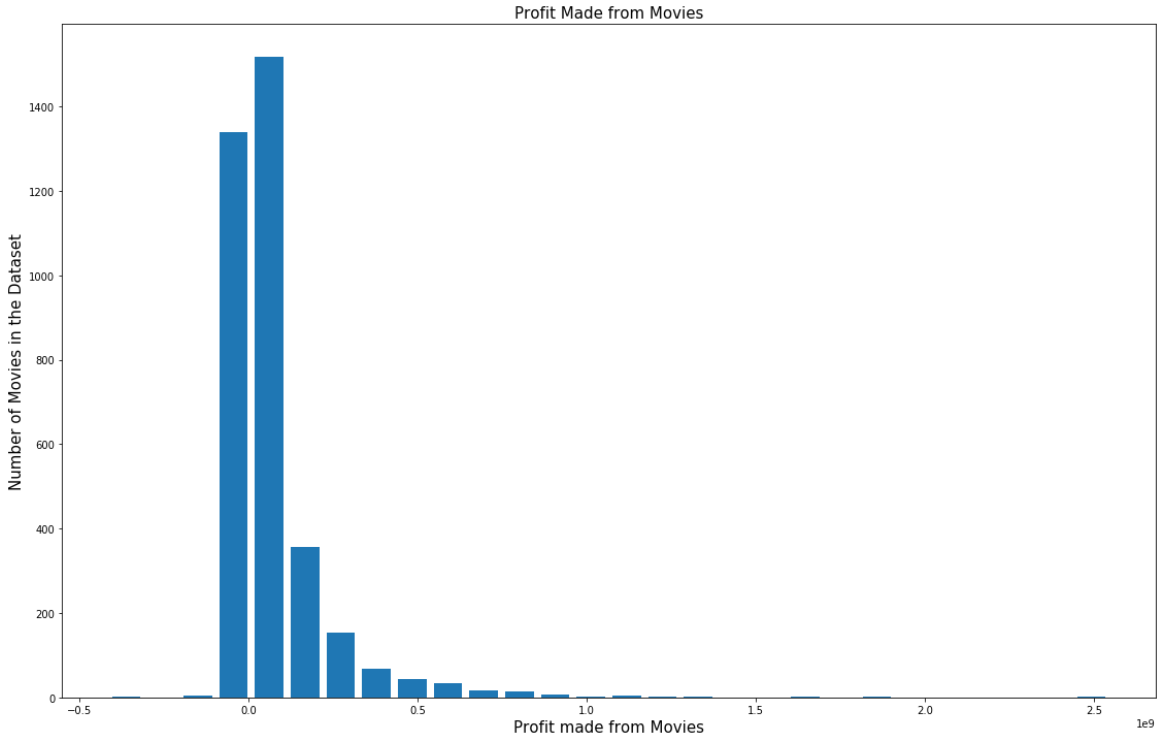
Out[15]:

	2107	8005
budget	1.8e+07	3e+06
revenue	871279	2.1e+07
profit	-1.71287e+07	1.8e+07
original_title	Carlos	Mickey's Christmas Carol
cast	Edgar RamÃrez|Alexander Scheer|Fadi Abi Samra...	Alan Young|Wayne Allwine|Clarence Nash|Hal Smi...
director	Olivier Assayas	Burny Mattinson
tagline	The man who hijacked the world	He's back! Mickey Mouse - in his first new mot...
runtime	338	26
genres	Crime|Drama|Thriller|History	Family|Animation
release_date	5/19/10	10/19/83
release_year	2010	1983

Plotting a histogram of runtime of movies

In [16]:

```
#giving the figure size(width, height)
plt.figure(figsize=(19,12))
plt.xlabel('Profit made from Movies', fontsize = 15)
plt.ylabel('Number of Movies in the Dataset', fontsize=15)
plt.title('Profit Made from Movies', fontsize=15)
plt.hist(df['profit'], rwidth = 0.8, bins =28)
plt.show()
```



Top 5 Movies Genres

In [17]:

```
#A functions that strips individual items based on the "|"
#and then creates a dictionary of items to produce top 5 items from the dictiona
ry

def calculate_top_5(column):
    list_stuffs = {}

    stuffs = df[column]
    stuffs = stuffs.str.split("|")
    stuffs = np.array(stuffs)
    for itemList in stuffs:
        for stuff in itemList:
                stuff = stuff.lstrip() #trim the whitespaces
                if stuff not in list_stuffs:
                    list_stuffs[stuff] = 1
                else:
                    list_stuffs[stuff] += 1



    sorted_list_stuffs = sorted(list_stuffs.items(), key = operator.itemgetter(1
), reverse = True)
    return sorted_list_stuffs[0:5]
```

In [18]:

```
calculate_top_5('genres')
```

Out[18]:

```
[('Drama', 1572),
 ('Comedy', 1255),
 ('Thriller', 1149),
 ('Action', 1046),
 ('Adventure', 720)]
```

Top 5 Actors that Appear Most

In [19]:

```
calculate_top_5('cast')
```

Out[19]:

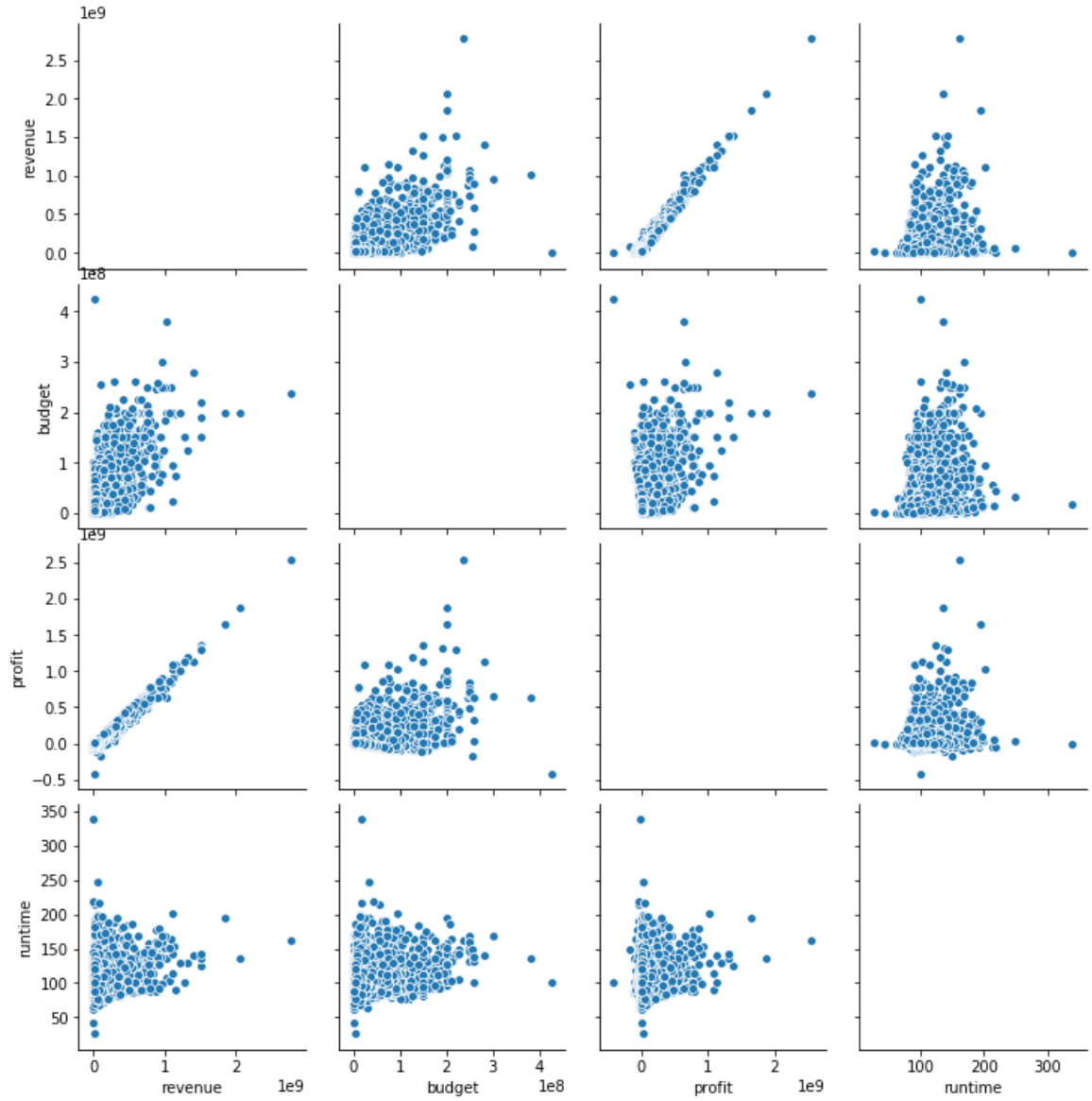
```
[('Robert De Niro', 50),
 ('Bruce Willis', 44),
 ('Samuel L. Jackson', 43),
 ('Nicolas Cage', 41),
 ('Matt Damon', 35)]
```

Correlation Between Datasets using a Pair plot

In [20]:

```
# Numeric columns from dataframe
pair_plot_df = df[['revenue', 'budget', 'profit', 'runtime']]

pair_plot = sns.pairplot(pair_plot_df, diag_kind="reg")
```



Conclusions

From the analysis we were able to clean a movie Dataset with so much details that gave a lot of insights about the movies on the IMDB. We highlighted some research questions in the introduction section of this project and answered them based on the dataset in the EDA section and depicted them using appropriate charts.