Investigate_a_Dataset

May 25, 2022

> **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. Once you complete this project, remove these **Tip** sections from your report before submission. 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:movies

1.1 Table of Contents

## Introduction

1.1.1 Dataset Description

2 This data set contains information about 10,000 movies collected from The Movie Database (TMDb), including user ratings and revenue. but if movie had a high vote and it's bad? that's mean that there are many factors to evaluate the dataset'

2.0.1 Question(s) for Analysis

the number of appearances per actor? how many movies had spread at that date? movies with higher votes count received a more ratings?

```
In [3]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        % matplotlib inline
```

```
In [26]: # Upgrade pandas to use dataframe.explode() function.
         !pip3 install --upgrade pandas
```

```
Requirement already up-to-date: pandas in /opt/conda/lib/python3.6/site-packages (1.1.5)
Requirement already satisfied, skipping upgrade: numpy>=1.15.4 in /opt/conda/lib/python3.6/site-
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in /opt/conda/lib/python3.6/site-p
Requirement already satisfied, skipping upgrade: python-dateutil>=2.7.3 in /opt/conda/lib/python
Requirement already satisfied, skipping upgrade: six>=1.5 in /opt/conda/lib/python3.6/site-packa
```

## Data Wrangling

2.0.2 General Properties

```
In [4]: df=pd.read_csv('tmdb-movies.csv')
        df.head()
```

```
Out[4]:        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]
```

2.0.3 Data Cleaning

in this section we print shape of data then we describe it and chek nulls, if there we drp it After discussing the structure of the data and any problems that need to be cleaned

```
In [5]: print(list(df.columns.values))
        print(df.shape)
        #describe
        print(df.describe())
        print(df.info())
        #check null
        data = df[df["cast"].isnull() == False]
        data = df[df["genres"].isnull() == False]
        #drop nulls
        df.dropna(axis=0, inplace=True)
```

```
['id', 'imdb_id', 'popularity', 'budget', 'revenue', 'original_title', 'cast', 'homepage', 'dire
(10866, 21)
```

```
                 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
<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
None
```

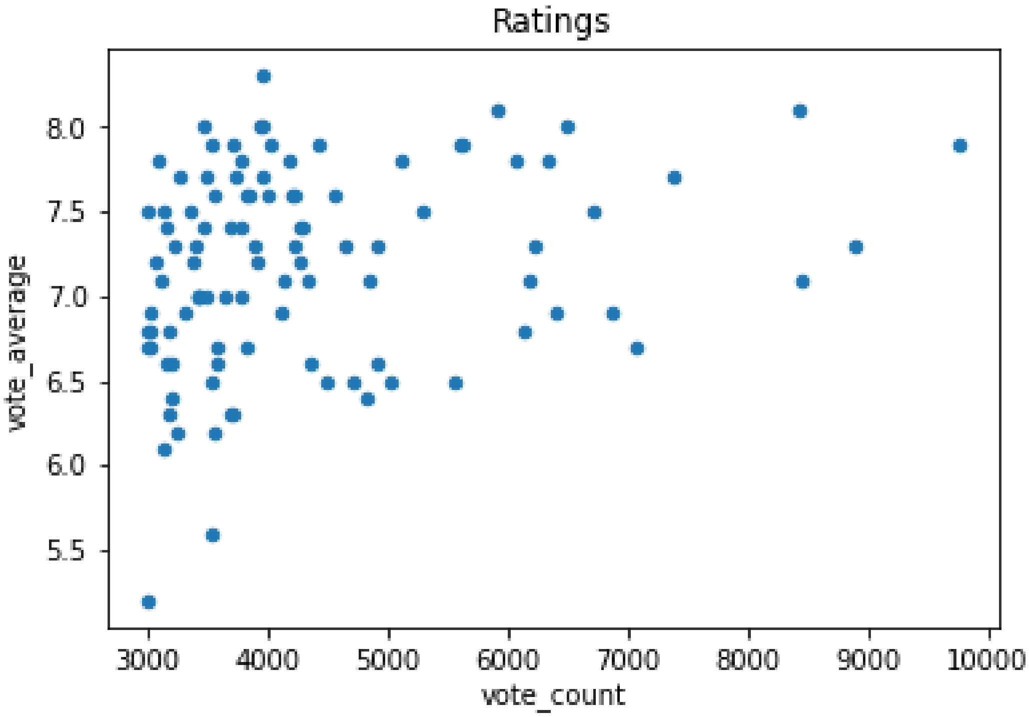
## Exploratory Data Analysis

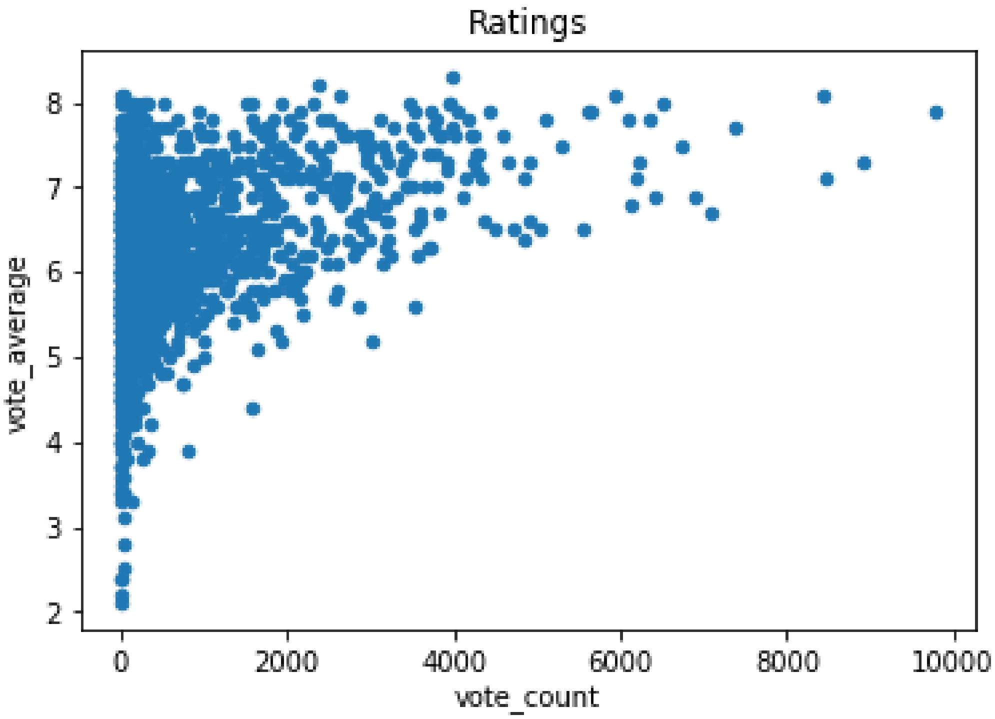
> **Tip**: Now that you've trimmed and cleaned your data, you're ready to move on to exploration. **Compute statistics** and **create visualizations** with the goal of addressing the research questions that you posed in the Introduction section. You should compute the relevant statistics throughout the analysis when an inference is made about the data. Note that at least two or more kinds of plots should be created as part of the exploration, and you must compare and show trends in the varied visualizations.

> **Tip**: - Investigate the stated question(s) from multiple angles. 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. You should explore at least three variables in relation to the primary question. This can be an exploratory relationship between three variables of interest, or looking at how two independent variables relate to a single dependent variable of interest. Lastly, you should perform both single-variable (1d) and multiple-variable (2d) explorations.

2.0.4 Research Question 1 #movies with higher votes count received a more ratings?

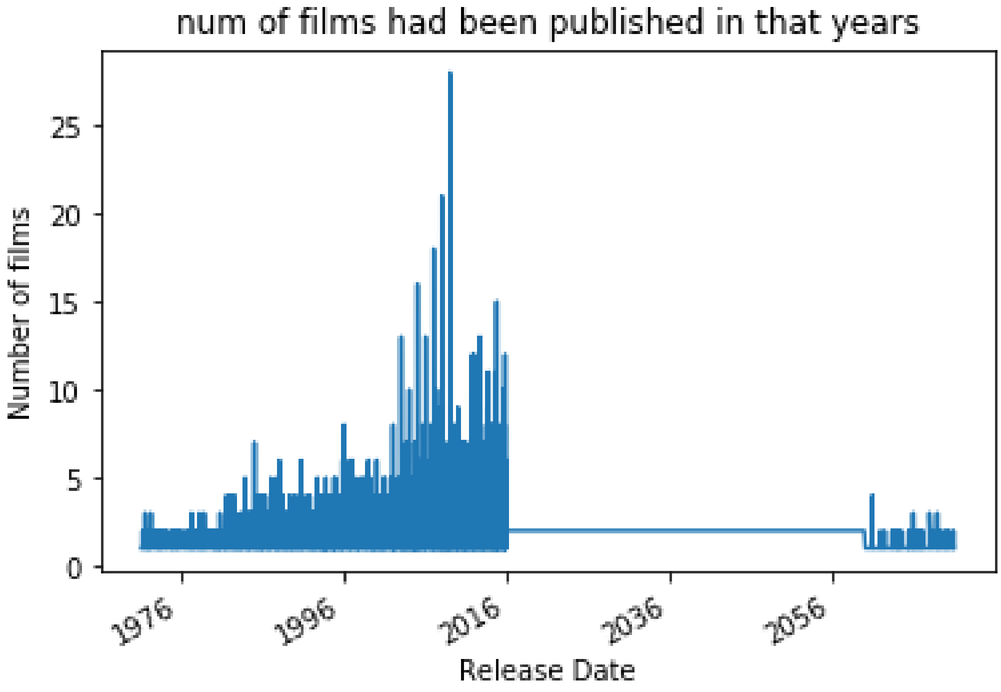
```
In [11]: df_ratings = df.loc[:, 'vote_count' : 'vote_average']
         df_vote = df_ratings[df_ratings['vote_count'] > 3000]
         df_vote.plot(x='vote_count', y='vote_average',title='Ratings', kind='scatter');
         df_ratings.plot(x='vote_count', y='vote_average',title='Ratings', kind='scatter');
```





in this 2 graphs we see that the count of votes often be less than 3000 ### Research Question 2 # how many movies had spread at that date?

```
In [7]: df['release_date']=pd.to_datetime(df['release_date'])
        show_count=df.groupby('release_date')[['id']].count()
        show_count['id'].plot()
        plt.ylabel('Number of films')
        plt.title('num of films had been published in that years')
        plt.xlabel('Release Date');
```



in general number of films in these years should be more than the previous years and the plot co ## Conclusions

> **Tip**: Finally, summarize your findings and the results that have been performed in relation to the question(s) provided at the beginning of the analysis. Summarize the results accurately, and point out where additional research can be done or where additional information could be useful.

This data is very rich in information, but it contained a set of obstacles, such as empty values that asked me to remove them, and also some films differ in the audience rating and arrangement, so I had to take into account that, but as all the data are useful and expressive

the second question shows that more films has been spreaded these years

> **Tip**: If you haven't done any statistical tests, do not imply any statistical conclusions. And make sure you avoid implying causation from correlation!

2.0.5 Limitations

All results are based on data, and since we do not know the validity of the data or if it is outdated, we must consider the data as indicators ## Submitting your Project

> **Tip**: 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).

Tip: 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.

Tip: 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 [8]: from subprocess import call
        call(['python', '-m', 'nbconvert', 'Investigate_a_Dataset.ipynb'])

Out[8]: 0

In [27]: actor_apperance={}
         actors=df['cast'].str.split("|")
         actors=np.array(actors)
         print(actors)

[list(['Chris Pratt', 'Bryce Dallas Howard', 'Irrfan Khan', "Vincent D'Onofrio", 'Nick Robinson'
 list(['Tom Hardy', 'Charlize Theron', 'Hugh Keays-Byrne', 'Nicholas Hoult', 'Josh Helman'])
 list(['Shailene Woodley', 'Theo James', 'Kate Winslet', 'Ansel Elgort', 'Miles Teller'])
 ...
 list(['John Belushi', 'Tim Matheson', 'John Vernon', 'Verna Bloom', 'Tom Hulce'])
 list(['Robbie Robertson', 'Rick Danko', 'Levon Helm', 'Richard Manuel', 'Garth Hudson'])
 list(['Burt Reynolds', 'Robert Klein', 'Adam West', 'Jan-Michael Vincent', 'Sally Field'])]


In [ ]:
```