Investigate_a_Dataset

July 5, 2022

1 Project: Investigate a Dataset - [TMDB]

1.1 Table of Contents

## Introduction This is a data analysis project for WGU C749 course.

1.1.1 Dataset Description

The data that we're going to use comes from TMDB 5000 Movie Dataset. The following is a list of all the columns found in the dataset:

- id
- imdb_id
- popularity - *Measure of a movie's popularity.*
- budget
- revenue
- original_title - *Movie title. We need this is easily identify the movie*
- cast- *list of cast members delimited by '|'*
- homepage
- director
- tagline
- keywords
- overview
- runtime - *duration of the movie*
- genres- *list of genres delimited by '|'*
- production_companies
- release_date
- vote_count
- vote_average
- release_year
- budget_adj - *the values here are adjusted for inflation*
- revenue_adj - *the values here are adjusted for inflation*

1.1.2 Question(s) for Analysis

1. What are the most popular movies?
2. Which genres are most popular from?
3. What movies have the highest budgets?
4. What movies have the highest revenue?
5. What movies are most profitable?
6. Describe the sweetspot for the runtime feature.
7. Are any of the features correlated?

Let's begin.

On this cell below, we're importing the packages/libraries that we will need for the project.

[1]:
```python
import pandas as pd
import numpy as np

# for visualizations
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

# to print out all the outputs
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

# set display options
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', None)
```

The cell below contains all of the program's functions.

[2]:
```python
def get_values(df, columns):
    """
    Take a dataframe and a list of columns and
    returns the value counts for the columns.
    """
    for column in columns:
        print(column)
        print('=====================================')
        print(df[column].value_counts(dropna=False))
        print('\n')

def show_values(df, param):
    """
    Takes a dataframe and parameters and
    calls the get_values function.
    """
    if param == 'all':
        get_values(df, df.columns)
    else:
        get_values(df, param)
```

[3]:
```
def calculate_toppers(df, column):
    """
    Sorts a dataframe by the supplied column name and
    lists the top 10 rows.
    """
    return df.sort_values(by = column, ascending = False).head(10)
```

[4]:
```
def pipe_counter(df, column):
    """
    Takes a dataframe, a column, and returns
    the top 10 rows of that column.
    """
    string_all = df[column].str.cat(sep = '|')
    series_all = pd.Series(string_all.split('|'))
    top5_all = series_all.value_counts(ascending = False)
    return top5_all.head()
```

## Data Wrangling

In this section, we will load in the data, check for cleanliness, and then trim and clean your dataset for analysis.

1.1.3 General Properties

[5]:
```
# read a csv file
df = pd.read_csv('../data/in/tmdb-movies.csv')
```

Let's get a feel for the dataset.

[6]:
```
df.shape
df.info()
```

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

```
<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
```

1.1.4 Data Cleaning

We only need certain columns. Let's create another dataframe that contains only the desired columns.

[7]:
```
df1 = df[['original_title',
          'popularity',
          'cast',
          'director',
          'runtime',
          'genres',
          'release_year',
          'budget_adj',
          'revenue_adj',
        ]]
```

Let's see how many rows we have.

[8]:
```
print('This dataframe has {} rows or records.'.format(df1.shape[0]))
```

```
This dataframe has 10866 rows or records.
```

Now, let's drop the duplicates.

[9]:
```
df1.drop_duplicates(keep ='first', inplace=True)
print('This dataframe now has {} rows or records.'.format(df1.shape[0]))
```

```
This dataframe now has 10865 rows or records.
```

```
C:\Users\Dd\AppData\Local\Temp\ipykernel_2728\1324486049.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df1.drop_duplicates(keep ='first', inplace=True)
```

Then, we will drop all the rows that has a NaN value. We will create another dataframe before then because of the large number of rows that are being dropped.

[10]:
```
df2 = df1.dropna()
print('This dataframe now has {} rows or records.'.format(df2.shape[0]))
```

```
This dataframe now has 10731 rows or records.
```

Finally, let's convert 0 into NaNs and drop them. Let's create another data that reflects this latest change.

[11]:
```
# creating a seperate list of revenue and budget column
nonzero =['budget_adj', 'revenue_adj']

#this will replace all the value from '0' to NAN in the list
df2[nonzero] = df2[nonzero].replace(0, np.NAN)

df3 = df2.dropna()
print('This dataframe now has {} rows or records.'.format(df3.shape[0]))
```

```
This dataframe now has 3849 rows or records.
```

```
C:\Users\Dd\AppData\Local\Temp\ipykernel_2728\1935603876.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df2[nonzero] = df2[nonzero].replace(0, np.NAN)
```

## Exploratory Data Analysis

Let's take a quick peek at the dataset.

[12]:
```
df3.head()
```

[12]:
```
                  original_title  popularity  \
0                 Jurassic World   32.985763
1             Mad Max: Fury Road   28.419936
2                      Insurgent   13.112507
3   Star Wars: The Force Awakens   11.173104
4                      Furious 7    9.335014
```

```
                                                                      cast  \
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vincent D'Onofrio|Nick Robinson
1        Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nicholas Hoult|Josh Helman
2           Shailene Woodley|Theo James|Kate Winslet|Ansel Elgort|Miles Teller
3             Harrison Ford|Mark Hamill|Carrie Fisher|Adam Driver|Daisy Ridley
4       Vin Diesel|Paul Walker|Jason Statham|Michelle Rodriguez|Dwayne Johnson

           director  runtime                                       genres  \
0   Colin Trevorrow      124  Action|Adventure|Science Fiction|Thriller
1     George Miller      120  Action|Adventure|Science Fiction|Thriller
2  Robert Schwentke      119         Adventure|Science Fiction|Thriller
3       J.J. Abrams      136   Action|Adventure|Science Fiction|Fantasy
4         James Wan      137                        Action|Crime|Thriller

   release_year    budget_adj   revenue_adj
0          2015  1.379999e+08  1.392446e+09
1          2015  1.379999e+08  3.481613e+08
2          2015  1.012000e+08  2.716190e+08
3          2015  1.839999e+08  1.902723e+09
4          2015  1.747999e+08  1.385749e+09
```

[13]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3849 entries, 0 to 10848
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   original_title  3849 non-null   object
 1   popularity      3849 non-null   float64
 2   cast            3849 non-null   object
 3   director        3849 non-null   object
 4   runtime         3849 non-null   int64
 5   genres          3849 non-null   object
 6   release_year    3849 non-null   int64
 7   budget_adj      3849 non-null   float64
 8   revenue_adj     3849 non-null   float64
dtypes: float64(3), int64(2), object(4)
memory usage: 300.7+ KB
```

[14]:
```
df3.describe()
```

[14]:
```
        popularity      runtime  release_year    budget_adj   revenue_adj
count  3849.000000  3849.000000   3849.000000  3.849000e+03  3.849000e+03
mean      1.192933   109.217459   2001.258249  4.429360e+07  1.372313e+08
```

```
std       1.475622    19.914141     11.285642  4.481360e+07  2.162018e+08
min       0.001117    15.000000   1960.000000  9.693980e-01  2.370705e+00
25%       0.463337    95.000000   1995.000000  1.316623e+07  1.843023e+07
50%       0.798582   106.000000   2004.000000  3.005030e+07  6.181393e+07
75%       1.374300   119.000000   2010.000000  6.076720e+07  1.634115e+08
max      32.985763   338.000000   2015.000000  4.250000e+08  2.827124e+09
```

1.1.5 Research Question 1 - What are the most popular movies?

[15]:
```
top_popular = calculate_toppers(df3, 'popularity')
top_popular
```

[15]:
```
                             original_title  popularity  \
0                            Jurassic World   32.985763
1                        Mad Max: Fury Road   28.419936
629                            Interstellar   24.949134
630                 Guardians of the Galaxy   14.311205
2                                 Insurgent   13.112507
631     Captain America: The Winter Soldier   12.971027
1329                              Star Wars   12.037933
632                               John Wick   11.422751
3              Star Wars: The Force Awakens   11.173104
633   The Hunger Games: Mockingjay - Part 1   10.739009

    cast  \
0          Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vincent D'Onofrio|Nick
Robinson
1                Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nicholas
Hoult|Josh Helman
629       Matthew McConaughey|Jessica Chastain|Anne Hathaway|Michael Caine|Casey
Affleck
630                     Chris Pratt|Zoe Saldana|Dave Bautista|Vin
Diesel|Bradley Cooper
2                     Shailene Woodley|Theo James|Kate Winslet|Ansel
Elgort|Miles Teller
631       Chris Evans|Scarlett Johansson|Sebastian Stan|Samuel L. Jackson|Robert
Redford
1329                 Mark Hamill|Harrison Ford|Carrie Fisher|Peter Cushing|Alec
Guinness
632                   Keanu Reeves|Michael Nyqvist|Alfie Allen|Willem Dafoe|Dean
Winters
3                       Harrison Ford|Mark Hamill|Carrie Fisher|Adam
Driver|Daisy Ridley
633   Jennifer Lawrence|Josh Hutcherson|Liam Hemsworth|Woody Harrelson|Donald
Sutherland

                         director  runtime  \
0                  Colin Trevorrow      124
1                    George Miller      120
629              Christopher Nolan      169
630                     James Gunn      121
2                Robert Schwentke      119
631        Joe Russo|Anthony Russo      136
1329                  George Lucas      121
632   Chad Stahelski|David Leitch      101
3                      J.J. Abrams      136
633              Francis Lawrence      123

                                         genres  release_year    budget_adj  \
0     Action|Adventure|Science Fiction|Thriller          2015  1.379999e+08
1     Action|Adventure|Science Fiction|Thriller          2015  1.379999e+08
629             Adventure|Drama|Science Fiction          2014  1.519800e+08
630            Action|Science Fiction|Adventure          2014  1.565855e+08
2            Adventure|Science Fiction|Thriller          2015  1.012000e+08
631            Action|Adventure|Science Fiction          2014  1.565855e+08
1329           Adventure|Action|Science Fiction          1977  3.957559e+07
632                             Action|Thriller          2014  1.842182e+07
3      Action|Adventure|Science Fiction|Fantasy          2015  1.839999e+08
633          Science Fiction|Adventure|Thriller          2014  1.151364e+08

       revenue_adj
0     1.392446e+09
1     3.481613e+08
629   5.726906e+08
630   7.122911e+08
2     2.716190e+08
631   6.583651e+08
1329  2.789712e+09
632   7.252661e+07
3     1.902723e+09
633   6.927528e+08
```

1.1.6 Research Question 2 - Which genres are most popular?

[16]:
```
pipe_counter(top_popular, 'genres')
```

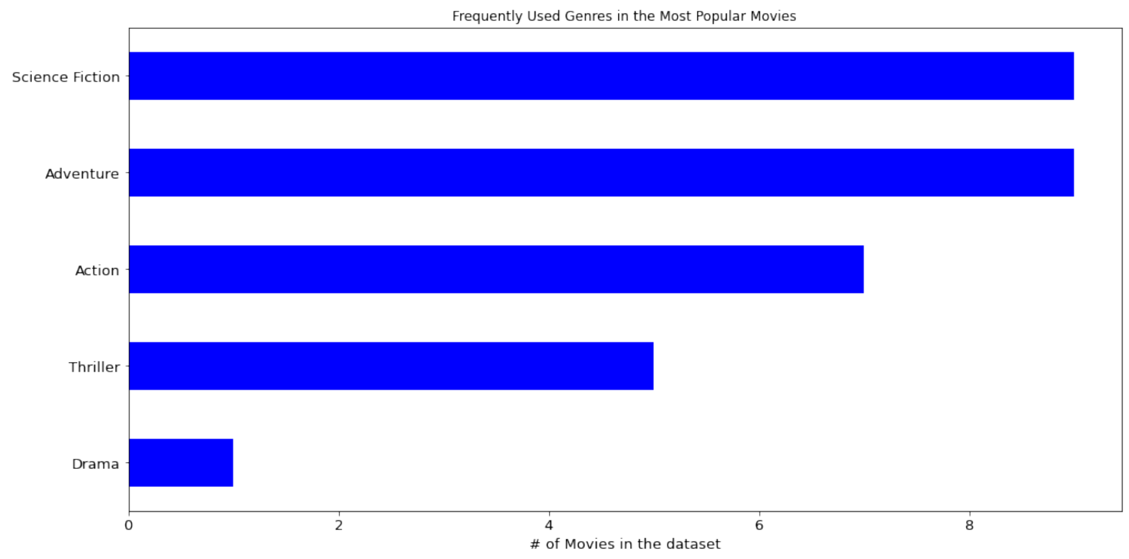
[16]:
```
Adventure          9
Science Fiction    9
Action             7
Thriller           5
Drama              1
dtype: int64
```

```
[17]: genres = pipe_counter(top_popular, 'genres')
      genres.sort_values(ascending = True, inplace = True)

      lt = genres.plot.barh(color = 'blue', fontsize = 13)
      lt.set(title = 'Frequently Used Genres in the Most Popular Movies')
      lt.set_xlabel('# of Movies in the dataset', color = 'black', fontsize = '13')
      lt.figure.set_size_inches(16, 8)
      plt.show()
```

```
[17]: [Text(0.5, 1.0, 'Frequently Used Genres in the Most Popular Movies')]
```

```
[17]: Text(0.5, 0, '# of Movies in the dataset')
```



1.1.7 Research Question 3 - What movies have the highest budgets?

```
[18]: top_budget = calculate_toppers(df3, 'budget_adj')
      top_budget
```

```
[18]:                                  original_title  popularity  \
      2244                            The Warrior's Way    0.250540
      3375  Pirates of the Caribbean: On Stranger Tides    4.955130
      7387     Pirates of the Caribbean: At World's End    4.965391
      6570                             Superman Returns    1.957331
      5231                                      Titanic    4.355219
      7394                                 Spider-Man 3    2.520912
      1929                                      Tangled    2.865684
      14                        Avengers: Age of Ultron    5.944927
      1389       Harry Potter and the Half-Blood Prince    5.076472
```

```
8089                                   Waterworld    1.232098

cast  \
2244                 Kate Bosworth|Jang Dong-gun|Geoffrey Rush|Danny Huston|Ti
Lung
3375             Johnny Depp|PenÃ©lope Cruz|Geoffrey Rush|Ian McShane|Kevin
McNally
7387             Johnny Depp|Orlando Bloom|Keira Knightley|Geoffrey Rush|Bill
Nighy
6570            Brandon Routh|Kevin Spacey|Kate Bosworth|James Marsden|Parker
Posey
5231           Kate Winslet|Leonardo DiCaprio|Frances Fisher|Billy Zane|Kathy
Bates
7394      Tobey Maguire|Kirsten Dunst|James Franco|Thomas Haden Church|Topher
Grace
1929                  Zachary Levi|Mandy Moore|Donna Murphy|Ron Perlman|M.C.
Gainey
14    Robert Downey Jr.|Chris Hemsworth|Mark Ruffalo|Chris Evans|Scarlett
Johansson
1389             Daniel Radcliffe|Rupert Grint|Emma Watson|Tom Felton|Julie
Walters
8089                 Kevin Costner|Chaim Girafi|Rick Aviles|R. D. Call|Zitto
Kazann

                       director  runtime  \
2244                 Sngmoo Lee      100
3375               Rob Marshall      136
7387             Gore Verbinski      169
6570               Bryan Singer      154
5231              James Cameron      194
7394                  Sam Raimi      139
1929  Nathan Greno|Byron Howard      100
14                  Joss Whedon      141
1389                David Yates      153
8089             Kevin Reynolds      135

                                     genres  release_year    budget_adj  \
2244  Adventure|Fantasy|Action|Western|Thriller          2010  4.250000e+08
3375                 Adventure|Action|Fantasy          2011  3.683713e+08
7387                 Adventure|Fantasy|Action          2007  3.155006e+08
6570   Adventure|Fantasy|Action|Science Fiction          2006  2.920507e+08
5231                   Drama|Romance|Thriller          1997  2.716921e+08
7394                 Fantasy|Action|Adventure          2007  2.713305e+08
1929                         Animation|Family          2010  2.600000e+08
14           Action|Adventure|Science Fiction          2015  2.575999e+08
1389                 Adventure|Fantasy|Family          2009  2.541001e+08
8089                         Adventure|Action          1995  2.504192e+08
```

```
       revenue_adj
2244  1.108757e+07
3375  9.904175e+08
7387  1.010654e+09
6570  4.230205e+08
5231  2.506406e+09
7394  9.369017e+08
1929  5.917949e+08
14    1.292632e+09
1389  9.492765e+08
8089  3.780875e+08
```

1.1.8 Research Question 4 - What movies have the highest revenue?

[19]:
```
top_revenue = calculate_toppers(df3, 'revenue_adj')
top_revenue
```

[19]:
```
                        original_title  popularity  \
1386                            Avatar    9.432768
1329                         Star Wars   12.037933
5231                           Titanic    4.355219
10594                      The Exorcist    2.010733
9806                              Jaws    2.563191
3          Star Wars: The Force Awakens   11.173104
8889        E.T. the Extra-Terrestrial    2.900556
8094                           The Net    1.136610
10110  One Hundred and One Dalmatians    2.631987
4361                      The Avengers    7.637767

cast  \
1386    Sam Worthington|Zoe Saldana|Sigourney Weaver|Stephen Lang|Michelle
Rodriguez
1329             Mark Hamill|Harrison Ford|Carrie Fisher|Peter Cushing|Alec
Guinness
5231            Kate Winslet|Leonardo DiCaprio|Frances Fisher|Billy Zane|Kathy
Bates
10594               Linda Blair|Max von Sydow|Ellen Burstyn|Jason Miller|Lee J.
Cobb
9806         Roy Scheider|Robert Shaw|Richard Dreyfuss|Lorraine Gary|Murray
Hamilton
3                   Harrison Ford|Mark Hamill|Carrie Fisher|Adam Driver|Daisy
Ridley
8889         Henry Thomas|Drew Barrymore|Robert MacNaughton|Dee Wallace|Peter
Coyote
8094            Sandra Bullock|Jeremy Northam|Dennis Miller|Wendy Gazelle|Ken
Howard
```

```
10110        Rod Taylor|J. Pat O'Malley|Betty Lou Gerson|Martha Wentworth|Ben
Wright
4361   Robert Downey Jr.|Chris Evans|Mark Ruffalo|Chris Hemsworth|Scarlett
Johansson

                                                director  runtime  \
1386                                       James Cameron      162
1329                                        George Lucas      121
5231                                       James Cameron      194
10594                                   William Friedkin      122
9806                                    Steven Spielberg      124
3                                           J.J. Abrams      136
8889                                    Steven Spielberg      115
8094                                       Irwin Winkler      114
10110  Clyde Geronimi|Hamilton Luske|Wolfgang Reitherman       79
4361                                         Joss Whedon      143

                                         genres  release_year    budget_adj  \
1386   Action|Adventure|Fantasy|Science Fiction          2009  2.408869e+08
1329           Adventure|Action|Science Fiction          1977  3.957559e+07
5231                     Drama|Romance|Thriller          1997  2.716921e+08
10594                     Drama|Horror|Thriller          1973  3.928928e+07
9806                  Horror|Thriller|Adventure          1975  2.836275e+07
3      Action|Adventure|Science Fiction|Fantasy          2015  1.839999e+08
8889   Science Fiction|Adventure|Family|Fantasy          1982  2.372625e+07
8094        Crime|Drama|Mystery|Thriller|Action          1995  3.148127e+07
10110          Adventure|Animation|Comedy|Family          1961  2.917944e+07
4361           Science Fiction|Action|Adventure          2012  2.089437e+08

        revenue_adj
1386   2.827124e+09
1329   2.789712e+09
5231   2.506406e+09
10594  2.167325e+09
9806   1.907006e+09
3      1.902723e+09
8889   1.791694e+09
8094   1.583050e+09
10110  1.574815e+09
4361   1.443191e+09
```

1.1.9 Research Question 5 - What movies are most profitable?

[20]:
```
df3['profit'] = df3.revenue_adj + df3.budget_adj
```

```
C:\Users\Dd\AppData\Local\Temp\ipykernel_2728\4085918358.py:1:
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df3['profit'] = df3.revenue_adj + df3.budget_adj
```

[21]:
```
df3.head()
```

[21]:
```
                 original_title  popularity  \
0                Jurassic World   32.985763
1            Mad Max: Fury Road   28.419936
2                     Insurgent   13.112507
3  Star Wars: The Force Awakens   11.173104
4                     Furious 7    9.335014

                                                                   cast  
\
0  Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vincent D'Onofrio|Nick Robinson
1        Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nicholas Hoult|Josh Helman
2           Shailene Woodley|Theo James|Kate Winslet|Ansel Elgort|Miles Teller
3             Harrison Ford|Mark Hamill|Carrie Fisher|Adam Driver|Daisy Ridley
4       Vin Diesel|Paul Walker|Jason Statham|Michelle Rodriguez|Dwayne Johnson

           director  runtime                                     genres  \
0   Colin Trevorrow      124  Action|Adventure|Science Fiction|Thriller
1     George Miller      120  Action|Adventure|Science Fiction|Thriller
2  Robert Schwentke      119         Adventure|Science Fiction|Thriller
3       J.J. Abrams      136   Action|Adventure|Science Fiction|Fantasy
4         James Wan      137                      Action|Crime|Thriller

   release_year    budget_adj   revenue_adj        profit
0          2015  1.379999e+08  1.392446e+09  1.530446e+09
1          2015  1.379999e+08  3.481613e+08  4.861612e+08
2          2015  1.012000e+08  2.716190e+08  3.728190e+08
3          2015  1.839999e+08  1.902723e+09  2.086723e+09
4          2015  1.747999e+08  1.385749e+09  1.560549e+09
```

[22]:
```
top_profit = calculate_toppers(df3, 'profit')
top_profit
```

[22]:
```
                     original_title  popularity  \
1386                         Avatar    9.432768
1329                      Star Wars   12.037933
5231                        Titanic    4.355219
10594                  The Exorcist    2.010733
3      Star Wars: The Force Awakens   11.173104
```

```
9806                             Jaws    2.563191
8889       E.T. the Extra-Terrestrial    2.900556
4361                     The Avengers    7.637767
8094                          The Net    1.136610
10110  One Hundred and One Dalmatians    2.631987

cast  \
1386    Sam Worthington|Zoe Saldana|Sigourney Weaver|Stephen Lang|Michelle
Rodriguez
1329             Mark Hamill|Harrison Ford|Carrie Fisher|Peter Cushing|Alec
Guinness
5231            Kate Winslet|Leonardo DiCaprio|Frances Fisher|Billy Zane|Kathy
Bates
10594               Linda Blair|Max von Sydow|Ellen Burstyn|Jason Miller|Lee J.
Cobb
3                   Harrison Ford|Mark Hamill|Carrie Fisher|Adam Driver|Daisy
Ridley
9806         Roy Scheider|Robert Shaw|Richard Dreyfuss|Lorraine Gary|Murray
Hamilton
8889         Henry Thomas|Drew Barrymore|Robert MacNaughton|Dee Wallace|Peter
Coyote
4361   Robert Downey Jr.|Chris Evans|Mark Ruffalo|Chris Hemsworth|Scarlett
Johansson
8094            Sandra Bullock|Jeremy Northam|Dennis Miller|Wendy Gazelle|Ken
Howard
10110        Rod Taylor|J. Pat O'Malley|Betty Lou Gerson|Martha Wentworth|Ben
Wright

                                                director  runtime  \
1386                                       James Cameron      162
1329                                        George Lucas      121
5231                                       James Cameron      194
10594                                   William Friedkin      122
3                                           J.J. Abrams      136
9806                                    Steven Spielberg      124
8889                                    Steven Spielberg      115
4361                                         Joss Whedon      143
8094                                       Irwin Winkler      114
10110  Clyde Geronimi|Hamilton Luske|Wolfgang Reitherman       79

                                      genres  release_year    budget_adj  \
1386   Action|Adventure|Fantasy|Science Fiction          2009  2.408869e+08
1329           Adventure|Action|Science Fiction          1977  3.957559e+07
5231                    Drama|Romance|Thriller          1997  2.716921e+08
10594                    Drama|Horror|Thriller          1973  3.928928e+07
3      Action|Adventure|Science Fiction|Fantasy          2015  1.839999e+08
9806                 Horror|Thriller|Adventure          1975  2.836275e+07
```

```
8889   Science Fiction|Adventure|Family|Fantasy          1982  2.372625e+07
4361           Science Fiction|Action|Adventure          2012  2.089437e+08
8094        Crime|Drama|Mystery|Thriller|Action          1995  3.148127e+07
10110         Adventure|Animation|Comedy|Family          1961  2.917944e+07

        revenue_adj        profit
1386   2.827124e+09  3.068011e+09
1329   2.789712e+09  2.829288e+09
5231   2.506406e+09  2.778098e+09
10594  2.167325e+09  2.206614e+09
3      1.902723e+09  2.086723e+09
9806   1.907006e+09  1.935369e+09
8889   1.791694e+09  1.815421e+09
4361   1.443191e+09  1.652135e+09
8094   1.583050e+09  1.614531e+09
10110  1.574815e+09  1.603994e+09
```

1.1.10 Research Question 6 - Describe the sweetspot for the runtime feature.

[23]:
```
plt.figure(figsize=(16,8), dpi = 100)
plt.xlabel('Runtime of the Movies', fontsize = 15)
plt.ylabel('# of Movies in the Dataset', fontsize=15)
plt.title('Runtime of all the movies', fontsize=15)
plt.hist(df3['runtime'], rwidth = 0.9, bins =35)
plt.show()
```

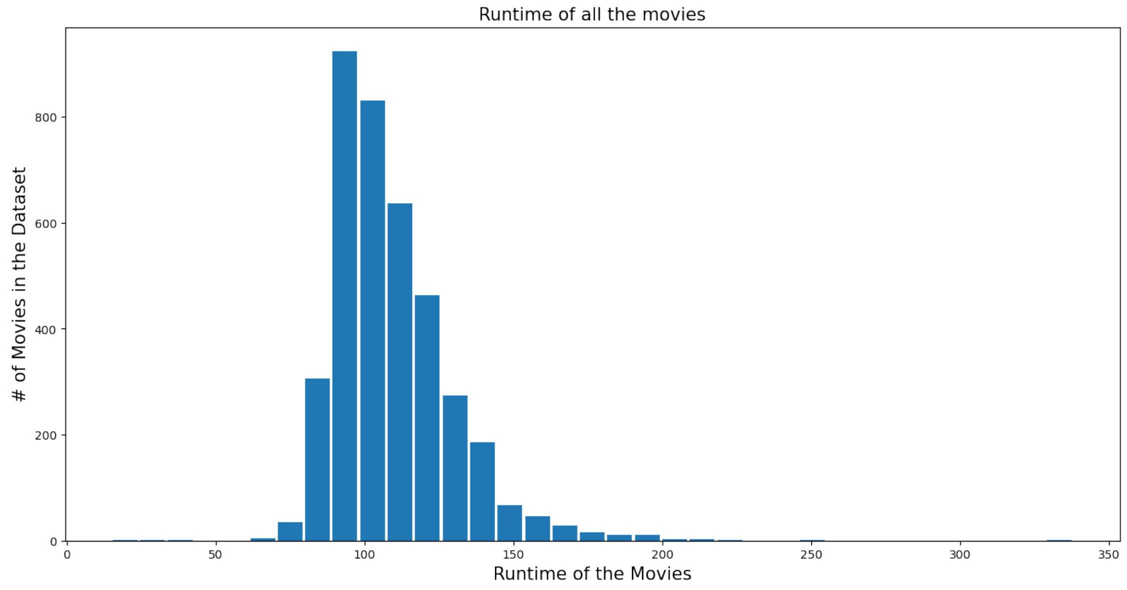
[23]:
```
<Figure size 1600x800 with 0 Axes>
```

[23]:
```
Text(0.5, 0, 'Runtime of the Movies')
```

[23]:
```
Text(0, 0.5, '# of Movies in the Dataset')
```

[23]:
```
Text(0.5, 1.0, 'Runtime of all the movies')
```

[23]:
```
(array([  1.,   1.,   1.,   0.,   0.,   4.,  35., 306., 923., 831., 637.,
        463., 274., 185.,  67.,  46.,  28.,  16.,  11.,  11.,   3.,   3.,
          1.,   0.,   0.,   1.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
          0.,   1.]),
 array([ 15.        ,  24.22857143,  33.45714286,  42.68571429,
         51.91428571,  61.14285714,  70.37142857,  79.6       ,
         88.82857143,  98.05714286, 107.28571429, 116.51428571,
        125.74285714, 134.97142857, 144.2       , 153.42857143,
        162.65714286, 171.88571429, 181.11428571, 190.34285714,
        199.57142857, 208.8       , 218.02857143, 227.25714286,
        236.48571429, 245.71428571, 254.94285714, 264.17142857,
        273.4       , 282.62857143, 291.85714286, 301.08571429,
        310.31428571, 319.54285714, 328.77142857, 338.        ]),
<BarContainer object of 35 artists>)
```



[24]:
```
df3['runtime'].describe()
```

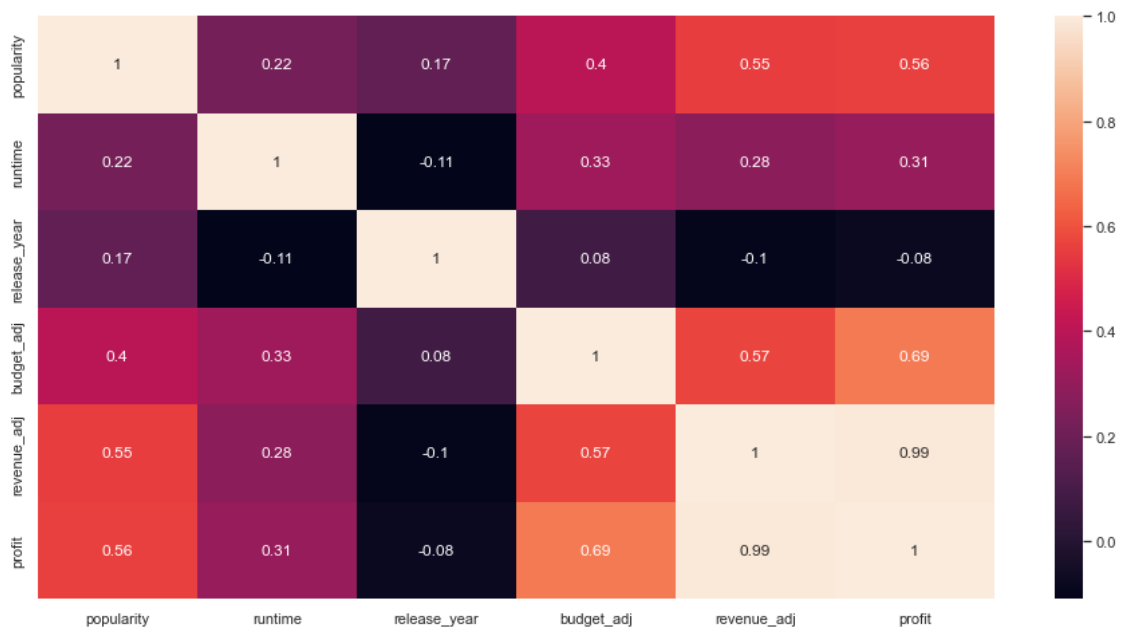
[24]:
```
count    3849.000000
mean      109.217459
std        19.914141
min        15.000000
25%        95.000000
50%       106.000000
75%       119.000000
max       338.000000
Name: runtime, dtype: float64
```

1.1.11 Research Question 7 - Are any of the features correlated?

[25]:
```
corr = round(df3.corr(),2)
```

[26]:
```
sns.set(rc = {'figure.figsize':(16,8)})
sns.heatmap(corr, annot=True)
```

[26]:
```
<AxesSubplot:>
```



## Conclusions

Although, we did not find any strong correlation between the selected variables, this dataset was still fun to explore.

1.1.12 Findings

1. Jurassic World is the most popular movie.
2. For the top ten most popular movies, adventure, science fiction, action, thriller, and drama are the most popular genres.
3. The Warrior's Way had the biggest budget.
4. And Avatar bringing in the most money.
5. But Jurassic World took the crown for profitability.
6. The average runtime of a movie in the most popular subset is 109 minutes.
7. No strong correlation on this dataset.

1.1.13 Limitations

Some of the analysis was done on a subset that consisted of the top ten most popular movies which may not be enough to conclude anything of value. Dropping all rows with NaN values significantly change the shape of the dataframe. Perhaps nextime, we should impute the missing values instead of droppoing them.

1.1.14 Next Steps

Modifying the parameters of the subset of data could greatly improved the result of this analysis. Instead of an arbitrary hardcoded number of 10, perhaps we could substiture it instead with a percentage or quartile.

Making a network graph of the cast would be interesting too.

```
[ ]:
```