TMDB Data Analysis

October 10, 2020

1 Project: TMDB Movie Data Analysis

1.1 Table of Contents

Introduction

> **Tip**: In this section of the report, provide a brief introduction to the dataset you've selected for analysis. At the end of this section, describe the questions that you plan on exploring over the course of the report. Try to build your report around the analysis of at least one dependent variable and three independent variables.
>
> If you haven't yet selected and downloaded your data, make sure you do that first before coming back here. If you're not sure what questions to ask right now, then make sure you familiarize yourself with the variables and the dataset context for ideas of what to explore.

This dataset shows the data of movies and information about their cast, rating, production companies, movie budget and various other attributes. We want to find out: - Actors with highest rating movies - Production Companies with highest rating movies - Do movies with high budgets recieve better rating than movies with low budgets - Which genres have the highest ratings - Which genre came out on top each year

[245]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pprint
%matplotlib inline
```

Data Wrangling

Data Quality Issues

- ☐ [‘cast’, ‘director’, ‘keywords’, ‘genres’, ‘production_companies’]has missing values
- ☒ release_date has str type instead of datetime
- ☒ Drop [‘imdb_id’, ‘homepage’, ‘tagline’, ‘keywords’, ‘overview’] columns.
- ☒ there is one duplicate row
- ☒ Replace ‘0’ with ‘NaN’ in [‘revenue’, ‘budget’, ‘runtime’, ‘budget_adj’, ‘revenue_adj’]

Data Tidiness

- ☒ Cast column has multiple values
- ☒ Genres columns has multiple values
- ☒ production_companies column has multiple values

1.1.1 General Properties

[4]:
```
# Load dataset into a dataframe
df = pd.read_csv('tmdb-movies.csv')
```

[5]:
```
# Overview of dataframe
df.head()
```

[5]:
```
       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]
```

[6]:
```python
# dataframe info
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
```

[8]:
```
df_clean = df.copy()
```

1.1.2 Data Cleaning

Drop unnecessary columns to analysis

[25]:
```
d_col = ['imdb_id', 'homepage', 'tagline', 'keywords', 'overview']
df_clean.drop(d_col, axis=1, inplace=True)
```

Fix values sperated by '|' issue by formating them into lists

[33]:
```
df_clean.cast = df.cast.str.split('|')
```

[13]:
```
df_clean.genres = df_clean.genres.str.split('|')
```

[21]:
```
df_clean.production_companies = df_clean.production_companies.str.split('|')
```

[103]:
```
df_clean.director = df_clean.director.str.split('|')
```

Expand the lists into columns

[43]:
```python
# Expand cast into columns
cast = df_clean.cast.apply(pd.Series)
# Rename cast colunmns
cast = cast.rename(columns = lambda x : 'cast_' + str(x+1))
```

[51]:
```python
# Concatenate the 2 dataframes
df_clean = pd.concat([df_clean,cast], axis=1)
```

[58]:
```python
# Expand Genres into columns
genres = df_clean.genres.apply(pd.Series)
# Rename genres columns
genres = genres.rename(columns = lambda x : 'genre_' + str(x+1))
```

[60]:
```python
# Concatenate the 2 dataframes
df_clean = pd.concat([df_clean, genres], axis=1)
```

[63]:
```python
# Expand production_companies into columns
p_c = df_clean.production_companies.apply(pd.Series)
# Rename columns
p_c = p_c.rename(columns= lambda x : 'company_' + str(x+1))
```

[65]:
```python
# Concatenate the 2 dataframes
df_clean = pd.concat([df_clean, p_c], axis=1)
```

[66]:
```python
# get a list of column names
list(df_clean)
```

[66]:
```
['id',
 'popularity',
 'budget',
 'revenue',
 'original_title',
 'cast',
 'director',
 'runtime',
 'genres',
 'production_companies',
 'release_date',
 'vote_count',
 'vote_average',
 'release_year',
 'budget_adj',
 'revenue_adj',
 'cast_1',
 'cast_2',
 'cast_3',
 'cast_4',
```

```
 'cast_5',
 'genre_1',
 'genre_2',
 'genre_3',
 'genre_4',
 'genre_5',
 'company_1',
 'company_2',
 'company_3',
 'company_4',
 'company_5']
```

[69]:
```
# Rearrange columns
df_clean = df_clean[
['id',
 'popularity',
 'budget',
 'revenue',
 'original_title',
 'cast',
 'cast_1',
 'cast_2',
 'cast_3',
 'cast_4',
 'cast_5',
 'director',
 'runtime',
 'genres',
 'genre_1',
 'genre_2',
 'genre_3',
 'genre_4',
 'genre_5',
 'production_companies',
 'company_1',
 'company_2',
 'company_3',
 'company_4',
 'company_5',
 'release_date',
 'vote_count',
 'vote_average',
 'release_year',
 'budget_adj',
 'revenue_adj']
]
```

[71]:
```
# drop columns with lists as data
df_clean.drop(['cast', 'genres', 'production_companies'], axis=1, inplace=True)
```

[72]:
```
# check if columns are arranged and dropped
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10866 entries, 0 to 10865
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              10866 non-null  int64
 1   popularity      10866 non-null  float64
 2   budget          10866 non-null  int64
 3   revenue         10866 non-null  int64
 4   original_title  10866 non-null  object
 5   cast_1          10790 non-null  object
 6   cast_2          10646 non-null  object
 7   cast_3          10556 non-null  object
 8   cast_4          10447 non-null  object
 9   cast_5          10134 non-null  object
 10  director        10822 non-null  object
 11  runtime         10866 non-null  int64
 12  genre_1         10843 non-null  object
 13  genre_2         8515 non-null   object
 14  genre_3         5079 non-null   object
 15  genre_4         1981 non-null   object
 16  genre_5         542 non-null    object
 17  company_1       9836 non-null   object
 18  company_2       6396 non-null   object
 19  company_3       3816 non-null   object
 20  company_4       2053 non-null   object
 21  company_5       1126 non-null   object
 22  release_date    10866 non-null  object
 23  vote_count      10866 non-null  int64
 24  vote_average    10866 non-null  float64
 25  release_year    10866 non-null  int64
 26  budget_adj      10866 non-null  float64
 27  revenue_adj     10866 non-null  float64
dtypes: float64(4), int64(6), object(18)
memory usage: 2.3+ MB
```

[74]:
```
# change release_date column type to datetime
df_clean.release_date = pd.to_datetime(df_clean.release_date)
```

[78]:
```
# show all columns
pd.set_option('display.max_columns', 28)
```

[86]:
```python
# check for duplicates
df_clean.duplicated().sum()
```

[86]: 1

[87]:
```python
# Drop duplicate row
df_clean.drop_duplicates(inplace=True)
```

[115]:
```python
# Replace '0' with NaN in columnns
df_clean.revenue.replace(0, np.NAN, inplace=True)
df_clean.budget.replace(0, np.NAN, inplace=True)
df_clean.runtime.replace(0, np.NAN, inplace=True)
df_clean.revenue_adj.replace(0, np.NAN, inplace=True)
df_clean.budget_adj.replace(0, np.NAN, inplace=True)
```

[122]:
```python
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              10865 non-null  int64
 1   popularity      10865 non-null  float64
 2   budget          5169 non-null   float64
 3   revenue         4849 non-null   float64
 4   original_title  10865 non-null  object
 5   cast_1          10789 non-null  object
 6   cast_2          10645 non-null  object
 7   cast_3          10555 non-null  object
 8   cast_4          10446 non-null  object
 9   cast_5          10133 non-null  object
 10  director        10821 non-null  object
 11  runtime         10834 non-null  float64
 12  genre_1         10842 non-null  object
 13  genre_2         8514 non-null   object
 14  genre_3         5078 non-null   object
 15  genre_4         1980 non-null   object
 16  genre_5         541 non-null    object
 17  company_1       9835 non-null   object
 18  company_2       6395 non-null   object
 19  company_3       3816 non-null   object
 20  company_4       2053 non-null   object
 21  company_5       1126 non-null   object
 22  release_date    10865 non-null  datetime64[ns]
 23  vote_count      10865 non-null  int64
 24  vote_average    10865 non-null  float64
```

```
 25   release_year     10865 non-null   int64
 26   budget_adj       5169 non-null    float64
 27   revenue_adj      4849 non-null    float64
dtypes: datetime64[ns](1), float64(7), int64(3), object(17)
memory usage: 2.4+ MB
```

[121]:
```
df_clean.head(20)
```

[121]:
```
        id  popularity       budget       revenue  \
0   135397   32.985763  150000000.0  1.513529e+09
1    76341   28.419936  150000000.0  3.784364e+08
2   262500   13.112507  110000000.0  2.952382e+08
3   140607   11.173104  200000000.0  2.068178e+09
4   168259    9.335014  190000000.0  1.506249e+09
5   281957    9.110700  135000000.0  5.329505e+08
6    87101    8.654359  155000000.0  4.406035e+08
7   286217    7.667400  108000000.0  5.953803e+08
8   211672    7.404165   74000000.0  1.156731e+09
9   150540    6.326804  175000000.0  8.537086e+08
10  206647    6.200282  245000000.0  8.806746e+08
11   76757    6.189369  176000003.0  1.839877e+08
12  264660    6.118847   15000000.0  3.686941e+07
13  257344    5.984995   88000000.0  2.436371e+08
14   99861    5.944927  280000000.0  1.405036e+09
15  273248    5.898400   44000000.0  1.557601e+08
16  260346    5.749758   48000000.0  3.257714e+08
17  102899    5.573184  130000000.0  5.186022e+08
18  150689    5.556818   95000000.0  5.423514e+08
19  131634    5.476958  160000000.0  6.505234e+08

                   original_title                 cast_1  \
0                  Jurassic World            Chris Pratt
1              Mad Max: Fury Road              Tom Hardy
2                       Insurgent       Shailene Woodley
3    Star Wars: The Force Awakens          Harrison Ford
4                       Furious 7             Vin Diesel
5                    The Revenant      Leonardo DiCaprio
6              Terminator Genisys  Arnold Schwarzenegger
7                     The Martian            Matt Damon
8                         Minions        Sandra Bullock
9                      Inside Out            Amy Poehler
10                        Spectre           Daniel Craig
11              Jupiter Ascending             Mila Kunis
12                     Ex Machina       Domhnall Gleeson
13                         Pixels           Adam Sandler
14        Avengers: Age of Ultron      Robert Downey Jr.
15              The Hateful Eight      Samuel L. Jackson
```

```
16                              Taken 3        Liam Neeson
17                              Ant-Man          Paul Rudd
18                           Cinderella         Lily James
19  The Hunger Games: Mockingjay - Part 2  Jennifer Lawrence

                cast_2                cast_3               cast_4  \
0   Bryce Dallas Howard           Irrfan Khan   Vincent D'Onofrio
1      Charlize Theron      Hugh Keays-Byrne      Nicholas Hoult
2           Theo James          Kate Winslet        Ansel Elgort
3          Mark Hamill         Carrie Fisher         Adam Driver
4          Paul Walker         Jason Statham  Michelle Rodriguez
5            Tom Hardy          Will Poulter    Domhnall Gleeson
6         Jason Clarke         Emilia Clarke         Jai Courtney
7     Jessica Chastain          Kristen Wiig         Jeff Daniels
8             Jon Hamm        Michael Keaton       Allison Janney
9        Phyllis Smith          Richard Kind           Bill Hader
10     Christoph Waltz          LÃ©a Seydoux       Ralph Fiennes
11      Channing Tatum             Sean Bean      Eddie Redmayne
12     Alicia Vikander           Oscar Isaac       Sonoya Mizuno
13   Michelle Monaghan        Peter Dinklage             Josh Gad
14     Chris Hemsworth          Mark Ruffalo          Chris Evans
15        Kurt Russell  Jennifer Jason Leigh      Walton Goggins
16      Forest Whitaker          Maggie Grace       Famke Janssen
17      Michael Douglas      Evangeline Lilly          Corey Stoll
18       Cate Blanchett        Richard Madden  Helena Bonham Carter
19      Josh Hutcherson        Liam Hemsworth      Woody Harrelson

               cast_5                           director  runtime  \
0       Nick Robinson                    Colin Trevorrow    124.0
1         Josh Helman                      George Miller    120.0
2         Miles Teller                  Robert Schwentke    119.0
3         Daisy Ridley                        J.J. Abrams    136.0
4       Dwayne Johnson                          James Wan    137.0
5        Paul Anderson  Alejandro GonzÃ¡lez IÃ±Ã¡rritu    156.0
6         J.K. Simmons                        Alan Taylor    125.0
7        Michael PeÃ±a                       Ridley Scott    141.0
8         Steve Coogan         Kyle Balda|Pierre Coffin     91.0
9          Lewis Black                        Pete Docter     94.0
10     Monica Bellucci                         Sam Mendes    148.0
11       Douglas Booth  Lana Wachowski|Lilly Wachowski    124.0
12       Corey Johnson                       Alex Garland    108.0
13         Kevin James                     Chris Columbus    105.0
14  Scarlett Johansson                        Joss Whedon    141.0
15       DemiÃ¡n Bichir                  Quentin Tarantino    167.0
16        Dougray Scott                    Olivier Megaton    109.0
17     Bobby Cannavale                        Peyton Reed    115.0
18   Holliday Grainger                    Kenneth Branagh    112.0
```

```
19     Elizabeth Banks               Francis Lawrence    136.0

            genre_1          genre_2          genre_3    genre_4 genre_5  \
0            Action        Adventure  Science Fiction   Thriller     NaN
1            Action        Adventure  Science Fiction   Thriller     NaN
2         Adventure  Science Fiction         Thriller        NaN     NaN
3            Action        Adventure  Science Fiction    Fantasy     NaN
4            Action            Crime         Thriller        NaN     NaN
5           Western            Drama        Adventure   Thriller     NaN
6   Science Fiction           Action         Thriller  Adventure     NaN
7             Drama        Adventure  Science Fiction        NaN     NaN
8            Family        Animation        Adventure     Comedy     NaN
9            Comedy        Animation           Family        NaN     NaN
10           Action        Adventure            Crime        NaN     NaN
11  Science Fiction          Fantasy           Action  Adventure     NaN
12            Drama  Science Fiction              NaN        NaN     NaN
13           Action           Comedy  Science Fiction        NaN     NaN
14           Action        Adventure  Science Fiction        NaN     NaN
15            Crime            Drama          Mystery    Western     NaN
16            Crime           Action         Thriller        NaN     NaN
17  Science Fiction           Action        Adventure        NaN     NaN
18          Romance          Fantasy           Family      Drama     NaN
19              War        Adventure  Science Fiction        NaN     NaN

                               company_1  \
0                      Universal Studios
1               Village Roadshow Pictures
2                    Summit Entertainment
3                               Lucasfilm
4                      Universal Pictures
5                     Regency Enterprises
6                      Paramount Pictures
7   Twentieth Century Fox Film Corporation
8                      Universal Pictures
9                    Walt Disney Pictures
10                      Columbia Pictures
11              Village Roadshow Pictures
12                              DNA Films
13                      Columbia Pictures
14                         Marvel Studios
15                   Double Feature Films
16  Twentieth Century Fox Film Corporation
17                         Marvel Studios
18                   Walt Disney Pictures
19                      Studio Babelsberg

                               company_2  \
```

```
0                    Amblin Entertainment
1            Kennedy Miller Productions
2                       Mandeville Films
3                  Truenorth Productions
4                          Original Film
5                             Appian Way
6                  Skydance Productions
7                 Scott Free Productions
8            Illumination Entertainment
9               Pixar Animation Studios
10                                Danjaq
11                    Dune Entertainment
12  Universal Pictures International (UPI)
13           Happy Madison Productions
14                           Prime Focus
15                  The Weinstein Company
16                              M6 Films
17                                   NaN
18                           Genre Films
19                           StudioCanal

                          company_3                           company_4  \
0                Legendary Pictures             Fuji Television Network
1                               NaN                                 NaN
2           Red Wagon Entertainment                             NeoReel
3                         Bad Robot                                 NaN
4               Media Rights Capital                             Dentsu
5                         CatchPlay                   Anonymous Content
6                               NaN                                 NaN
7                 Mid Atlantic Films               International Traders
8                               NaN                                 NaN
9   Walt Disney Studios Motion Pictures                              NaN
10                              B24                                 NaN
11             Anarchos Productions                        Warner Bros.
12                            Film4                                 NaN
13                              NaN                                 NaN
14           Revolution Sun Studios                                 NaN
15                       FilmColony                                 NaN
16                           Canal+                          EuropaCorp
17                              NaN                                 NaN
18                 Beagle Pug Films        Allison Shearmur Productions
19                        Lionsgate  Walt Disney Studios Motion Pictures

            company_5 release_date  vote_count  vote_average  release_year  \
0              Dentsu   2015-06-09        5562           6.5          2015
1                 NaN   2015-05-13        6185           7.1          2015
2                 NaN   2015-03-18        2480           6.3          2015
```

```
3                    NaN   2015-12-15        5292           7.5           2015
4        One Race Films   2015-04-01        2947           7.3           2015
5  New Regency Pictures   2015-12-25        3929           7.2           2015
6                    NaN   2015-06-23        2598           5.8           2015
7     TSG Entertainment   2015-09-30        4572           7.6           2015
8                    NaN   2015-06-17        2893           6.5           2015
9                    NaN   2015-06-09        3935           8.0           2015
10                   NaN   2015-10-26        3254           6.2           2015
11                   NaN   2015-02-04        1937           5.2           2015
12                   NaN   2015-01-21        2854           7.6           2015
13                   NaN   2015-07-16        1575           5.8           2015
14                   NaN   2015-04-22        4304           7.4           2015
15                   NaN   2015-12-25        2389           7.4           2015
16                CinÃ©+   2015-01-01        1578           6.1           2015
17                   NaN   2015-07-14        3779           7.0           2015
18                   NaN   2015-03-12        1495           6.8           2015
19           Color Force   2015-11-18        2380           6.5           2015
```

```
      budget_adj   revenue_adj
0   1.379999e+08  1.392446e+09
1   1.379999e+08  3.481613e+08
2   1.012000e+08  2.716190e+08
3   1.839999e+08  1.902723e+09
4   1.747999e+08  1.385749e+09
5   1.241999e+08  4.903142e+08
6   1.425999e+08  4.053551e+08
7   9.935996e+07  5.477497e+08
8   6.807997e+07  1.064192e+09
9   1.609999e+08  7.854116e+08
10  2.253999e+08  8.102203e+08
11  1.619199e+08  1.692686e+08
12  1.379999e+07  3.391985e+07
13  8.095996e+07  2.241460e+08
14  2.575999e+08  1.292632e+09
15  4.047998e+07  1.432992e+08
16  4.415998e+07  2.997096e+08
17  1.195999e+08  4.771138e+08
18  8.739996e+07  4.989630e+08
19  1.471999e+08  5.984813e+08
```

Exploratory Data Analysis

1.1.3 Research Question 1: Which genres are most popular from year to year?

[134]:
```
# Create list of genres in dataset
genres = pd.unique(df_clean[['genre_1', 'genre_2', 'genre_3', 'genre_3',␣
↪'genre_4', 'genre_5']].values.ravel('K')).tolist()
```

[149]:
```
len(genres)
```

[149]: 21

Note: Every movie in this dataset has more than one genre listed, so when calculating the average rating for each genre through the years movies may be calculated more than once accross different genres.

[145]:
```
# Create list of release_years in dataset
years = df_clean.release_year.unique().tolist()
```

[203]:
```
# Getting the average rating for each genre in each year listed in the dataset
# create list to store dicts in it which will be used later to construct a␣
↪dataframe
years_list = []
# loop through each genre for each year
for year in years:
    temp_dict= {}
    temp_dict.update({'Year' : year})
    avg_rate = []
    temp_df = df_clean[df_clean.release_year == year]
    for genre in genres:
#       Get the average rating for each genre in each year
        temp1_df = temp_df[temp_df.genre_1 == genre]
        avg_rate.append(temp1_df.vote_average.mean())
        temp1_df = temp_df[temp_df.genre_2 == genre]
        avg_rate.append(temp1_df.vote_average.mean())
        temp1_df = temp_df[temp_df.genre_3 == genre]
        avg_rate.append(temp1_df.vote_average.mean())
        temp1_df = temp_df[temp_df.genre_4 == genre]
        avg_rate.append(temp1_df.vote_average.mean())
        temp1_df = temp_df[temp_df.genre_5 == genre]
        avg_rate.append(temp1_df.vote_average.mean())
#       Append list of dict where the key is the key is the genre and value is␣
↪the average rate for each genre
        temp_dict.update({genre: np.nanmean(avg_rate)})
    years_list.append(temp_dict)
```

[205]:
```
# Create a dataframe of each year and each genre average rating
year_genre_df = pd.DataFrame(years_list)
```

[223]:
```python
# Drop nan column
year_genre_df.drop(np.nan, axis=1, inplace=True)
```

[224]:
```python
year_genre_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56 entries, 0 to 55
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Year             56 non-null     int64
 1   Action           56 non-null     float64
 2   Adventure        56 non-null     float64
 3   Western          56 non-null     float64
 4   Science Fiction  56 non-null     float64
 5   Drama            56 non-null     float64
 6   Family           56 non-null     float64
 7   Comedy           56 non-null     float64
 8   Crime            56 non-null     float64
 9   Romance          56 non-null     float64
 10  War              56 non-null     float64
 11  Mystery          56 non-null     float64
 12  Thriller         56 non-null     float64
 13  Fantasy          56 non-null     float64
 14  History          56 non-null     float64
 15  Animation        56 non-null     float64
 16  Horror           56 non-null     float64
 17  Music            56 non-null     float64
 18  Documentary      56 non-null     float64
 19  TV Movie         56 non-null     float64
 20  Foreign          56 non-null     float64
dtypes: float64(20), int64(1)
memory usage: 9.3 KB
```

[226]:
```python
year_genre_df
```

[226]:
```
   Year    Action  Adventure   Western  Science Fiction     Drama    Family  \
0  2015  5.667830   5.833887  6.051758         6.020137  6.010789  6.092039
1  2014  6.246564   6.020051  5.998223         6.001323  5.993221  6.081321
2  1977  6.650000   6.540873  6.446786         6.349490  6.380079  6.435974
3  2009  5.912923   5.979953  5.979953         5.864602  5.963752  5.941454
4  2010  5.842133   6.177322  6.142714         5.852415  5.895067  5.904924
5  1999  5.857458   5.887744  5.780195         5.622918  5.706903  5.710609
6  2001  5.642708   5.644618  5.655694         5.485582  5.570320  5.624711
7  2008  5.339773   5.583232  5.631736         5.512735  5.579638  5.618732
8  2011  5.722271   5.992994  6.053358         5.934782  5.957945  6.016795
9  2002  5.944468   6.040686  6.158388         6.014440  6.041751  5.992603
```

10	1994	5.686988	5.749108	5.749513	5.684388	5.738186	5.740547
11	2012	5.800840	5.858877	5.717102	5.534151	5.655857	5.708191
12	2003	5.917285	6.041239	6.074704	6.039769	6.055665	6.020832
13	1997	5.874442	5.954476	5.954476	5.885571	5.942377	5.902609
14	2013	5.789969	6.077997	6.041088	5.994167	5.998266	6.027258
15	1985	5.986869	5.981768	6.198956	6.225009	6.223842	6.273875
16	2005	5.740048	5.667155	5.783724	5.672038	5.743663	5.749444
17	2006	6.014647	5.971307	5.954177	5.884840	5.924009	5.826800
18	2004	5.786630	5.725967	5.579516	5.644818	5.725954	5.707026
19	1972	6.611111	6.574306	6.518651	6.363056	6.392479	6.446815
20	1980	6.466667	6.176250	5.940750	6.079107	6.076300	6.041590
21	2007	6.139260	6.051230	6.009985	5.956683	5.979656	5.978314
22	1979	6.103333	6.187000	6.060625	5.898750	6.074631	6.099672
23	1984	5.813529	5.973134	5.973134	5.908531	5.983033	6.051578
24	1983	5.969231	6.030147	5.926378	5.895079	5.962373	6.067748
25	1995	5.645635	5.912071	6.049657	5.816067	5.897608	5.903801
26	1992	6.122527	6.209655	6.351954	5.903637	6.026580	6.008140
27	1981	6.200000	6.184127	6.184127	6.079293	6.108938	6.165285
28	1996	5.842857	5.859555	5.859555	5.761232	5.744980	5.711161
29	2000	6.003988	6.062584	6.010067	5.762267	5.894673	5.893981
30	1982	6.246032	6.293849	6.145387	5.949675	6.098438	6.150263
31	1998	5.818333	5.914970	5.935529	5.751871	5.884781	5.849274
32	1989	6.340000	6.253259	6.380674	6.114447	6.132331	6.138242
33	1991	5.226389	5.757188	5.645750	5.632679	5.711381	5.799003
34	1988	5.961420	6.060943	6.060943	5.721556	5.772032	5.820595
35	1987	6.466667	6.328175	6.237153	6.200556	6.199960	6.219666
36	1968	6.383333	6.541667	6.507143	6.530000	6.518571	6.456471
37	1974	6.089394	6.141364	6.410909	6.196162	6.301494	6.305739
38	1975	6.168750	6.495500	6.447188	6.401042	6.348915	6.364818
39	1962	6.170000	6.038000	6.298571	6.167500	6.188939	6.111948
40	1964	6.466667	6.364444	6.148333	6.143333	6.203750	6.323000
41	1971	6.533333	6.172917	6.191667	6.253788	6.322262	6.394479
42	1990	5.811111	5.954167	6.147917	6.072619	5.984524	6.036491
43	1961	6.033333	6.078571	6.052083	6.046970	6.155272	6.123950
44	1960	6.050000	6.444444	6.176190	6.353333	6.341685	6.391088
45	1976	5.960000	5.961000	6.115000	5.995909	6.065427	6.112705
46	1993	6.006111	6.024563	6.005265	5.882601	5.979854	5.952615
47	1967	6.085000	6.097333	6.183810	6.063667	6.199499	6.137093
48	1963	6.466667	6.510667	6.483889	6.436190	6.355333	6.416944
49	1986	6.085630	6.162954	6.178181	6.112331	6.138389	6.200124
50	1973	6.546154	6.487692	6.512637	6.516346	6.613819	6.631968
51	1970	6.620000	6.598000	6.632222	6.540833	6.564687	6.556961
52	1965	6.150000	6.156250	6.126389	6.194792	6.200167	6.269372
53	1969	5.040476	5.703571	5.692143	5.424725	5.539006	5.590655
54	1978	6.462626	6.214646	6.083983	5.815379	5.951461	5.832835
55	1966	5.959091	5.925433	5.917803	5.917541	6.006523	6.094963

	Comedy	Crime	Romance	War	Mystery	Thriller	Fantasy	\
0	6.089792	6.070865	6.073456	6.084876	6.054229	6.019411	6.019237	
1	6.139910	6.098711	6.118192	6.155106	6.128025	6.107857	6.119851	
2	6.455457	6.458757	6.462881	6.431407	6.381567	6.347344	6.292475	
3	5.915351	5.857387	5.880754	5.936226	5.904133	5.874240	5.887882	
4	5.891521	5.914869	5.920496	5.991447	5.978493	5.958634	5.977696	
5	5.737421	5.807580	5.825071	5.848784	5.899719	5.897055	5.894175	
6	5.599133	5.659866	5.691965	5.744620	5.756878	5.756150	5.775988	
7	5.642268	5.701592	5.744627	5.832073	5.844990	5.829843	5.804350	
8	6.010776	6.046886	6.036515	6.032365	6.031614	5.984627	5.989044	
9	5.972695	5.996626	5.996191	6.027351	6.028977	6.000109	6.015695	
10	5.740838	5.798881	5.823303	5.874709	5.903705	5.899567	5.886735	
11	5.738132	5.739671	5.755245	5.810500	5.744410	5.708884	5.757795	
12	5.978206	5.993329	5.948730	6.009375	6.000940	5.982677	5.990271	
13	5.929550	5.968182	5.967317	5.992524	6.027616	6.041017	5.983526	
14	6.019907	6.010341	6.042086	6.066429	6.031161	6.001376	6.003074	
15	6.240038	6.217204	6.218835	6.187100	6.230878	6.213251	6.195124	
16	5.737222	5.790318	5.805939	5.817700	5.813835	5.784723	5.753221	
17	5.818735	5.851344	5.881793	5.907906	5.890051	5.872128	5.876507	
18	5.728477	5.770508	5.797185	5.827558	5.805024	5.807176	5.819217	
19	6.434088	6.462517	6.508944	6.508944	6.511065	6.515599	6.497184	
20	6.053058	6.086669	6.077099	6.081517	6.056756	6.043761	6.071342	
21	5.952975	5.998072	5.986202	5.969141	5.971661	5.966341	5.965777	
22	6.128687	6.188990	6.130741	6.175359	6.194265	6.211102	6.211102	
23	6.023984	6.047003	6.058427	6.050562	6.021570	5.996725	5.979796	
24	6.033426	6.012814	5.955317	5.991926	6.007652	6.016366	5.948205	
25	5.916375	5.955679	5.996357	6.036721	6.073086	6.042140	6.025211	
26	6.020609	6.030602	6.060737	6.034050	6.025645	6.013777	6.030433	
27	6.195136	6.193000	6.163264	6.226925	6.201510	6.190386	6.208367	
28	5.720423	5.823234	5.829343	5.876754	5.896206	5.901178	5.873540	
29	5.869692	5.878317	5.829590	5.863289	5.860395	5.852907	5.863391	
30	6.130660	6.116981	6.119811	6.166920	6.167780	6.172680	6.248815	
31	5.870023	5.929882	5.982755	6.010563	6.041332	6.007768	5.981200	
32	6.125797	6.112906	6.114873	6.159752	6.109299	6.074977	6.089081	
33	5.720849	5.783750	5.787087	5.803556	5.798853	5.811092	5.813583	
34	5.854063	5.920374	5.937772	5.889170	5.948857	5.951254	5.994357	
35	6.223945	6.223252	6.240035	6.262751	6.285743	6.253894	6.239919	
36	6.450500	6.454242	6.461282	6.459286	6.519778	6.530260	6.492429	
37	6.313116	6.345580	6.332110	6.338568	6.385813	6.414477	6.407715	
38	6.373025	6.364354	6.395472	6.383559	6.371613	6.350357	6.375330	
39	6.059840	6.086145	6.115030	6.140937	6.186024	6.202700	6.212952	
40	6.307692	6.313438	6.269211	6.331667	6.300652	6.338036	6.316613	
41	6.393847	6.406511	6.398068	6.421420	6.435930	6.434041	6.446984	
42	6.052155	6.042199	6.069586	6.032684	6.057227	6.025401	6.042509	
43	6.203482	6.227126	6.225452	6.268011	6.287011	6.287011	6.240907	
44	6.396667	6.341481	6.368095	6.292609	6.292609	6.381282	6.400476	
45	6.127799	6.240115	6.242906	6.224913	6.282422	6.320470	6.313777	

46	5.962897	6.002018	6.034254	6.039288	6.060505	6.035589	5.999225
47	6.088333	6.169015	6.173234	6.214305	6.250282	6.237343	6.183564
48	6.303926	6.322716	6.282222	6.335169	6.324701	6.327605	6.329050
49	6.190361	6.175023	6.142381	6.147053	6.166853	6.156636	6.113713
50	6.704673	6.739311	6.755518	6.755518	6.742004	6.727399	6.715187
51	6.560083	6.537029	6.553765	6.571722	6.583384	6.577824	6.548761
52	6.324123	6.344814	6.289759	6.278966	6.240199	6.215172	6.215172
53	5.555538	5.702976	5.692049	5.650883	5.652418	5.625068	5.534524
54	5.841052	5.931187	5.948602	5.996473	6.017967	6.035483	6.009716
55	6.114111	6.142452	6.182990	6.157802	6.175740	6.184563	6.171855

	History	Animation	Horror	Music	Documentary	TV Movie	Foreign
0	6.048491	6.070349	6.006383	6.035249	6.067581	6.062360	6.062360
1	6.142278	6.191332	6.105528	6.118206	6.139459	6.123967	6.123967
2	6.299498	6.306529	6.269492	6.265119	6.278481	6.278481	6.278481
3	5.913488	5.949942	5.911518	5.910746	5.933692	5.958146	5.942033
4	6.061313	6.084201	6.017069	6.038106	6.056915	6.036536	6.024111
5	5.903773	5.934301	5.892867	5.928261	5.961138	5.968430	5.977818
6	5.804000	5.812125	5.793414	5.815391	5.830386	5.825237	5.821573
7	5.888302	5.893499	5.836996	5.889516	5.917534	5.895626	5.899119
8	5.993972	6.004565	5.967596	5.979290	6.021403	6.020301	6.046876
9	6.042355	6.049542	6.027826	6.060013	6.083378	6.052589	6.040041
10	5.916092	5.939305	5.935962	5.940521	5.960835	5.950028	5.949269
11	5.777794	5.797555	5.746362	5.792147	5.840970	5.835785	5.833340
12	6.013883	6.031790	6.005310	6.025567	6.046266	6.017563	5.998806
13	6.024836	6.040210	6.020252	6.016047	6.017379	6.018670	6.020947
14	6.019294	6.038919	5.958046	6.012502	6.062656	6.066222	6.066222
15	6.191298	6.180306	6.169017	6.170068	6.205567	6.205567	6.205567
16	5.792153	5.816428	5.739642	5.790225	5.818180	5.829305	5.838339
17	5.887547	5.897851	5.870194	5.897170	5.929215	5.906798	5.906897
18	5.834995	5.856918	5.828204	5.839444	5.874877	5.863023	5.907103
19	6.497184	6.490385	6.462511	6.504177	6.530524	6.530524	6.530524
20	6.081260	6.081260	6.085867	6.097547	6.097547	6.104733	6.102846
21	5.974974	5.983974	5.940009	5.964251	5.980367	5.991857	5.979805
22	6.218047	6.212729	6.199739	6.229461	6.229461	6.229461	6.229461
23	6.017790	6.003142	5.980389	6.010487	6.033987	6.043746	6.043746
24	5.976023	6.010855	6.016098	6.015821	6.015821	6.006443	5.993828
25	6.048473	6.063107	6.028548	6.028548	6.056901	6.051493	6.038214
26	6.045608	6.048240	6.029327	6.044890	6.069973	6.069973	6.083411
27	6.231039	6.248548	6.207477	6.265178	6.262920	6.230940	6.230940
28	5.884364	5.893291	5.872460	5.895467	5.920108	5.923210	5.945769
29	5.872880	5.885675	5.855606	5.901462	5.918349	5.906658	5.904621
30	6.267730	6.328428	6.290219	6.306812	6.321234	6.315497	6.317857
31	5.997152	6.008550	5.961916	6.005231	6.011336	5.977549	5.957353
32	6.120967	6.144326	6.112431	6.135417	6.159458	6.159458	6.163783
33	5.840245	5.855421	5.823265	5.825837	5.848539	5.848539	5.872803
34	5.996919	6.023633	5.989462	6.001866	6.047838	6.048653	6.040212

35	6.278036	6.286713	6.274051	6.291720	6.291720	6.309443	6.309443
36	6.506184	6.500897	6.503293	6.505595	6.505595	6.505595	6.505595
37	6.375074	6.378362	6.391124	6.417526	6.441729	6.414135	6.414135
38	6.397982	6.377573	6.374780	6.380406	6.396798	6.396798	6.396798
39	6.224187	6.224187	6.196634	6.172493	6.172493	6.172493	6.182126
40	6.338333	6.341618	6.315270	6.307875	6.307875	6.307875	6.290610
41	6.448459	6.452344	6.450197	6.453764	6.453764	6.461816	6.461816
42	6.053691	6.062333	6.050870	6.059587	6.083655	6.065445	6.073545
43	6.257623	6.268322	6.279383	6.274847	6.274847	6.274847	6.247768
44	6.389140	6.389140	6.438788	6.443529	6.443529	6.443529	6.399429
45	6.339368	6.339368	6.344153	6.336285	6.351370	6.341339	6.333919
46	6.034624	6.057636	6.041505	6.066382	6.088104	6.088104	6.045143
47	6.183564	6.178789	6.179035	6.148107	6.166446	6.166446	6.166446
48	6.323249	6.334330	6.328393	6.328393	6.328393	6.328393	6.355142
49	6.105229	6.144944	6.116408	6.106972	6.106972	6.106972	6.106972
50	6.735988	6.741181	6.709581	6.714009	6.720509	6.702275	6.702275
51	6.550040	6.559129	6.546704	6.542589	6.545868	6.544932	6.538033
52	6.193912	6.236520	6.206338	6.246536	6.246536	6.282154	6.285175
53	5.574435	5.595890	5.589009	5.582831	5.582831	5.607117	5.607117
54	6.009264	6.018145	6.018086	6.079001	6.081373	5.987964	6.015455
55	6.166778	6.214107	6.146256	6.146256	6.169912	6.169912	6.187370

[233]:
```
# Setting 'year' column as index
year_genre_df = year_genre_df.set_index('Year')
```

[239]:
```
year_genre_df.sort_index(inplace=True)
```

[270]:
```
popular_genres = year_genre_df.idxmax(axis='columns')
```

[272]:
```
unpopular_genres = year_genre_df.idxmin(axis='columns')
```

[283]:
```
pop_genres = pd.DataFrame(popular_genres, columns=['Best Genre'])
```

[284]:
```
pop_genres = pd.concat([pop_genres, pd.DataFrame(unpopular_genres,␣
↪columns=['Worst Genre'])], axis=1)
```

[285]:
```
pop_genres
```

[285]:

	Best Genre	Worst Genre
Year		
1960	Adventure	Action
1961	Mystery	Action
1962	Western	Adventure
1963	Adventure	Romance
1964	Action	Science Fiction
1965	Crime	Western
1966	Animation	Science Fiction

1967	Mystery	Science Fiction
1968	Adventure	Action
1969	Adventure	Action
1970	Western	Crime
1971	Action	Adventure
1972	Action	Science Fiction
1973	Romance	Adventure
1974	Documentary	Action
1975	Adventure	Action
1976	Documentary	Action
1977	Action	Music
1978	Action	Science Fiction
1979	Music	Science Fiction
1980	Action	Western
1981	Music	Science Fiction
1982	Animation	Science Fiction
1983	Family	Science Fiction
1984	Romance	Action
1985	Family	Adventure
1986	Family	Action
1987	Action	Drama
1988	Adventure	Science Fiction
1989	Western	Thriller
1990	Western	Action
1991	Foreign	Action
1992	Western	Science Fiction
1993	Documentary	Science Fiction
1994	Documentary	Science Fiction
1995	Mystery	Action
1996	Foreign	Family
1997	Thriller	Action
1998	Mystery	Science Fiction
1999	Foreign	Science Fiction
2000	Adventure	Science Fiction
2001	Documentary	Science Fiction
2002	Western	Action
2003	Western	Action
2004	Foreign	Western
2005	Foreign	Adventure
2006	Action	Comedy
2007	Action	Horror
2008	Documentary	Action
2009	Adventure	Crime
2010	Adventure	Action
2011	Western	Action
2012	Adventure	Science Fiction
2013	Adventure	Action

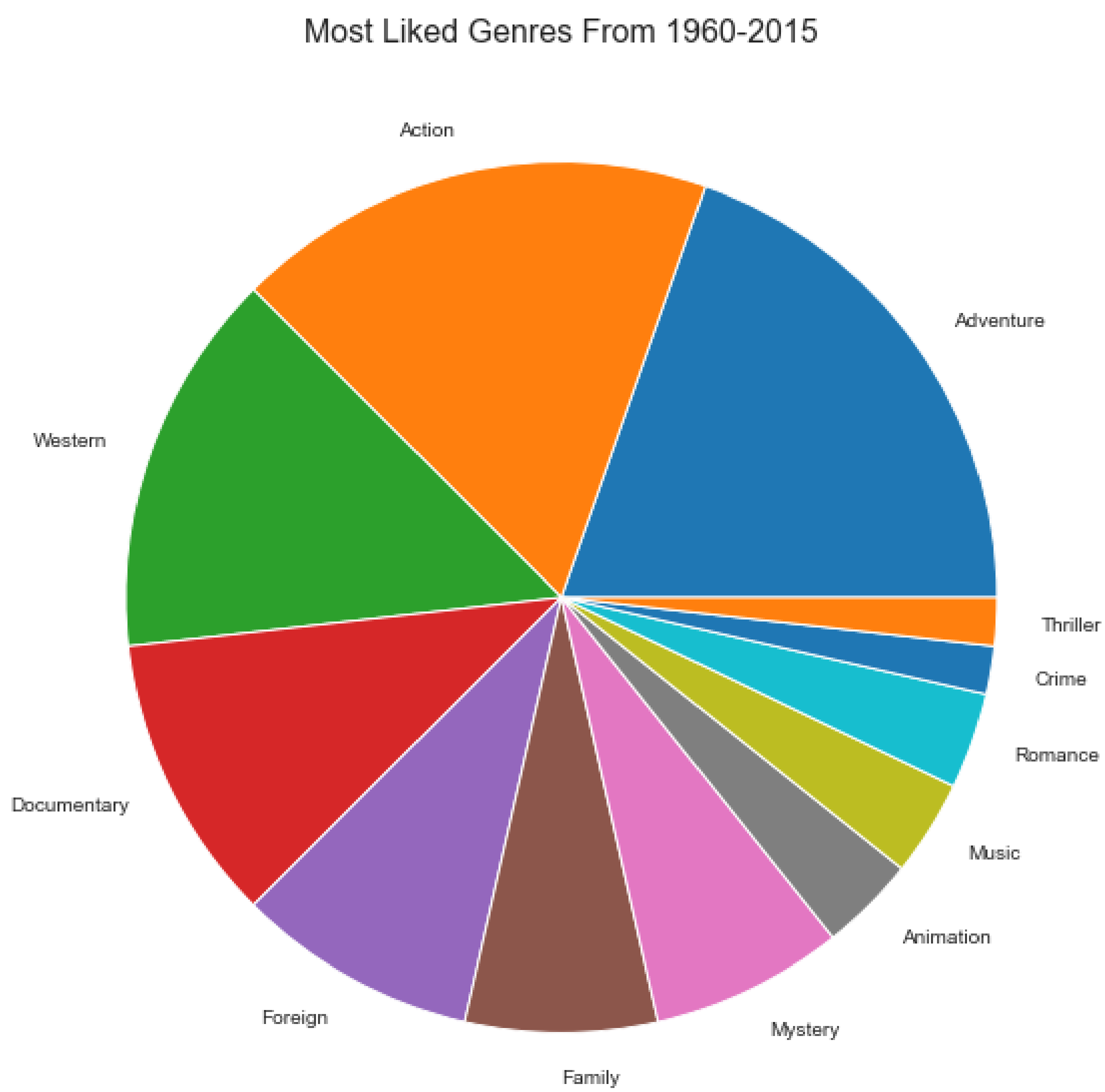
```
2014       Action            Drama
2015       Family           Action
```

[288]:
```python
# Save dataframe as a csv file
pop_genres.to_csv('genres_popularity.csv')
```

[389]:
```python
sns.set_style('darkgrid')
```

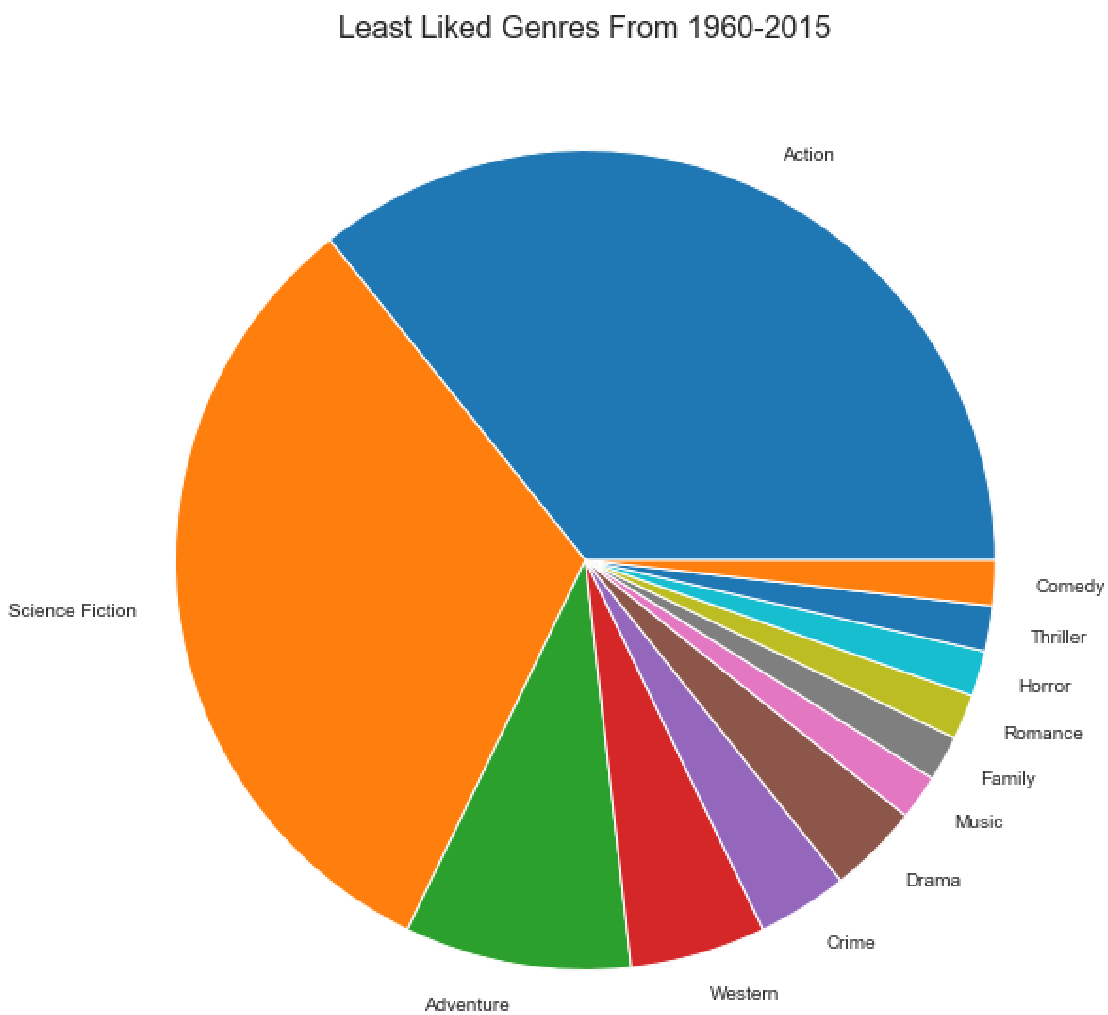
[390]:
```python
# plot a pie chart to show the most popular movies over the years
pop_genres['Best Genre'].value_counts().plot.pie(label='', figsize=(12,10))
plt.title('Most Liked Genres From 1960-2015', fontsize=16)
```

[390]:
```
Text(0.5, 1.0, 'Most Liked Genres From 1960-2015')
```



[391]:
```
# plot a pie chart to show the most popular movies over the years
pop_genres['Worst Genre'].value_counts().plot.pie(label='', figsize=(12,10))
plt.title('Least Liked Genres From 1960-2015', fontsize=16)
```

[391]: Text(0.5, 1.0, 'Least Liked Genres From 1960-2015')



1.1.4 Research Question 2: What kinds of properties are associated with movies that have high revenues?

[306]:
```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10865 entries, 0 to 10865
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              10865 non-null  int64
 1   popularity      10865 non-null  float64
 2   budget          5169 non-null   float64
 3   revenue         4849 non-null   float64
 4   original_title  10865 non-null  object
 5   cast_1          10789 non-null  object
 6   cast_2          10645 non-null  object
 7   cast_3          10555 non-null  object
 8   cast_4          10446 non-null  object
 9   cast_5          10133 non-null  object
 10  director        10821 non-null  object
 11  runtime         10834 non-null  float64
 12  genre_1         10842 non-null  object
 13  genre_2         8514 non-null   object
 14  genre_3         5078 non-null   object
 15  genre_4         1980 non-null   object
 16  genre_5         541 non-null    object
 17  company_1       9835 non-null   object
 18  company_2       6395 non-null   object
 19  company_3       3816 non-null   object
 20  company_4       2053 non-null   object
 21  company_5       1126 non-null   object
 22  release_date    10865 non-null  datetime64[ns]
 23  vote_count      10865 non-null  int64
 24  vote_average    10865 non-null  float64
 25  release_year    10865 non-null  int64
 26  budget_adj      5169 non-null   float64
 27  revenue_adj     4849 non-null   float64
dtypes: datetime64[ns](1), float64(7), int64(3), object(17)
memory usage: 2.4+ MB
```

[311]:
```
# drop rows where data about budget and revenue is nor available
revenue_df = df_clean[df_clean.budget_adj.notna() & df_clean.revenue_adj.
 ↪notna()]
```

[328]:
```
# suppress scientific notation
pd.set_option('display.float_format', lambda x: '%.5f' % x)
pd.options.display.float_format = '{:,}'.format
```

[329]:
```
revenue_df
```

[329]:

```
           id           popularity        budget         revenue  \
0      135397            32.985763 150,000,000.0 1,513,528,810.0
1       76341            28.419936 150,000,000.0   378,436,354.0
2      262500            13.112507 110,000,000.0   295,238,201.0
3      140607   11.173103999999999 200,000,000.0 2,068,178,225.0
4      168259             9.335014 190,000,000.0 1,506,249,360.0
…        …                   …             …               …
10822     396    0.670273999999999   7,500,000.0    33,736,689.0
10828    5780 0.40273000000000003   3,000,000.0    13,000,000.0
10829    6644 0.39566799999999996   4,653,000.0     6,000,000.0
10835    5923 0.29991100000000004  12,000,000.0    20,000,000.0
10848    2161 0.20725700000000002   5,115,000.0    12,000,000.0

                        original_title           cast_1  \
0                       Jurassic World      Chris Pratt
1                   Mad Max: Fury Road        Tom Hardy
2                            Insurgent  Shailene Woodley
3         Star Wars: The Force Awakens    Harrison Ford
4                            Furious 7       Vin Diesel
…                                    …                …
10822  Who's Afraid of Virginia Woolf?  Elizabeth Taylor
10828                     Torn Curtain      Paul Newman
10829                        El Dorado       John Wayne
10835                  The Sand Pebbles    Steve McQueen
10848                Fantastic Voyage     Stephen Boyd

                      cast_2             cast_3               cast_4  \
0        Bryce Dallas Howard        Irrfan Khan   Vincent D'Onofrio
1            Charlize Theron  Hugh Keays-Byrne      Nicholas Hoult
2                 Theo James       Kate Winslet        Ansel Elgort
3                Mark Hamill      Carrie Fisher         Adam Driver
4                Paul Walker      Jason Statham  Michelle Rodriguez
…                          …                  …                    …
10822         Richard Burton       George Segal        Sandy Dennis
10828          Julie Andrews       Lila Kedrova      HansjÃ¶rg Felmy
10829         Robert Mitchum         James Caan       Charlene Holt
10835  Richard Attenborough     Richard Crenna      Candice Bergen
10848           Raquel Welch      Edmond O'Brien   Donald Pleasence

                   cast_5           director  runtime    genre_1  \
0           Nick Robinson    Colin Trevorrow    124.0     Action
1             Josh Helman      George Miller    120.0     Action
2            Miles Teller   Robert Schwentke    119.0  Adventure
3            Daisy Ridley        J.J. Abrams    136.0     Action
4          Dwayne Johnson          James Wan    137.0     Action
…                       …                  …       …          …
10822      Agnes Flanagan       Mike Nichols    131.0      Drama
```

```
10828  Tamara Toumanova   Alfred Hitchcock    128.0    Mystery
10829           Paul Fix       Howard Hawks    120.0     Action
10835  Emmanuelle Arsan        Robert Wise    182.0     Action
10848  Arthur O'Connell  Richard Fleischer    100.0  Adventure

               genre_2          genre_3   genre_4  genre_5  \
0            Adventure  Science Fiction  Thriller      NaN
1            Adventure  Science Fiction  Thriller      NaN
2      Science Fiction         Thriller       NaN      NaN
3            Adventure  Science Fiction   Fantasy      NaN
4                Crime         Thriller       NaN      NaN
…                    …                …         …        …
10822              NaN              NaN       NaN      NaN
10828         Thriller              NaN       NaN      NaN
10829          Western              NaN       NaN      NaN
10835        Adventure            Drama       War  Romance
10848  Science Fiction              NaN       NaN      NaN

                                    company_1                    company_2  \
0                           Universal Studios         Amblin Entertainment
1                    Village Roadshow Pictures  Kennedy Miller Productions
2                       Summit Entertainment              Mandeville Films
3                                   Lucasfilm       Truenorth Productions
4                          Universal Pictures                Original Film
…                                           …                            …
10822                    Chenault Productions                          NaN
10828                      Universal Pictures                          NaN
10829                      Paramount Pictures           Laurel Productions
10835  Twentieth Century Fox Film Corporation            Solar Productions
10848  Twentieth Century Fox Film Corporation                          NaN

                     company_3                 company_4       company_5  \
0           Legendary Pictures  Fuji Television Network          Dentsu
1                          NaN                       NaN             NaN
2      Red Wagon Entertainment                   NeoReel             NaN
3                    Bad Robot                       NaN             NaN
4         Media Rights Capital                    Dentsu  One Race Films
…                            …                         …               …
10822                      NaN                       NaN             NaN
10828                      NaN                       NaN             NaN
10829                      NaN                       NaN             NaN
10835  Robert Wise Productions                       NaN             NaN
10848                      NaN                       NaN             NaN

      release_date  vote_count  vote_average  release_year  \
0       2015-06-09        5562           6.5          2015
1       2015-05-13        6185           7.1          2015
```

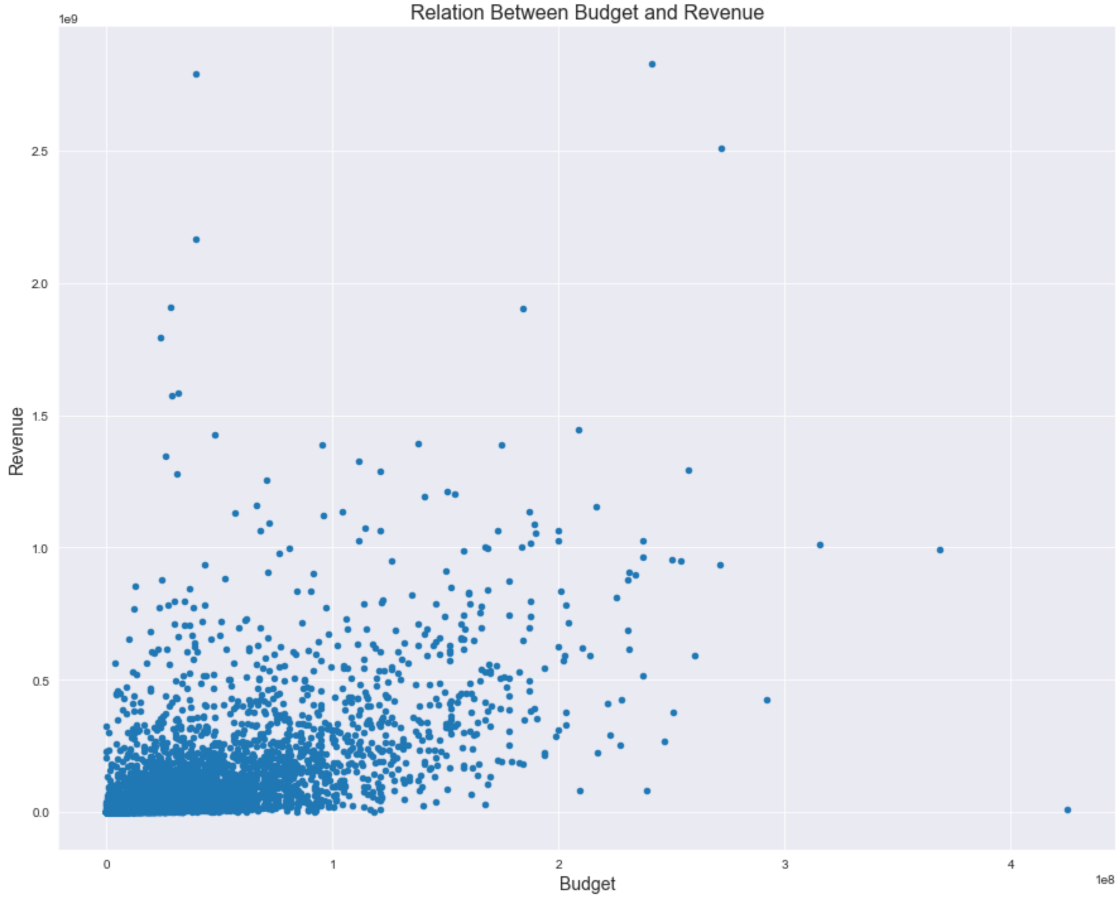
```
2       2015-03-18        2480           6.3          2015
3       2015-12-15        5292           7.5          2015
4       2015-04-01        2947           7.3          2015
…            …         …           …           …
10822   2066-06-21          74           7.5          1966
10828   2066-07-13          46           6.3          1966
10829   2066-12-17          36           6.9          1966
10835   2066-12-20          28           7.0          1966
10848   2066-08-24          42           6.7          1966

                budget_adj          revenue_adj
0        137,999,939.280026   1,392,445,892.5238
1        137,999,939.280026   348,161,292.489031
2      101,199,955.47201899   271,619,025.407628
3        183,999,919.040035 1,902,723,129.80182
4      174,799,923.08803302 1,385,748,801.47052
…                     …                  …
10822   50,385,110.1922359   226,643,572.371492
10828   20,154,044.0768943 87,334,190.99987571
10829   31,258,922.3632632   40,308,088.1537887
10835   80,616,176.3075775 134,360,293.84596202
10848 34,362,645.151104905   80,616,176.3075775

[3854 rows x 28 columns]
```

[319]:
```
revenue_df.budget_adj.corr(revenue_df.revenue_adj)
```

[319]: 0.5704510195812402

[392]:
```
# Scatter plot to represent relatinship between budget and revenue of a movie
revenue_df.plot.scatter(x='budget_adj', y='revenue_adj', figsize=(15,12))
plt.title("Relation Between Budget and Revenue", fontsize=16)
plt.xlabel("Budget", fontsize=14)
plt.ylabel("Revenue", fontsize=14)
```

[392]: Text(0, 0.5, 'Revenue')



The above scatter plot and correlation of 0.5705 shows that amount spent on making a movie is somewhat correlated to the movie achieving high revenues

[332]:
```
revenue_df.revenue_adj.describe()
```

[332]:
```
count                  3,854.0
mean     137,064,690.30304146
std      216,111,351.44431075
min          2.37070528956505
25%       18,357,350.356732048
50%        61,730,679.07895175
75%          163,257,654.555831
max       2,827,123,750.41189
Name: revenue_adj, dtype: float64
```

[342]:
```
# classifying movies into categories depending on their revenue
edges= [2.37070528956505, 18357350.356732048, 61730679.07895175, 163257654.
↪555831, 2827123750.41189]
ratings =['Low', 'Below Average', 'Above Average', 'High']
```

```python
# create a new column to store the movie category in
revenue_df['Movie_Rating'] = pd.cut(revenue_df.revenue_adj, bins=edges,␣
↪labels=ratings)
```

```
<ipython-input-342-783556001afe>: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
  revenue_df['Movie_Rating'] = pd.cut(revenue_df.revenue_adj, bins=edges,
labels=ratings)
```

[348]:
```python
revenue_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3854 entries, 0 to 10848
Data columns (total 29 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              3854 non-null   int64
 1   popularity      3854 non-null   float64
 2   budget          3854 non-null   float64
 3   revenue         3854 non-null   float64
 4   original_title  3854 non-null   object
 5   cast_1          3850 non-null   object
 6   cast_2          3846 non-null   object
 7   cast_3          3846 non-null   object
 8   cast_4          3838 non-null   object
 9   cast_5          3816 non-null   object
 10  director        3853 non-null   object
 11  runtime         3854 non-null   float64
 12  genre_1         3854 non-null   object
 13  genre_2         3205 non-null   object
 14  genre_3         2112 non-null   object
 15  genre_4         873 non-null    object
 16  genre_5         259 non-null    object
 17  company_1       3808 non-null   object
 18  company_2       2924 non-null   object
 19  company_3       1972 non-null   object
 20  company_4       1188 non-null   object
 21  company_5       692 non-null    object
 22  release_date    3854 non-null   datetime64[ns]
 23  vote_count      3854 non-null   int64
 24  vote_average    3854 non-null   float64
 25  release_year    3854 non-null   int64
 26  budget_adj      3854 non-null   float64
 27  revenue_adj     3854 non-null   float64
```

```
 28  Movie_Rating    3853 non-null   category
dtypes: category(1), datetime64[ns](1), float64(7), int64(3), object(17)
memory usage: 877.1+ KB
```

[371]:

```
properties_list= []
for rating in ratings:
    temp_df = revenue_df[revenue_df.Movie_Rating == rating]
    avg_runtime = temp_df.runtime.mean()
    avg_rating = temp_df.vote_average.mean()
    avg_budget = temp_df.budget_adj.mean()
    frequent_actor = temp_df.loc[:, 'cast_1':'cast_5'].stack().value_counts().
 ↪idxmax()
    frequent_director = temp_df.director.value_counts().idxmax()
    frequent_genre = temp_df.loc[:, 'genre_1':'genre_5'].stack().value_counts().
 ↪idxmax()
    prod_comp = temp_df.loc[:, 'company_1':'company_5'].stack().value_counts().
 ↪idxmax()
    properties_list.append(
    {'Revenue_Bin' : rating,
     'Average_Runtime' : avg_runtime,
     'Average_Rate' : avg_rating,
     'Average_Budget' : avg_budget,
     'Frequent_Actor' : frequent_actor,
     'Frequent_Director' : frequent_director,
     'Frequent_Genre' : frequent_genre,
     'Production_Company' : prod_comp
    })
```

[372]:

```
properties_df = pd.DataFrame(properties_list)
```

[375]:

```
properties_df.round(2)
```

[375]:

```
     Revenue_Bin  Average_Runtime  Average_Rate  Average_Budget  \
0            Low           103.22          5.96    16,000,828.2
1  Below Average            106.4          6.07   29,470,726.03
2  Above Average            109.6          6.19   44,490,497.99
3           High           117.66          6.46   86,992,079.79

  Frequent_Actor  Frequent_Director Frequent_Genre  Production_Company
0   Willem Dafoe  Richard Linklater          Drama        Warner Bros.
1 Robert De Niro         Wes Craven          Drama  Universal Pictures
2 Robert De Niro     Clint Eastwood          Drama        Warner Bros.
3     Tom Cruise   Steven Spielberg         Action        Warner Bros.
```

[399]:

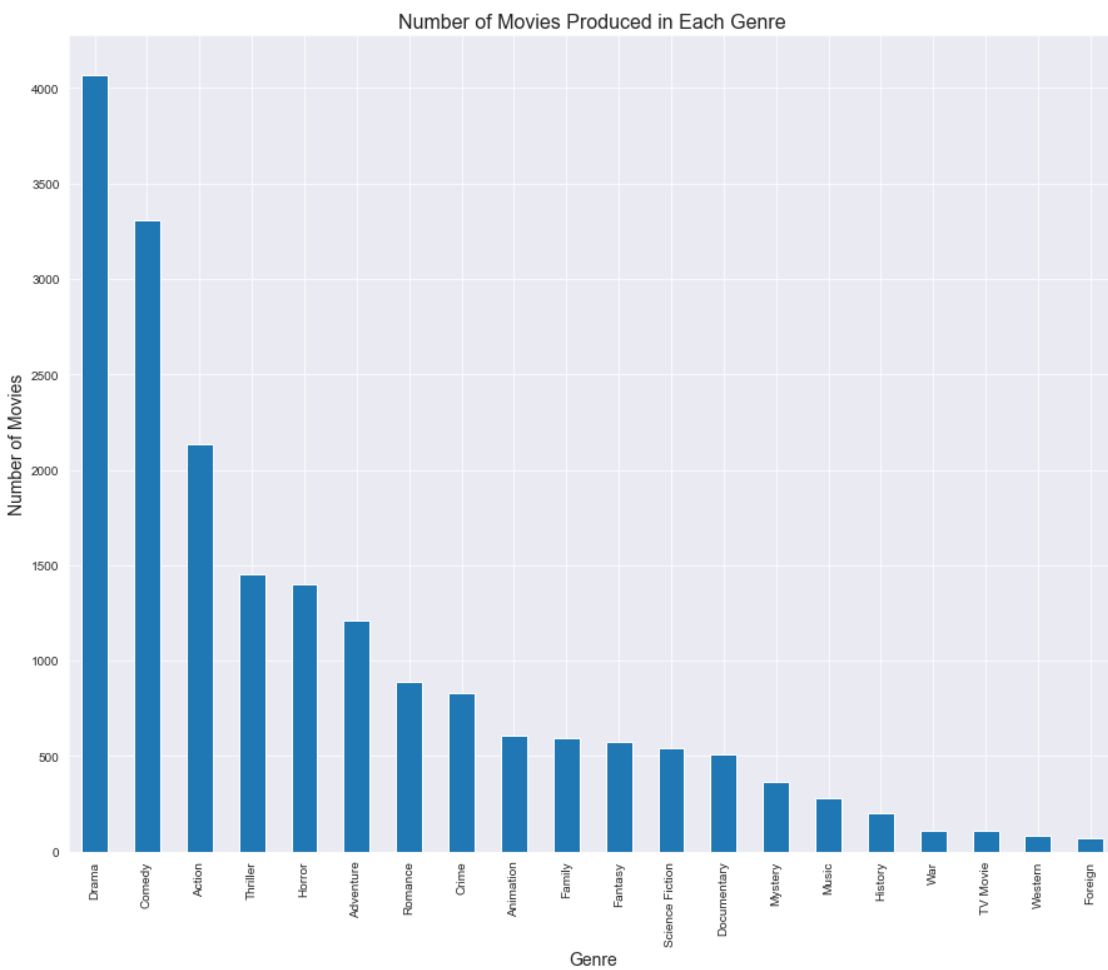
```
properties_df.to_csv('properties.csv')
```

1.1.5 Research Question 3: What are the most produced genres over time?

[395]:
```
# create a series of number of movies produced in each genre
prod_gen= df_clean.loc[:, 'genre_1':'genre_2'].stack().value_counts()
```

[398]:
```
# Plot a bar chart showing number of movies in each genre
prod_gen.plot.bar(figsize=(15,12))
plt.title('Number of Movies Produced in Each Genre', fontsize=16)
plt.xlabel('Genre',fontsize=14)
plt.ylabel('Number of Movies',fontsize=14)
```

[398]: Text(0, 0.5, 'Number of Movies')



Conclusions

Note: Every movie in this dataset has more than one genre listed, so when calculating the any average for each genre through the years movies may be calculated more than once accross different genres.

In the first part of the analysis I attempted to find the most popular genres from year to year. The results of the analysis were based on the average vote by users for each genre. It showed over the years people favored action and adventure movies. It is note worthy that western movies was close to them and that is due to low amount of movies produced in that particular genre.

In the second part of the analysis I wanted to find properties associated with high revenue. I questioned wether if spending more on the movie will yield higher revenue. I found out that this is somewhat true, as the correlation between these two variables where close to 0.5. Other factors that affect the revenues such as director, genre or the presence of a certain actor in the cast that attracts people to see the movie.

The last part examined the number of movies produced over the years. This was to give a sense on how the numbers of movies relate to the whole analysis.

[ ]: