Investigate a dataset

Dataset investigated

- TMDb movie data

Questions to investigate

1. Number of films per year

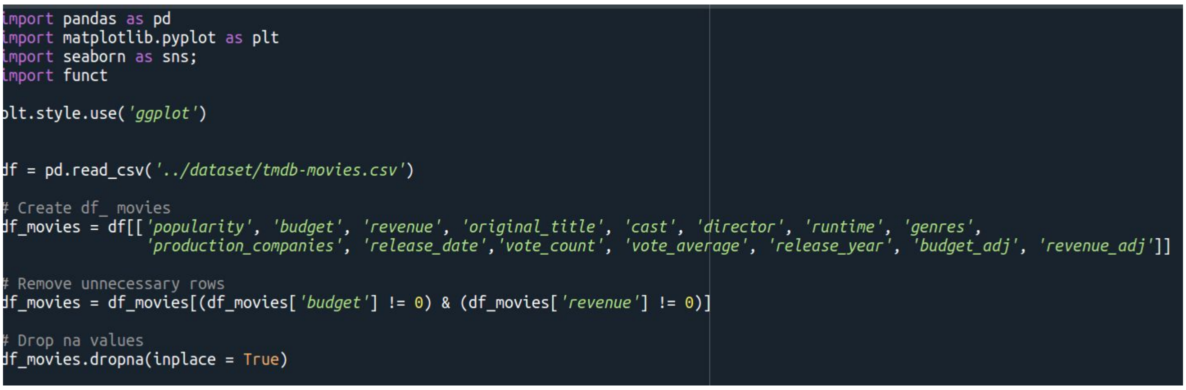
2. Which genres are most popular from years

3. What kinds of properties are associated with movies that have high revenues

    - Calculate correlation
    - See relationships between columns with more correlation
    - review the relationships between genders, directors, actors and producers

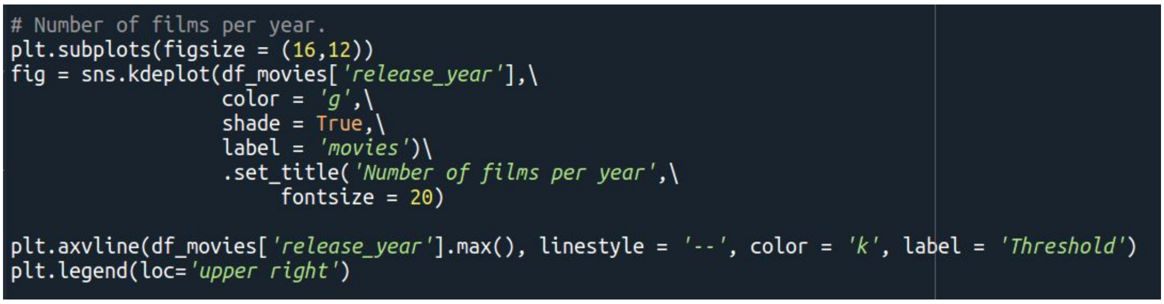
Description of steps

1. Number of films per year:
   First use the necessary columns, then remove rows with "0" in budget or revenue, lastly remove null values.



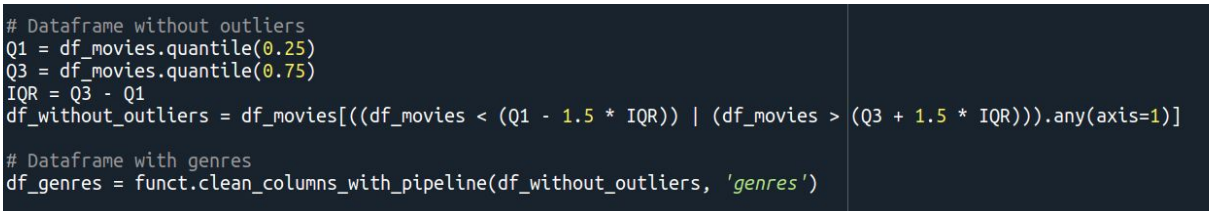
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns;
import funct

plt.style.use('ggplot')


df = pd.read_csv('../dataset/tmdb-movies.csv')

# Create df_ movies
df_movies = df[['popularity', 'budget', 'revenue', 'original_title', 'cast', 'director', 'runtime', 'genres',
                'production_companies', 'release_date','vote_count', 'vote_average', 'release_year', 'budget_adj', 'revenue_adj']]

# Remove unnecessary rows
df_movies = df_movies[(df_movies['budget'] != 0) & (df_movies['revenue'] != 0)]

# Drop na values
df_movies.dropna(inplace = True)
```

Then plot:

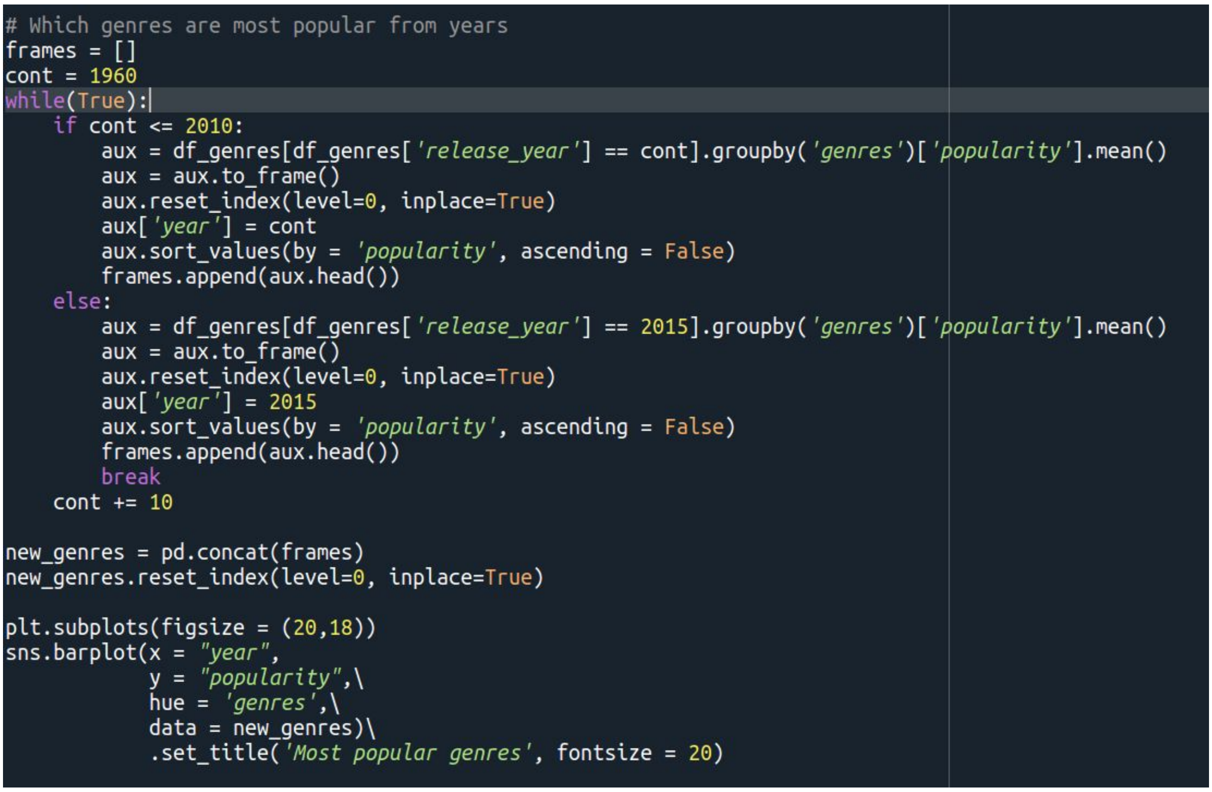


```
# Number of films per year.
plt.subplots(figsize = (16,12))
fig = sns.kdeplot(df_movies['release_year'],\
                  color = 'g',\
                  shade = True,\
                  label = 'movies')\
                  .set_title('Number of films per year',\
                       fontsize = 20)

plt.axvline(df_movies['release_year'].max(), linestyle = '--', color = 'k', label = 'Threshold')
plt.legend(loc='upper right')
```

2. Which genres are most popular from years:
   remove outliers, since we are going to calculate the mean, and create a new df that we are going to use to plot

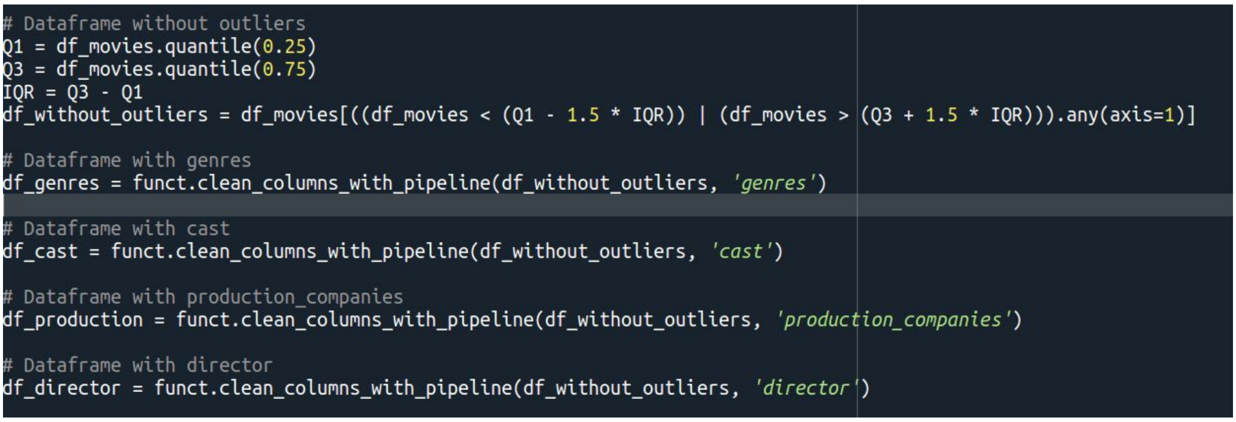


```
# Dataframe without outliers
Q1 = df_movies.quantile(0.25)
Q3 = df_movies.quantile(0.75)
IQR = Q3 - Q1
df_without_outliers = df_movies[((df_movies < (Q1 - 1.5 * IQR)) | (df_movies > (Q3 + 1.5 * IQR))).any(axis=1)]

# Dataframe with genres
df_genres = funct.clean_columns_with_pipeline(df_without_outliers, 'genres')
```



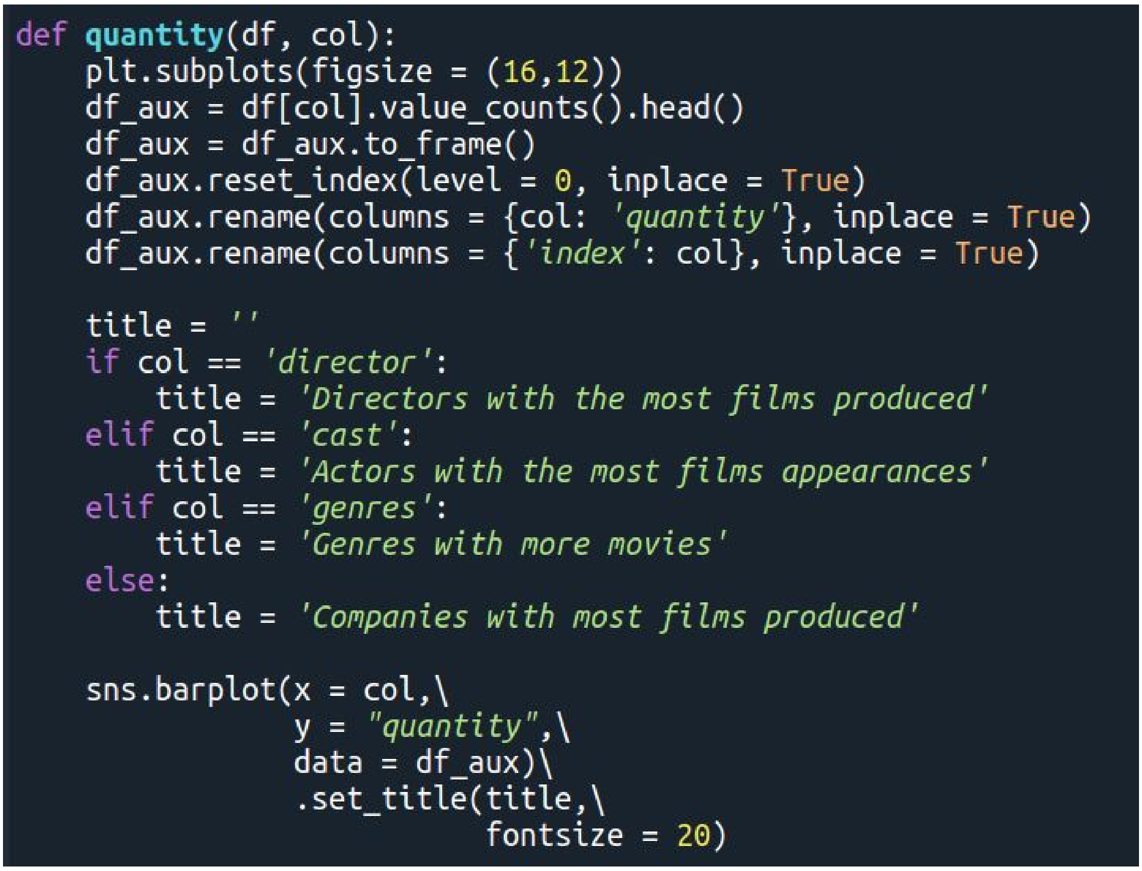
```
# Which genres are most popular from years
frames = []
cont = 1960
while(True):
    if cont <= 2010:
        aux = df_genres[df_genres['release_year'] == cont].groupby('genres')['popularity'].mean()
        aux = aux.to_frame()
        aux.reset_index(level=0, inplace=True)
        aux['year'] = cont
        aux.sort_values(by = 'popularity', ascending = False)
        frames.append(aux.head())
    else:
        aux = df_genres[df_genres['release_year'] == 2015].groupby('genres')['popularity'].mean()
        aux = aux.to_frame()
        aux.reset_index(level=0, inplace=True)
        aux['year'] = 2015
        aux.sort_values(by = 'popularity', ascending = False)
        frames.append(aux.head())
        break
    cont += 10

new_genres = pd.concat(frames)
new_genres.reset_index(level=0, inplace=True)

plt.subplots(figsize = (20,18))
sns.barplot(x = "year",
            y = "popularity",\
            hue = 'genres',\
            data = new_genres)\
            .set_title('Most popular genres', fontsize = 20)
```

3. What kinds of properties are associated with movies that have high revenues: remove outliers, since we are going to calculate the mean, and create a new df that we are going to use to plot, create respective functions.



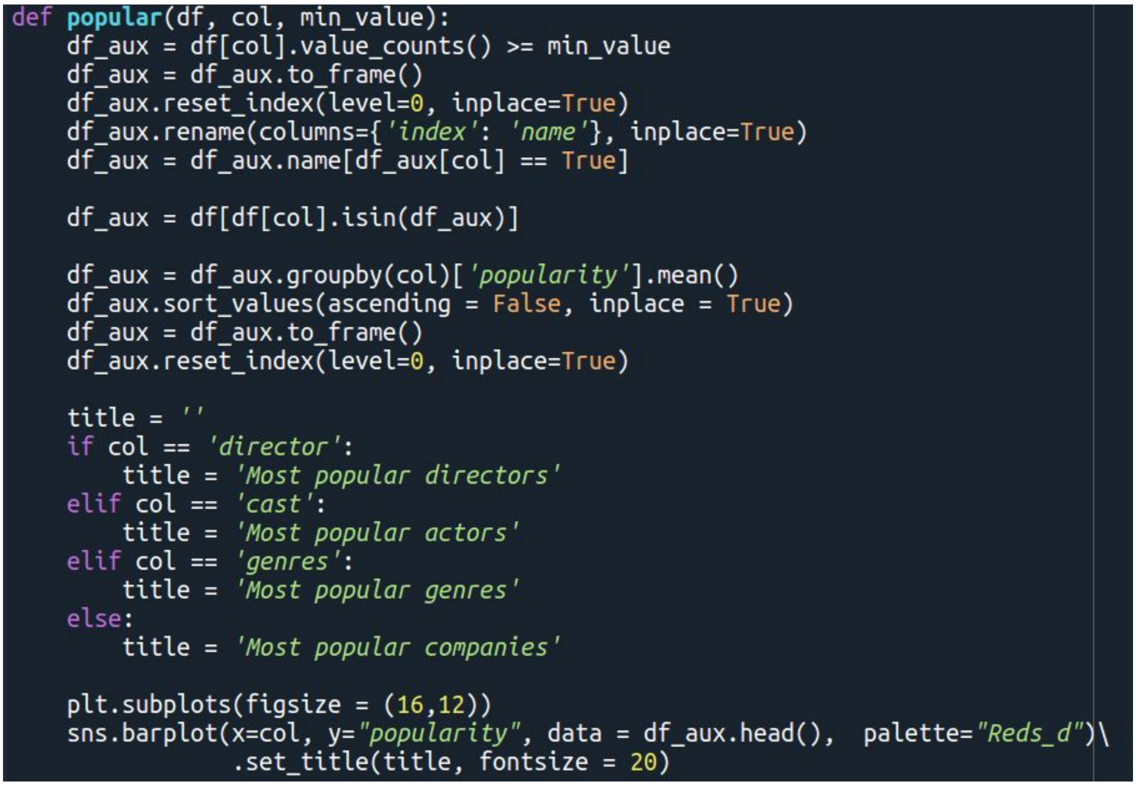
```
# Dataframe without outliers
Q1 = df_movies.quantile(0.25)
Q3 = df_movies.quantile(0.75)
IQR = Q3 - Q1
df_without_outliers = df_movies[((df_movies < (Q1 - 1.5 * IQR)) | (df_movies > (Q3 + 1.5 * IQR))).any(axis=1)]

# Dataframe with genres
df_genres = funct.clean_columns_with_pipeline(df_without_outliers, 'genres')

# Dataframe with cast
df_cast = funct.clean_columns_with_pipeline(df_without_outliers, 'cast')

# Dataframe with production_companies
df_production = funct.clean_columns_with_pipeline(df_without_outliers, 'production_companies')

# Dataframe with director
df_director = funct.clean_columns_with_pipeline(df_without_outliers, 'director')
```

Quantity:



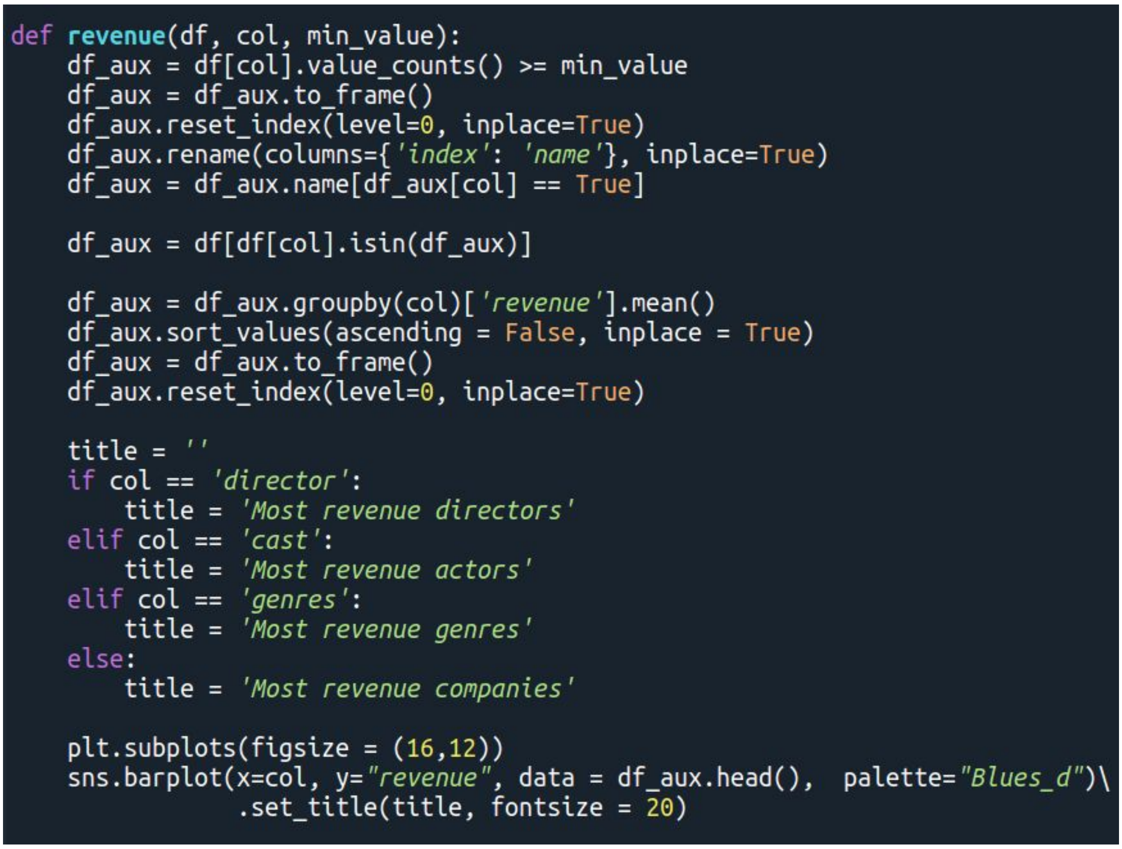
```
def quantity(df, col):
    plt.subplots(figsize = (16,12))
    df_aux = df[col].value_counts().head()
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level = 0, inplace = True)
    df_aux.rename(columns = {col: 'quantity'}, inplace = True)
    df_aux.rename(columns = {'index': col}, inplace = True)

    title = ''
    if col == 'director':
        title = 'Directors with the most films produced'
    elif col == 'cast':
        title = 'Actors with the most films appearances'
    elif col == 'genres':
        title = 'Genres with more movies'
    else:
        title = 'Companies with most films produced'

    sns.barplot(x = col,\
                y = "quantity",\
                data = df_aux)\
                .set_title(title,\
                           fontsize = 20)
```

Popularity:



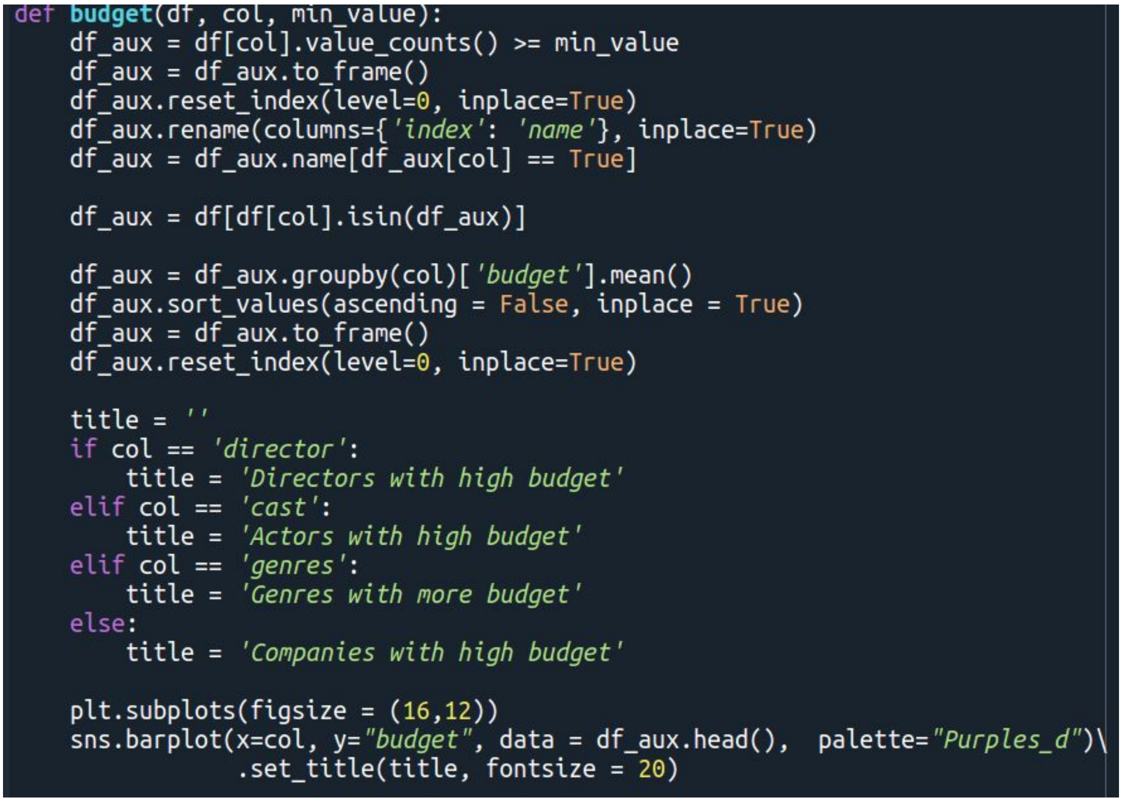
```
def popular(df, col, min_value):
    df_aux = df[col].value_counts() >= min_value
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)
    df_aux.rename(columns={'index': 'name'}, inplace=True)
    df_aux = df_aux.name[df_aux[col] == True]

    df_aux = df[df[col].isin(df_aux)]

    df_aux = df_aux.groupby(col)['popularity'].mean()
    df_aux.sort_values(ascending = False, inplace = True)
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)

    title = ''
    if col == 'director':
        title = 'Most popular directors'
    elif col == 'cast':
        title = 'Most popular actors'
    elif col == 'genres':
        title = 'Most popular genres'
    else:
        title = 'Most popular companies'

    plt.subplots(figsize = (16,12))
    sns.barplot(x=col, y="popularity", data = df_aux.head(),  palette="Reds_d")\
                .set_title(title, fontsize = 20)
```

Revenue:



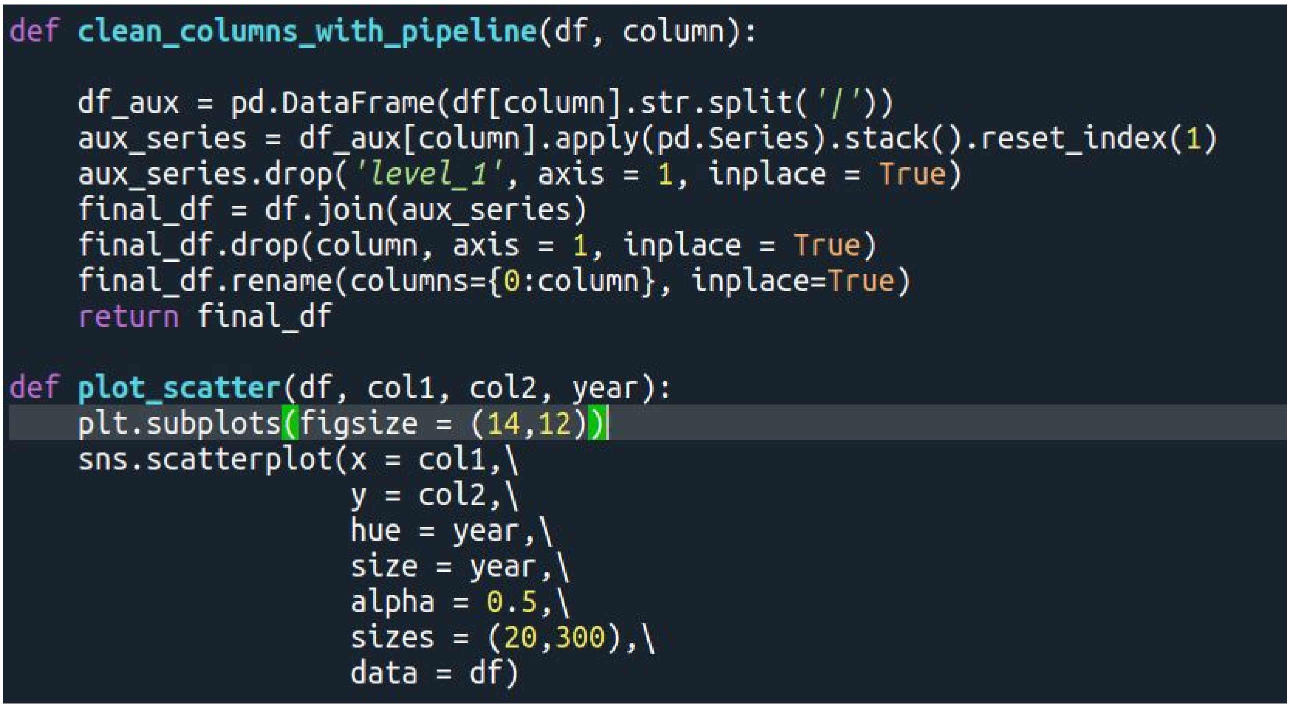
```
def revenue(df, col, min_value):
    df_aux = df[col].value_counts() >= min_value
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)
    df_aux.rename(columns={'index': 'name'}, inplace=True)
    df_aux = df_aux.name[df_aux[col] == True]

    df_aux = df[df[col].isin(df_aux)]

    df_aux = df_aux.groupby(col)['revenue'].mean()
    df_aux.sort_values(ascending = False, inplace = True)
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)

    title = ''
    if col == 'director':
        title = 'Most revenue directors'
    elif col == 'cast':
        title = 'Most revenue actors'
    elif col == 'genres':
        title = 'Most revenue genres'
    else:
        title = 'Most revenue companies'

    plt.subplots(figsize = (16,12))
    sns.barplot(x=col, y="revenue", data = df_aux.head(),  palette="Blues_d")\
                .set_title(title, fontsize = 20)
```

Budget:



```
def budget(df, col, min_value):
    df_aux = df[col].value_counts() >= min_value
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)
    df_aux.rename(columns={'index': 'name'}, inplace=True)
    df_aux = df_aux.name[df_aux[col] == True]

    df_aux = df[df[col].isin(df_aux)]

    df_aux = df_aux.groupby(col)['budget'].mean()
    df_aux.sort_values(ascending = False, inplace = True)
    df_aux = df_aux.to_frame()
    df_aux.reset_index(level=0, inplace=True)

    title = ''
    if col == 'director':
        title = 'Directors with high budget'
    elif col == 'cast':
        title = 'Actors with high budget'
    elif col == 'genres':
        title = 'Genres with more budget'
    else:
        title = 'Companies with high budget'

    plt.subplots(figsize = (16,12))
    sns.barplot(x=col, y="budget", data = df_aux.head(),  palette="Purples_d")\
                .set_title(title, fontsize = 20)
```

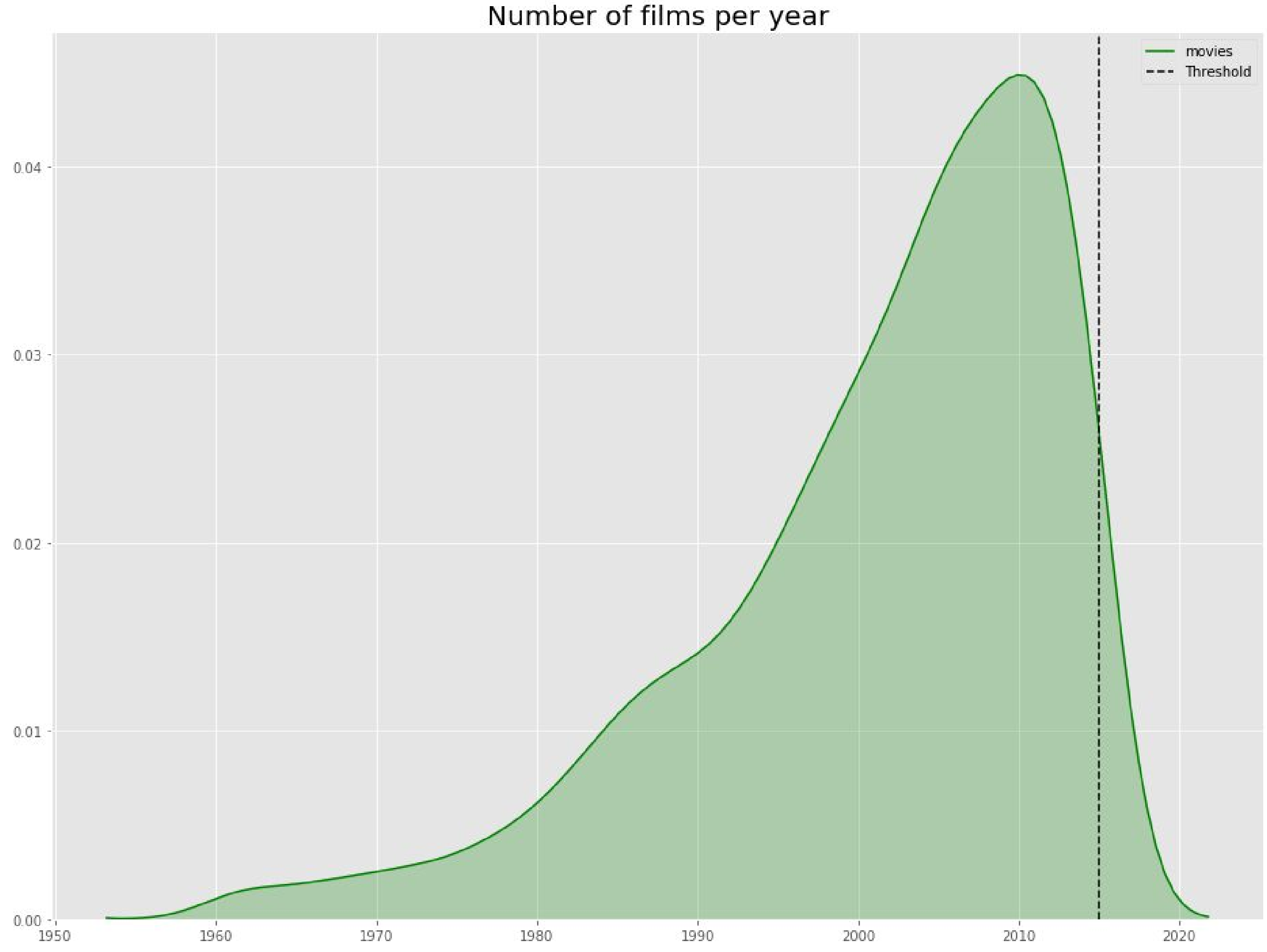
Function to create new columns, from those with "|", and function to plot scatter:



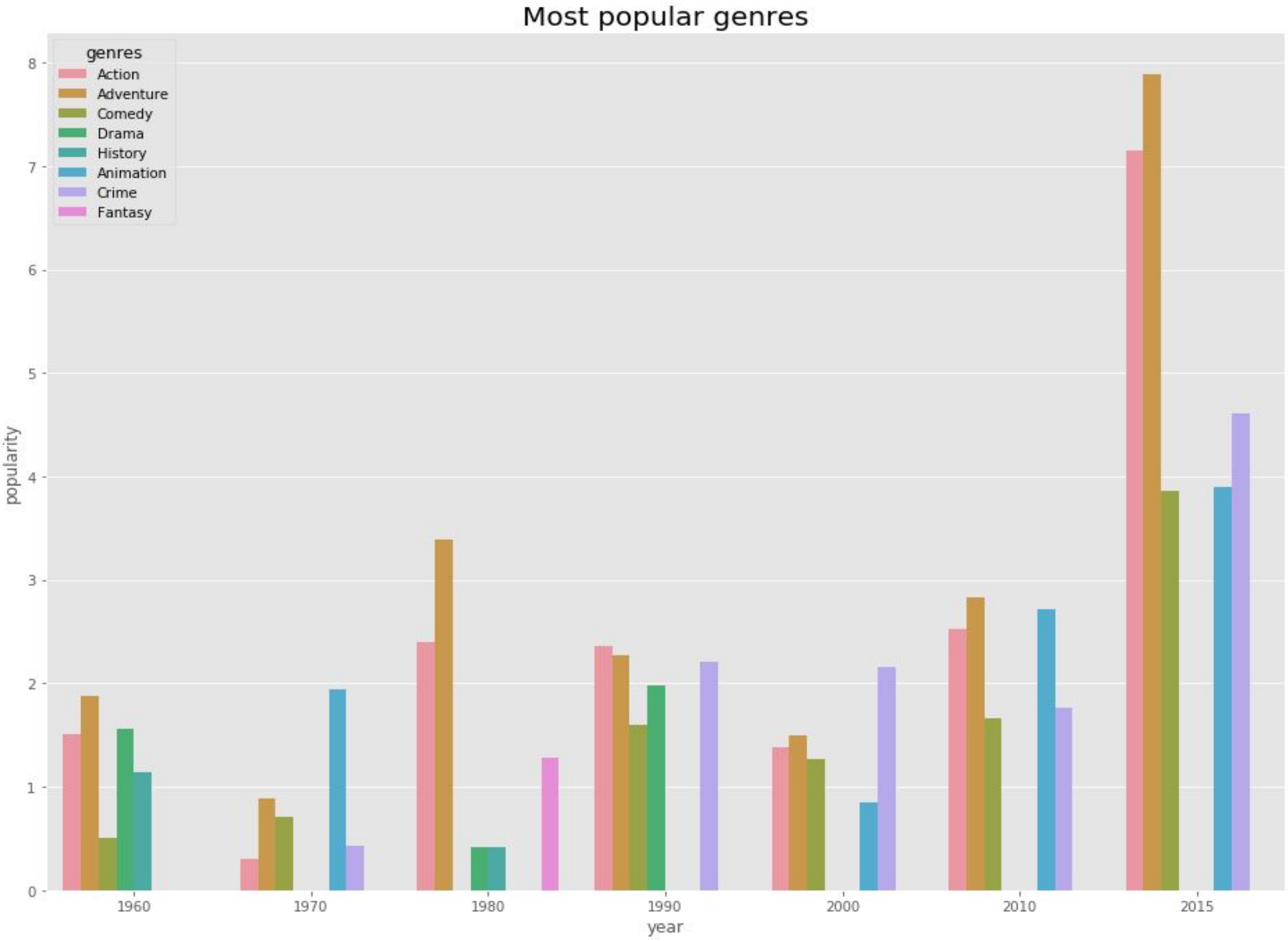
```
def clean_columns_with_pipeline(df, column):

    df_aux = pd.DataFrame(df[column].str.split('|'))
    aux_series = df_aux[column].apply(pd.Series).stack().reset_index(1)
    aux_series.drop('level_1', axis = 1, inplace = True)
    final_df = df.join(aux_series)
    final_df.drop(column, axis = 1, inplace = True)
    final_df.rename(columns={0:column}, inplace=True)
    return final_df

def plot_scatter(df, col1, col2, year):
    plt.subplots(figsize = (14,12))
    sns.scatterplot(x = col1,\
                    y = col2,\
                    hue = year,\
                    size = year,\
                    alpha = 0.5,\
                    sizes = (20,300),\
                    data = df)
```

Results

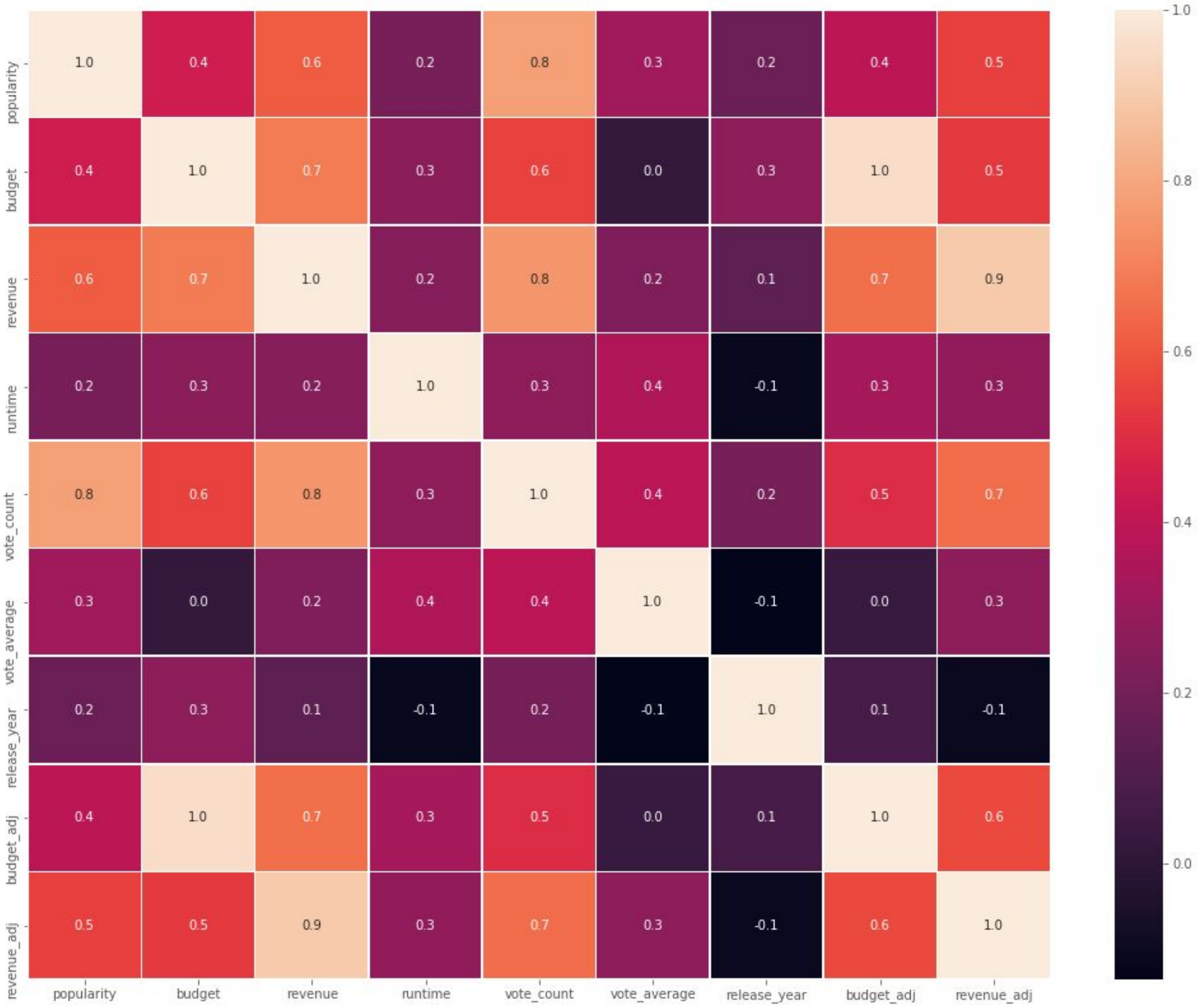
Observations:
We can see that the creation of movies increases exponentially, until 2015, due to lack of data, it decreases a little.



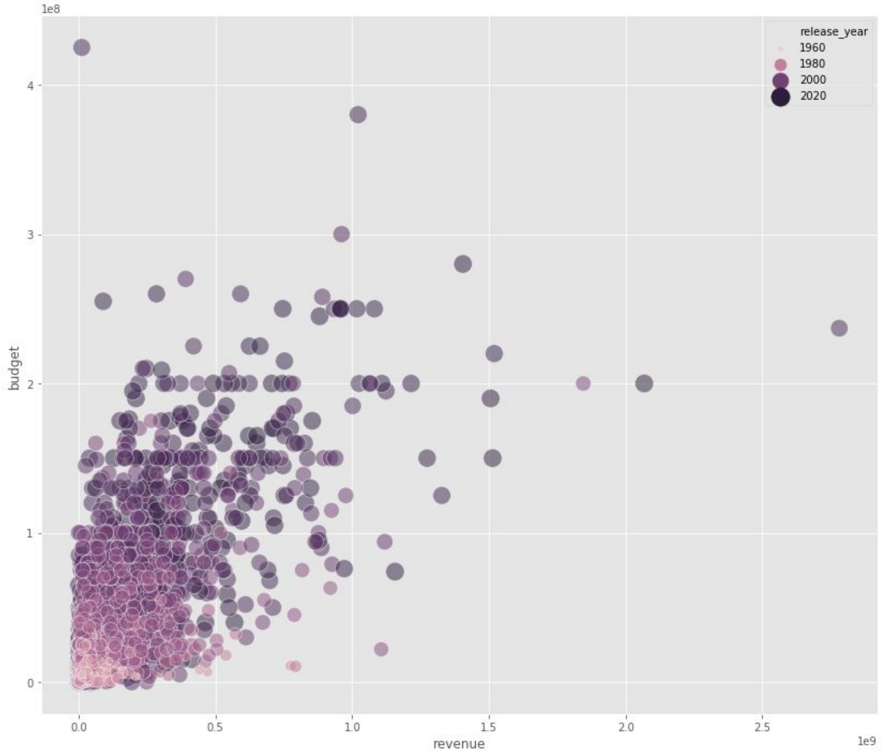
Observations:
Here we observe a change of genres over the years, but always action and adventure remain strong, we also see how genres such as animation and crime rise after the 2000s.

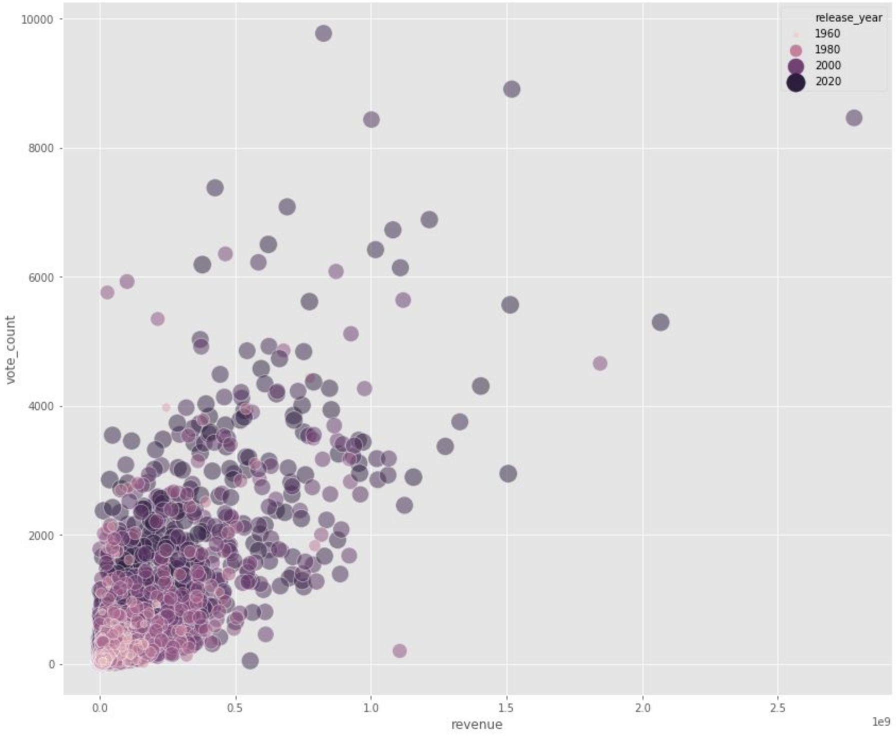


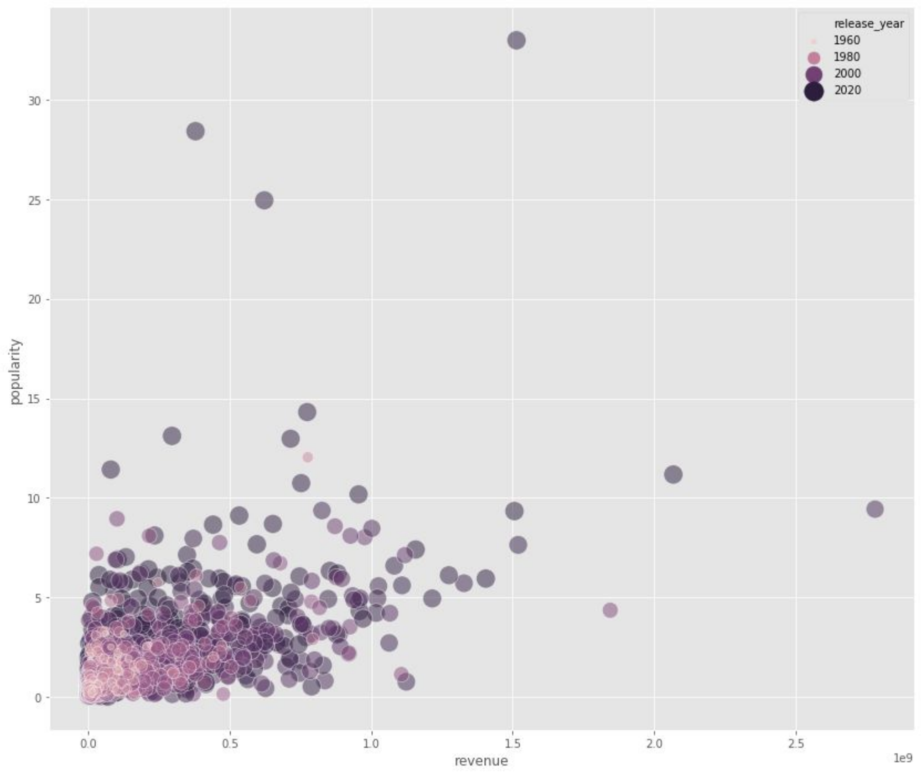
Observations:
Here we see the correlation between several columns, but the one that interests us for now is "revenue".



Now we are going to see graphs of the relationship between the columns that have the highest correlation with "revenue"





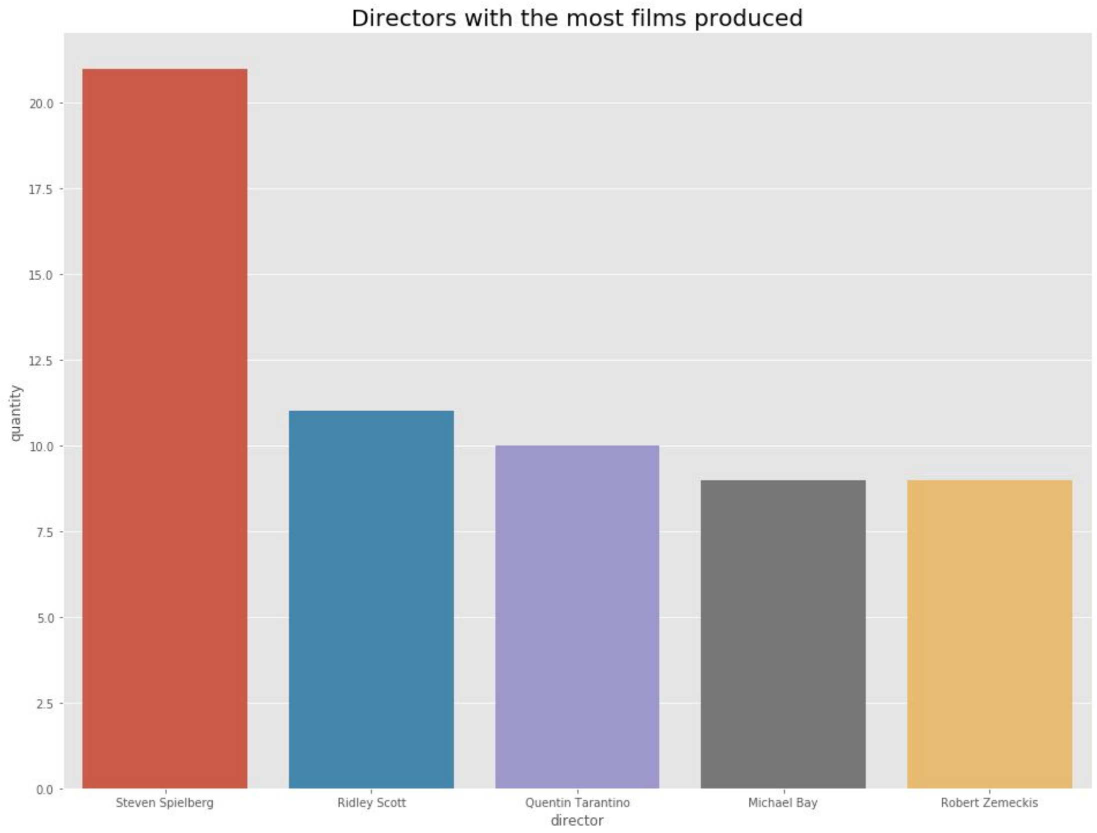


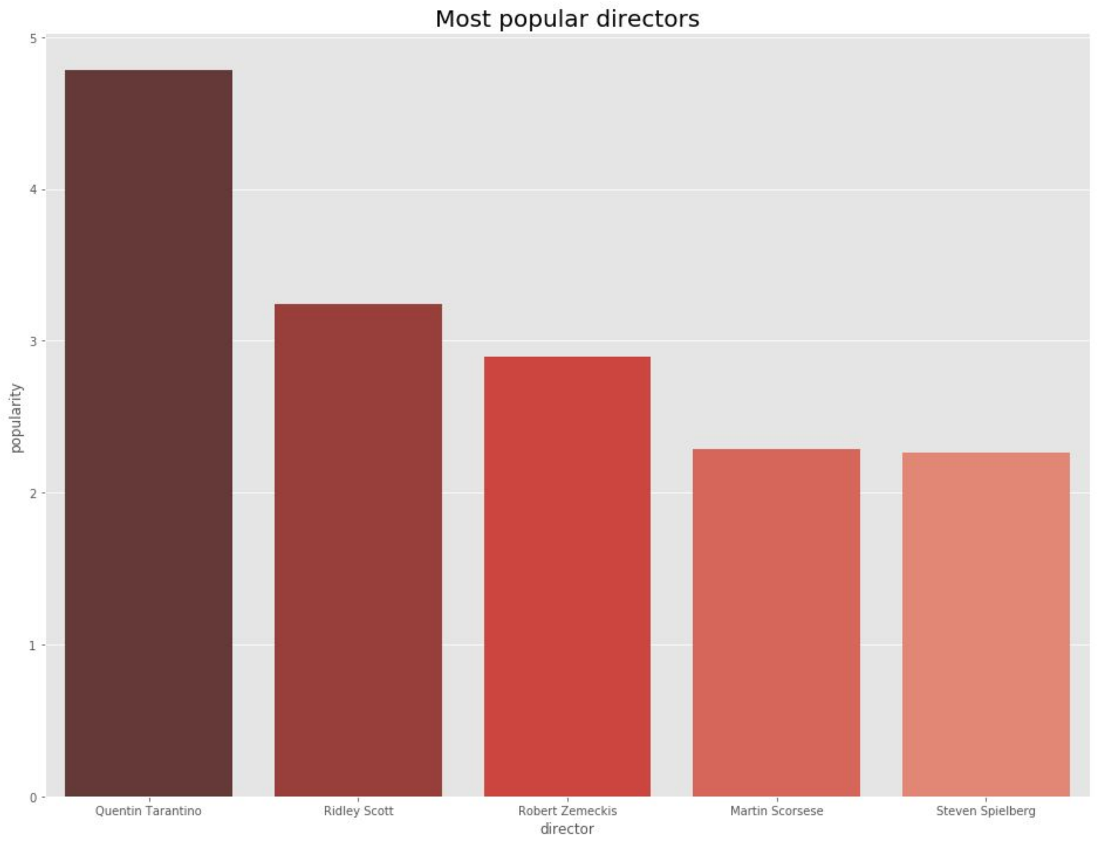
Now we are going to proceed using the columns [budget, revenue, popularity] since they seem to me the most interesting, this process we are going to do with all the columns that have a "|" in the original dataset.

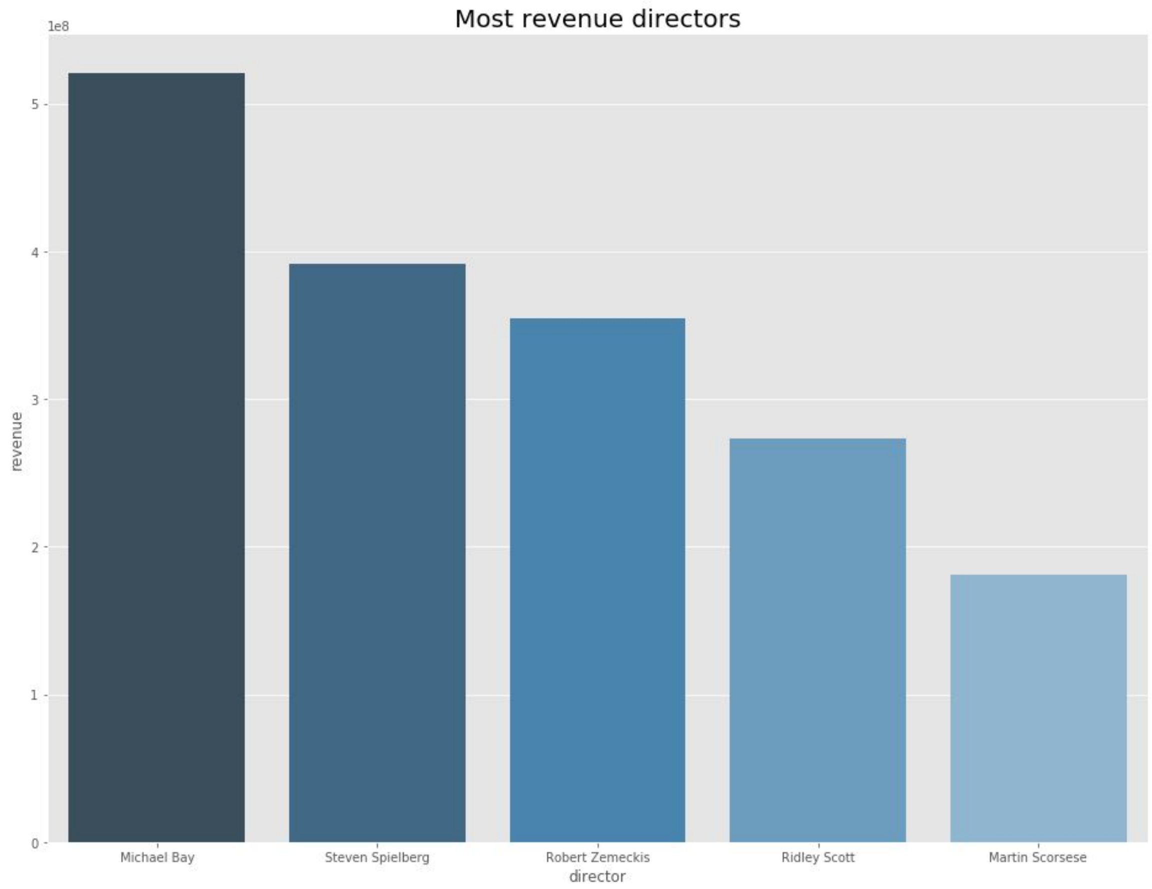
Another consideration is that you calculate this taking into account the number of films, the only exception column of this is that of genres.

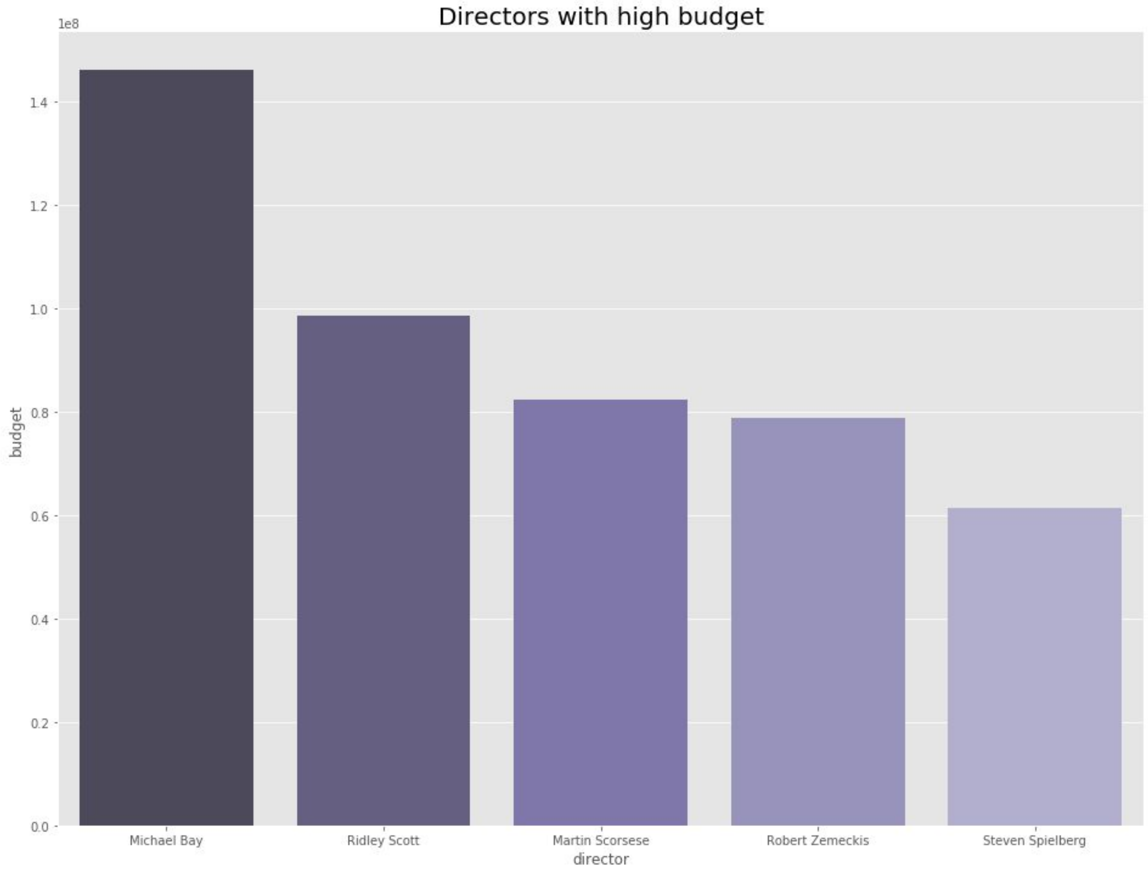
As a last dimension, I want to comment that in these graphs I want to show how most of the time the variables that have the highest "budget" and "popularity" have a good "revenue"

Directors:

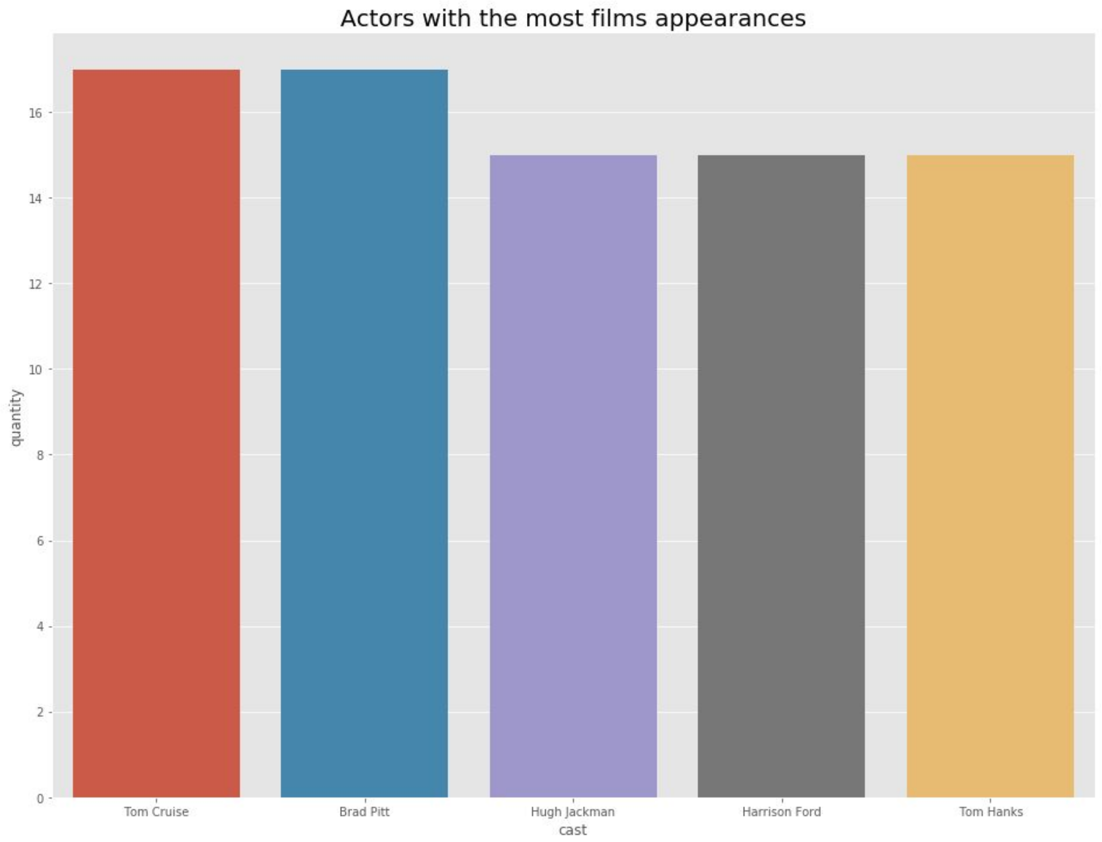

Directors with the most films produced
quantity
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Steven Spielberg
Ridley Scott
Quentin Tarantino
Michael Bay
Robert Zemeckis
director

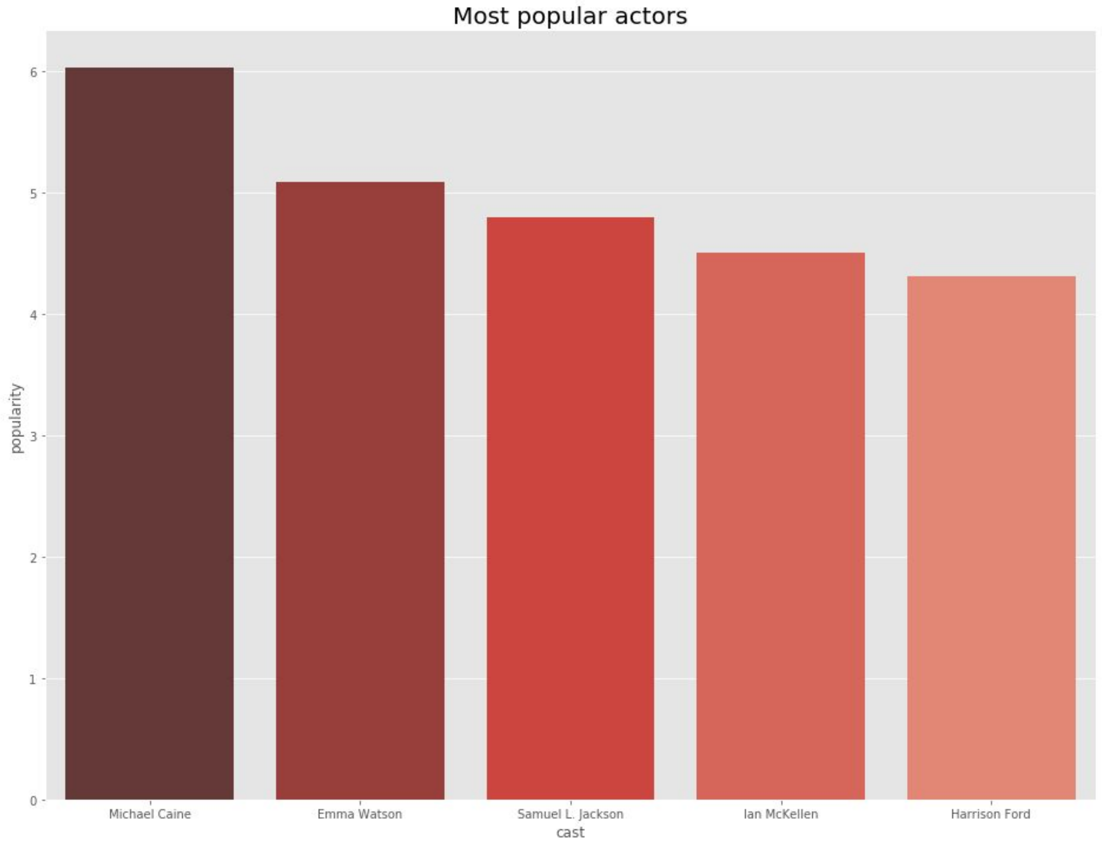

Most popular directors
popularity
0
1
2
3
4
5
Quentin Tarantino
Ridley Scott
Robert Zemeckis
Martin Scorsese
Steven Spielberg
director

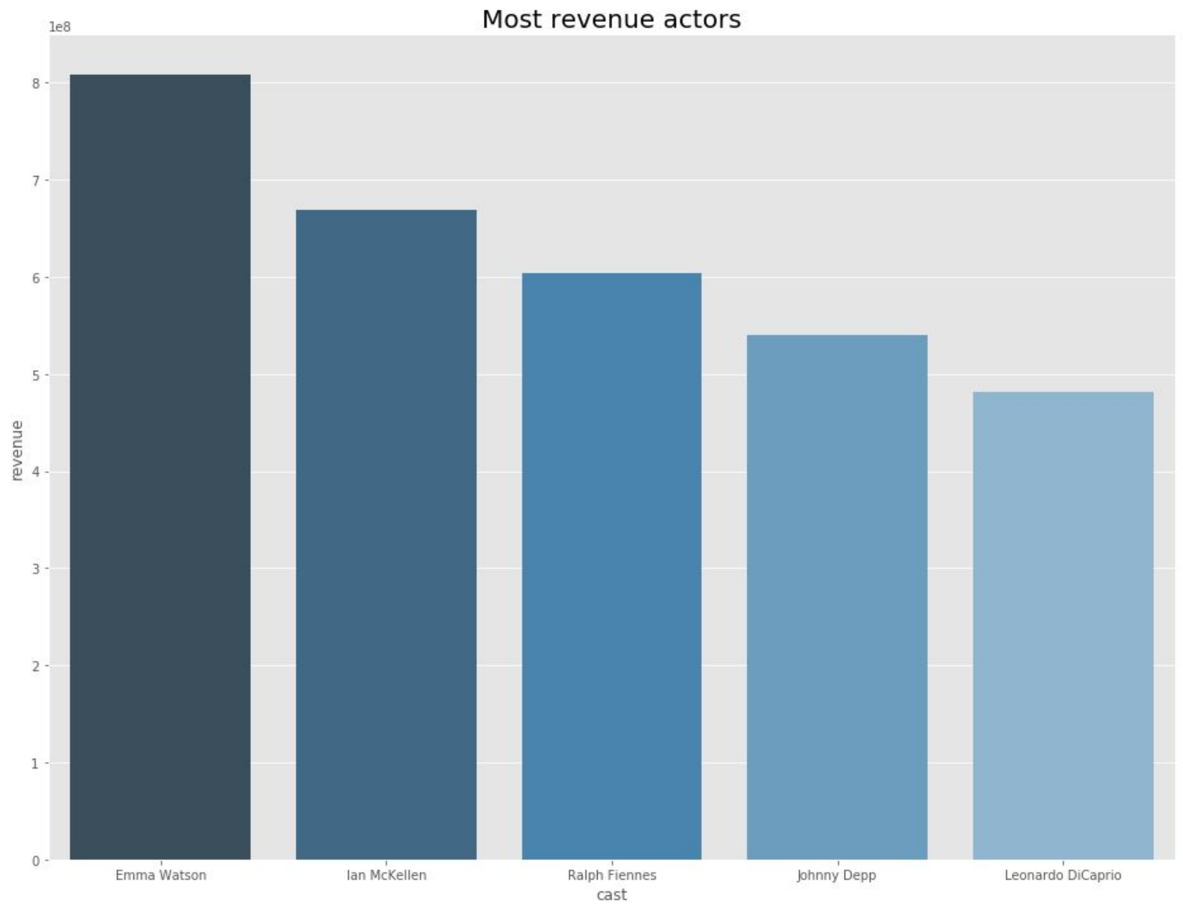

Most revenue directors
1e8
revenue
0
1
2
3
4
5
Michael Bay
Steven Spielberg
Robert Zemeckis
Ridley Scott
Martin Scorsese
director

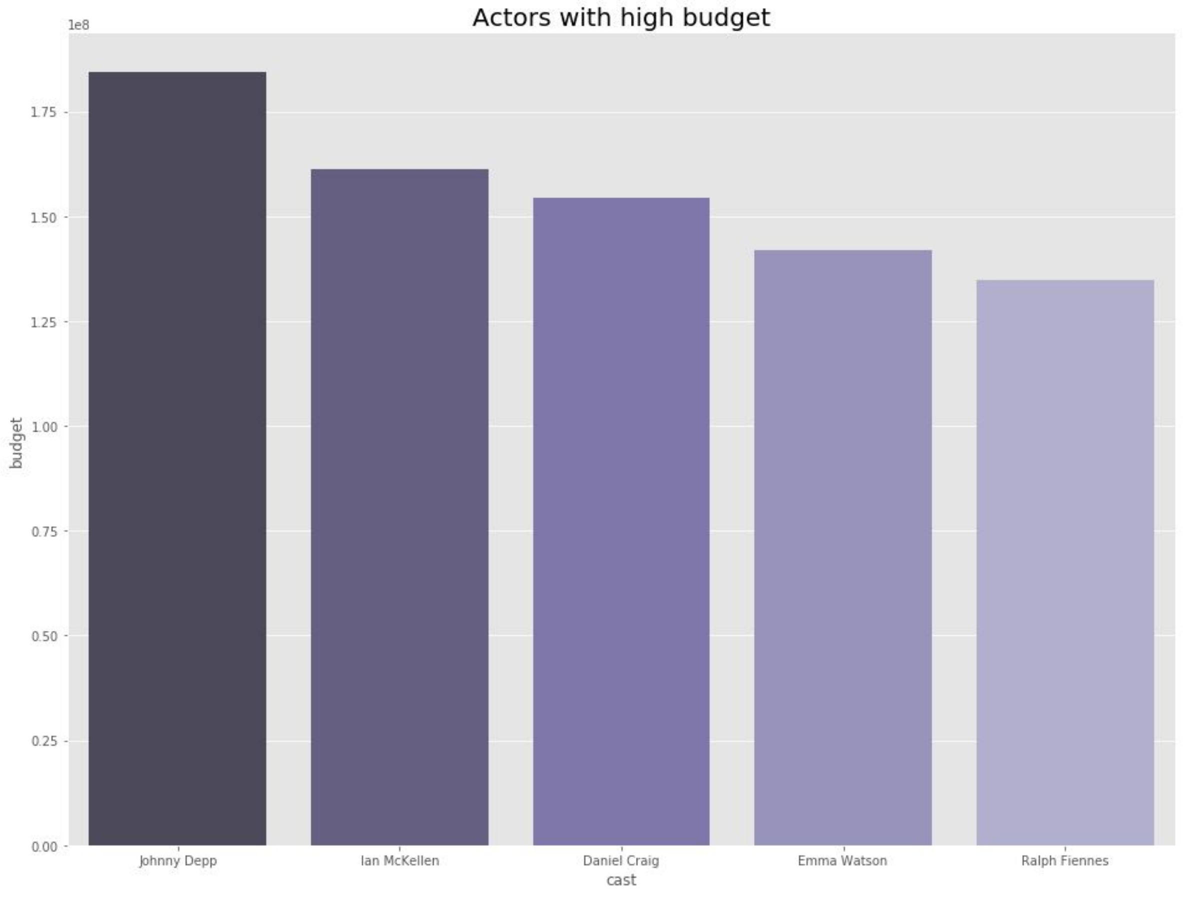

Directors with high budget
1e8
budget
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
Michael Bay
Ridley Scott
Martin Scorsese
Robert Zemeckis
Steven Spielberg
director

Actors:

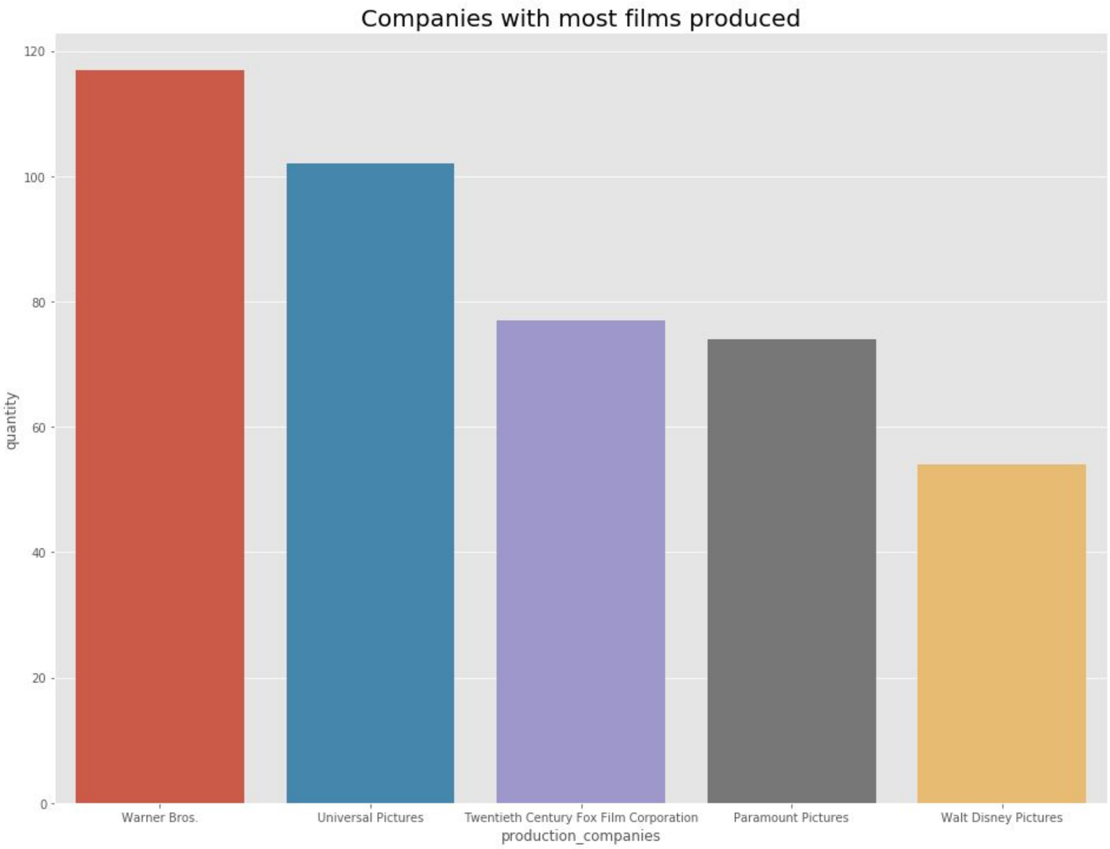

Actors with the most films appearances
quantity
0
2
4
6
8
10
12
14
16
Tom Cruise
Brad Pitt
Hugh Jackman
Harrison Ford
Tom Hanks
cast

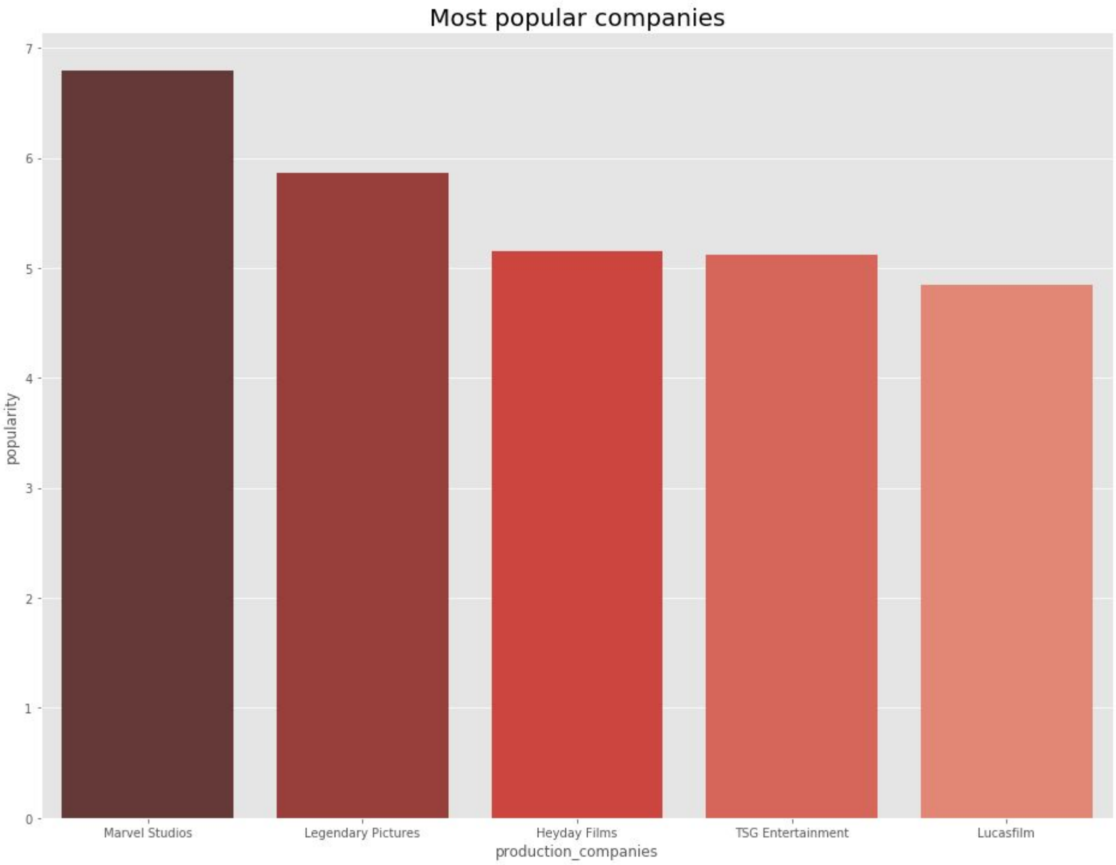

Most popular actors
popularity
0
1
2
3
4
5
6
Michael Caine
Emma Watson
Samuel L. Jackson
Ian McKellen
Harrison Ford
cast

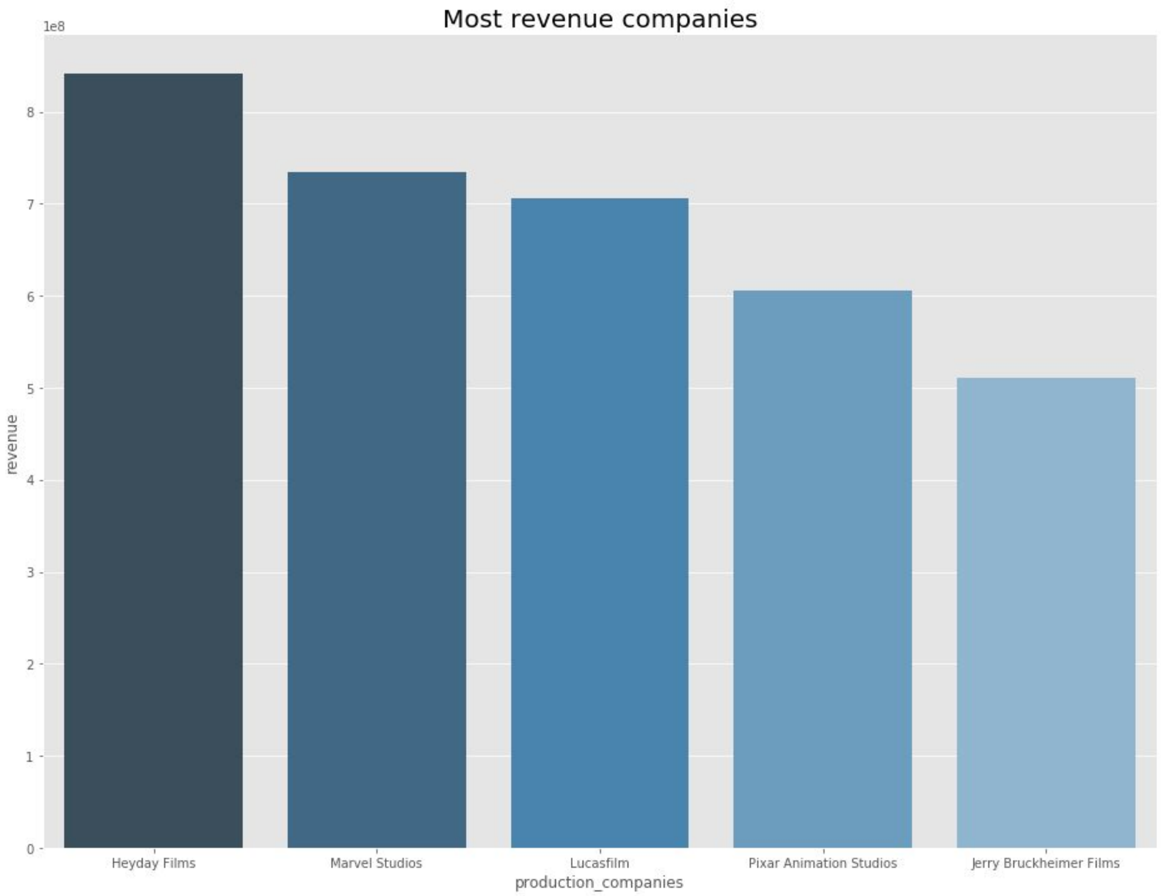

Most revenue actors

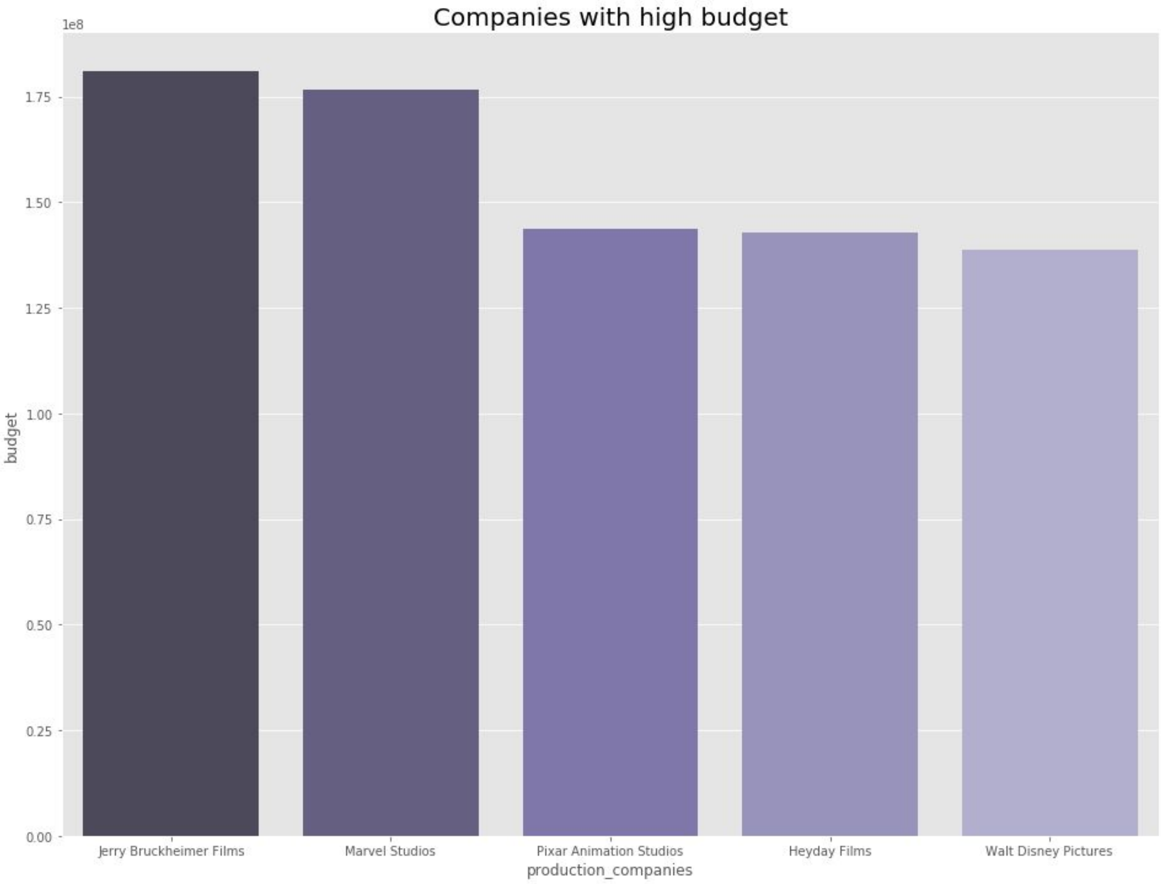

Actors with high budget

Production companies:

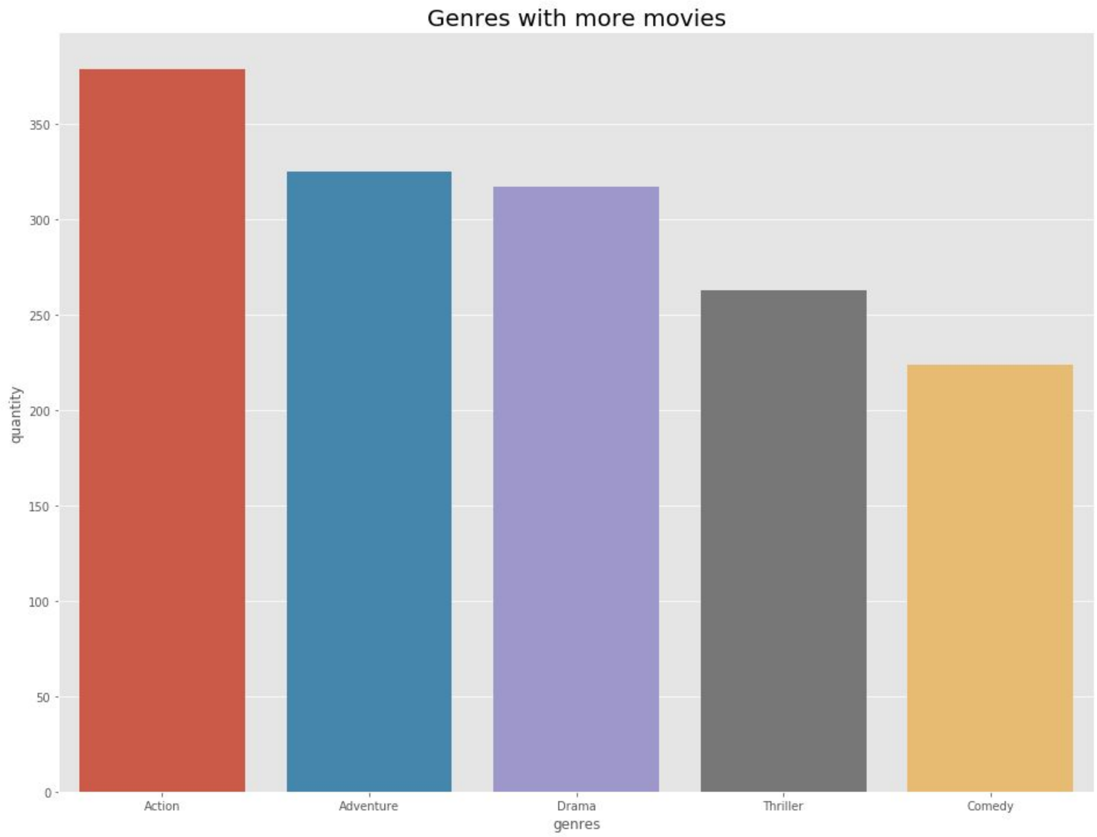


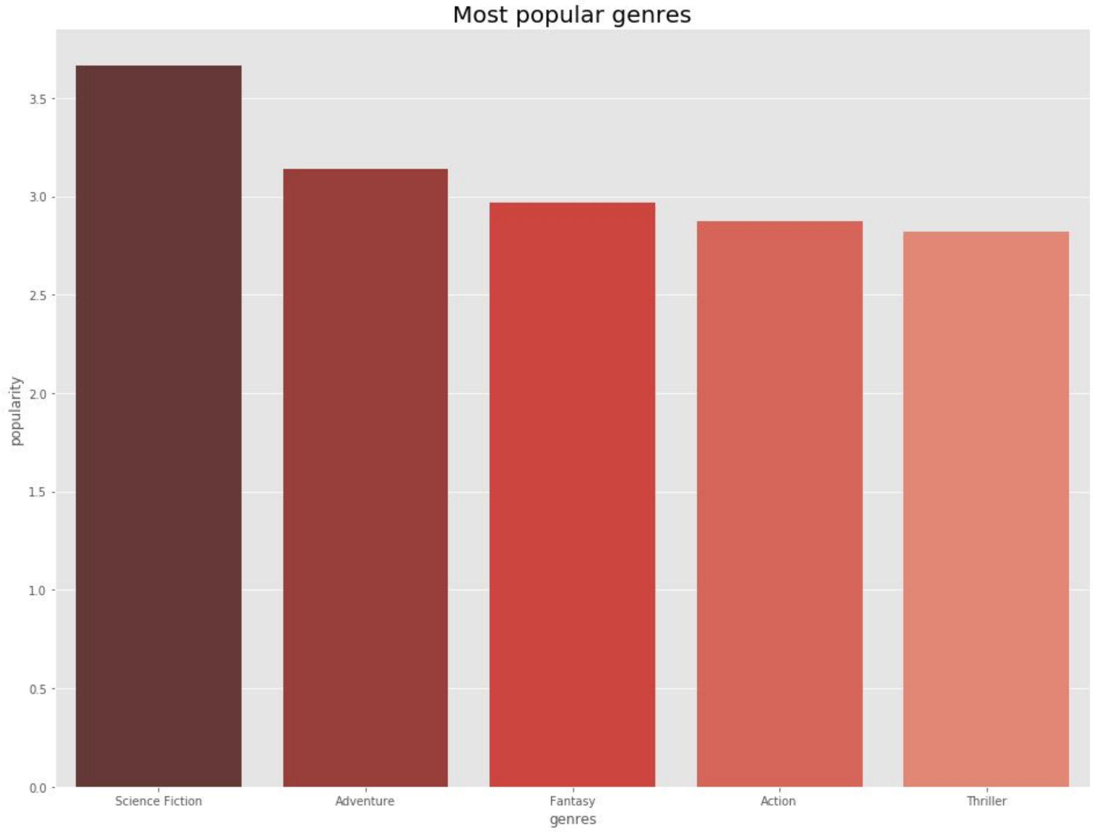


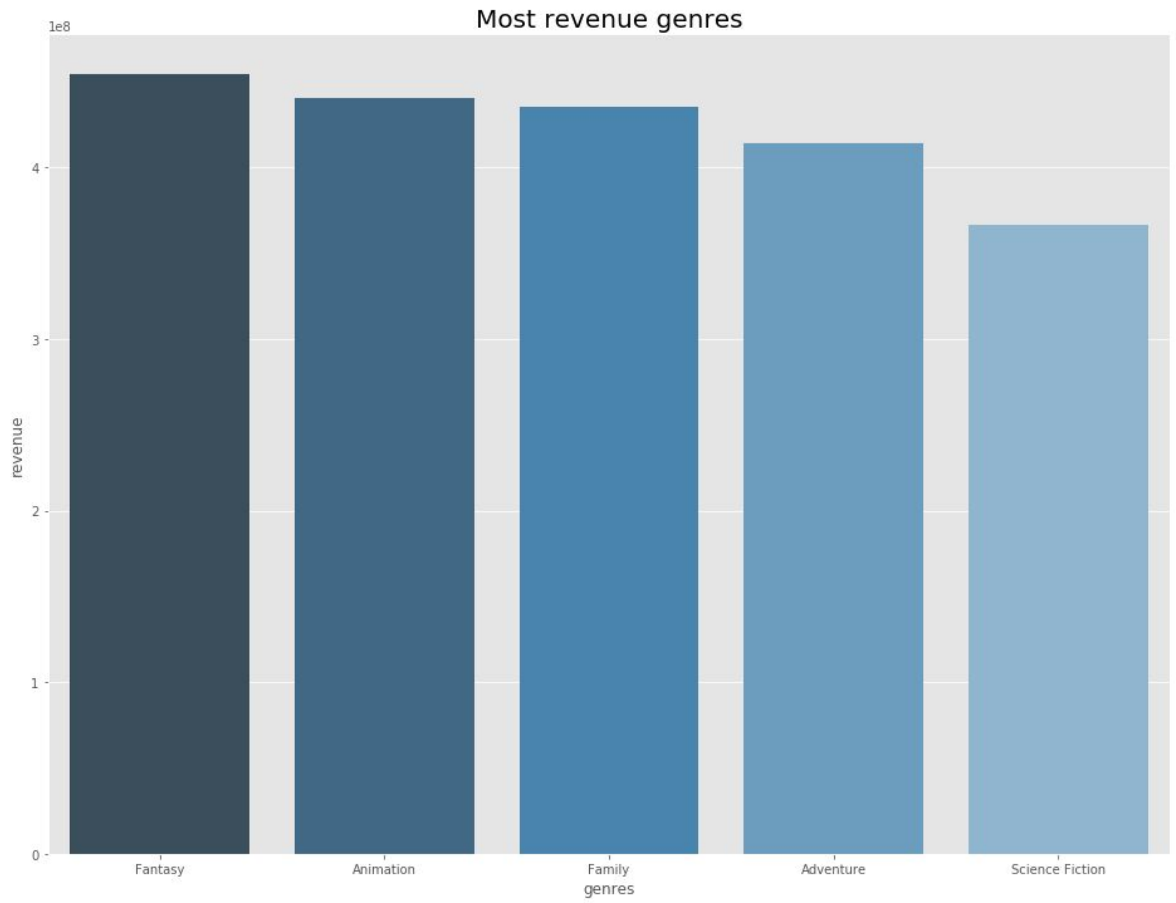

Most revenue companies
1e8
revenue
0
1
2
3
4
5
6
7
8
Heyday Films
Marvel Studios
Lucasfilm
Pixar Animation Studios
Jerry Bruckheimer Films
production_companies

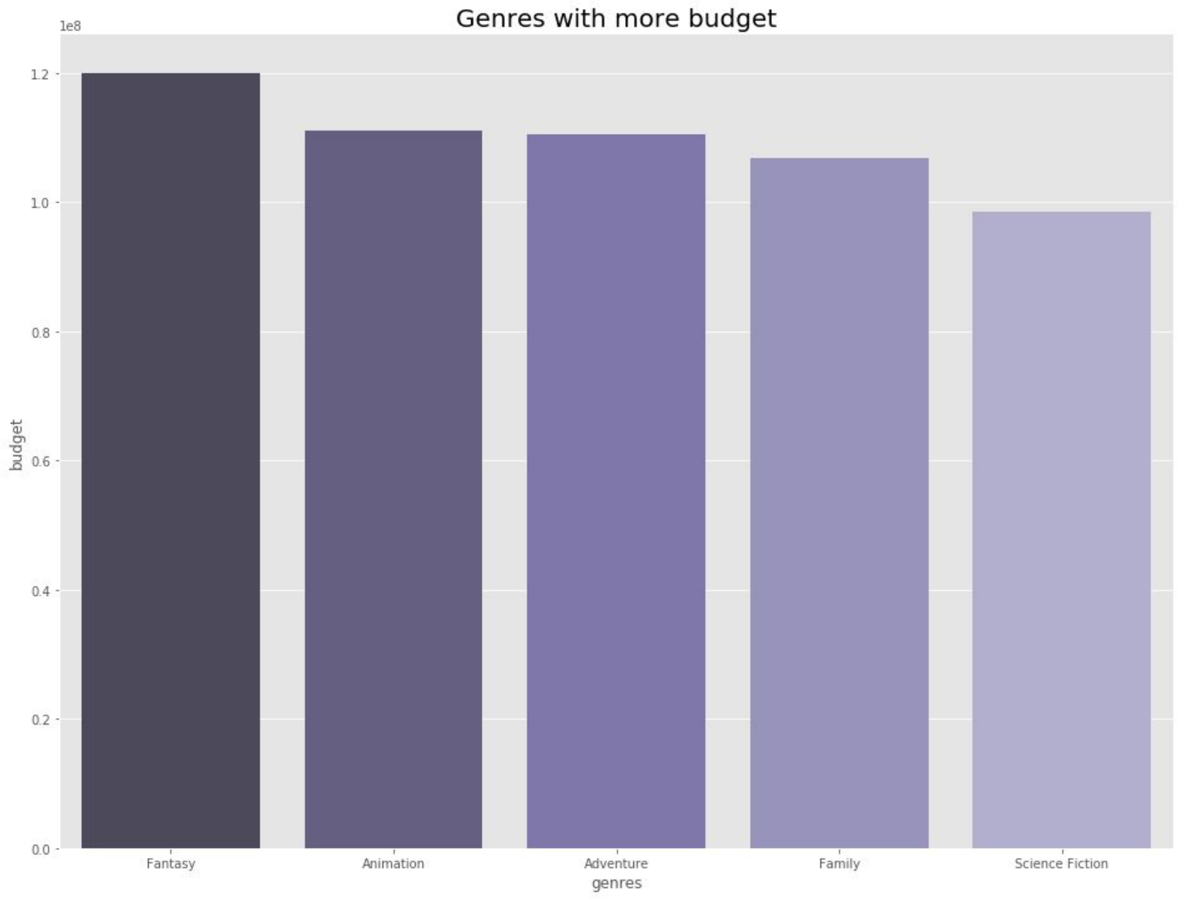

Companies with high budget
1e8
budget
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
Jerry Bruckheimer Films
Marvel Studios
Pixar Animation Studios
Heyday Films
Walt Disney Pictures
production_companies

Genres:


Genres with more movies
quantity
0
50
100
150
200
250
300
350
Action
Adventure
Drama
Thriller
Comedy
genres


Most popular genres
popularity
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Science Fiction
Adventure
Fantasy
Action
Thriller
genres


Most revenue genres
1e8
revenue
0
1
2
3
4
Fantasy
Animation
Family
Adventure
Science Fiction
genres


Genres with more budget
1e8
budget
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Fantasy
Animation
Adventure
Family
Science Fiction
genres

References:

- https://seaborn.pydata.org/generated/seaborn.kdeplot.html
- https://seaborn.pydata.org/generated/seaborn.heatmap.html
- https://seaborn.pydata.org/generated/seaborn.barplot.html