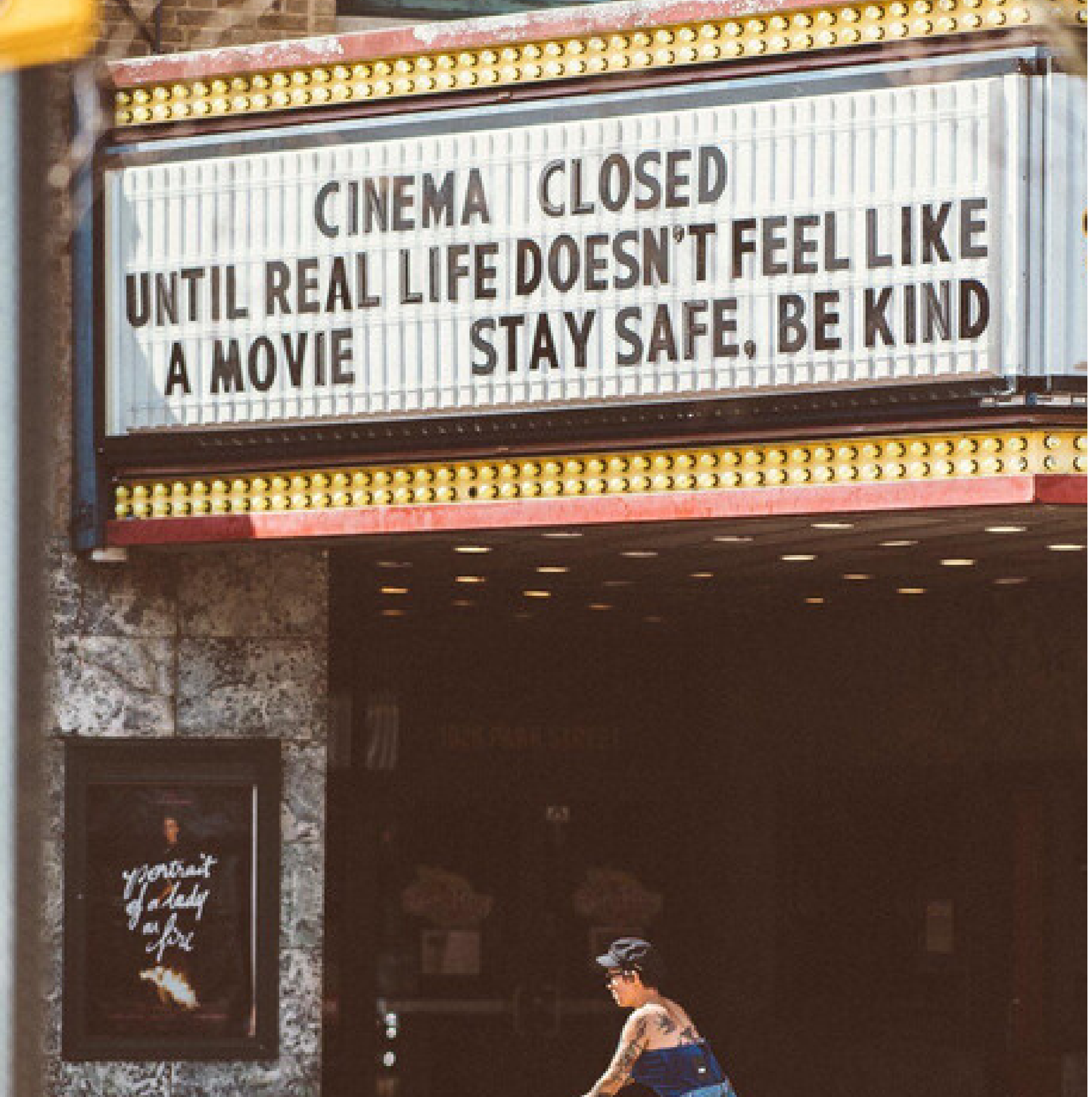


Analysing the Oscars

Are Oscar winners....

....the best rated?

....the most expensive to make?

... yielding best box office revenues?

By Rita Pereira

Motivation

Presentation Pipeline

DATA SOURCES

A brief history of a semi professional data thief

Source: Kaagle
Oscars nominees and winners from 1962-2015

Source: The Movie Database
information on the movies popularity, ratins, number of votes

Source: The numbers website
information on the movies budgets, domestic and world wide box office

Source: Kaagle
Dataset on 45k movies - used to get reliable budget data on oscar movies

The Academy dataset

1) Select 'Best Picture' from a 114 awards dataset:

```
oscars = oscars[(oscars['Award']=='Best Picture')]
oscars
```

	Year	Ceremony	Award	Winner	Name	Film
3924	1962	35	Best Picture	1.0	Lawrence of Arabia	Sam Spiegel, Producer
3925	1962	35	Best Picture	NaN	The Longest Day	Darryl F. Zanuck, Producer
3926	1962	35	Best Picture	NaN	Meredith Willson's The Music Man	Morton Da Costa, Producer
3927	1962	35	Best Picture	NaN	Mutiny on the Bounty	Aaron Rosenberg, Producer
3928	1962	35	Best Picture	NaN	To Kill a Mockingbird	Alan J. Pakula, Producer

List with movie titles

getting ready to request API

```
newlist2 =[]
for element in newlist:
    newlist2.append("+".join(element))
print(newlist2)
```

```
['Lawrence+of+Arabia', 'The+Longest+Day', "Meredith+Willson's+The+Music+Man", 'Mutiny+on+the+Bounty', 'To+Kill+a+Mockingbird', 'America+America', 'Cleopatra', 'How+the+West+Was+Won', 'Lilies+of+the+Field', 'Tom+Jones', 'Becket', 'Dr.+Strangelove+or:+How+I+Learned+to+Stop+Worrying+and+Love+the+Bomb', 'Mary+Poppins', 'My+Fair+Lady', 'Zorba+the+Greek', 'Darling', 'Doctor+Zhivago', 'Ship+of+Fools', 'The+Sound+of+Music', 'A+Thousand+Clowns', 'Alfie', 'A+Man+for+All+Seasons', 'The+Russians+Are+Coming+The+Russians+Are+Coming', 'The+Sand+Pebbles', "Who's+Afraid+of+Virginia+Woolf?", 'Bonnie+and+Clyde', 'Doctor+Dolittle', 'The+Graduate', "Guess+Who's+Coming+to+Dinner", 'In+the+Heat+of+the+Night', 'Funny+Girl', 'The+Lion+in+Winte
```

The Movie DB - API

1) Used the list of lists to pull information from the API:

```
def super_function(newlist2):
    df = pd.DataFrame([])
    for title in newlist2:
        try:
            api_key = "5d6ee3c337c9ed1dd8636a0112113fd2"
            response = requests.get('https://api.themoviedb.org/3/search/movie?api_key=' +api_key+ "&query=" + title)
            movie = pd.DataFrame(response.json()['results'])
            df = df.append(movie, sort=False )
        except ValueError:
            pass
    return df


TMDB = super_function(newlist2)
TMDB
```

	popularity	vote_count	video	poster_path	id	adult	backdrop_path	original_language	original_title	genre_
0	15.211	1652	False	/j21UXCOq0EbV8EPTtmPn1CGePfB.jpg	947	False	/lENerYvUcfl9GloIqA1emj5gb4k.jpg	en	Lawrence of Arabia	[12, 107

2392 rows x 14 columns

2) Data cleaning:

- filtered by title
- eliminated rows with zero number of reviews, null popularity and release date
- I had to do a better matching --> get year from 'Resease date' and create new column with year and title (same in inital dataset)

```
TMDB_data = TMDB[TMDB['title'].isin(list_titles2)]
```

564 rows x 14 columns

433 rows x 14 columns

3) Merge datasets

270 rows x 17 columns

Web Scrape

```
url = 'https://www.the-numbers.com/movie/budgets/all'

response = requests.get(url)
html = response.content
soup = BeautifulSoup(html, "lxml")
table = soup.find_all('td')

rows = [row.text.strip().split("\n") for row in table]
rows
```

```
[['1'],
 ['Apr 23, 2019'],
 ['Avengers: Endgame'],
 ['$400,000,000'],
 ['$858,373,000'],
 ['$2,797,800,564'],
```

```
new_list = []
for i in range(0, len(list_rows), 6):
    new_list.append(list_rows[i : i+6])

print(new_list)
```

```
[['1', 'Apr 23, 2019', 'Avengers: Endgame', '$400,000,000', '$858,373,000', '$2,797,800,56
4'], ['2', 'May 20, 2011', 'Pirates of the Caribbean: On Stranger Tides', '$379,000,000', '$2
41,063,875', '$1,045,663,875'], ['3', 'Apr 22, 2015', 'Avengers: Age of Ultron', '$365,000,00
0', '$459,005,868', '$1,403,013,963'], ['4', 'Dec 16, 2015', 'Star Wars Ep. VII: The Force Aw
```

THE NUMBERS®

Where Data and the Movie Business Meet

News | Box Office | Home Video | Movies | People | Research Tools | Our Services | My Numbers

Tweet | Share 53 | Like 53 | Share

Movie Budgets

Note: Budget numbers for movies can be both difficult to find and unreliable. Studios and film-makers often try to keep the information secret and will use accounting tricks to inflate or reduce announced budgets.

This chart shows the budget of every film in our database, where we have it. The data we have is, to the best of our knowledge, accurate but there are gaps and disputed figures. If you have additional information or corrections, please let us know at corrections@the-numbers.com.

Our movie profit and loss records, based on this budget information, can be found here.

	Release Date	Movie	Production Budget	Domestic Gross	Worldwide Gross
1	Apr 23, 2019	**Avengers: Endgame**	$400,000,000	$858,373,000	$2,797,800,564
2	May 20, 2011	**Pirates of the Caribbean: On Stranger Tides**	$379,000,000	$241,063,875	$1,045,663,875
3	Apr 22, 2015	**Avengers: Age of Ultron**	$365,000,000	$459,005,868	$1,403,013,963
4	Dec 16, 2015	**Star Wars Ep. VII: The Force Awakens**	$306,000,000	$936,662,225	$2,068,223,624

	Release Date	Movie	Production Budget	Domestic Gross	Worldwide Gross
0	Apr 23, 2019	Avengers: Endgame	$400,000,000	$858,373,000	$2,797,800,564
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$379,000,000	$241,063,875	$1,045,663,875
2	Apr 22, 2015	Avengers: Age of Ultron	$365,000,000	$459,005,868	$1,403,013,963
3	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	$306,000,000	$936,662,225	$2,068,223,624

100 rows

x

8 columns

Web Scrape - do IT for the loop

```
def scrape_machine(url):
    df = pd.DataFrame([])
    try:
        response = requests.get(url)
        html = response.content
        soup = BeautifulSoup(html, "lxml")
        table = soup.find_all('td')
        rows = [row.text.strip().split("\n") for row in table]
        list_rows = sum(rows, [])
        new_list = []
        for i in range(0, len(list_rows), 6):
            new_list.append(list_rows[i : i+6])
        page = pd.DataFrame(new_list)
        df = df.append(page, sort=False )
        return df
    except ValueError:
        pass
```

```
url = 'https://www.the-numbers.com/movie/budgets/all/'

df_final = pd.DataFrame([])

for i in range(101,6001,100):
    df = scrape_machine(url+str(i))
    df_final = df_final.append(df)
```

	Release Date	Movie	Production Budget	Domestic Gross	Worldwide Gross
0	Apr 23, 2019	Avengers: Endgame	$400,000,000	$858,373,000	$2,797,800,564
1	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$379,000,000	$241,063,875	$1,045,663,875
2	Apr 22, 2015	Avengers: Age of Ultron	$365,000,000	$459,005,868	$1,403,013,963
3	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	$306,000,000	$936,662,225	$2,068,223,624

x 5,900

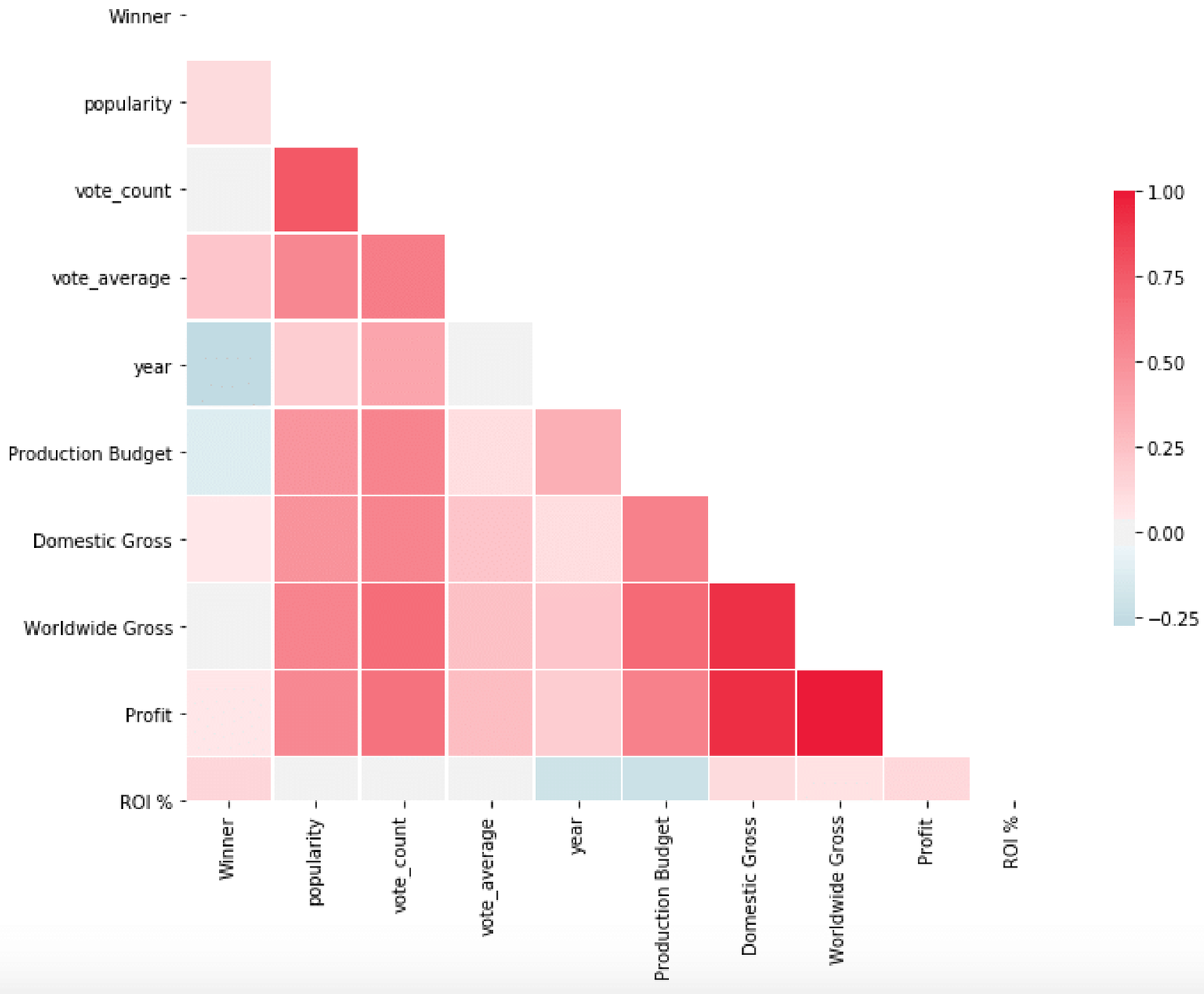
New variables

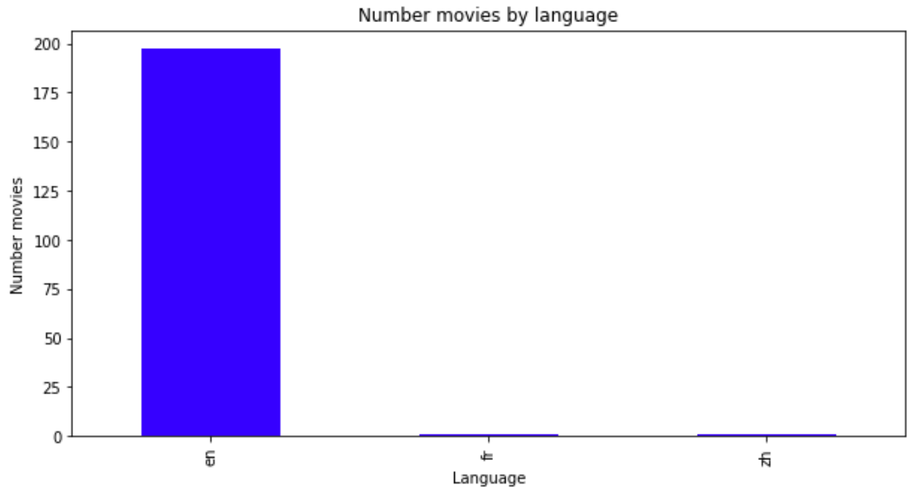
- Profit
- Return on investment (Profit/Budget)

id	Release Date	title	Production Budget	Domestic Gross	Worldwide Gross	Profit	ROI %
1	Apr 23, 2019	Avengers: Endgame	400000000	858373000	2797800564	2397800564	599.450141
2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	379000000	241063875	1045663875	666663875	175.900759
3	Apr 22, 2015	Avengers: Age of Ultron	365000000	459005868	1403013963	1038013963	284.387387
4	Dec 16, 2015	Star Wars Ep. VII: The Force Awakens	306000000	936662225	2068223624	1762223624	575.890073
5	Apr 25, 2018	Avengers: Infinity War	300000000	678815482	2048359754	1748359754	582.786585

Insights

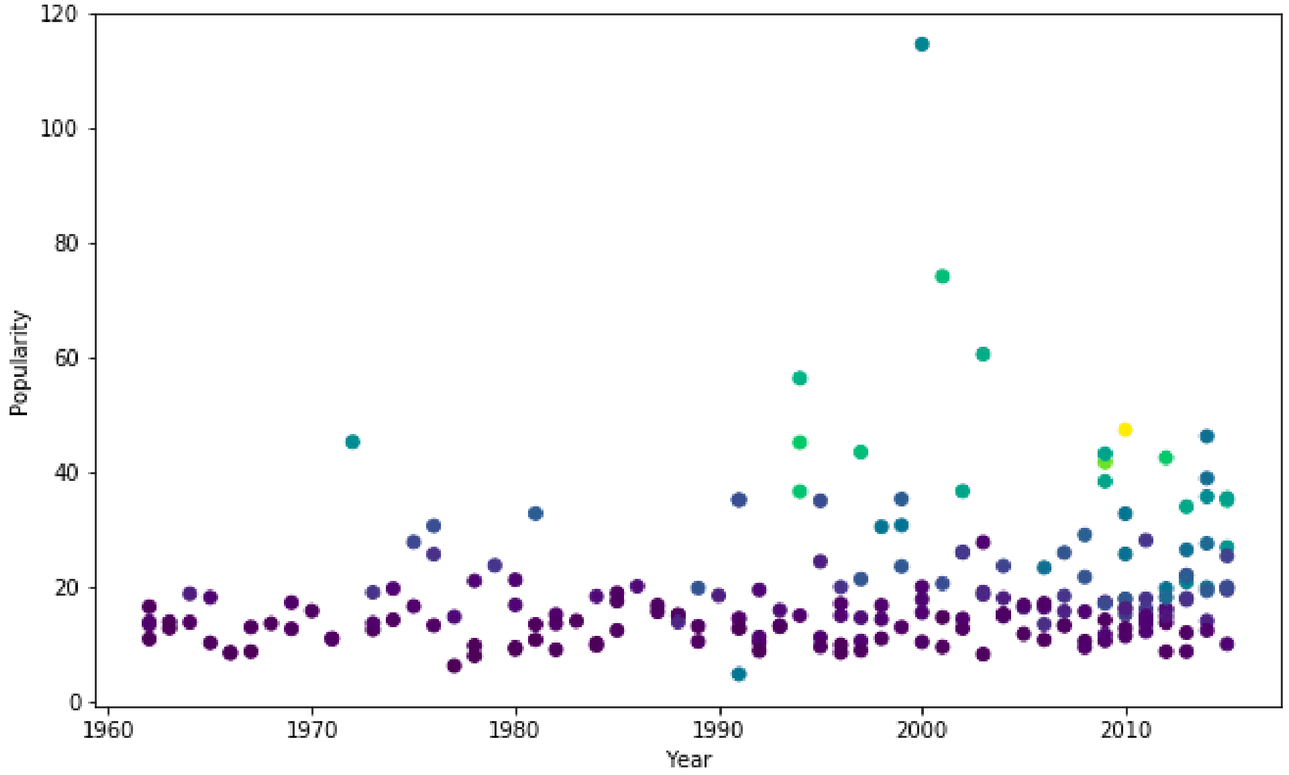
Correlation between variables:

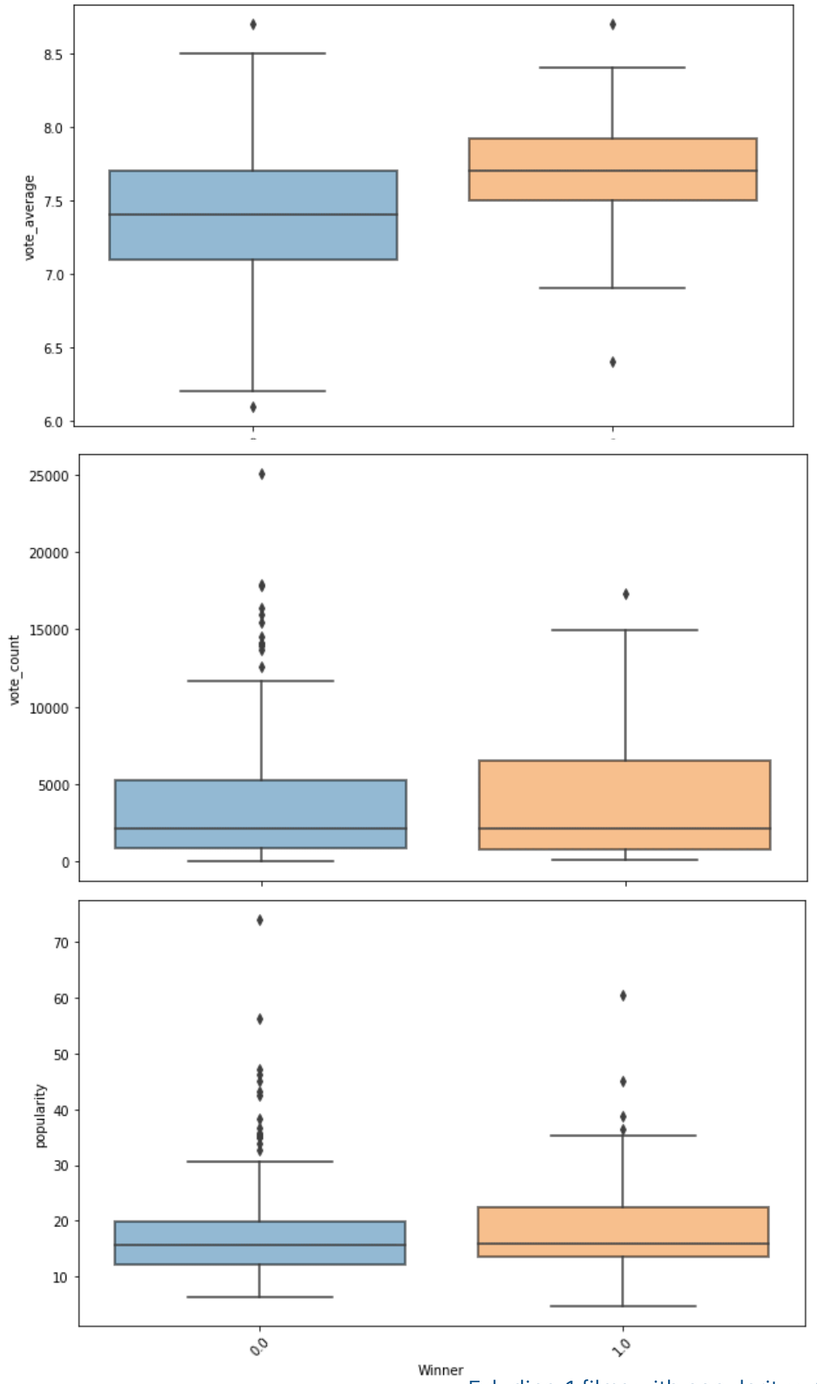




Oscar movies and "the people"

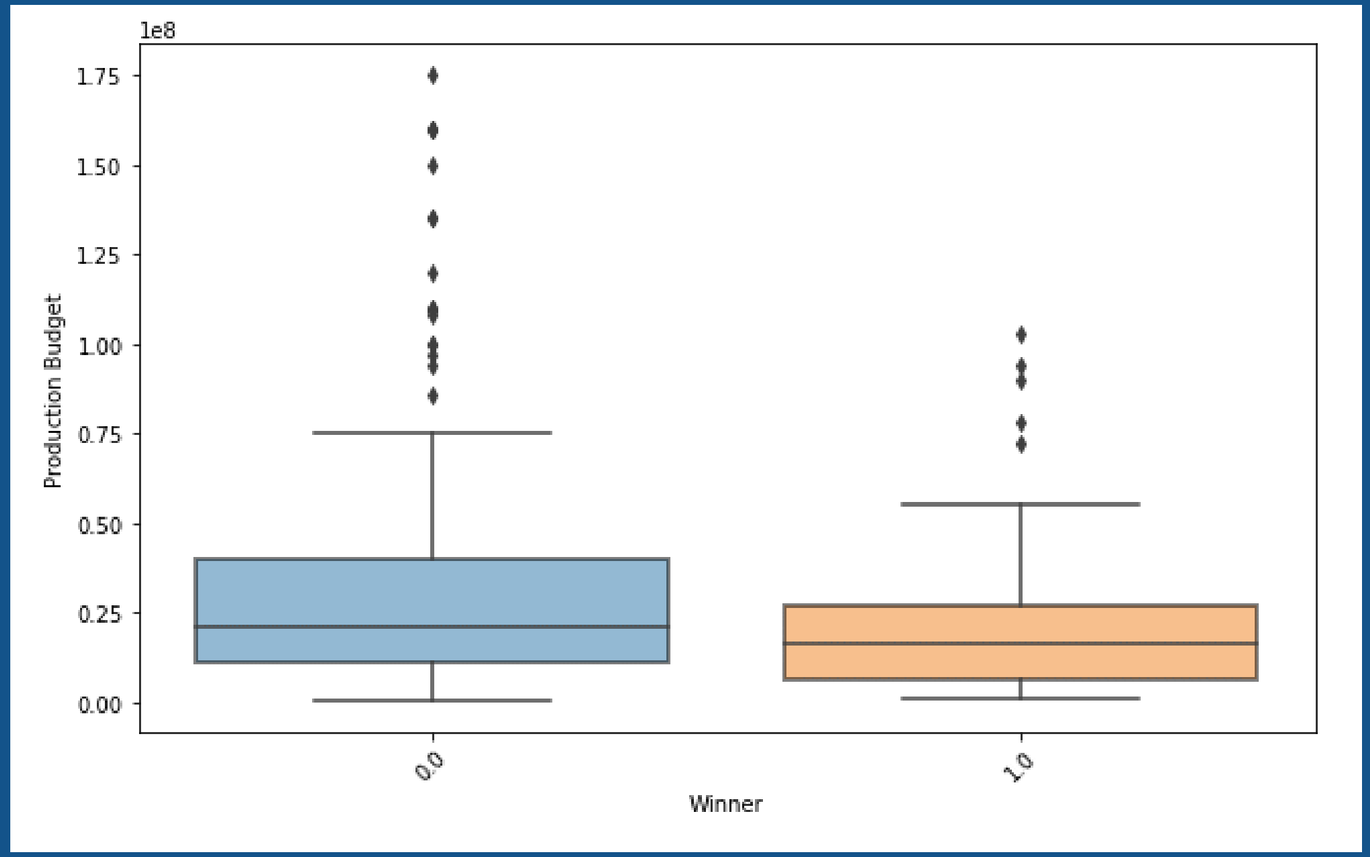
Popularity over the years - painting by number of votes



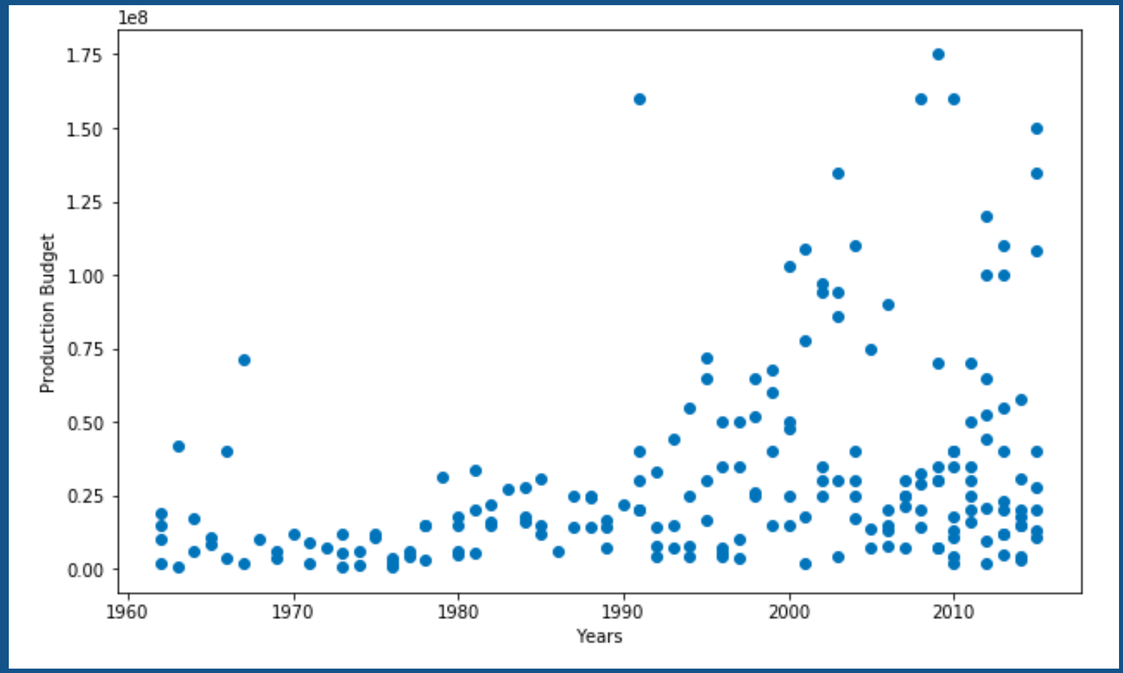


Exluding 1 films with popularity > 100

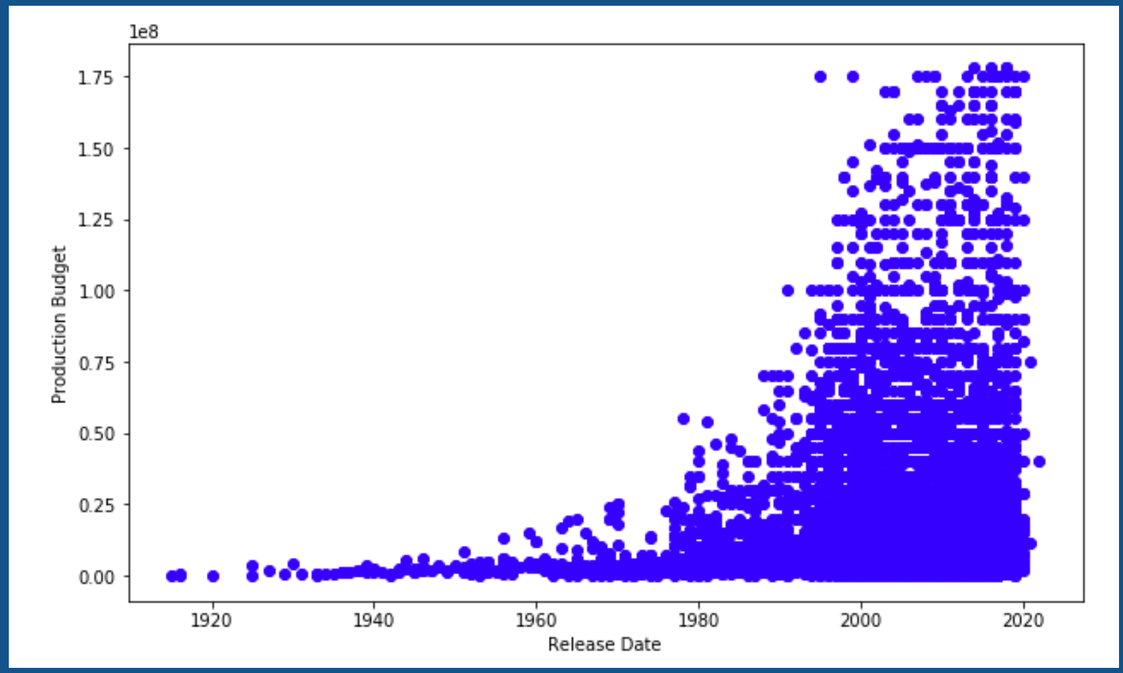
Production Budget



Oscar movies



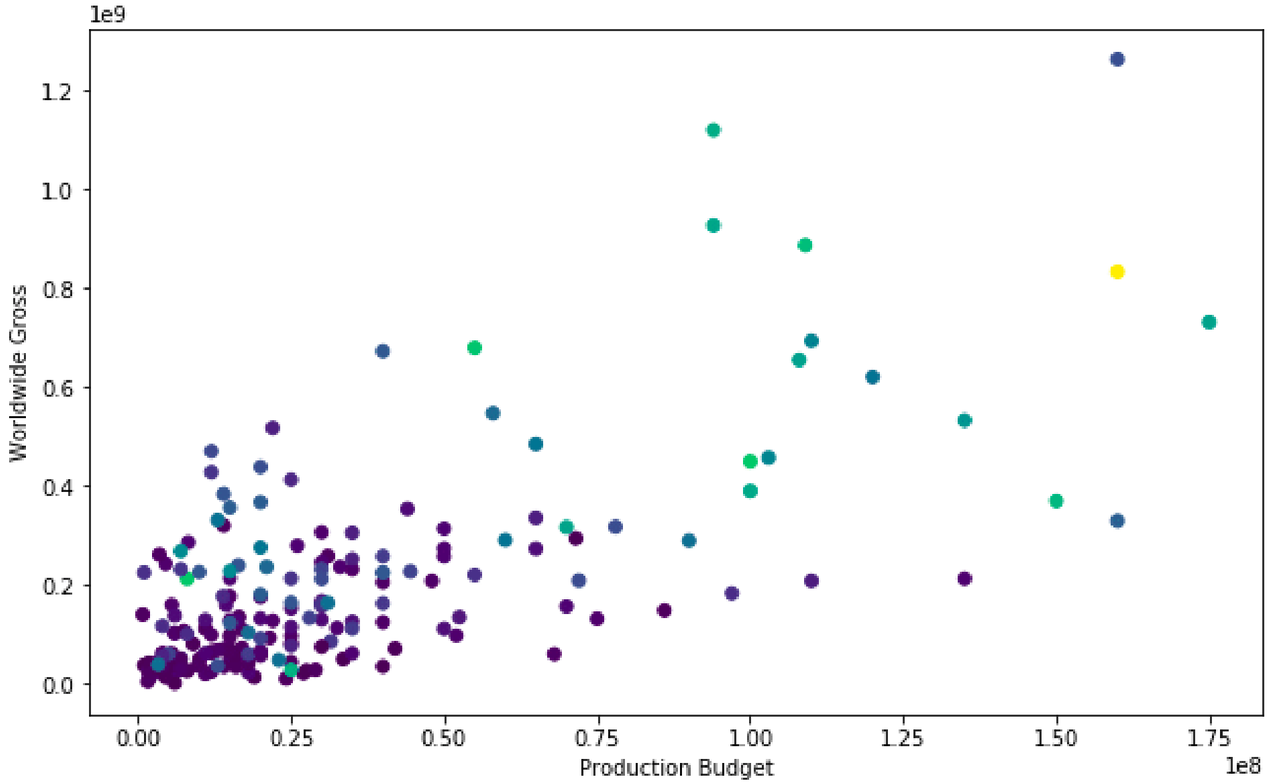
Movies that didn't go to the oscars



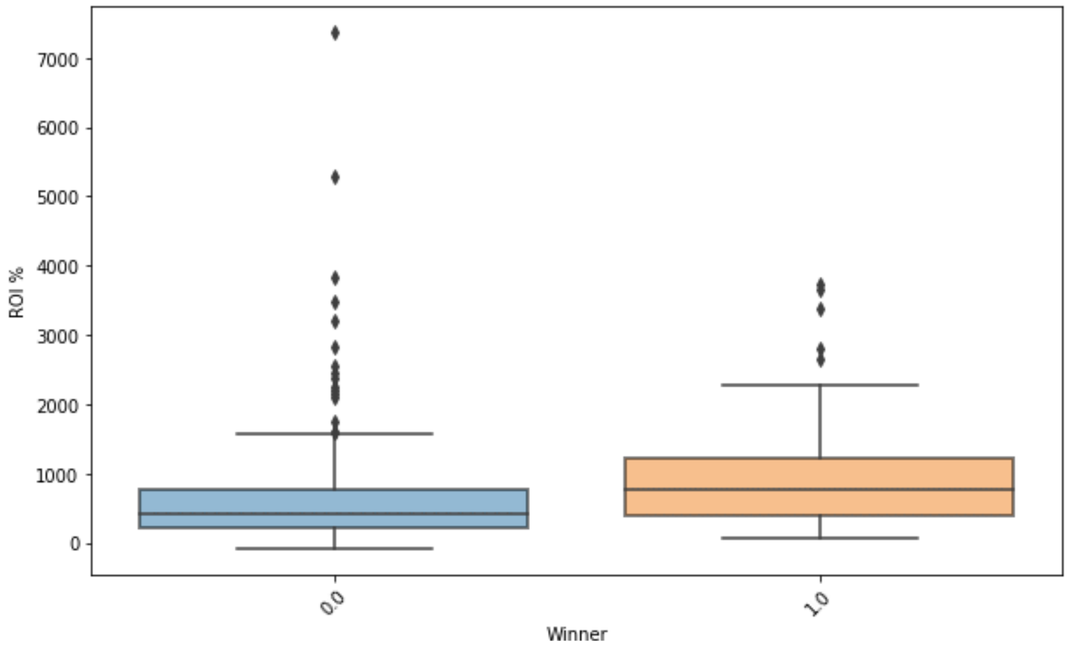
5702 scraped movies

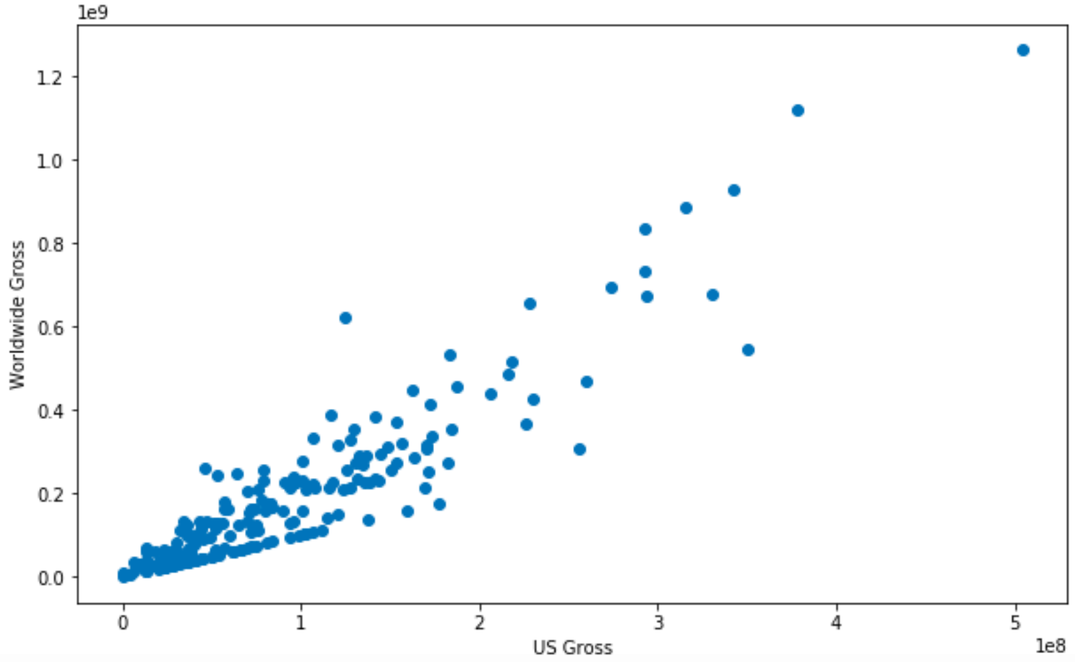
Oscar movies - Box Office

Box office vs Budget - painting by different vote count



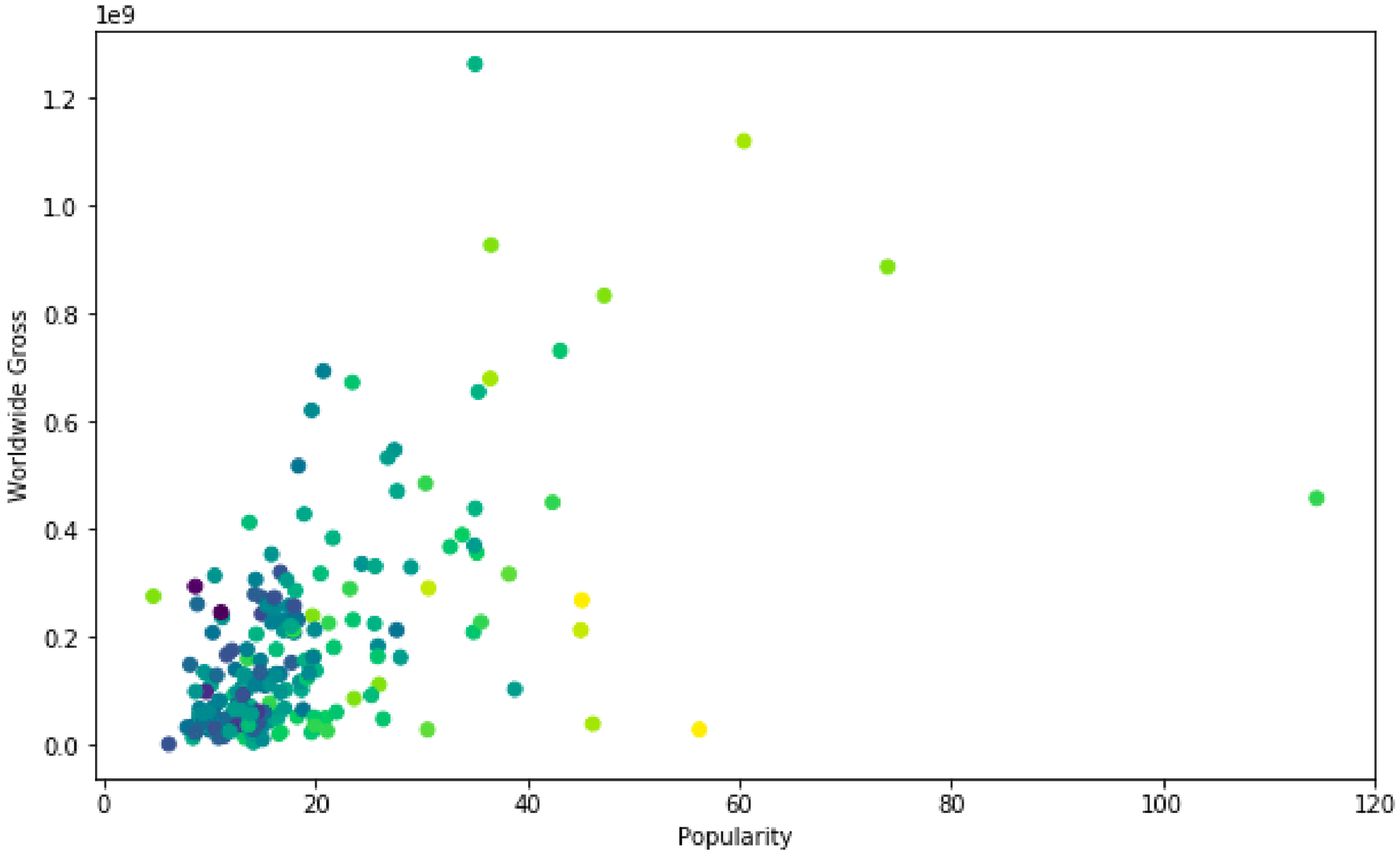
Exluding 2 films with ROI > 10,000





All together now

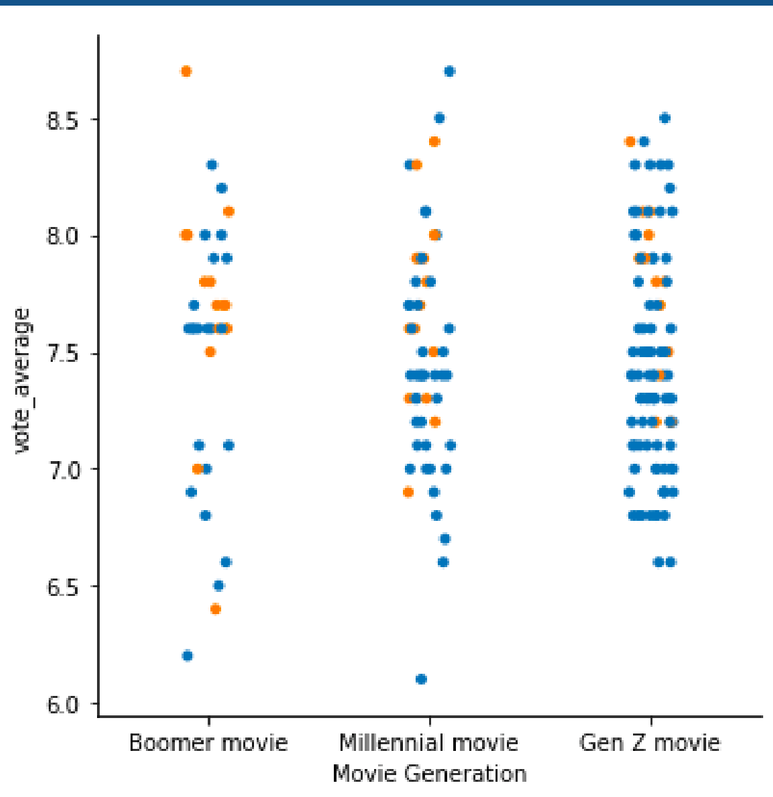
Box Office vs Popularity - paininting by number of vote average



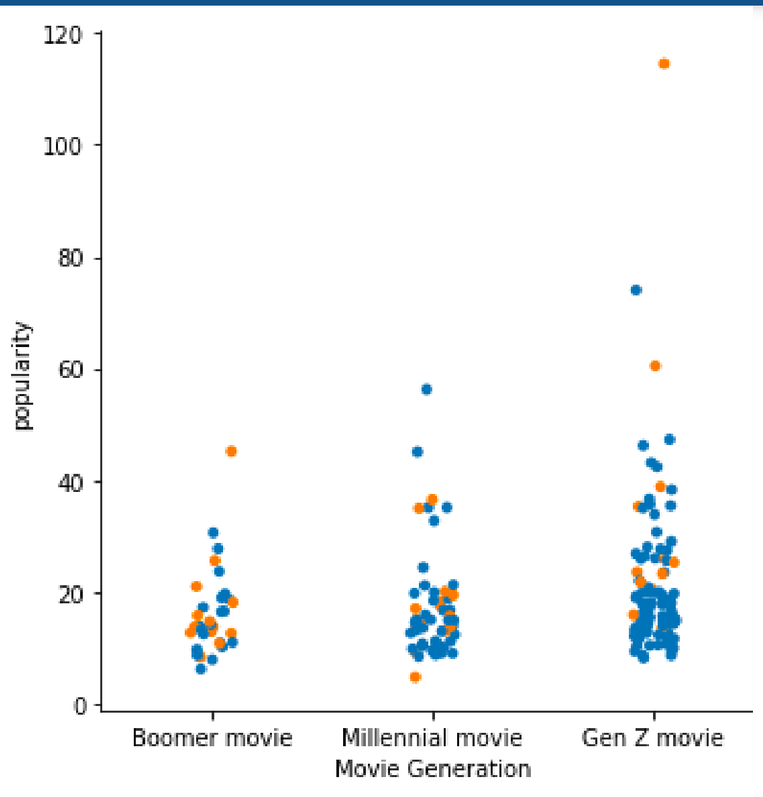
Generation comparison

```
movies_oscars.loc[(movies_oscars['year'] < 1980), 'Movie Generation'] = 'Boomer movie'
movies_oscars.loc[(movies_oscars['year'] >= 1980) & (movies_oscars['year'] < 1998), 'Movie Generation'] = 'Millennial r
movies_oscars.loc[(movies_oscars['year'] > 1998), 'Movie Generation'] = 'Gen Z movie'
```

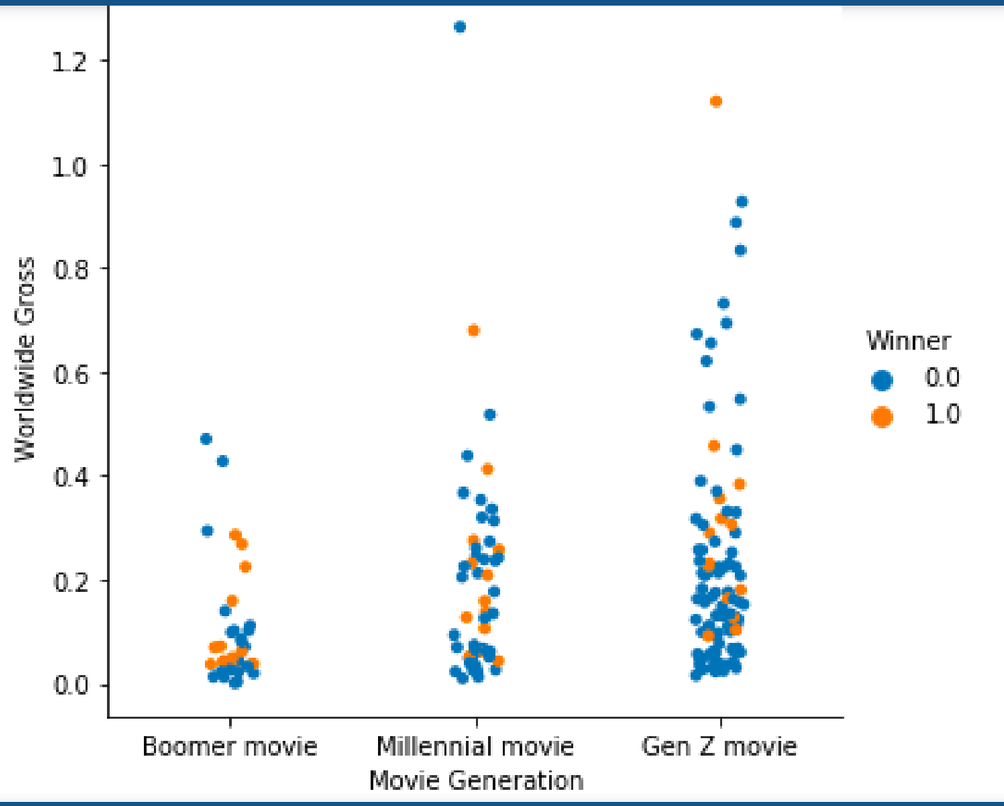
Vote Average



Popularity



Worldwide box office



Main challenges & Strengths

Something (not that) funny that happened during the project

What would I have done differently?


That's all Folks!