Project 2

By: Katie Greisiger Frost

Date submitted: March 1, 2022

Background

Project 1 investigated the correlation between horror films and their poor ratings from RottenTomatoes.com. However, there were limits to the data, such as box office information, that were not explored. in Project 2, I will explore a set of horror films to see if similar correlations exist from IMDB.com ratings, and expand the correlative relationships between ratings and production budgets or box office performance with data from the-numbers.com.

```
##Set up
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

##import the-numbers box office data
df_numbers = pd.read_excel("the_numbers_horror.xlsx",
usecols =
['Released_yr','Released_worldwide_yr','Title','Theatrical_
distributor',

'Genre','Production_budget','Opening_weekend_theaters','The
atrical_engagements',

'Opening_weekend_revenue','Domestic_box_office','Infl_adj_b
ox_office',

'International_box_office','Worldwide_box_office'])
```

```
##import imdb titles and titleid data
```

```
df_imdb_titles = pd.read_csv("imdb_titles-ABOR3428.tsv",
sep="\t", usecols = ['titleId','title','isOriginalTitle'],
low_memory=False)
df_imdb_titles =
df_imdb_titles[df_imdb_titles.isOriginalTitle == '1']
df_imdb_titles = df_imdb_titles.reset_index(drop=True)

##review head of imdb import titles
df_imdb_titles.head(3)
```

	titleId	title	isOriginalTitle
0	tt0000001	Carmencita	1
1	tt0000002	Le clown et ses chiens	1
2	tt0000003	Pauvre Pierrot	1

```
##import imdb ratings data based on titleid
df_imdb_ratings = pd.read_csv("imdb_ratings.tsv", sep="\t",
low_memory=False)

##review head of imdb import ratings
df_imdb_ratings.head(3)
```

	tconst	averageRating	numVotes
0	tt0000001	5.7	1864
1	tt0000002	6.0	244
2	tt0000003	6.5	1632

Datasets

The dataset containing box office and production budget information is from the-numbers.com. It includes 100 observations and 20 columns with metadata of horror films, and will act as the *parent* dataset to the two imdb.com datasets. The imdb datasets are *much* larger than the parent set, so I sliced them to get just the columns I needed for imdb ratings. I also excluded non-originl titles from the imdb titles' set. Next, the datasets will be merged into one dataset (imdb sets merged by 'titleId' and imdb merged to the-numbers by 'Title').

```
##merge imdb datasets based on titleId and tconst
```

```
df_imdb_merged=df_imdb_titles.set_index('titleId').join(df_
imdb_ratings.set_index('tconst'), how='inner')
df_imdb_merged.head(3)
```

	title	isOriginalTitle	averageRating	numVotes
tt0000001	Carmencita	1	5.7	1864
tt0000002	Le clown et ses chiens	1	6.0	244
tt0000003	Pauvre Pierrot	1	6.5	1632

```
##merge the-numbers to merged imdb dataset based on Title
values
df_final_merged = pd.merge(
    df_numbers, df_imdb_merged,
    left_on=['Title',
df_numbers.groupby('Title').cumcount()],
    right_on=['title',
df_imdb_merged.groupby('title').cumcount()]
)

##Review datatypes of columns
df_final_merged.dtypes
```

```
Released_yr                 int64
Released_worldwide_yr       int64
Title                      object
Theatrical_distributor     object
Genre                      object
Production_budget           int64
Opening_weekend_theaters    int64
Theatrical_engagements      int64
Opening_weekend_revenue     int64
Domestic_box_office         int64
Infl_adj_box_office         int64
International_box_office    int64
Worldwide_box_office        int64
title                      object
isOriginalTitle            object
averageRating             float64
numVotes                    int64
dtype: object
```

Methods and Results

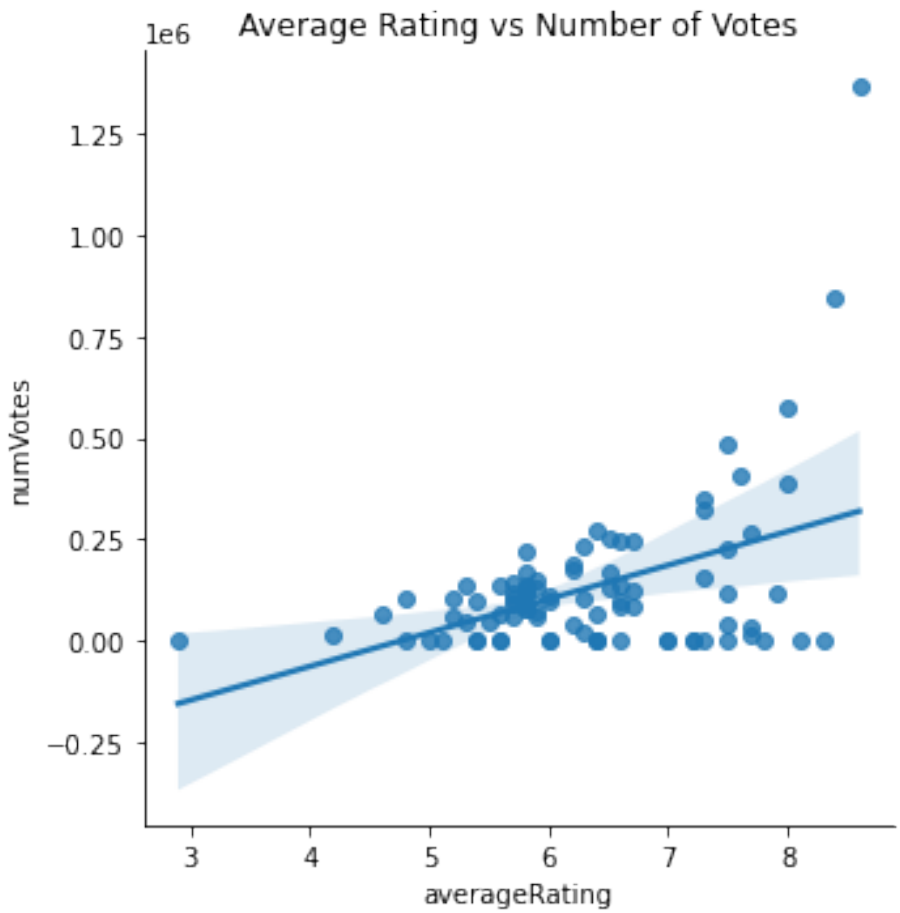
Of the 100 horror films in the-numbers dataset, I was able to join 88 of them to the merged imdb dataset by the title strings. This will be the final merged set that is used in the analysis. I first explored the top rated, top budget, and top worldwide box office titles to loosely look for trends.

I don't see any of the same titles in the top 5 average ratings and top 5 number of rating votes. Between the highest budget list and the highest worldwide box office list, I see that Title *Candyman* is in both lists. Perhaps this is signaling to a positive correlation between worldwide box office and production budget.

Relationships were further visualized using scatter subplots using Seaborn.

```
##Explore raltionship between rating and number of votes
sns.lmplot(x='averageRating',y='numVotes', data =
df_final_merged).set(title = "Average Rating vs Number of
Votes")
```

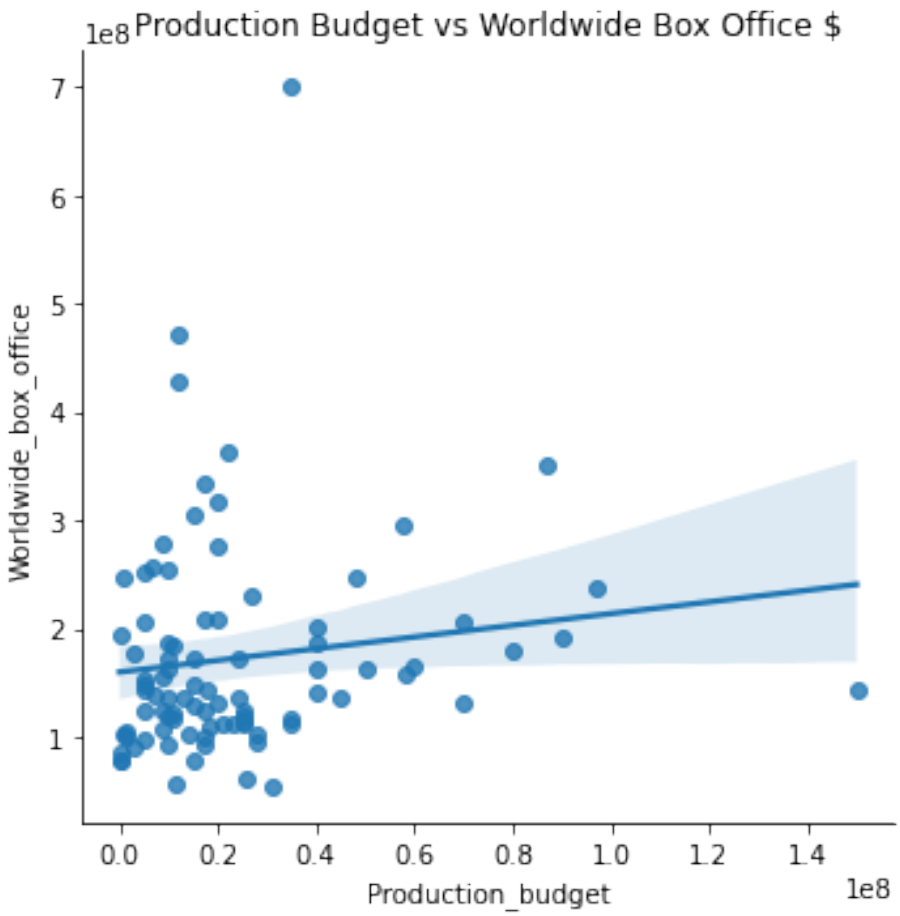
```
<seaborn.axisgrid.FacetGrid at 0x262ea345130>
```



When comparing the average ratings to the number of votes, the regression line is progressing upward to the right as the average rating value increases. This suggests that there may be a positive correlation between ratings and the number of ratings.

```
##Explore raltionship between budget and worldwide box
office
sns.lmplot(x='Production_budget',y='Worldwide_box_office',
data = df_final_merged).set(title = "Production Budget vs
Worldwide Box Office $")
```

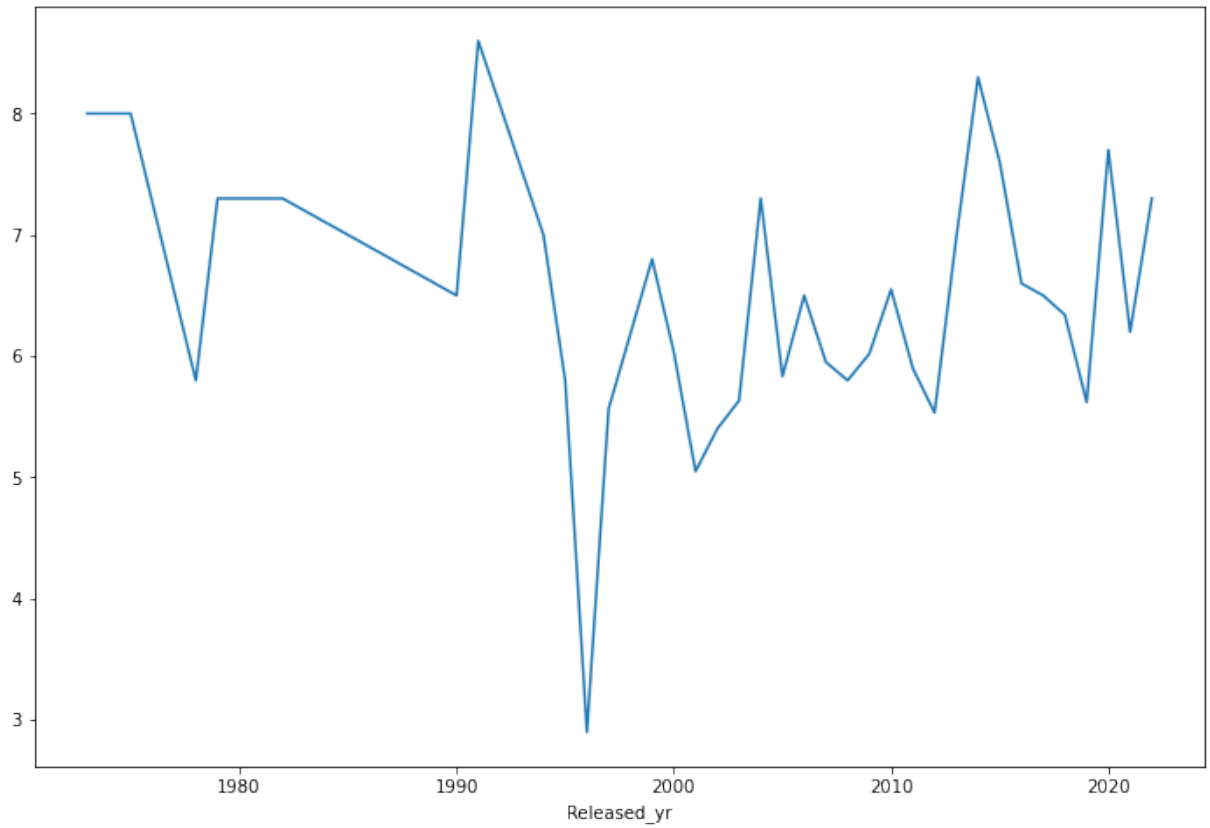
```
<seaborn.axisgrid.FacetGrid at 0x262e0a8f0d0>
```



When comparing the production budget to the worldwide box office (US$), the regression line is progressing upward to the right just slightly as the production budget increases. The plot suggests that there may be a positive correlation between budgets and box office, however it also may be illustrating a couple of outliers, as well.

Next, I explored average ratings by year and the results do not seem to support an easily-identifiable trend, as seen in the below line chart.

```
fig, ax = plt.subplots(figsize=(12,8))
df_final_merged.groupby(['Released_yr']).mean()
['averageRating'].plot(ax=ax)
<AxesSubplot:xlabel='Released_yr'>
```



As a reminder and comparison, the RottenTomatoes.com dataset utilized in Project 1 was aggregated to review the *overall average* (mean of tomatometer and audience rating) of horror films by year released. To recreate the Project 1 bar chart highlighting overall mean by year released, I first import and manipulate the RottenTomatoes.com dataset so only horror film data is utilized.

Note The IMDB rating system differs from Rotten Tomatoes in that ratings are cast from a 1 to 10 range; 10 being the highest rating. For the purpose of this project, a 1 IMDB will be similar to a 10% rating from Rotten Tomatoes, 2 to 20%, 3 to 30%, and so on.

```
##import Rotten Tomatoes data from Project 1
df_rt_movies = pd.read_csv("rotten_tomatoes_movies.csv",
usecols =
['rotten_tomatoes_link','movie_title','genres','original_re
lease_date','tomatometer_rating','audience_rating'])

#year released
df_rt_movies['year_released'] =
```

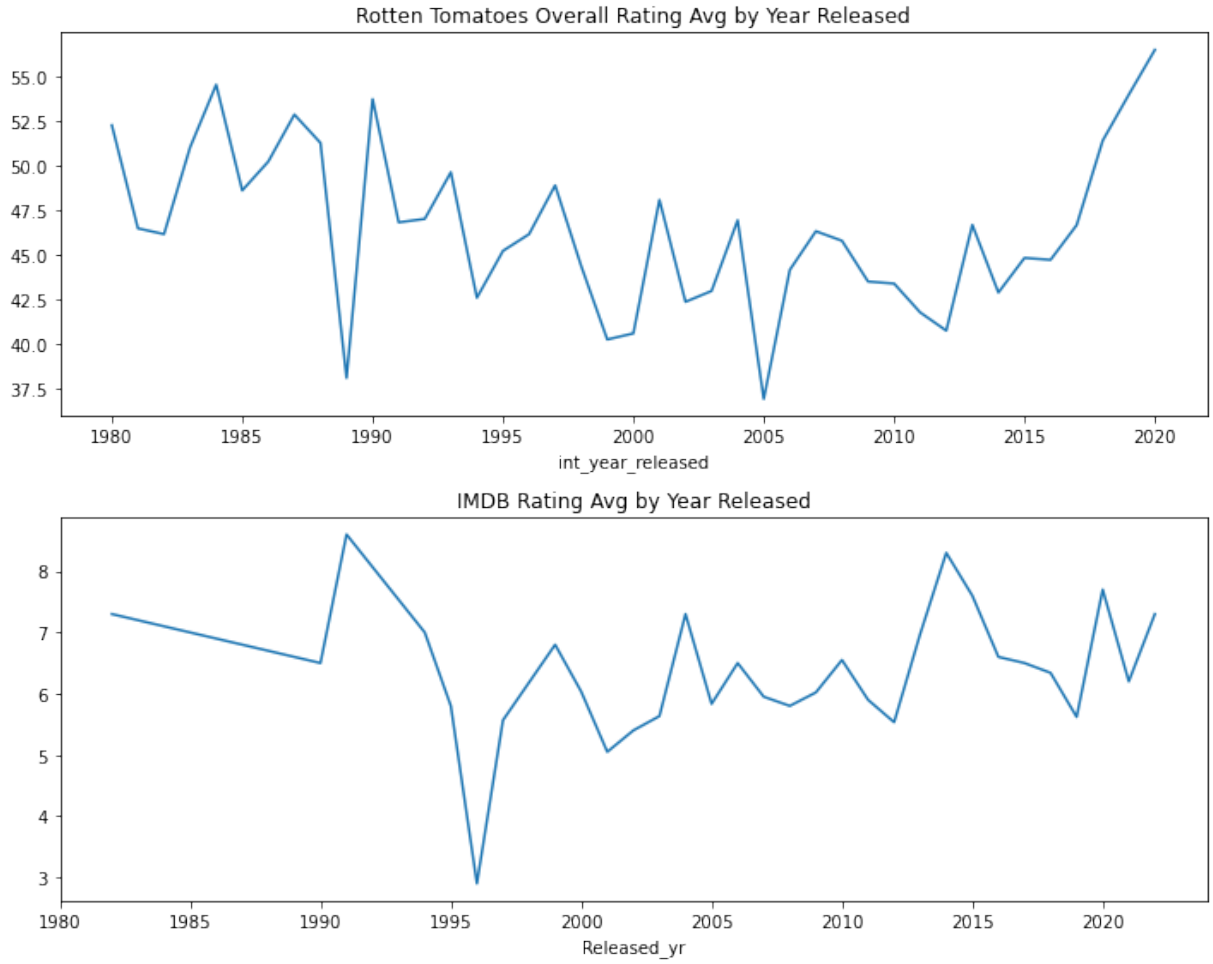
```
df_rt_movies['original_release_date'].str[-4:]

#find overall average rating
df_rt_movies['overall_rating'] =
(df_rt_movies['tomatometer_rating'] +
df_rt_movies['audience_rating']) / 2

#exclude isHorror == False
df_rt_movies['isHorror'] =
df_rt_movies['genres'].str.contains('Horror',case=True,rege
x=False)
df_rt_movies = df_rt_movies[df_rt_movies.isHorror == True]
df_rt_movies = df_rt_movies.reset_index(drop=True)

df_rt_movies.head(5)
```

	rotten_tomatoes_link	movie_title	genres	original_release_date	tomatometer_rating	audience_rating	year_released	overall_rating	isHorror
0	m/10002114-dark_water	Dark Water	Art House & International, Horror, Mystery & S...	1/19/2002	80.0	66.0	2002	73.0	True
1	m/10003925-dead_end	Dead End	Comedy, Horror, Mystery & Suspense	9/26/2003	75.0	60.0	2003	67.5	True
2	m/10004504-ultraviolet	Ultraviolet	Action & Adventure, Drama, Horror, Science Fic...	3/3/2006	8.0	30.0	2006	19.0	True
3	m/10004684-malevolence	Malevolence	Horror	9/10/2004	33.0	83.0	2004	58.0	True
4	m/10004697-eternal	Eternal	Horror, Mystery & Suspense	8/26/2004	21.0	39.0	2004	30.0	True

Next, we review a chart highlighting the overall mean ratings by year released from RottenTomatoes.com to compare to Project 2 mean rating scores from IMDB. First the year released column values in the

Rotten Tomatoes set needed to be coerced to 0 if NaN and then to an interger value so that the range of years released could be similar to IMDB (i.e., >= 1980).

```
##coerce year_released NaN to 0 values
import numpy as np
df_rt_movies[['int_year_released']] =
df_rt_movies[['year_released']].fillna(0)
df_rt_movies['int_year_released'] =
pd.to_numeric(df_rt_movies['int_year_released'])
df_rt_movies['yr_1980_or_greater'] =
df_rt_movies['int_year_released'] > 1979
df_rt_movies2 =
df_rt_movies[df_rt_movies.yr_1980_or_greater == True]
df_rt_movies2 = df_rt_movies2.reset_index(drop=True)
df_final_merged[['int_Released_yr']] =
df_final_merged[['Released_yr']].fillna(0)
df_final_merged['int_Released_yr'] =
pd.to_numeric(df_final_merged['int_Released_yr'])
df_final_merged['yr_1980_or_greater'] =
df_final_merged['int_Released_yr'] > 1979
df_final_merged4 =
df_final_merged[df_final_merged.yr_1980_or_greater == True]
df_final_merged4 = df_final_merged4.reset_index(drop=True)
fig = plt.figure(num=1, clear=True, figsize= (10,8))
ax1 = fig.add_subplot(2, 1, 1)
ax2 = fig.add_subplot(2, 1, 2)
df_rt_movies2.groupby(['int_year_released']).mean()
['overall_rating'].plot(ax=ax1)
df_final_merged4.groupby(['Released_yr']).mean()
['averageRating'].plot(ax=ax2)
ax1.set_title("Rotten Tomatoes Overall Rating Avg by Year
Released")
ax2.set_title("IMDB Rating Avg by Year Released")
fig.tight_layout()
```

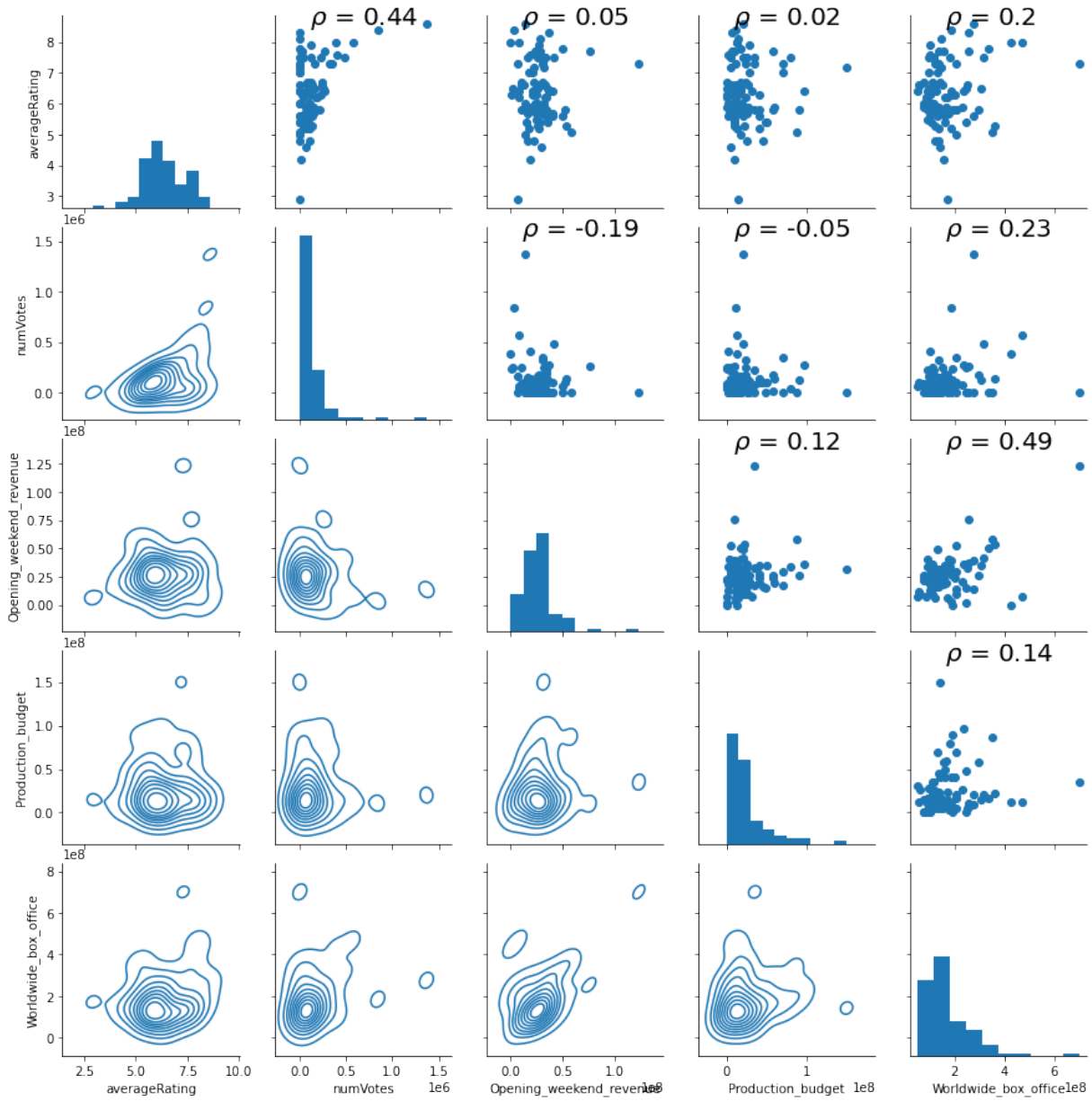


Interestingly, we see some similar trends in the two line charts above. Both experience a steep drop in mean ratings around years 1990 and 1995; several released years from the IMDB dataset have mean ratings > 6, which is equivocally greater than the highest mean ratings for Rotten Tomatoes (~55%); and similarly the ratings after 2020 are on their way up in both charts.

The exploration completed above helped to narrow in on a focus for the final analyses and results. A Seaborn PairGrid method was used to plot relationships between numerical variables, while a function utilized to add p values to the grid. Variables "averageRating", "numVotes", "Opening_weekend_revenue", "Production_budget", and "Worldwide_box_office" will be included in the correlative analysis.

```
##Correlogram 1
df_correl_1 =
df_final_merged[["averageRating","numVotes","Opening_weeken
```

```
d_revenue",

"Production_budget","Worldwide_box_office"]].copy()

#coefficient calculation function
def corr(x, y, **kwargs):

    # Calculate the value
    coef = np.corrcoef(x, y)[0][1]
    # Make the label
    label = r'$\rho$ = ' + str(round(coef, 2))

    # Add the label to the plot
    ax = plt.gca()
    ax.annotate(label, xy = (0.2, 0.95), size = 20,
xycoords = ax.transAxes)

# Create a pair grid instance
grid = sns.PairGrid(data=df_correl_1)

# Map the plots to the locations
grid = grid.map_upper(plt.scatter)
grid = grid.map_upper(corr)
grid = grid.map_lower(sns.kdeplot)
grid = grid.map_diag(plt.hist, bins = 10)
```



Looking at the above correlation output, we see that just one of the variable relationships shows a statistically significant finding in the subset of horror films data. The relationship between fields "averageRating" and "Production_budget" show a positive correlation ($p=0.02$).

Conclusion

Similar to the conclusion in Project 1, there doesn't seem to yet be a concrete way to predict the ratings of horror films on IMDB.com based off of analysis of variable relationships. This project was, of course, just a subset of 88 horror films, so a future study could utilize larger, developer datasets from IMDB APIs, and/or explore relatioships between horror film ratings and other associated genres, actors, directors, and more.

Appendix

Data sources:

https://datasets.imdbws.com/ ('title.akas.tsv.gz' and 'title.ratings.tsv.gz')

https://www.the-numbers.com/movies/report-builder (Genre = 'Horror')

https://www.rottentomatoes.com/ by way of https://www.kaggle.com/stefanoleone992/rotten-tomatoesmovies-and-critic-reviews-dataset (Project 1 dataset)