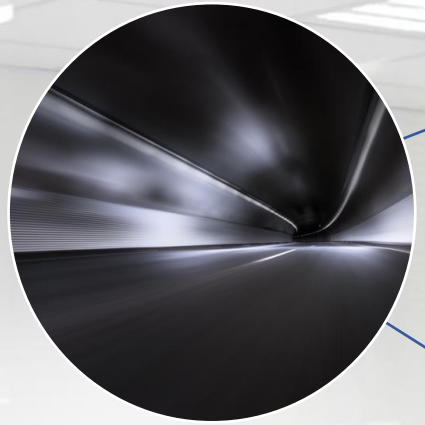
Does a higher budget determine how lucrative a **movie's profit** will be?

Team #3

Datasets

- Budget
- Revenue
- Genres

- Movie Titles
- IMDb Ratings



- Popularity - Only 9,043 data available out of 84,294 titles

- Netflix titles -568 titles with data available for only 44 original titles


Pulling data
Act 2 - Cass
PRODUCTION
DIRECTOR
CAMERA
SCENE
TAKE

Pull OMDB

```
# # make URL
url_omdb = "http://www.omdbapi.com/?apikey="+ omdb_key + "&i="

error_count = 0

for index, row in OMDB_titles_clean_df.iterrows():
    try:
        movie_data = requests.get(url_omdb + str(movie_titles_clean.tconst[index])).json()
        try:
            OMDB_titles_clean_df.loc[index, 'Metascore'] = movie_data['Metascore']
            OMDB_titles_clean_df.loc[index, 'imdbRating'] = movie_data['imdbRating']
            OMDB_titles_clean_df.loc[index, 'imdbVotes'] = movie_data['imdbVotes']
            OMDB_titles_clean_df.loc[index, 'Title'] = movie_data['Title']
        except (IndexError, KeyError, ValueError):
            error_count +=1
    # Added for OMDB errors when their system returns JSONDecodeError (ValueError on their side - years 2016, 2017)
    except(ValueError, TypeError):
        error_count +=1
```

```
# change title name to have + instead of ' '
TMDB_movies_df['primaryTitle'] = TMDB_movies_df['primaryTitle'].str.replace(" ", "+")


# ******Error 1: need to remove # from the beginning of titles for TMDB to work

# variable cause starswith() wasn't happy with '#'
pound_sign = '#'

# make dataframe for pound sign = True (startswith() returns True/False)
replace_pound_df = TMDB_movies_df.iloc[:, 0:3]
replace_pound_df.primaryTitle = replace_pound_df.primaryTitle.str.startswith(pound_sign)

# make df for ONLY the True values + primaryTitle from TMDB_movies_df
pound_true_df = replace_pound_df.loc[replace_pound_df.primaryTitle == True]
pound_true_df['TITLE'] = TMDB_movies_df['primaryTitle']

# Fix titles to not have # in the front & clean up columns
pound_true_df['TITLE'] = pound_true_df['TITLE'].str.replace(pound_sign, "")
pound_true_clean_df = pound_true_df.drop(columns=['primaryTitle', 'startYear'])
pound_true_clean_df = pound_true_clean_df.rename(columns={'TITLE': 'primaryTitle'})

# Merge 2 dfs, replace blank primaryTitle_y values with na so you can do fillna into a
# nice new clean has correct info column & delete primaryTitle_y/x
titles_combined_df = pd.merge(TMDB_movies_df, pound_true_clean_df, how='outer', on='tconst')
titles_combined_df['primaryTitle_y'] = titles_combined_df['primaryTitle_y'].str.replace(" ", "nan")
titles_combined_df["primaryTitle"] = titles_combined_df["primaryTitle_y"].fillna(titles_combined_df["primaryTitle_x"])
titles_fixed_df = titles_combined_df.drop(columns=['primaryTitle_y', 'primaryTitle_x'])


# FINALLY make movie titles into a list so you can run it
movies = titles_fixed_df['primaryTitle'].tolist()
```


Cleaning data
Act 3 – Sriven
PRODUCTION
DIRECTOR
CAMERA
SCENE
TAKE

CLEANING if budget = 0, revenue = 0, IMDB_id not found

- This is to help keep the file size down by dropping rows we cannot use or cannot match up

In [24]:
```
movie_info_pulled_df = TMDB_df.copy()
movie_info_pulled_df.head()

movie_info_pulled_df = movie_info_pulled_df[movie_info_pulled_df.budget != 0]
movie_info_pulled_df = movie_info_pulled_df[movie_info_pulled_df.revenue != 0]
movie_info_pulled_df = movie_info_pulled_df.dropna(subset=['imdb_id'])

final_number = movie_info_pulled_df.imdb_id.count()
```

Save results as a CSV

In [25]:
```
total_errors = beginning_number - final_number

file_outpath_FINAL = f"Resources/TMDB_pull_FINAL_{year}_dropped_movies_{total_errors}.csv"
movie_info_pulled_df.to_csv(file_outpath_FINAL)
movie_info_pulled_df.head(2)
```

Out[25]:

	ID	imdb_id	release_date	budget	revenue	genres	original_language	original_title	origin_country	production_countries name	spoken_languages name
3	504562	tt0385887	2019-10-31	26000000	18377736	Drama	English	Motherless Brooklyn	US	United States of America	English
17	586776	tt10011102	2019-03-08	1000	1000	Action	हिन्दी	The Sholay Girl	IN	India	हिन्दी

Clean Years

2015

In [117]:
```
def clean_year(file):
    del file["Unnamed: 0"]
    del file["ID"]
    del file["original_title"]
    return file

def profit(file):
    file["Profit%"] = round((file['revenue']-file['budget'])/file['budget']*100,2)

clean_year(movies_2015).head(2)
```

Out[117]:

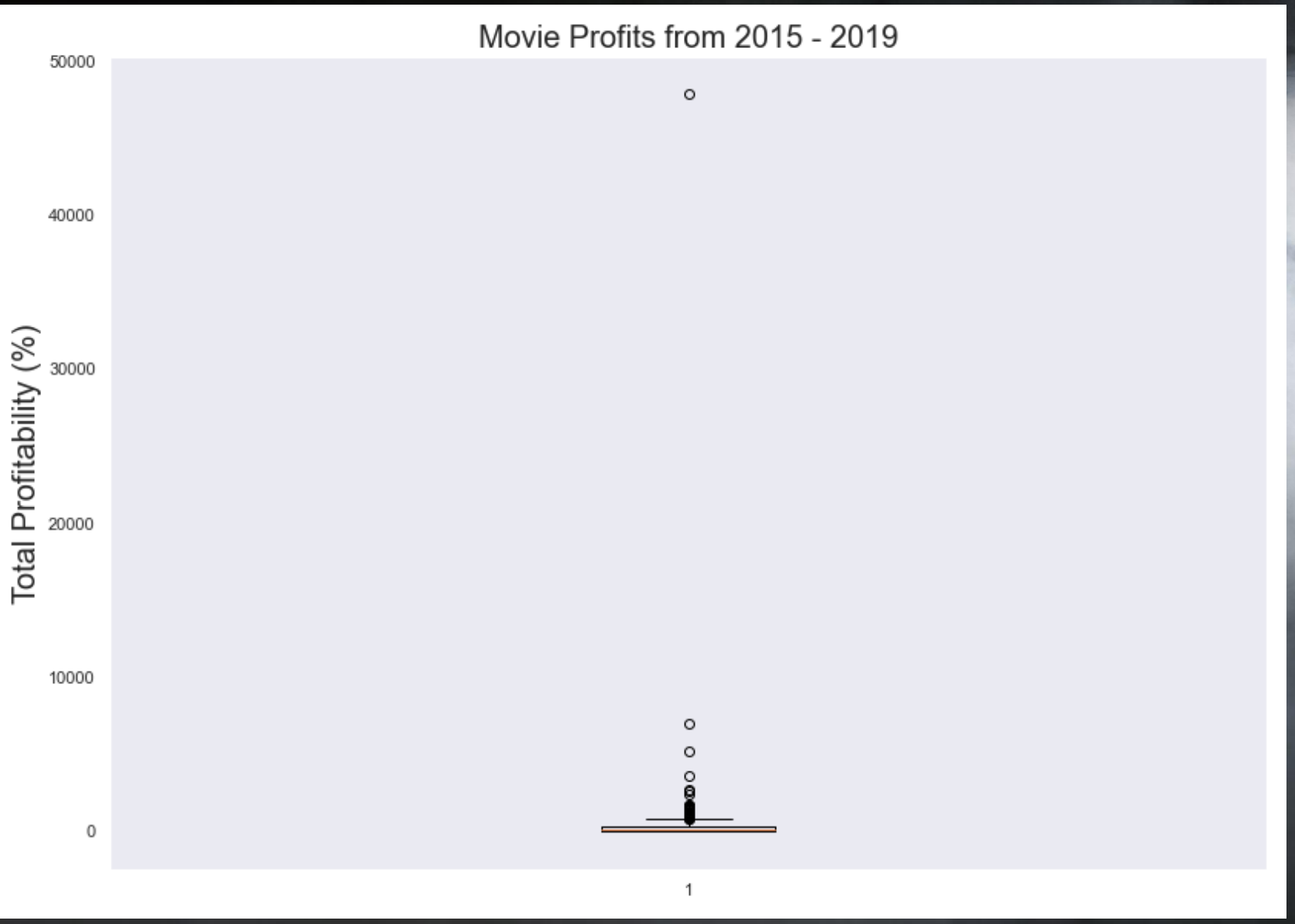
	imdb_id	release_date	budget	revenue	genres	original_language	origin_country	production_countries name	spoken_languages name
0	tt0810819	2015-01-01	15000000	64191523	Drama	Français	DE	Germany	Français
1	tt0884732	2015-01-16	23000000	79799880	Comedy	English	US	United States of America	English

In [118]:
```
movies_2015.reset_index().head()
movies_2015 = movies_2015.drop(index=123)
movies_2015 = movies_2015.rename(columns={'imdb_id':'tconst'})
movies_2015['budget'] = pd.to_numeric(movies_2015['budget'])
movies_2015['revenue'] = pd.to_numeric(movies_2015['revenue'])
profit(movies_2015)
movies_2015.head(2)
```

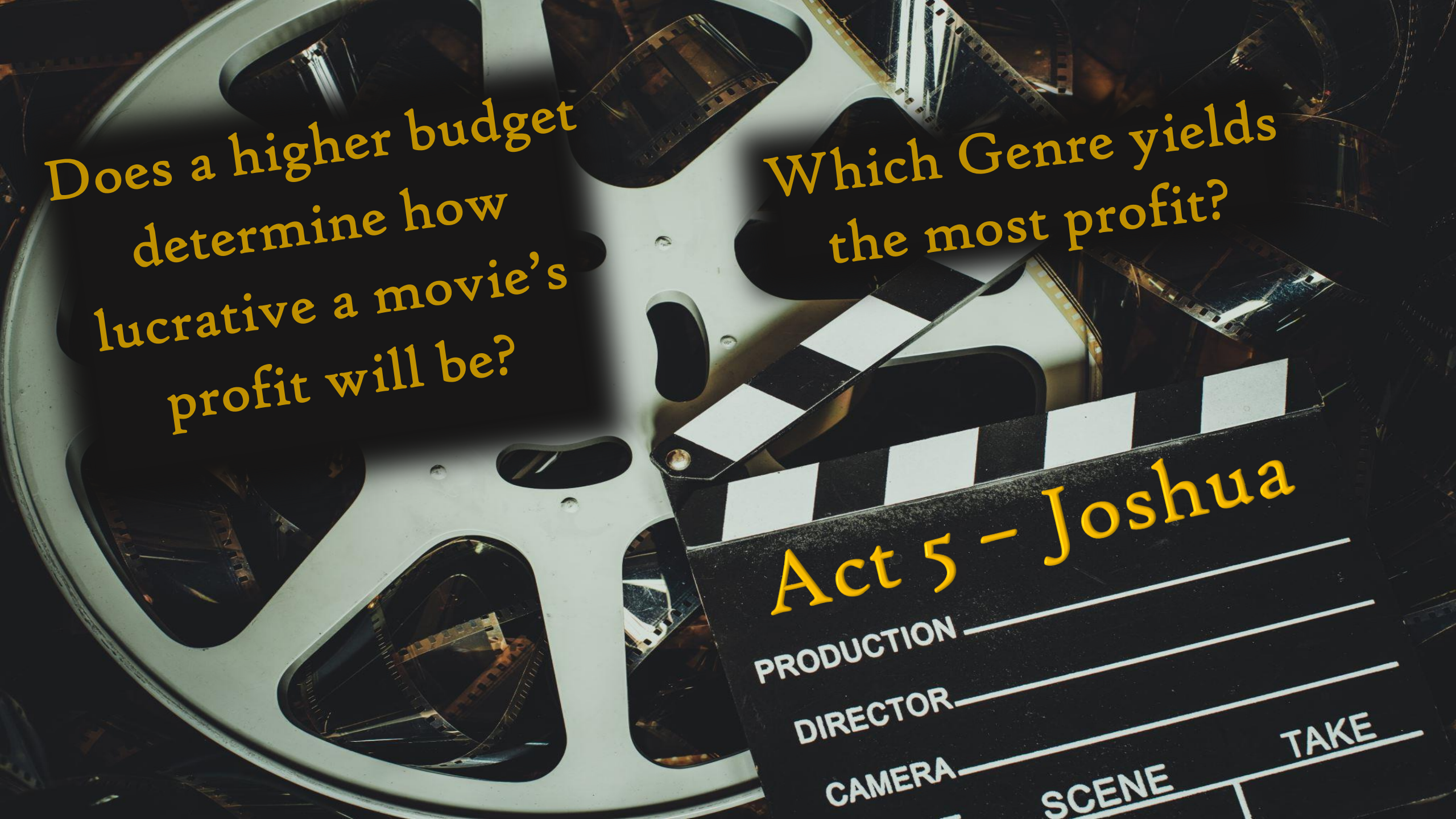
Out[118]:

	tconst	release_date	budget	revenue	genres	original_language	origin_country	production_countries name	spoken_languages name	Profit%
0	tt0810819	2015-01-01	15000000	64191523	Drama	Français	DE	Germany	Français	327.94
1	tt0884732	2015-01-16	23000000	79799880	Comedy	English	US	United States of America	English	246.96

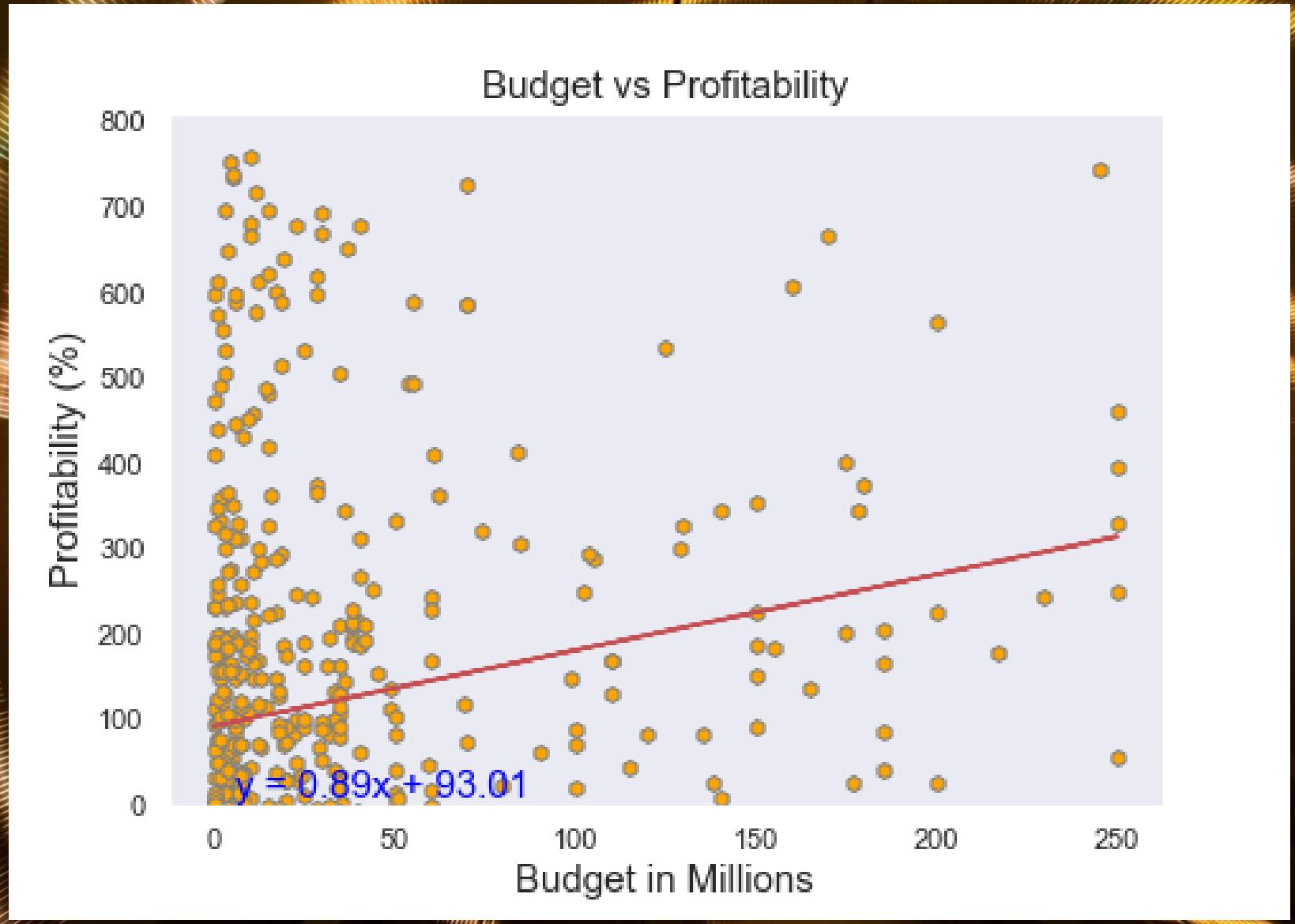
Identifying Outliers

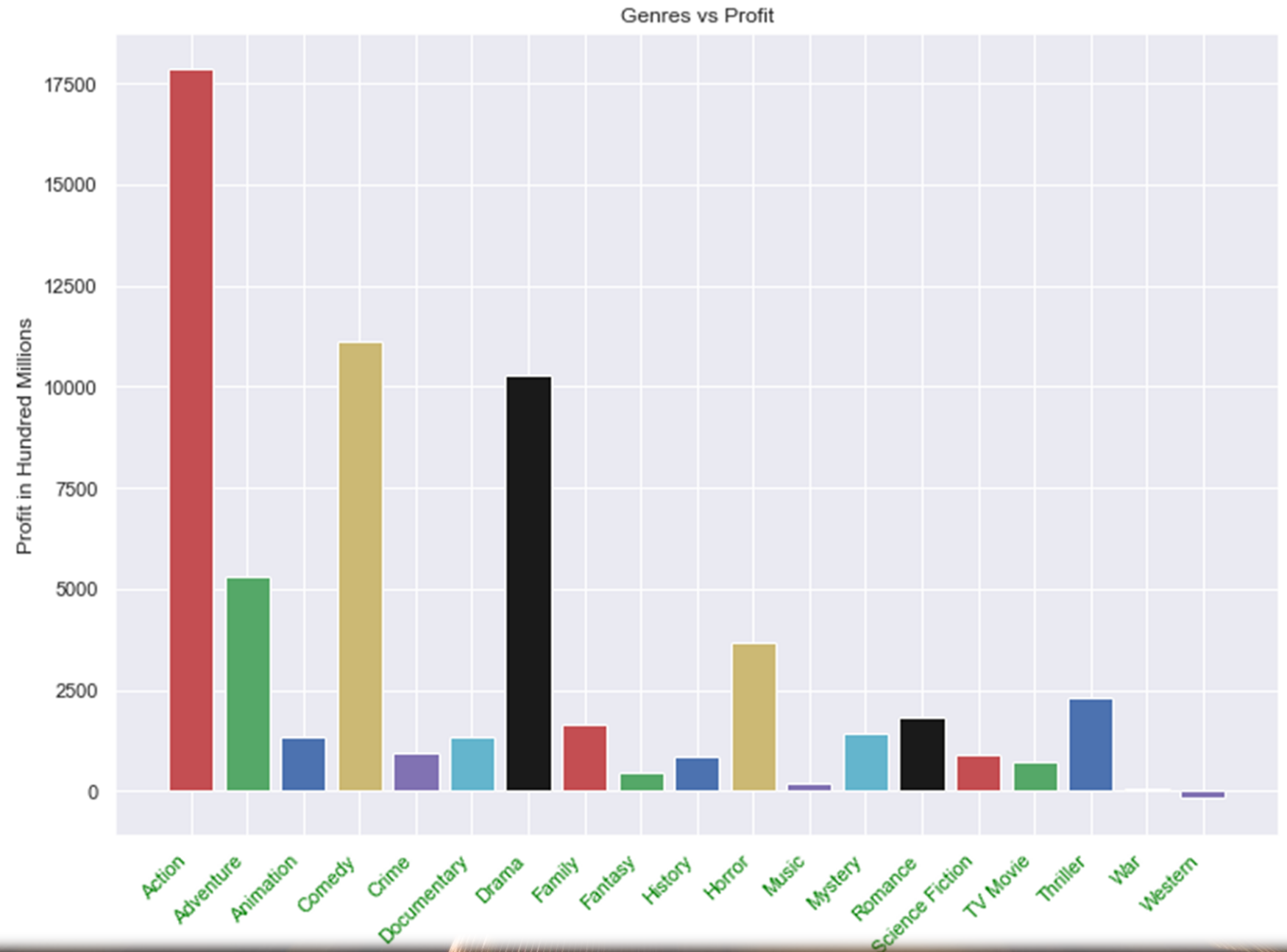


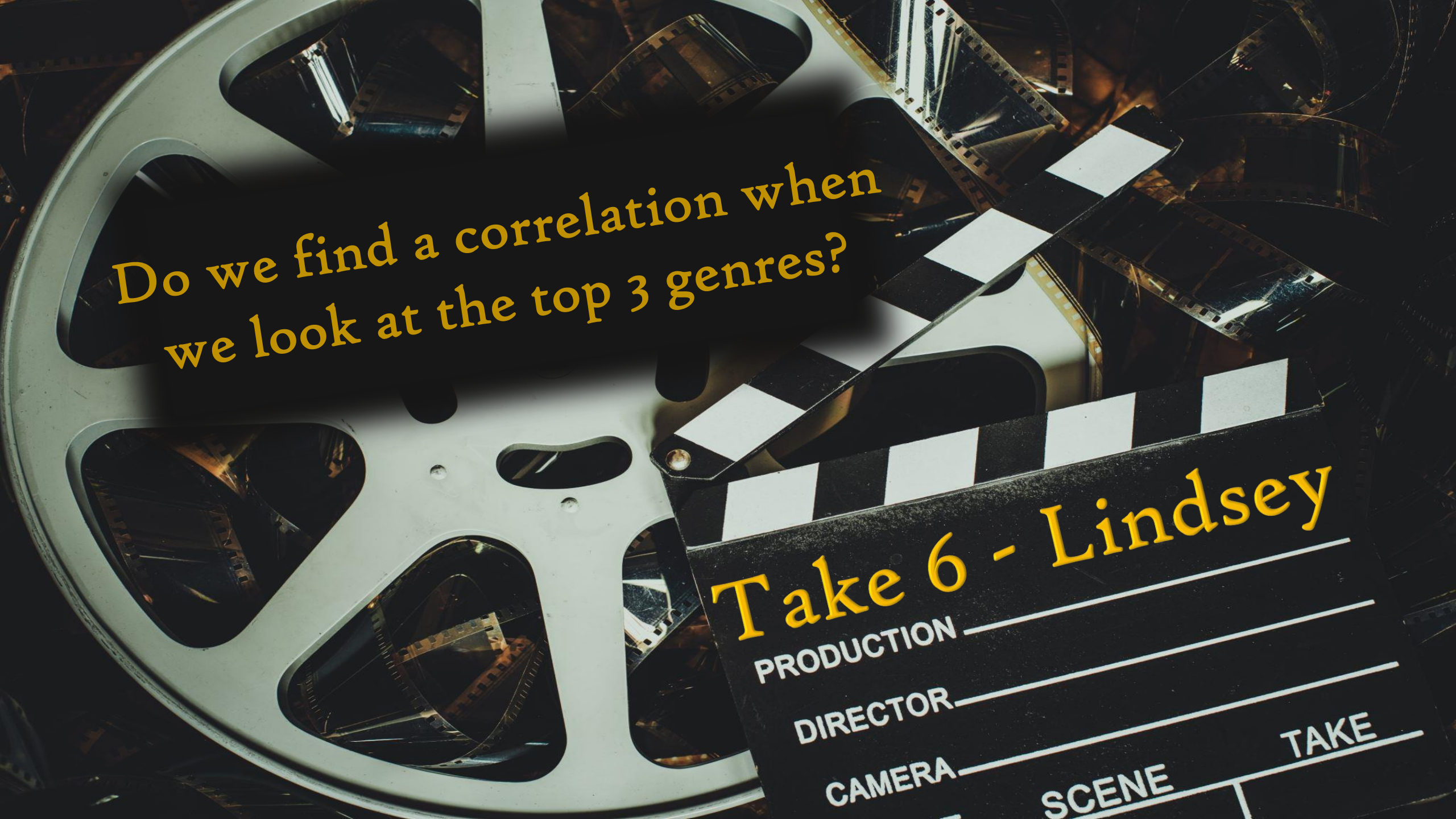
Remove outliers for analysis

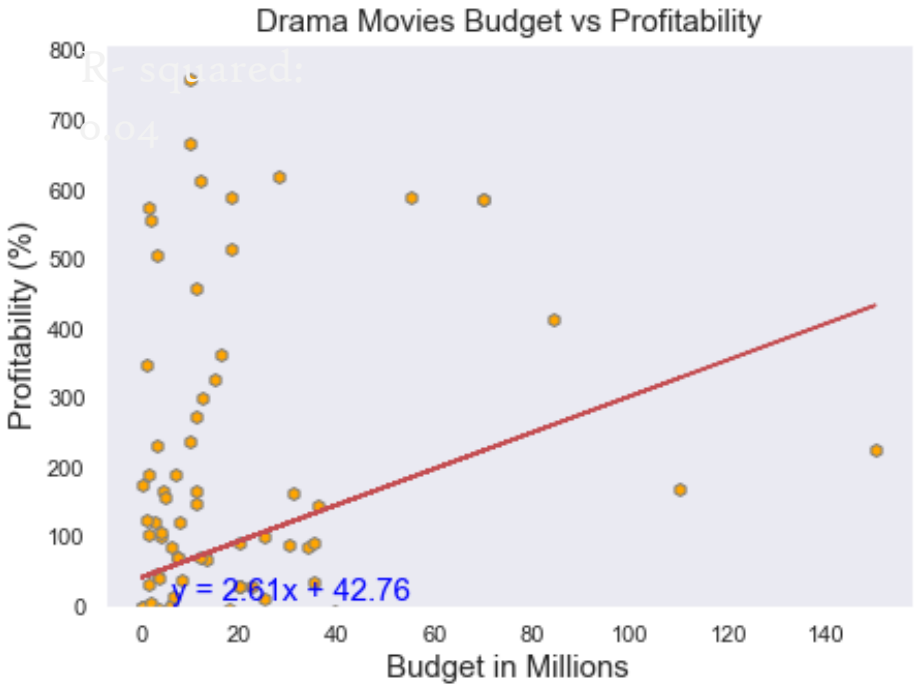

Does a higher budget determine how lucrative a movie's profit will be?
Which Genre yields the most profit?
Act 5 - Joshua
PRODUCTION
DIRECTOR
CAMERA
SCENE
TAKE

R- squared:
0.04

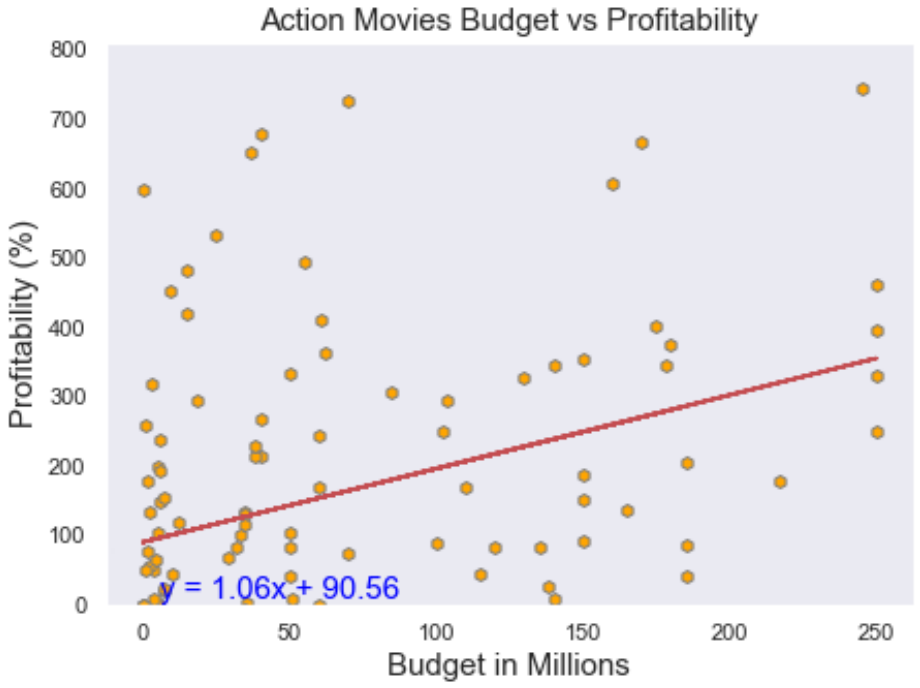



Genres vs Profit
Profit in Hundred Millions
17500
15000
12500
10000
7500
5000
2500
0
Action
Adventure
Animation
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Horror
Music
Mystery
Romance
Science Fiction
TV Movie
Thriller
War
Western

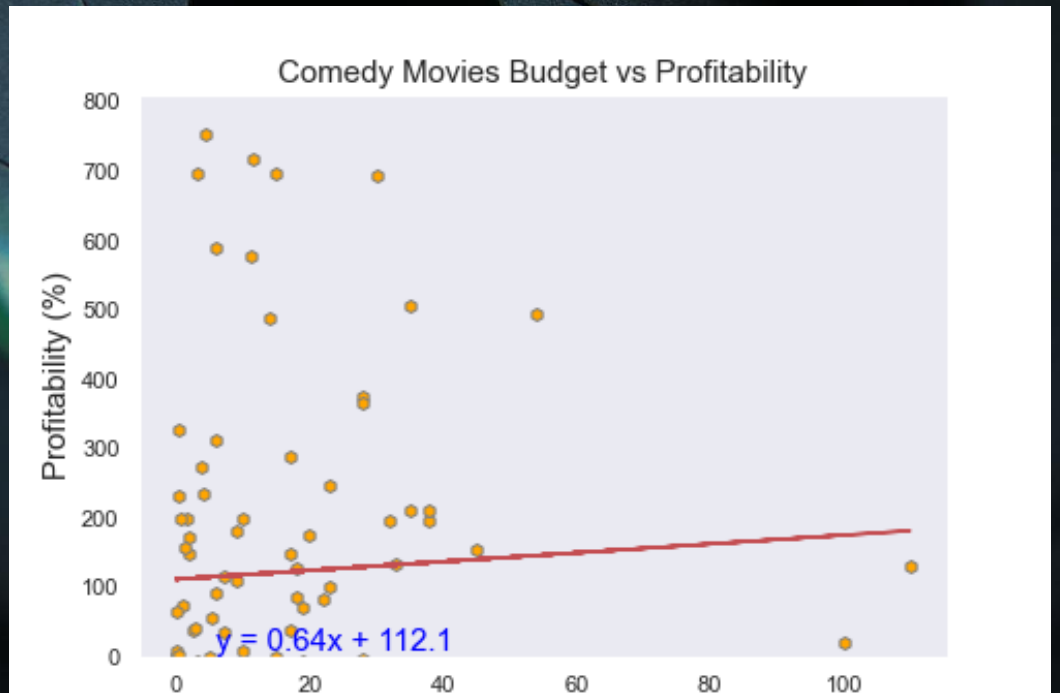

Do we find a correlation when we look at the top 3 genres?
Take 6 - Lindsey
PRODUCTION
DIRECTOR
CAMERA
SCENE
TAKE



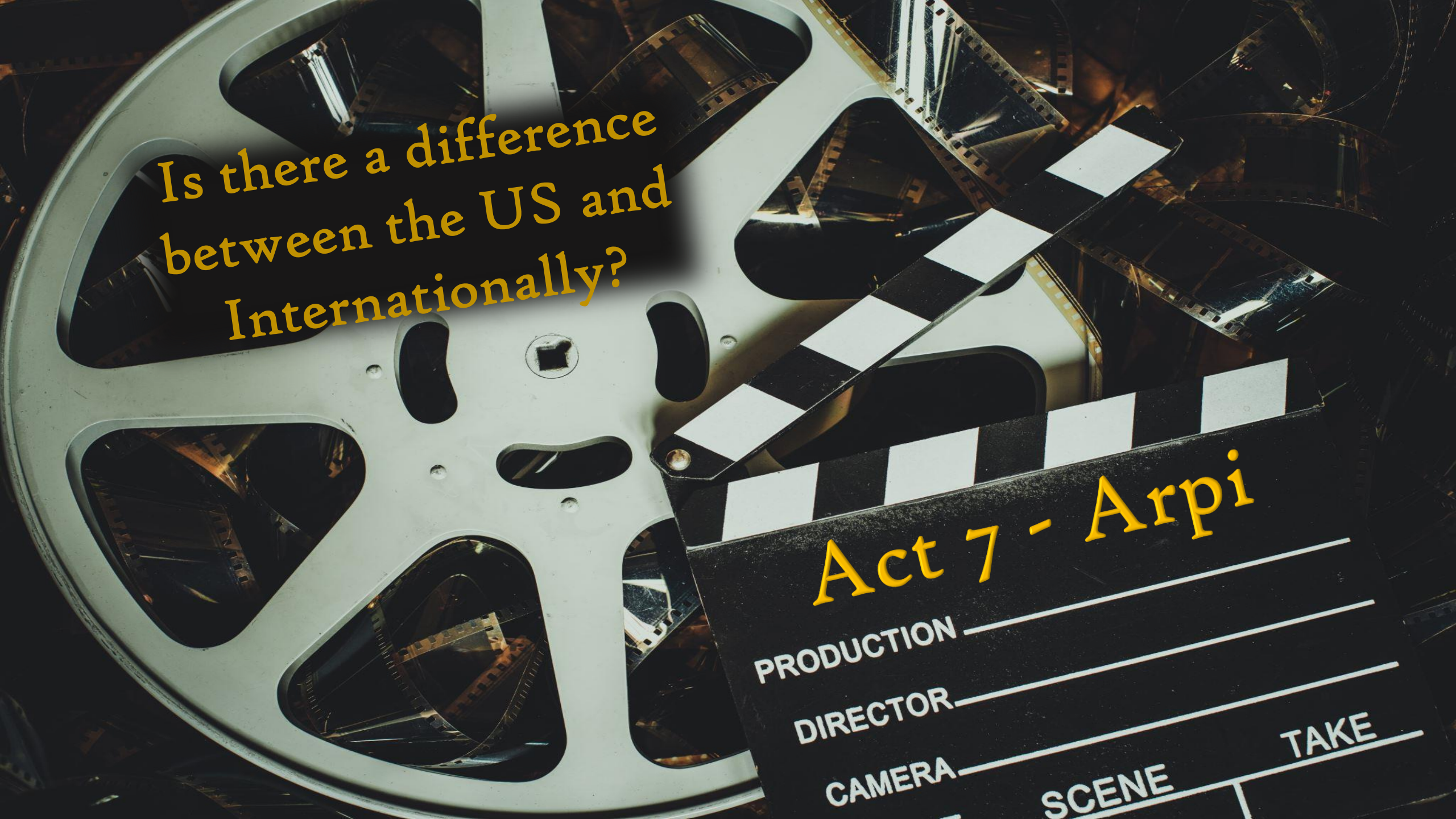
Comedy
R- squared: 0.00



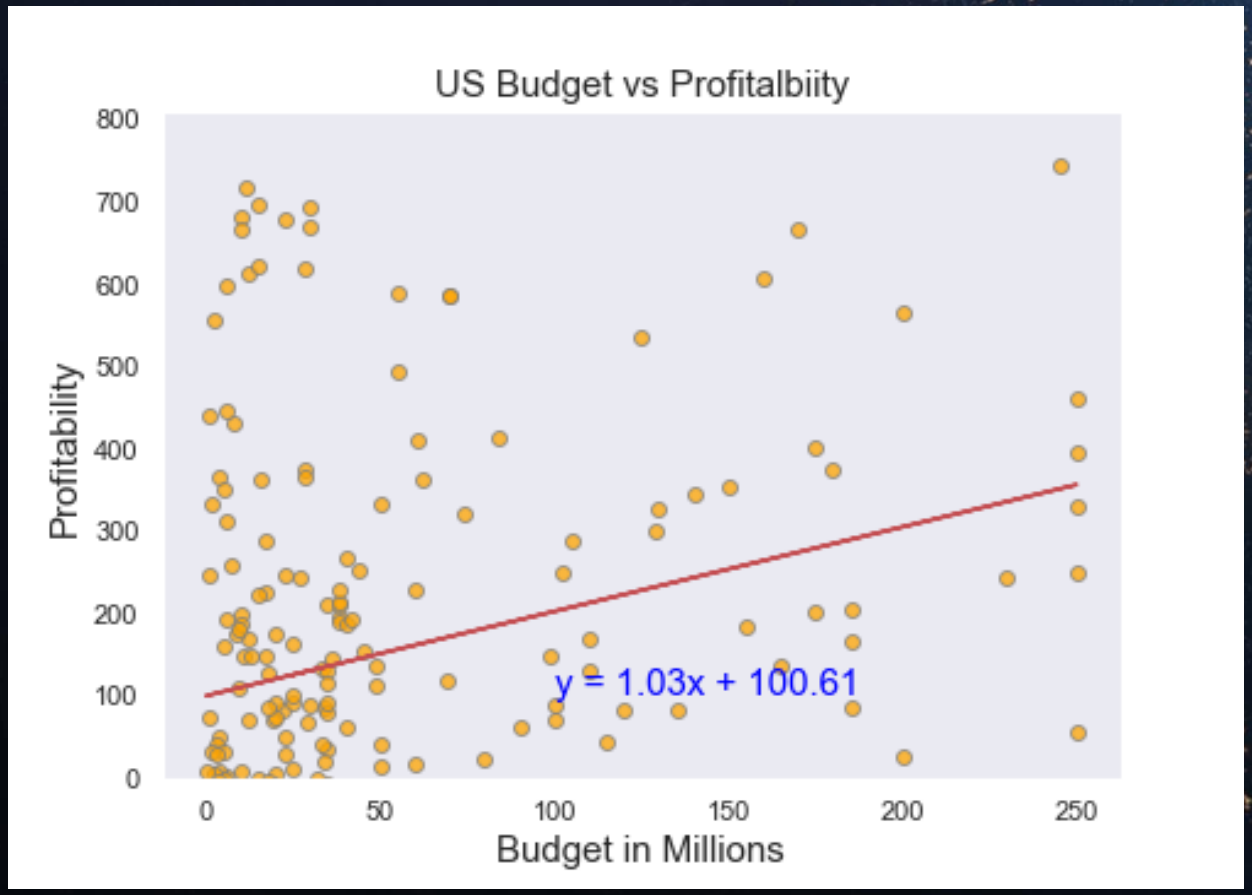
Drama
R- squared: 0.07



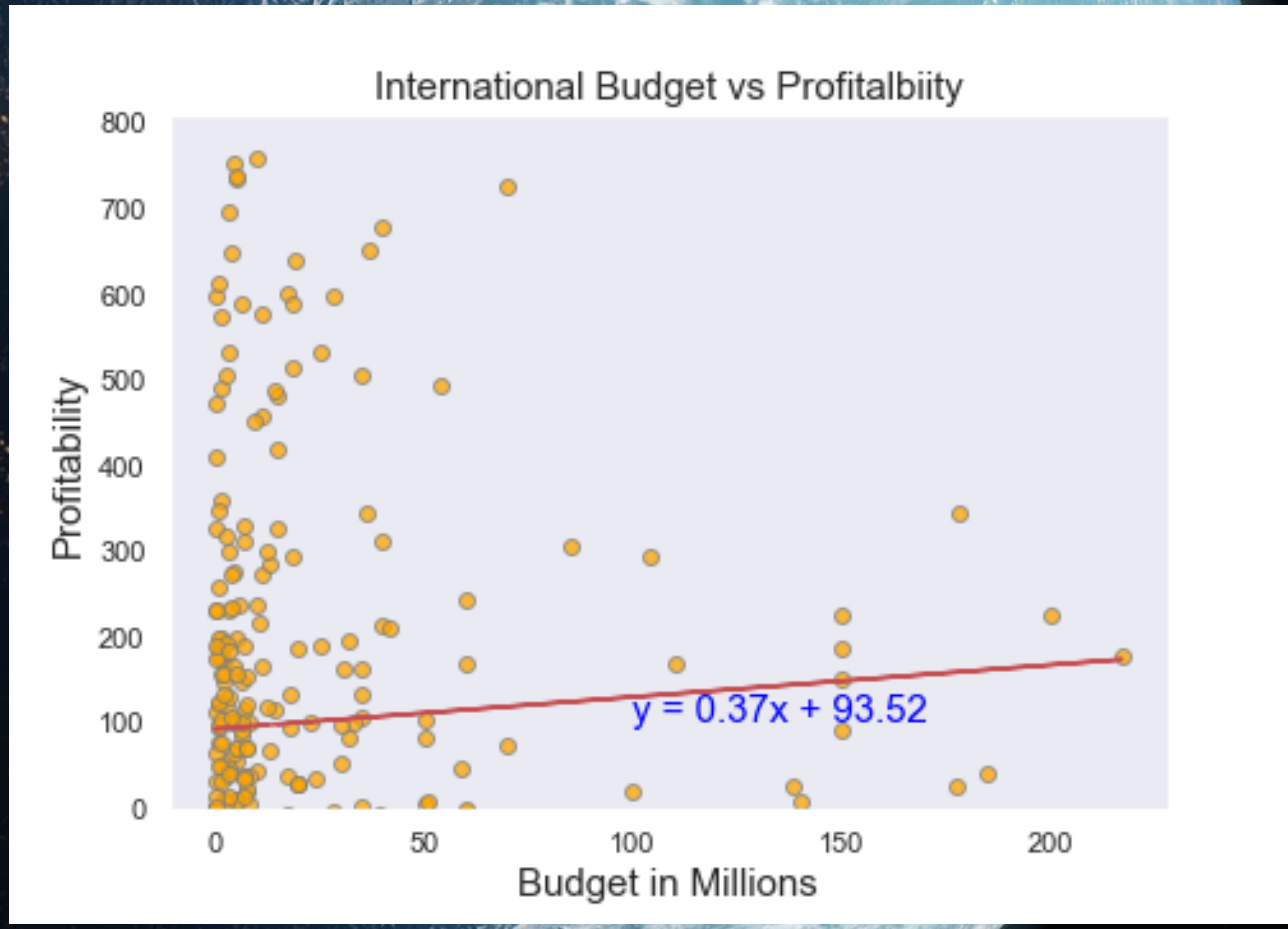
Action
R- squared: 0.13


Is there a difference between the US and Internationally?
Act 7 - Arpi
PRODUCTION
DIRECTOR
CAMERA
SCENE
TAKE

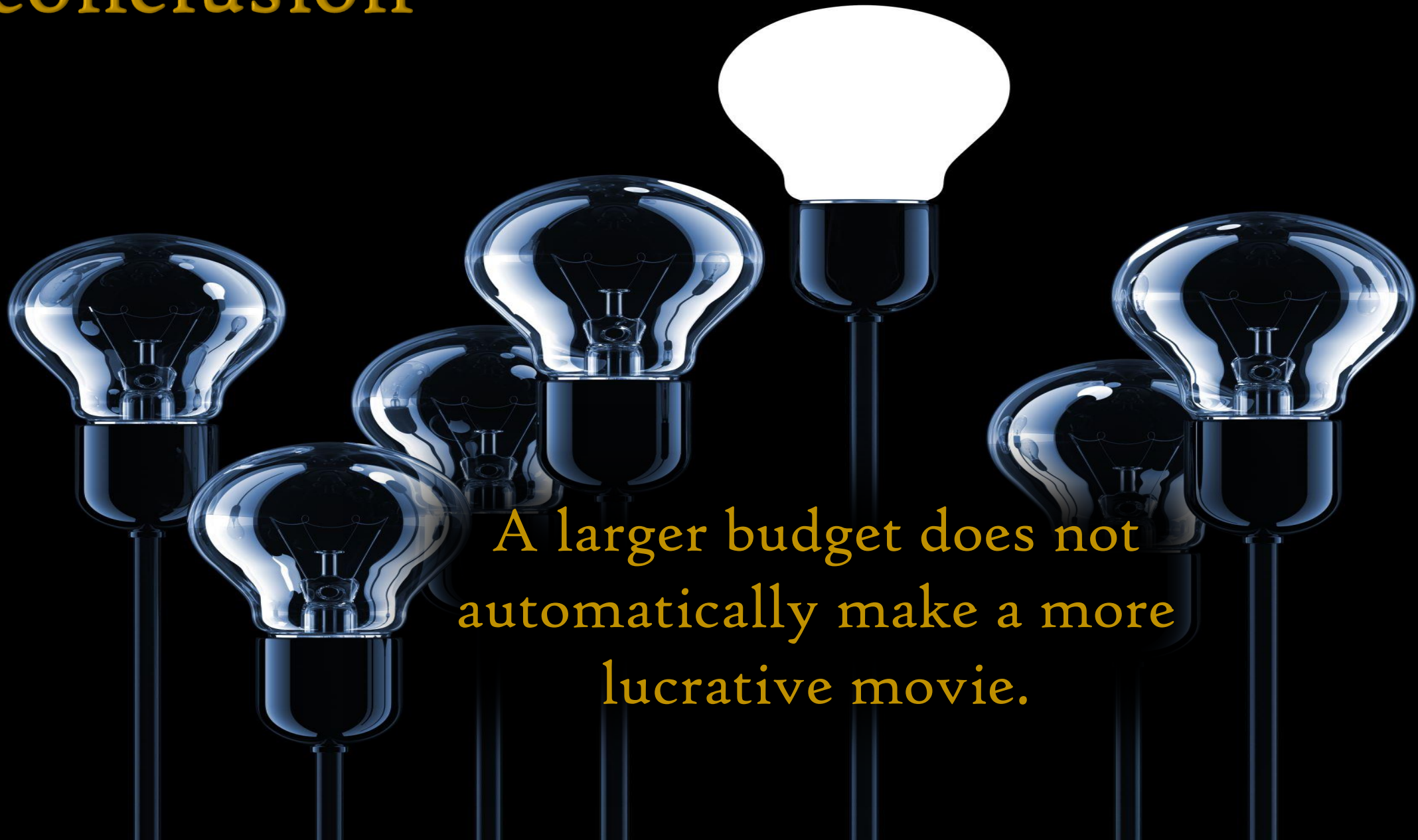
Us
R- squared: 0.08



International
R- squared: 0.00



In conclusion



A larger budget does not automatically make a more lucrative movie.

Starring

Amandeep Brar
Arpine Bankikyan
Cassie Folkers
Lindsey Giron
Ricardo Negrete
Sriven Ankam

Featuring

THE MOVIE DB

IMDb

OMDb API
The Open Movie Database

kaggle