Data Insights to Microsoft Movies studio

~by Jackson Munene

Next

Problem Statement

Microsoft

has decided to create a new **movie studio**, but they don't know anything about movie industries.

They need valuable insights from specific data from these three movie companies:

- Box Office Mojo
- The Numbers
- The Movies DB

GOALS

As a data scientist, here are some of the important milestones before getting to the insights

Understanding the Datasets

Cleaning the datasets

Finding patterns and visual representations of data

1) Box Office Mojo

This is how the first five records of the csv data from Box Office Mojo looks like:

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010

There are only five features: Title, studio, domestic gross, foreign gross and year. This data defines the total amount of money a specific movie generated in the year of its release, both nationally and internationally.

.info()

This is how the high level information about the data from Box Office Mojo:

There are a total of 3,387 records in the dataset, and 5 features. 3 of the features are in string data type, 1(year) is an integer and 1(domestic gross) is a floating value. Notice the inconsistency with the dataset datatype:

- Foreign gross is keyed in like a string. We need to change it to a floating value to perform calculations with it.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3387 entries, 0 to 3386
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   title           3387 non-null   object
 1   studio          3382 non-null   object
 2   domestic_gross  3359 non-null   float64
 3   foreign_gross   2037 non-null   object
 4   year            3387 non-null   int64
dtypes: float64(1), int64(1), object(3)
memory usage: 132.4+ KB
```

```
3   foreign_gross    2037 non-null    float64
```

Statistical Measures

Here is min, max, mean, median, standard deviation, count and quantiles of both **domestic** and **foreign gross**

	foreign_gross	domestic_gross
count	2037.0	3359.0
mean	74872810.2	28745845.1
std	137410600.8	66982498.2
min	600.0	100.0
25%	3700000.0	120000.0
50%	18700000.0	1400000.0
75%	74900000.0	27900000.0
max	960500000.0	936700000.0

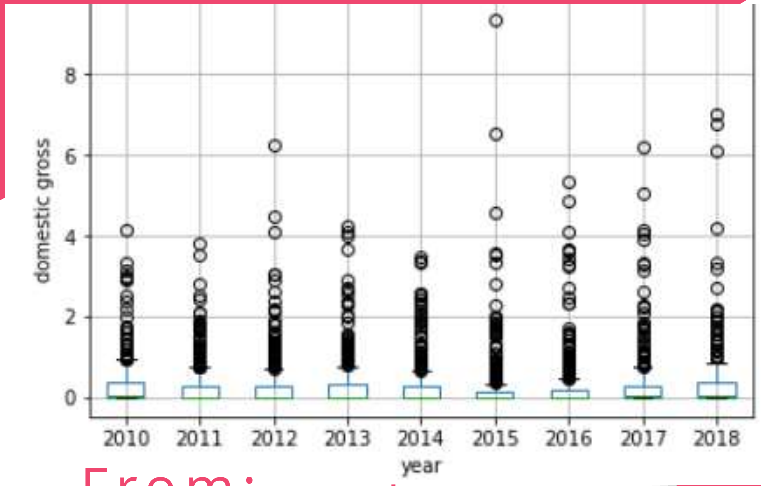
Notice the outliers, from min to the 25th percentile.

We change that by removing the outliers through getting data between 30% and 70% to look like as follows.

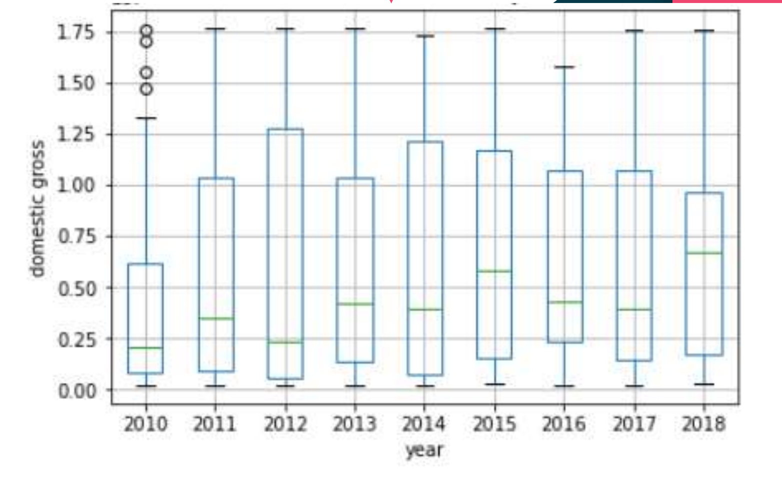
min	5300000.0	195000.0
25%	8500000.0	1100000.0

Domestic Gross

From:



To:

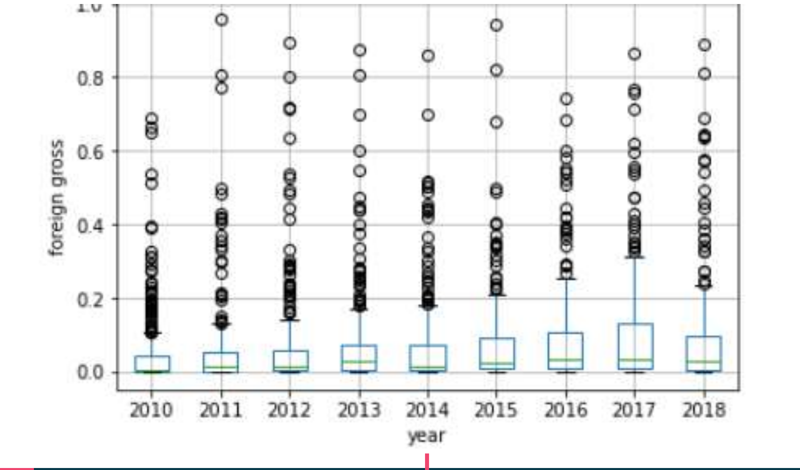


Another way to visualize these outliers is through:

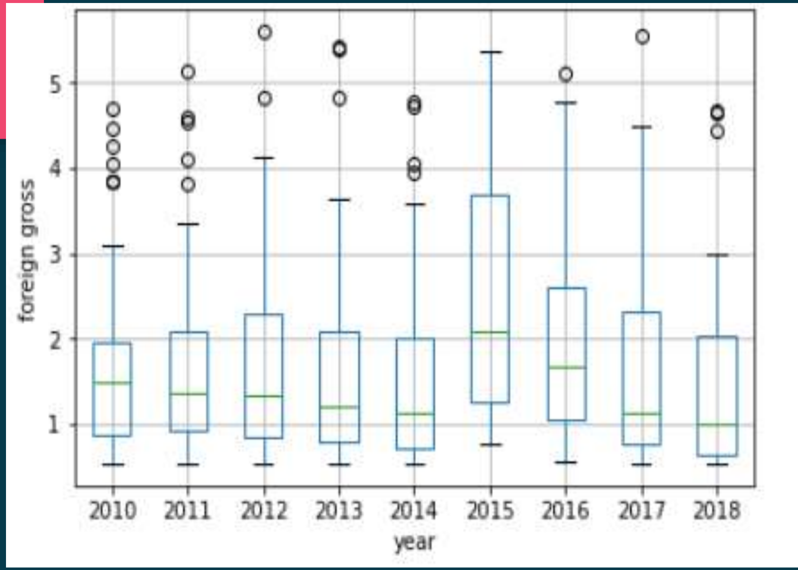
Box Plots

Foreign Gross

From:



To:

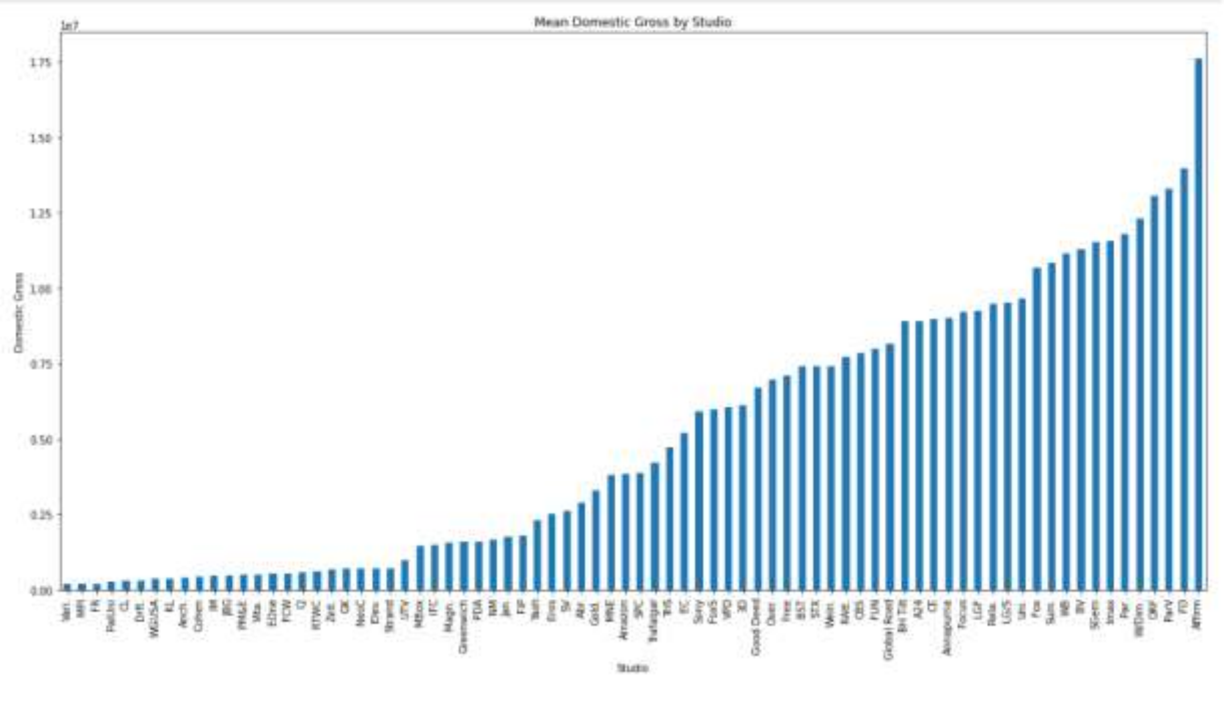


Which studio has the highest domestic gross?

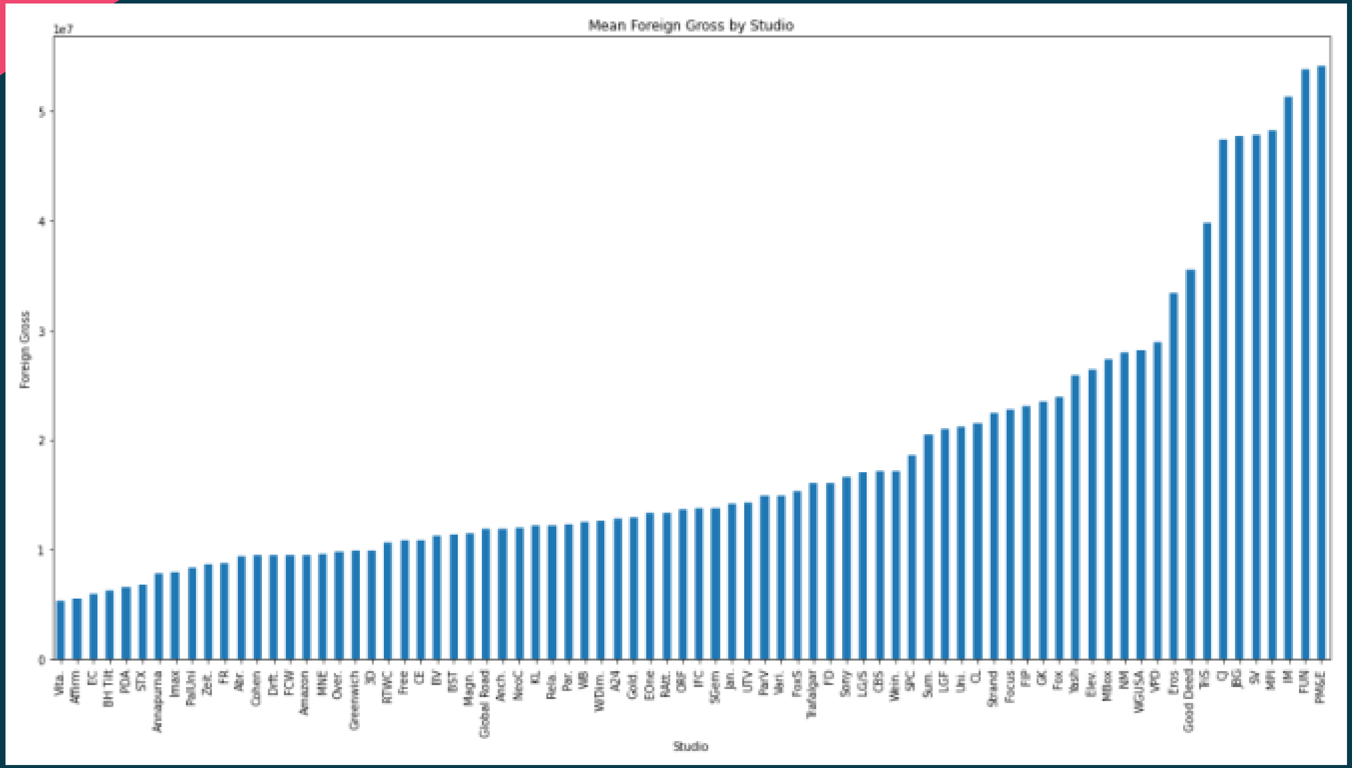
A bar graph representation for categorical feature(Studio) will be needed for this.

- 1st we'll group the **domestic gross** column by **studio.**
- Get the mean of the grouped data
- Sort the values in ascending order
- Then plot the bar graph.

We can conclude that after we removed the outliers, **Vari.** and **MPI** studios have the least while Affirm has the highest **domestic Gross**.



Which studio has the highest foreign gross?



The same will be done, but we only use Foreign Gross column in the data

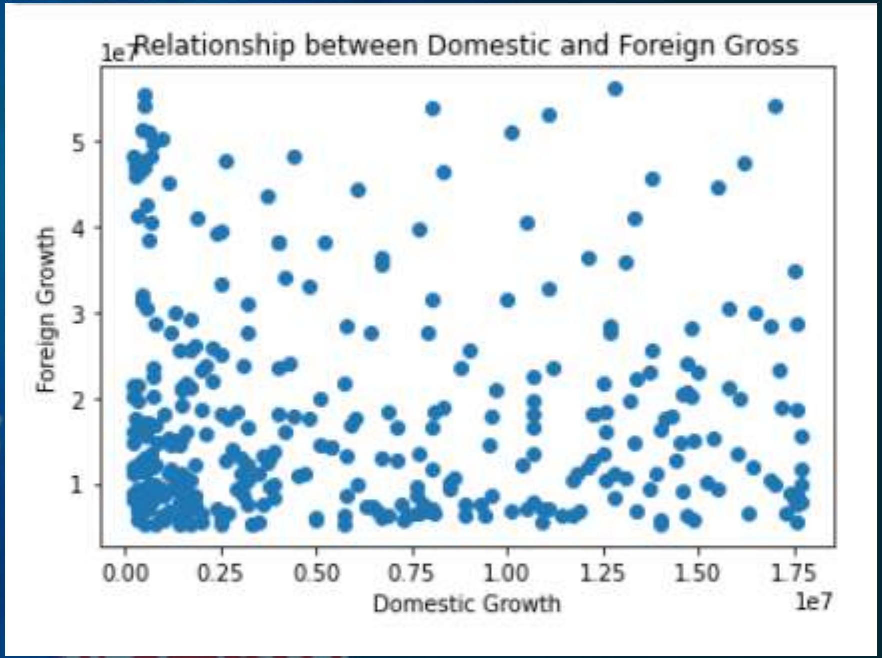
- 1st we'll group the **foreign gross** column by **studio.**
- Get the mean of the grouped data
- Sort the values in ascending order
- Then plot the bar graph.

Here, the opposite is true from the observations such that **Affirm** generates the least while **MPI** is one of the highest **Foreign gross** generator.

Gross Relationship

As much as the studios which generate the most domestic gross perform so low in foreign gross, there is no concrete connection between **domestic** and **foreign gross**.

Their correlation is: -0.01509821

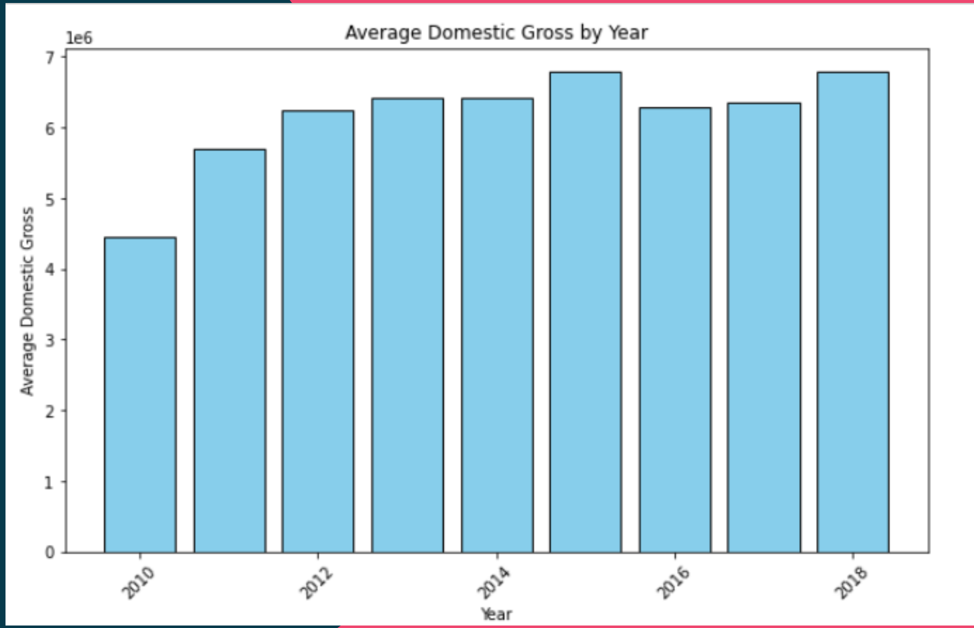


Domestic Gross growth

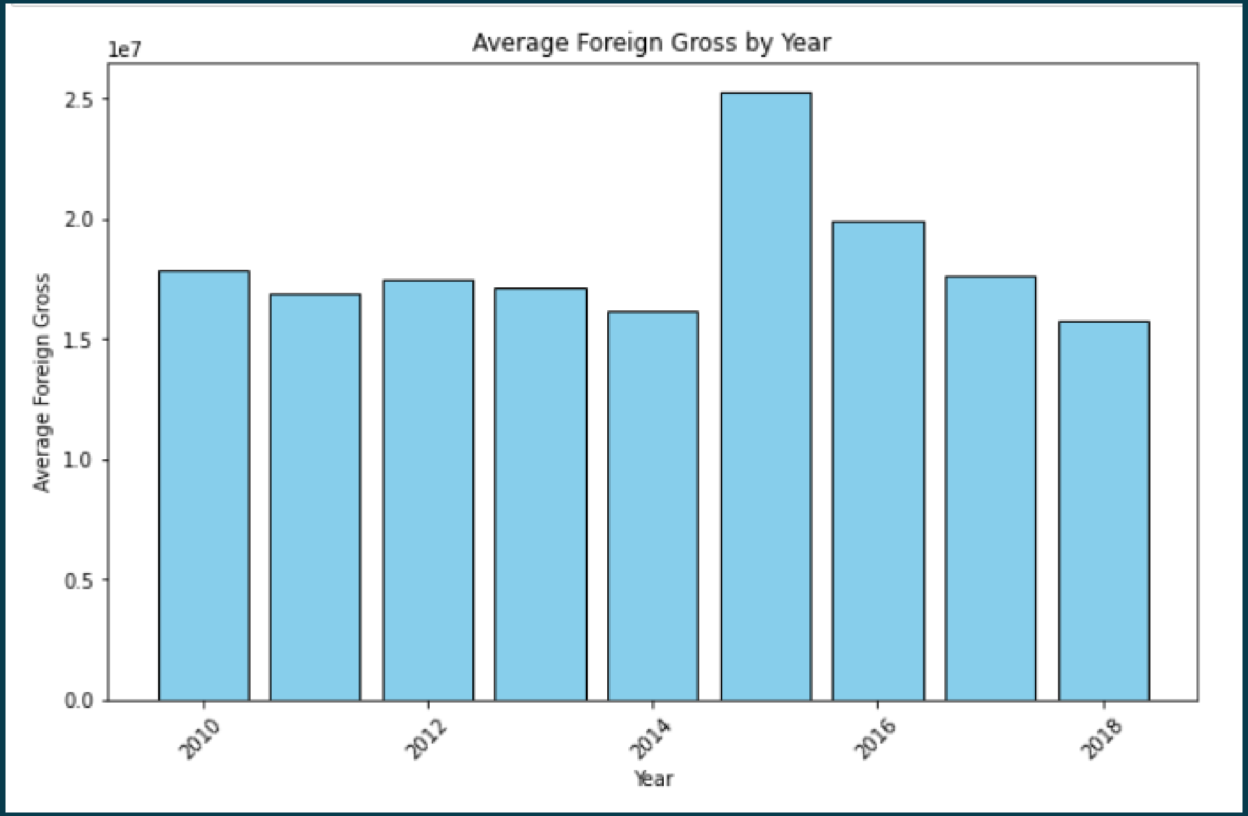
A bar graph representation for the average Domestic gross per year will be needed for this.

- 1st we'll group the **domestic gross** column by **year.**
- Get the mean of the grouped data
- Then plot the bar graph.

Notice a slight positive gradual growth over the years.



Foreign Gross growth



The same is not true with foreign gross.

- 1st we'll group the **Foreign gross** by **year.**
- Get the mean of the grouped data
- Then plot the bar graph.

In the foreign gross, there is not so much improvements, only that in 2015, the gross went up.

Recommendations

- Microsoft should consider partnering with **Affirm** studio to generate high Domestic gross and also **MPI** and **FUN** studios for high foreign gross.
- Microsoft should also find a way to automate data entry to minimize mistakes like getting outliers because of wrong data entry.
- The stakeholders should concentrate on both domestic and foreign gross, because they are both independent.

2) The Numbers

Here is a slight peek into The Numbers csv file:

There are a lot of things that have to be changed like:

- Removing **$** and comma(**,**) signs from **Production Budget**, **Domestic gross** and **worldwide gross.**
- We will also need to add few more features like **domestic profits**, **year** and **worldwide profits.**

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
3	4	May 1, 2015	Avengers: Age of Ultron	$330,600,000	$459,005,868	$1,403,013,963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	$317,000,000	$620,181,382	$1,316,721,747

release_date	movie	production_budget	domestic_gross	worldwide_gross	domestic_profits	worldwide_profits	year
2009-12-18	Avatar	425000000.0	760507625.0	2.776345e+09	335507625.0	2.351345e+09	2009
2011-05-20	Pirates of the Caribbean: On Stranger Tides	410600000.0	241063875.0	1.045664e+09	-169536125.0	6.350639e+08	2011
2019-06-07	Dark Phoenix	350000000.0	42762350.0	1.497624e+08	-307237650.0	-2.002376e+08	2019
2015-05-01	Avengers: Age of Ultron	330600000.0	459005868.0	1.403014e+09	128405868.0	1.072414e+09	2015
2017-12-15	Star Wars Ep. VIII: The Last Jedi	317000000.0	620181382.0	1.316722e+09	303181382.0	9.997217e+08	2017

.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5782 non-null   int64
 1   release_date       5782 non-null   object
 2   movie              5782 non-null   object
 3   production_budget  5782 non-null   object
 4   domestic_gross     5782 non-null   object
 5   worldwide_gross    5782 non-null   object
dtypes: int64(1), object(5)
memory usage: 271.2+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5782 entries, 0 to 5781
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5782 non-null   int64
 1   release_date       5782 non-null   datetime64[ns]
 2   movie              5782 non-null   object
 3   production_budget  5782 non-null   float64
 4   domestic_gross     5782 non-null   float64
 5   worldwide_gross    5782 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(1)
memory usage: 271.2+ KB
```

The data has 5,782 rows and 6 columns which are all objects except from the Id.

There are a lot of things that have to be changed like:

- Changing data types of **production budget**, **domestic gross**, and **worldwide gross** to floating values.
- **Release date** should also be in datetime datatype.

MOVIE NIGHT

Statistical Measures

Before cleaning the dataset, notice the outliers from 75% percentile to max in all of the features.

We have to clean that by dropping the data so it looks as follows:

	production_budget	domestic_gross	worldwide_gross	domestic_profits	worldwide_profits
count	5782.0	5782.0	5.782000e+03	5782.0	5.782000e+03
mean	31587757.1	41873326.9	9.148746e+07	10285569.8	5.989970e+07
std	41812076.8	68240597.4	1.747200e+08	49921366.5	1.460889e+08
min	1100.0	0.0	0.000000e+00	-307237650.0	-2.002376e+08
25%	5000000.0	1429534.5	4.125415e+06	-9132757.0	-2.189071e+06
50%	17000000.0	17225945.0	2.798445e+07	-348775.5	8.550286e+06
75%	40000000.0	52348661.5	9.764584e+07	17781444.0	6.096850e+07
max	425000000.0	936662225.0	2.776345e+09	630662225.0	2.351345e+09

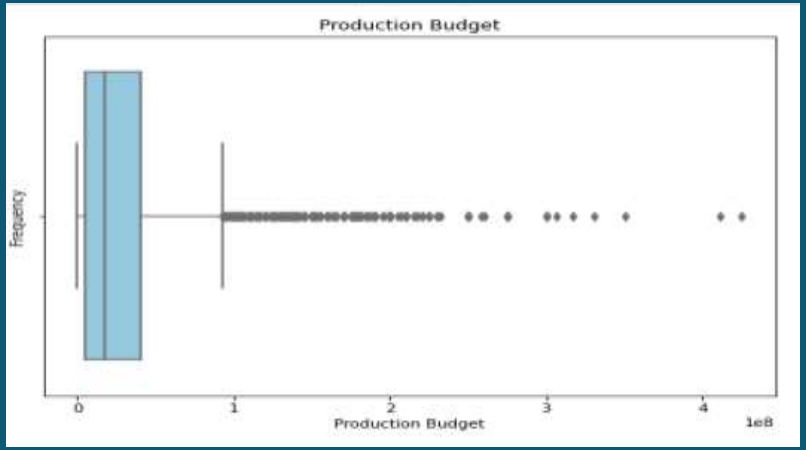
75%	25000000.0	27367976.5	39545462.2	9794648.8	22113796.0
max	35000000.0	42469946.0	76086711.0	33157856.0	62727492.0

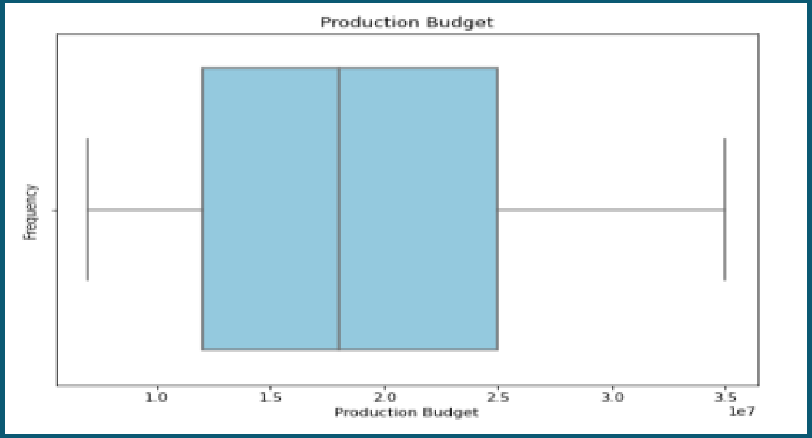
Below is another way to visualize the outliers and what will happen to the data when those outliers are removed.

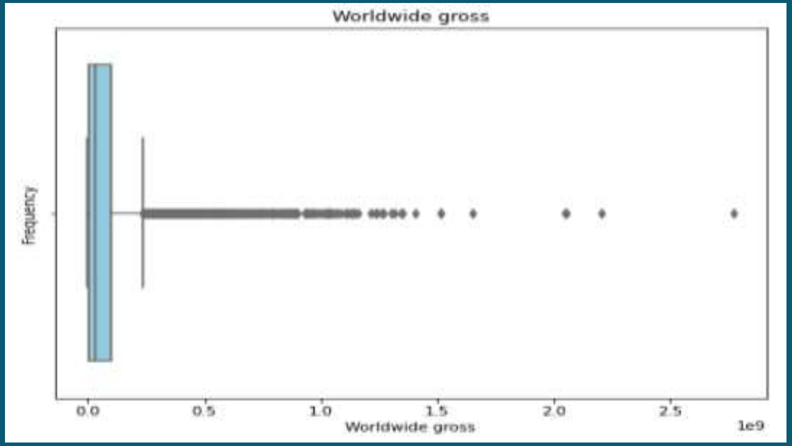
before

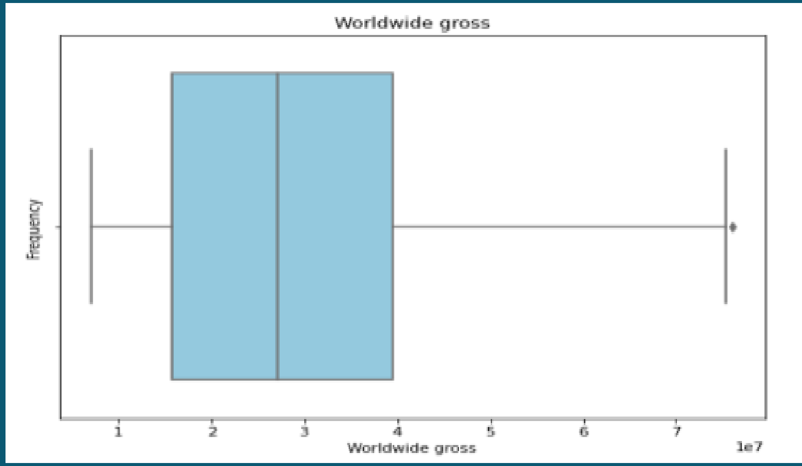
after

Box plots





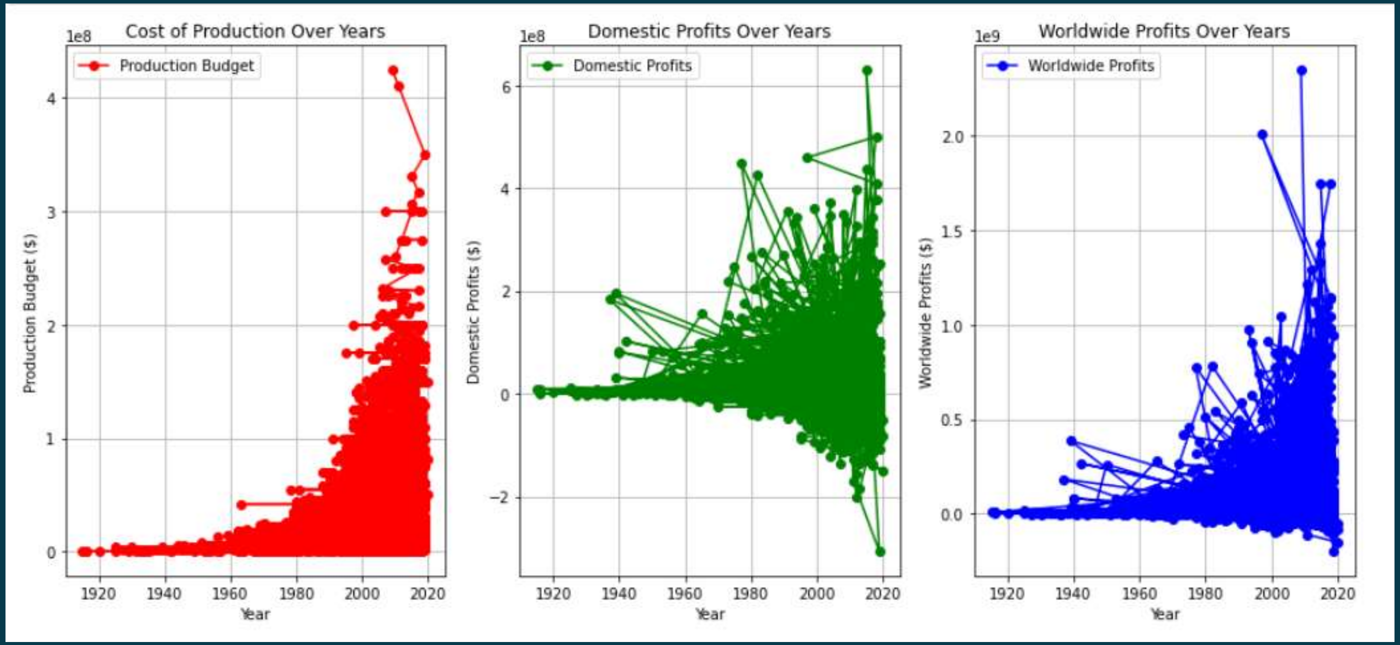




Does Production budget affect Profits?

The higher the **production budget**, the higher the **worldwide profits**, but the same is not true with **domestic profits**.

The correlation between **production budget** and **worldwide profits** is **0.6087521**.
While the correlation between the **product budget** and **domestic budget** is **0.099742.**



3) The Movie DB

This is how the first five records of the csv data from The Movie DB looks like:

They have a lot more features than the other datasets, and they also have **average vote counts** of each movie and their **popularity**.

	Unnamed: 0	genre_ids	id	original_language	original_title	popularity	release_date	title	vote_average	vote_count
0	0	[12, 14, 10751]	12444	en	Harry Potter and the Deathly Hallows: Part 1	33.533	2010-11-19	Harry Potter and the Deathly Hallows: Part 1	7.7	10788
1	1	[14, 12, 16, 10751]	10191	en	How to Train Your Dragon	28.734	2010-03-26	How to Train Your Dragon	7.7	7610
2	2	[12, 28, 878]	10138	en	Iron Man 2	28.515	2010-05-07	Iron Man 2	6.8	12368
3	3	[16, 35, 10751]	862	en	Toy Story	28.005	1995-11-22	Toy Story	7.9	10174
4	4	[28, 878, 12]	27205	en	Inception	27.920	2010-07-16	Inception	8.3	22186

.info()

In this high level overview of the data, we find that:

- We have 10 columns
- 26,517 records
- A mix of floating values and strings.
- No null values

The just need to change **release date** datatype to datetime.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26517 entries, 0 to 26516
Data columns (total 10 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         26517 non-null  int64
 1   genre_ids          26517 non-null  object
 2   id                 26517 non-null  int64
 3   original_language  26517 non-null  object
 4   original_title     26517 non-null  object
 5   popularity         26517 non-null  float64
 6   release_date       26517 non-null  object
 7   title              26517 non-null  object
 8   vote_average       26517 non-null  float64
 9   vote_count         26517 non-null  int64
dtypes: float64(2), int64(3), object(5)
memory usage: 2.0+ MB
```

```
 6   release_date       26517 non-null  datetime64[ns]
```

Statistical Measures

Notice the outliers, from min to the 25th percentile, and 75th to max.

We change that by removing the outliers through getting data between:

- 0.05 and 0.65 quantiles for **popularity**
- 0.3 and 0.8 quantiles for **Average votes**
- 0.005 and 0.65 quantiles for **vote count.**

	popularity	vote_average	vote_count
count	26517.000000	26517.000000	26517.000000
mean	3.130912	5.991281	194.224837
std	4.355229	1.852946	960.961095
min	0.600000	0.000000	1.000000
25%	0.600000	5.000000	2.000000
50%	1.374000	6.000000	5.000000
75%	3.694000	7.000000	28.000000
max	80.773000	10.000000	22186.000000

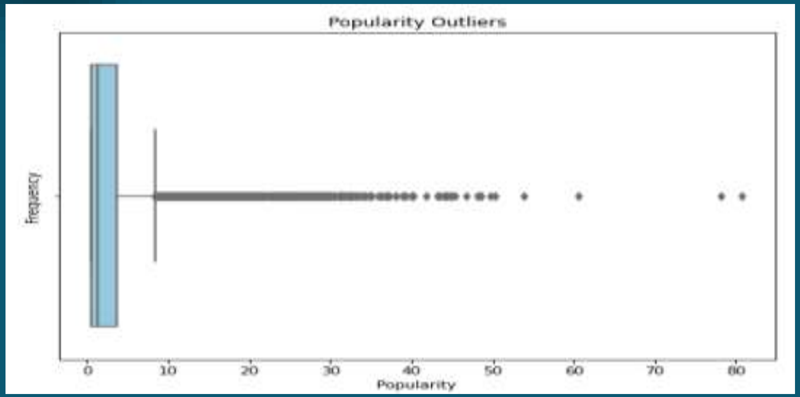
min	0.600000	5.200000	1.000000
25%	0.600000	6.000000	1.000000
50%	0.711000	6.200000	3.000000
75%	1.257000	7.000000	5.000000
max	2.235000	7.300000	13.000000

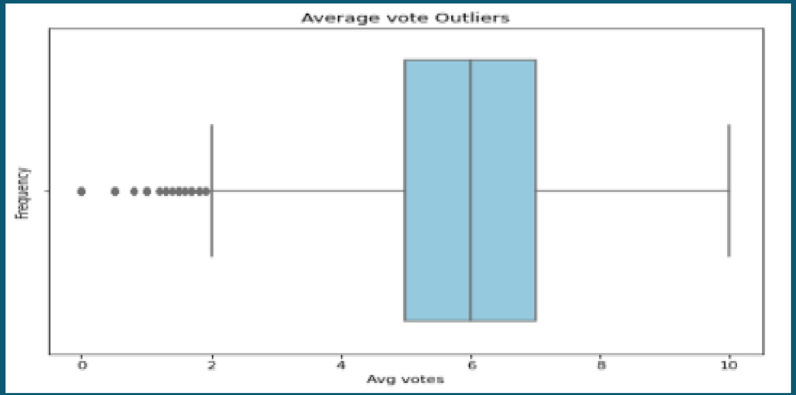
before

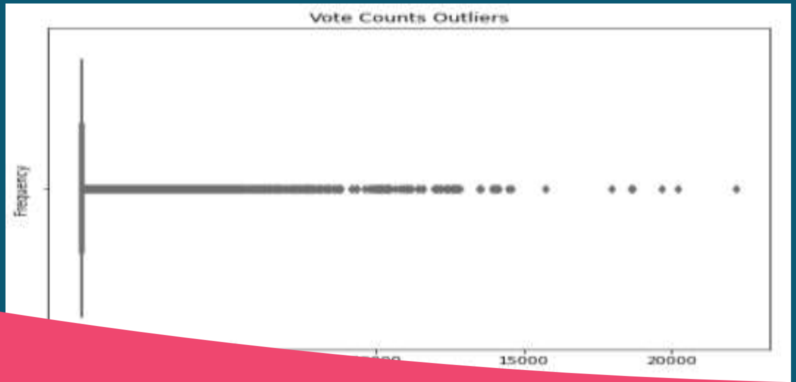
Let us visualize the outliers with a

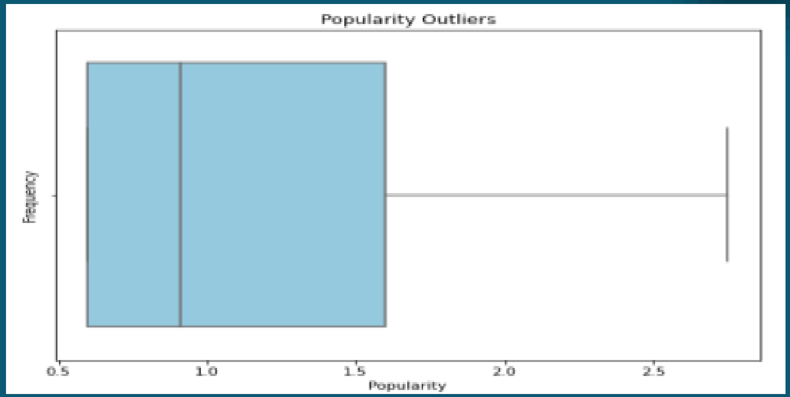
Box plots

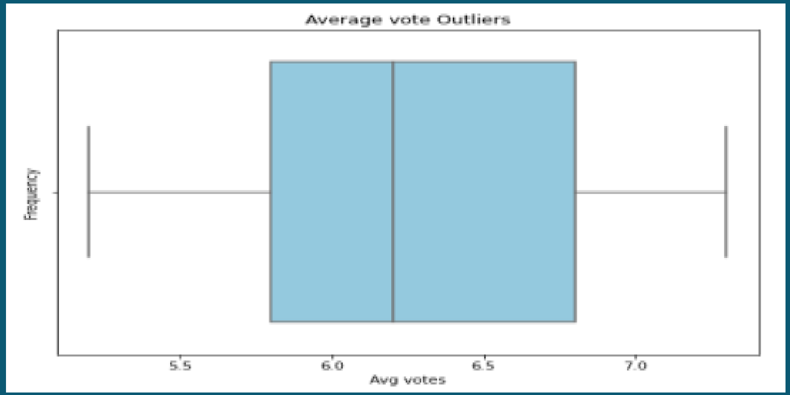
after

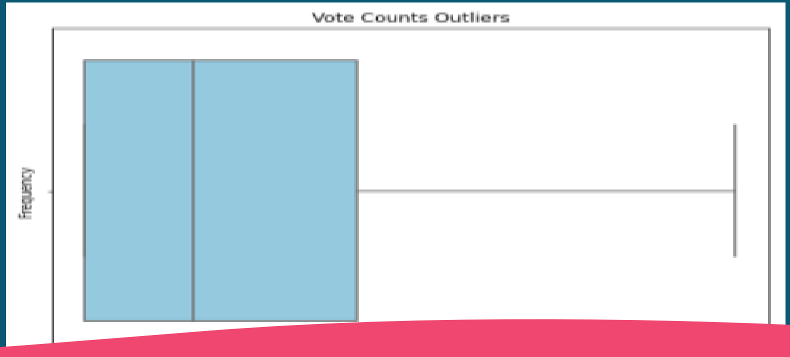










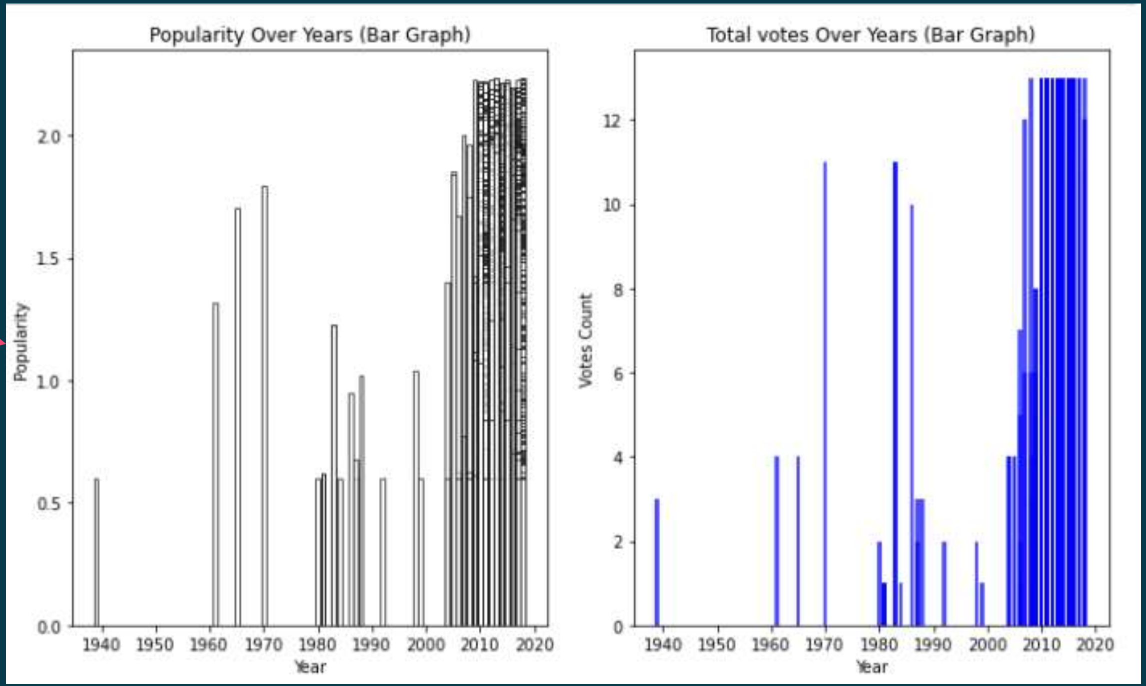


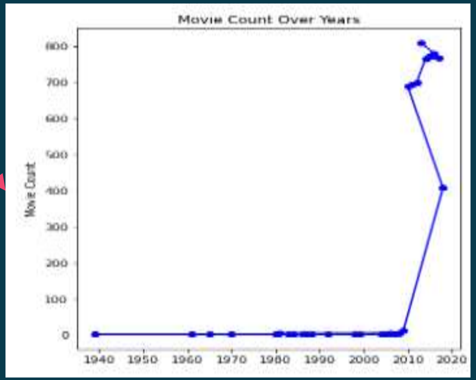
Does Vote count affect popularity?

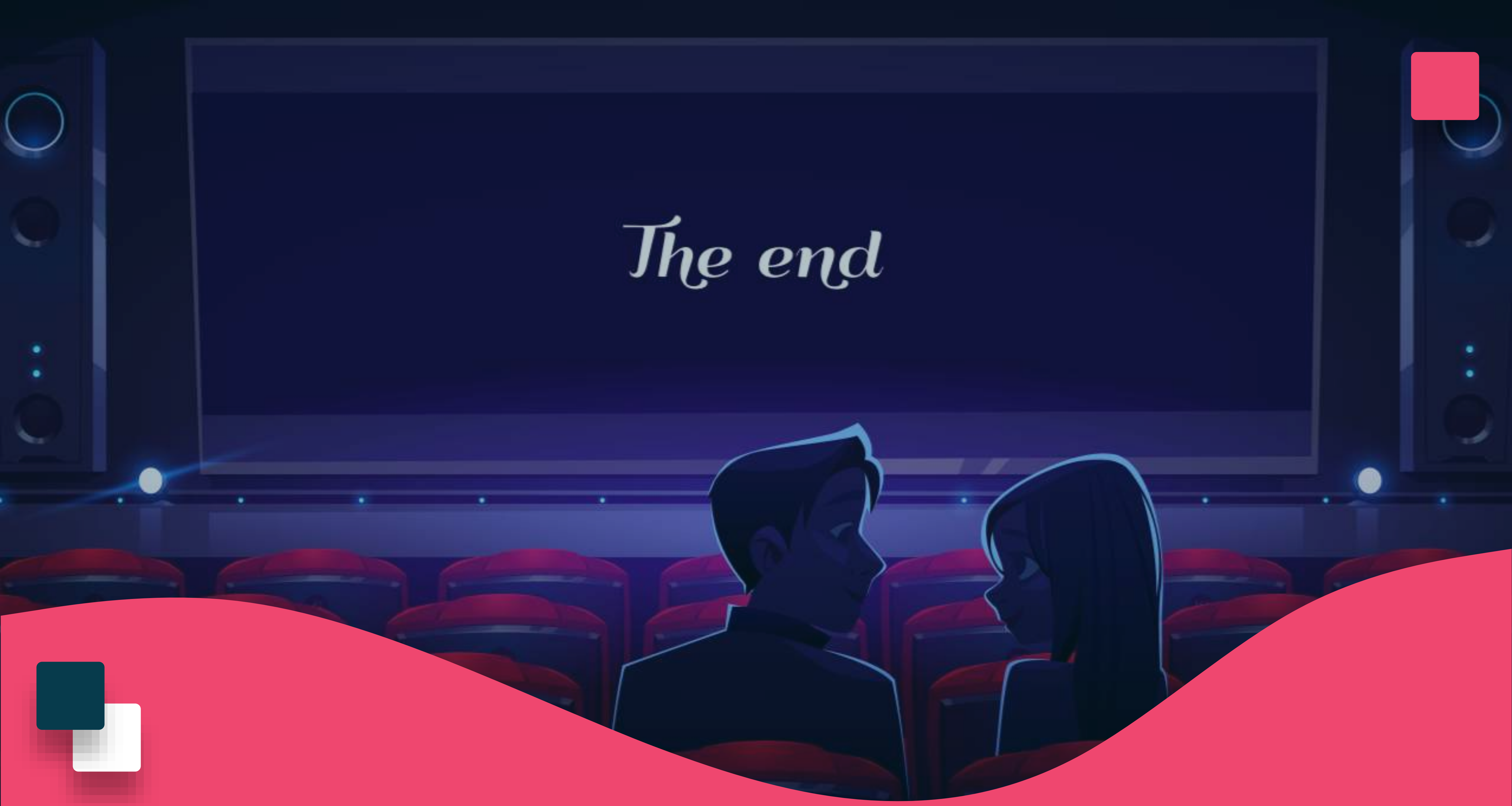
Vote count is positively correlated to popularity, but it is not a strong correlation.

Correlation = 0.549394219698477.

The movie DB released less movies from 1940 to 2005, but started releasing a lot of movies from 2005 onwards, and gained a lot of votes and popularity there after.






The end