فاز اول پروژه

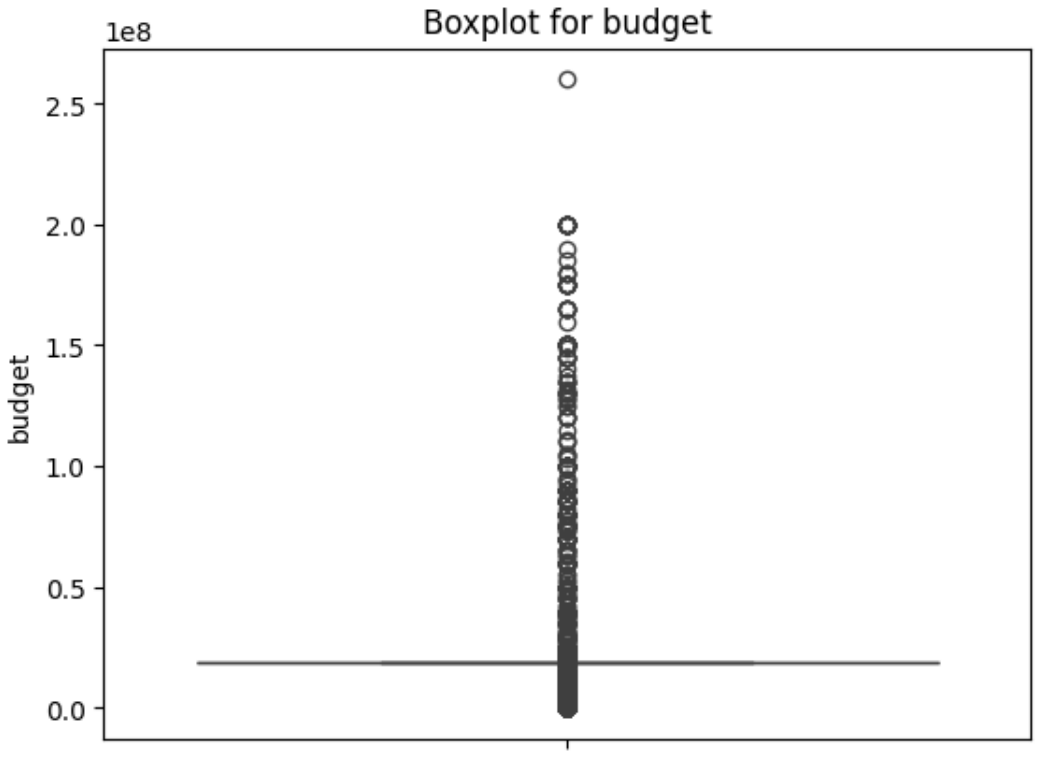
درس مبانی داده کاوی

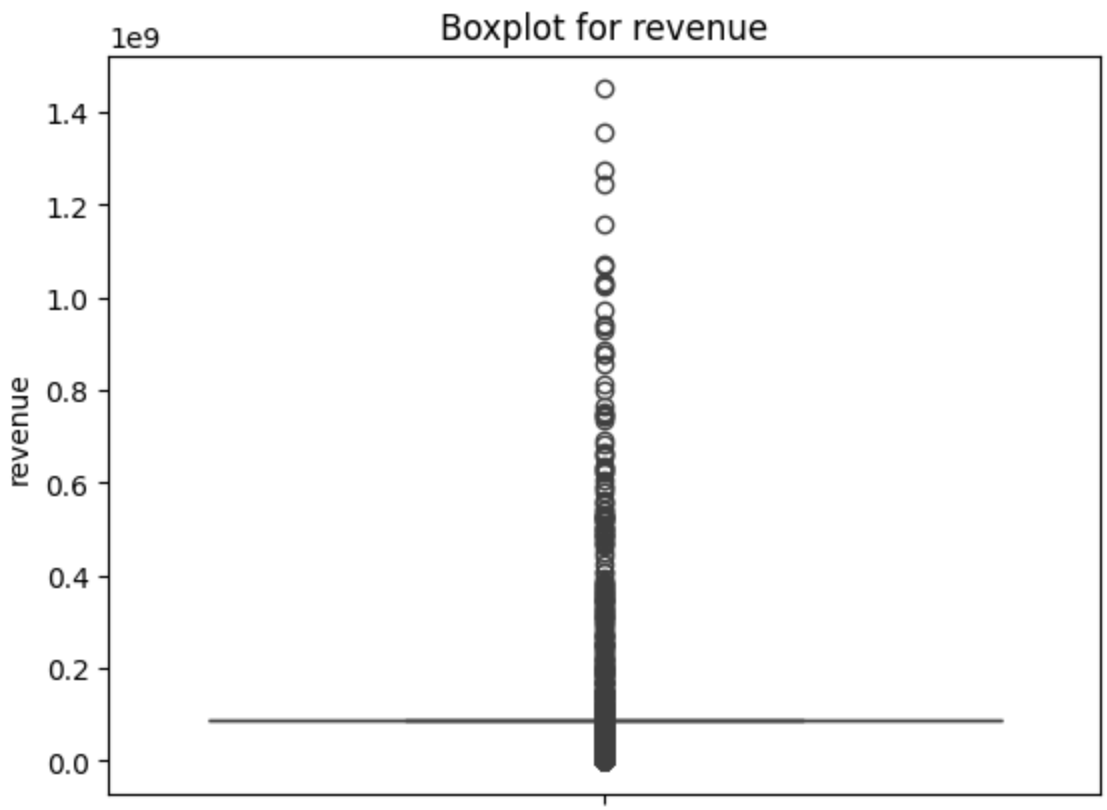
بخش اول :

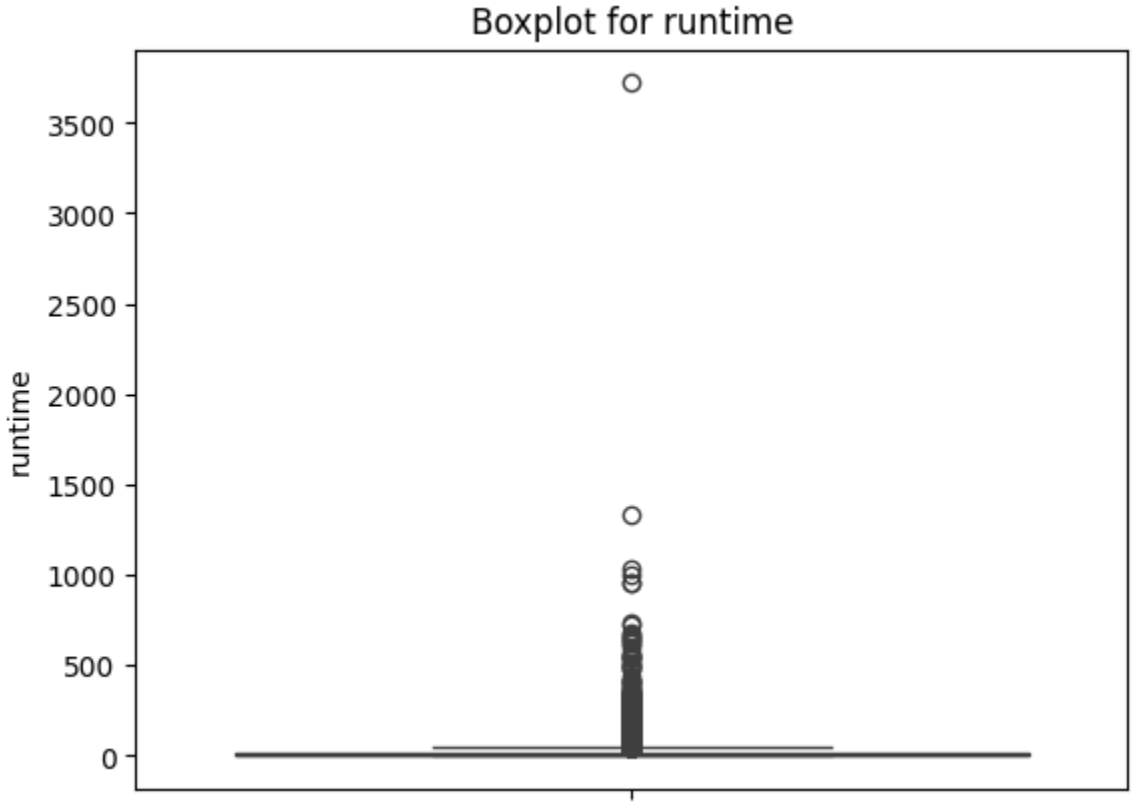
مقادیر اشتباه : ابتدا مقادیر صفر در دیتاست را با میانگین مقادیر دیگر جایگزین میکنیم.

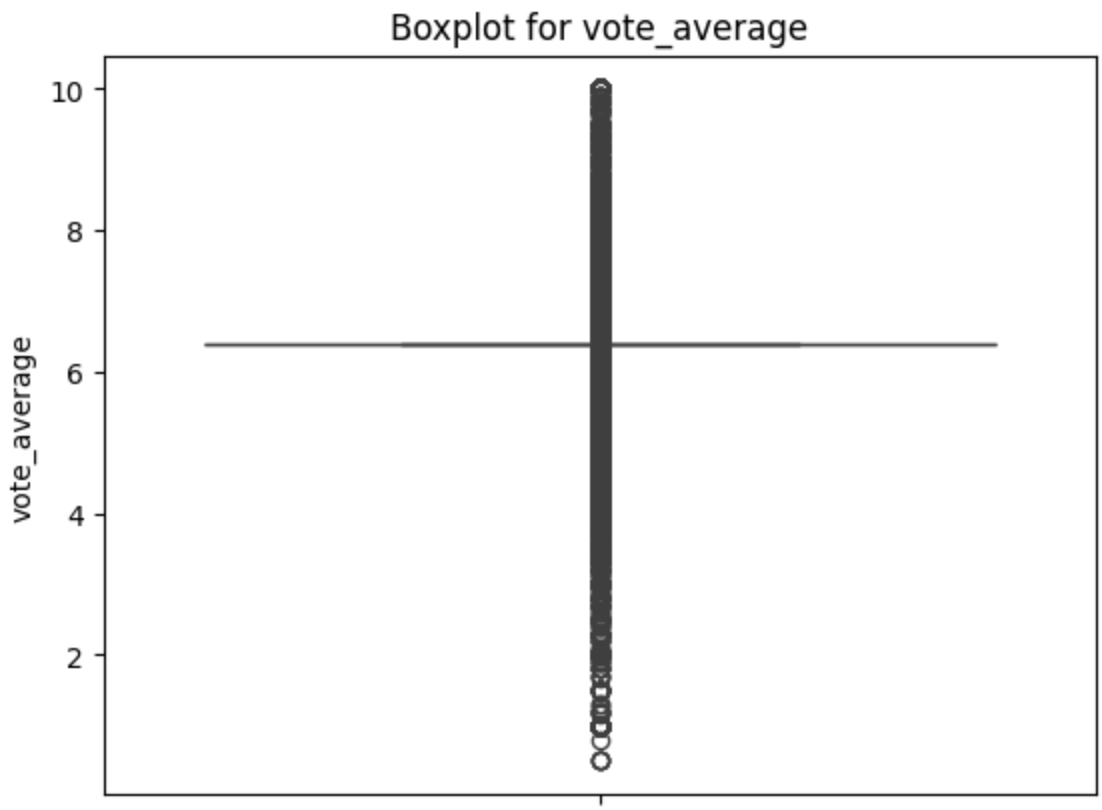
	Name	Type	Range	Min	Max	Mean	Mode	Median
0	budget	float64	1.0 :260000000.0	1.0	2.600000e+08	1.866362e+07	0 1.866362e+07 Name: budget, dtype: float64	1.866362e+07
1	revenue	float64	1.0 :1450026933.0	1.0	1.450027e+09	8.799786e+07	0 8.799786e+07 Name: revenue, dtype: float64	8.799786e+07
2	runtime	int64	0 :3720	0.0	3.720000e+03	2.090141e+01	0 0 Name: runtime, dtype: int64	7.000000e+00
3	vote_average	float64	0.5 :10.0	0.5	1.000000e+01	6.376701e+00	0 6.376701 Name: vote_average, dtype: float64	6.376701e+00
4	vote_count	float64	1.0 :19463.0	1.0	1.946300e+04	9.913720e+01	0 99.137201 Name: vote_count, dtype: float64	9.913720e+01

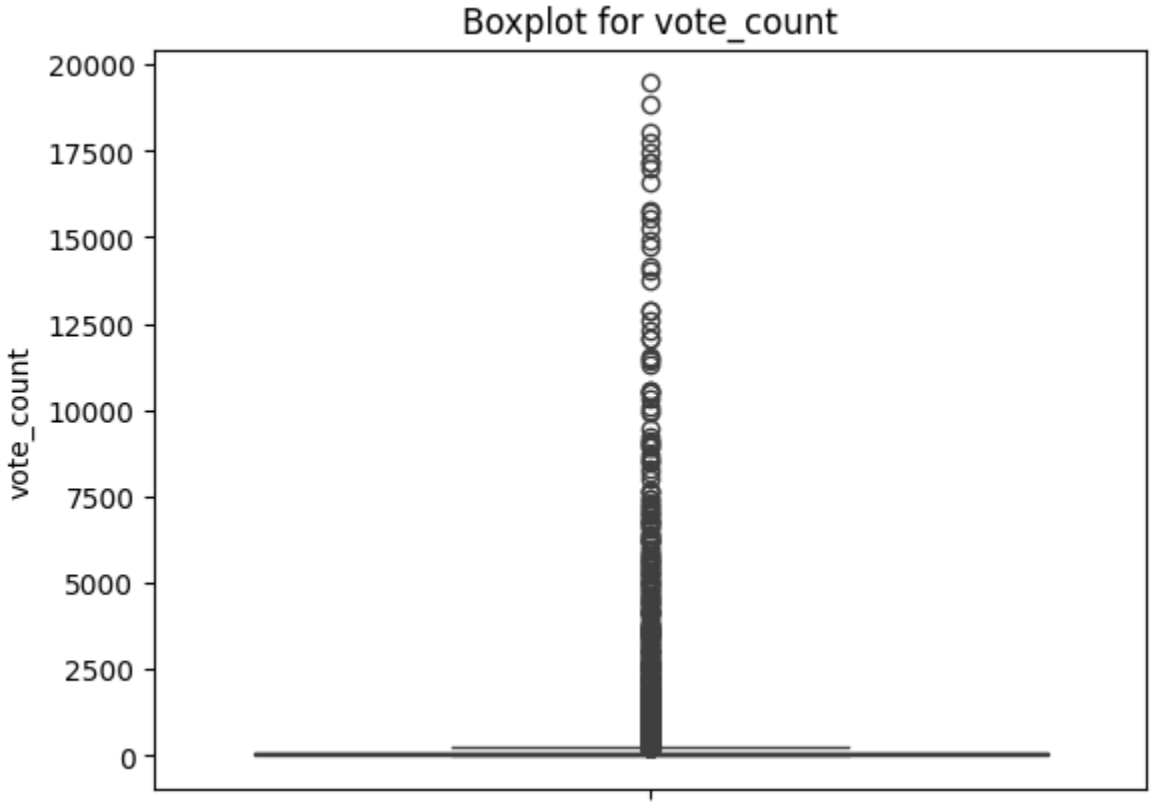
مقادیر دورافتاده : نمودار باکس پلات را برای مقادیر عددی رسم میکنیم.




Boxplot for revenue
1e9
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
revenue


Boxplot for runtime
3500
3000
2500
2000
1500
1000
500
0
runtime


Boxplot for vote_average
vote_average
10
8
6
4
2


Boxplot for vote_count
vote_count
20000
17500
15000
12500
10000
7500
5000
2500
0

بسیاری از داده ها outlier هستند.

بخش دوم :

تحلیل و بررسی کیفیت داده‌ها

مرحله ۱: بررسی مقادیر گم‌شده

در این مرحله، مقادیر گم‌شده در ستون‌های بودجه، درآمد و تاریخ انتشار شناسایی می‌شوند. برای این کار، ابتدا تعداد مقادیر گم‌شده در هر ستون محاسبه شده و سپس درصد مقادیر گم‌شده نسبت به کل داده‌ها بدست می‌آید.

کد زیر تعداد و درصد مقادیر گم‌شده را محاسبه و نمایش می‌دهد:

```
missing_values = df[['revenue', 'budget', 'release_date']].isnull().sum()
missing_percentage = (missing_values / len(df)) * 100
print("Missing values and their percentages:\n", missing_percentage)
```

مرحله ۲: شناسایی داده‌های پرت

داده‌های پرت (outliers) در ستون‌های درآمد و بودجه شناسایی می‌شوند .برای این کار، از شاخص IQR (Interquartile Range)استفاده می‌شود تا مقادیر بسیار دور از حد معمول را تشخیص دهیم.

کد زیر برای شناسایی داده‌های پرت در هر ستون نوشته شده است:

```
def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    outliers = df[(df[column] < (Q1 - 1.5 * IQR)) | (df[column] > (Q3 + 1.5 * IQR))]
    return outliers
```

مرحله ۳: بررسی ناسازگاری‌ها و خطاها در تاریخ انتشار

تاریخ‌های انتشار بررسی می‌شوند تا اطمینان حاصل شود که هیچ تاریخ انتشاری در زمان حال یا آینده وجود ندارد. به عنوان مثال، اگر تاریخی در ستون `release_date` بزرگتر از تاریخ فعلی باشد، آن را نامعتبر در نظر می‌گیریم.

کد زیر برای شناسایی تاریخ‌های نامعتبر و بودجه یا درآمد منفی نوشته شده است:

```
today = datetime.today()

invalid_dates = df[pd.to_datetime(df['release_date'], errors='coerce') > today]
print("Invalid dates in date_release:\n", invalid_dates)

invalid_budget = df[df['budget'] < 0]
invalid_revenue = df[df['revenue'] < 0]
print(f"Invalid budget values:\n")
display(invalid_budget)
print(f"Invalid revenue values:\n")
display(invalid_revenue)
```

ناسازگاری ها:

این قسمت از کد به دنبال ردیف‌هایی می‌گردد که در آنها مقدار بودجه (budget) موجود نیست ولی مقدار درآمد (revenue) موجود است. وجود درآمد بدون مقدار بودجه می‌تواند نشان‌دهنده ناسازگاری یا مشکل در داده‌ها باشد. این ناسازگاری‌ها نمایش داده می‌شوند تا بتوان آن‌ها را بررسی و اصلاح کرد:

```
inconsistent_budget_revenue = df[(df['budget'].isna()) & (~df['revenue'].isna())]
display(f"Inconsistent rows between budget and revenue:\n{inconsistent_budget_revenue}")

inconsistent_release_date_imdb_id = df[(df['release_date'].isna()) & (~df['imdb_id'].isna())]
display(f"Inconsistent rows between release_date and revenue:\n{inconsistent_release_date_imdb_id}")
```

توزیع فیلم‌ها در دهه‌های مختلف :

در این بخش، فیلم‌ها بر اساس دهه انتشار دسته‌بندی می‌شوند. دهه‌ها به عنوان بازه‌های ۱۰ ساله تعریف می‌شوند و تعداد فیلم‌ها در هر دهه محاسبه می‌شود. این توزیع به نمایش تعداد فیلم‌ها در هر دوره زمانی کمک می‌کند

```
decade_bins = pd.cut(df['release_year'], bins=np.arange(1900, 2030, 10))
decade_distribution = df.groupby(decade_bins).size()
display(f"Number of movies in each decade:\n{decade_distribution}")
```

محاسبه نسبت مقادیر معتبر در ستون‌های بودجه، درآمد و تاریخ انتشار:

این قسمت از کد درصد مقادیر معتبر (ناموجود نبودن) را برای ستون‌های بودجه، درآمد و تاریخ انتشار محاسبه و نمایش می‌دهد. برای مثال، نسبت معتبر بودجه نشان می‌دهد چند درصد از فیلم‌ها مقدار بودجه دارند. این مقادیر برای ارزیابی کیفیت کلی داده‌ها مهم هستند

```
valid_budget_ratio = df['budget'].notna().mean() * 100
valid_revenue_ratio = df['revenue'].notna().mean() * 100
valid_date_release_ratio = df['release_date'].notna().mean() * 100

print(f"Valid budget ratio: {valid_budget_ratio:.2f}%")
print(f"Valid revenue ratio: {valid_revenue_ratio:.2f}%")
print(f"Valid date_release ratio: {valid_date_release_ratio:.2f}%")
```

پیشنهادات

پیشنهادات برای بهبود کیفیت داده‌ها:

- پر کردن مقادیر گم‌شده با میانگین یا میانه‌ی داده‌ها برای ستون‌های budget وrevenue
- استفاده از اطلاعات تاریخی معتبر برای پر کردن مقادیر گم‌شده درdate_release

حذف یا تصحیح مقادیر پرت (outliers) مانند مقادیر منفی در ستون‌های budget و revenue

برخی از خروجی های این بخش:

```
Missing values and their percentages:
 revenue         0.000000
budget           0.000000
release_date     4.113967
dtype: float64
Outliers in revenue:
```

```
Number of movies in each decade:
release_year
(1900, 1910]       68
(1910, 1920]      314
(1920, 1930]     1098
(1930, 1940]     2155
(1940, 1950]     1853
(1950, 1960]     1928
(1960, 1970]     2226
(1970, 1980]     2709
(1980, 1990]     3696
(1990, 2000]     4307
(2000, 2010]     7329
(2010, 2020]    15092
dtype: int64
Valid budget ratio: 100.00%
Valid revenue ratio: 100.00%
Valid date_release ratio: 95.89%
```

مشکلات مربوط به داده‌ها در جدول می‌توانند شامل مسائل مختلفی باشند که در زمینه‌های زیر خلاصه می‌شوند:

۱. Single Instance (تک نمونه‌ای بودن داده)

– مشکل: داده‌ها تنها شامل یک نمونه از هر نوع (یا کلاس) هستند. این مسئله می‌تواند منجر به مشکل در تحلیل داده‌ها و ایجاد مدل‌های یادگیری ماشین شود، زیرا داده‌های کافی برای شناسایی الگوها و روابط وجود ندارد.

– توضیح: در یادگیری ماشین و تحلیل داده‌ها، دسترسی به داده‌های متنوع و نمونه‌های متعدد از هر نوع ضروری است تا مدل بتواند به درستی تعمیم داده‌ها را یاد بگیرد. تک نمونه‌ای بودن باعث کاهش قدرت تعمیم‌دهی مدل و افزایش خطای آن در پیش‌بینی می‌شود.

۲. Single Schema (یک‌دستی یا یکسان بودن ساختار داده)

– مشکل: ساختار و فرمت داده‌ها به صورت یکسان است و تفاوتی در میان داده‌های مختلف مشاهده نمی‌شود. این مسئله ممکن است باعث شود که داده‌ها محدودیت در تنوع داشته باشند و برخی از ویژگی‌های لازم را نداشته باشند.

– توضیح: وقتی همه داده‌ها از یک نوع ساختار پیروی می‌کنند، امکان وجود ناهماهنگی و تنوع در داده‌ها کاهش می‌یابد. این مسئله باعث می‌شود که داده‌های مورد نیاز برای یادگیری کامل مدل و پوشش شرایط مختلف موجود نباشند و مدل به داده‌های واقعی قابل تعمیم نباشد. به عنوان مثال، داده‌های مختلف در پایگاه‌های داده‌ای از ساختارهای مختلف استفاده می‌کنند و نیاز است که داده‌ها به شکل استاندارد یا در قالب‌های متنوع‌تری نگهداری شوند.

بخش سوم:

- **پر کردن مقادیر گم‌شده در ستون‌های مختلف با میانگین:**

در اینجا، مقادیر گم‌شده در ستون‌های `vote_average`، `runtime` و `popularity`با استفاده از میانگین (`mean`)آن ستون‌ها جایگزین می‌شوند. استفاده از میانگین برای پر کردن مقادیر گم‌شده عددی می‌تواند به همگن‌سازی داده‌ها کمک کند

```
imputer_mean = SimpleImputer(strategy='mean')
df_clean['vote_average'] = imputer_mean.fit_transform(df_clean[['vote_average']])

imputer_median = SimpleImputer(strategy='mean')
df_clean['runtime'] = imputer_median.fit_transform(df_clean[['runtime']])

imputer_mode = SimpleImputer(strategy='mean')
df_clean['popularity'] = imputer_mode.fit_transform(df_clean[['popularity']])
```

- **جایگزینی مقادیر گم‌شده در ستون‌های تاریخ انتشار و ژانر با مقدار پرتکرار:**

در این بخش، مقادیر گم‌شده در ستون‌های `release_date` و`genres`با مقدار پرتکرار (`mode`) جایگزین می‌شوند. این روش برای ستون‌هایی که داده‌های متنی یا تاریخ هستند مناسب است.

```
most_frequent_date = df['release_date'].mode()[0]

df['release_date'].fillna(most_frequent_date, inplace=True)

most_frequent_genre = df['genres'].mode()[0]


df['genres'].fillna(most_frequent_genre, inplace=True)
```

- **نرمال‌سازی ستون‌های عددی:**

```
scaler = MinMaxScaler()
df_clean[['vote_average', 'runtime', 'popularity']] = scaler.fit_transform(df_clean[['vote_average', 'runtime', 'popularity']])
```

- **جایگزینی داده‌های پرت با میانگین بدون در نظر گرفتن ۱۰٪ بالا و پایین:**

```
def replace_outlier(data):
    lower_bound = data.quantile(0.1)
    upper_bound = data.quantile(0.9)


    filtered_data = data[(data >= lower_bound) & (data <= upper_bound)]
    mean_without_extremes = filtered_data.mean()


    data.apply(lambda x: mean_without_extremes if x < lower_bound or x > upper_bound else x)

replace_outlier(df_clean['budget'])

replace_outlier(df_clean['revenue'])
```

- **ایجاد ستون های جدید:**

```
def convert_to_jalali(gregorian_date):
    gregorian_date_obj = pd.to_datetime(gregorian_date)

    jalali_date = jdatetime.date.fromgregorian(day=gregorian_date_obj.day,
                                               month=gregorian_date_obj.month,
                                               year=gregorian_date_obj.year)
    return jalali_date


df_clean['persian_date']=df['release_date'].apply(convert_to_jalali)
```

- **استخراج سال و فصل انتشار:**

```
df_clean['release_year'] = pd.to_datetime(df_clean['release_date']).dt.year
df_clean['release_season'] = pd.to_datetime(df_clean['release_date']).dt.month % 12 // 3 + 1
```

- **رمزگذاری ستون ژانر:**

```
#matni be adadi
df_clean['genre_encoded'] = LabelEncoder().fit_transform(df_clean['genres'])
```

این خط ژانرهای مختلف را به مقادیر عددی تبدیل می‌کند تا در تحلیل‌های عددی استفاده شوند.

- دسته‌بندی طول فیلم‌ها:

```
df_clean['runtime_category'] = pd.cut(df_clean['runtime'], bins=[0, 40,80, 120, 160,200], labels=['Very Short', 'Short Feature', 'Standard Feature', 'Long Feature','Epic'])
```

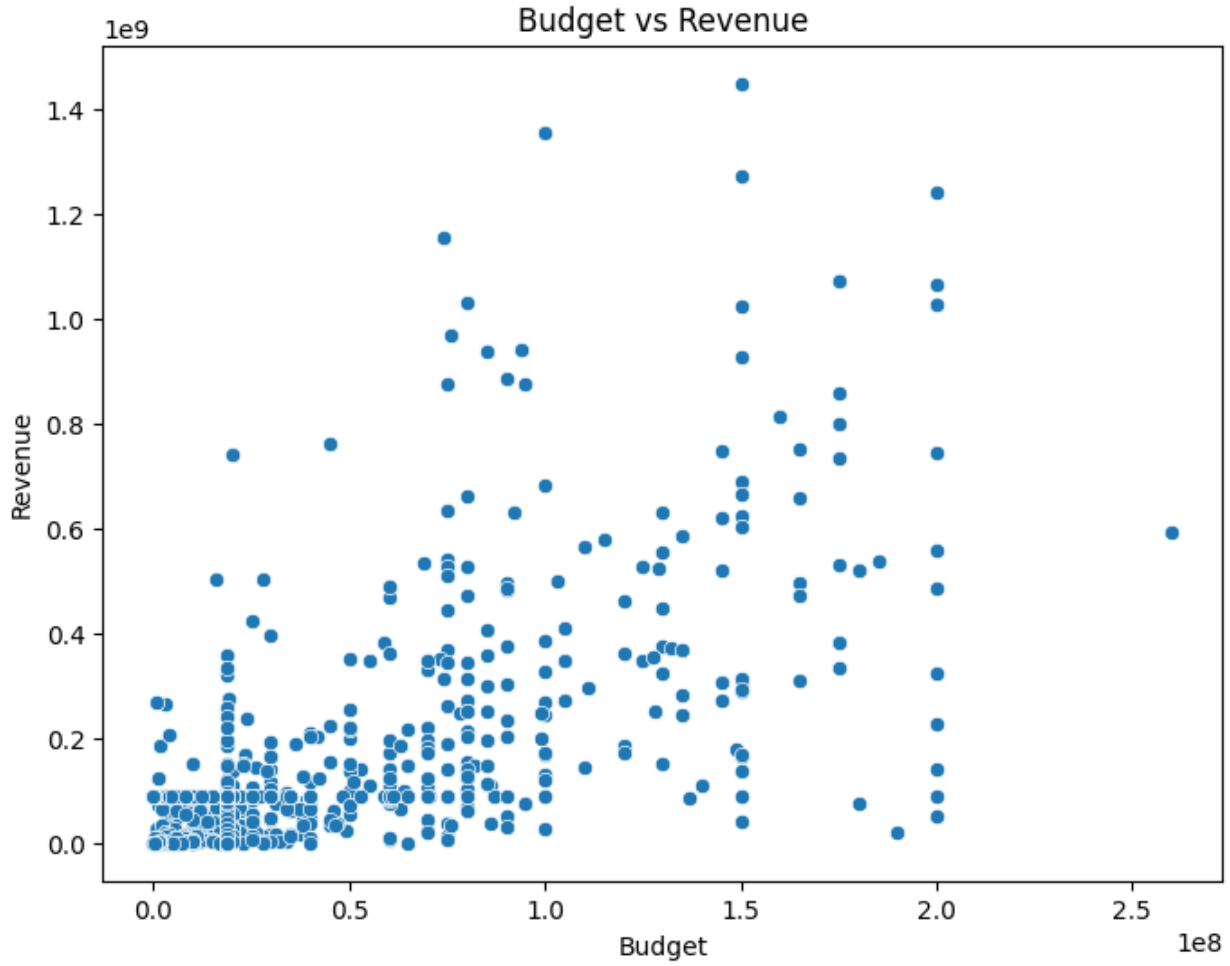
پیش‌پردازش توضیحات (overview) برای تحلیل متن:

```
stop_words = set(stopwords.words('english'))
ps = PorterStemmer()
lemmatizer = WordNetLemmatizer()

def preprocess_text(text):
    words = text.split()
    words = [word for word in words if word.lower() not in stop_words]
    words = [ps.stem(word) for word in words]
    words = [lemmatizer.lemmatize(word) for word in words]
    return ' '.join(words)


df_clean['overview'] = df_clean['overview'].astype(str)
df_clean['overview_processed'] = df_clean['overview'].apply(preprocess_text)
```

خروجی ها :

```
Missing Data:
 id                         0
title                       1
vote_average                0
vote_count                  0
status                      0
release_date             2137
revenue                     0
runtime                     0
adult                       0
backdrop_path           36110
budget                      0
homepage                43692
imdb_id                 22393
original_language           0
original_title              1
overview                 6079
popularity                  0
poster_path             14011
tagline                 47267
genres                      0
production_companies    22547
production_countries    12245
spoken_languages        18127
release_year             2137
```

مصورسازی:



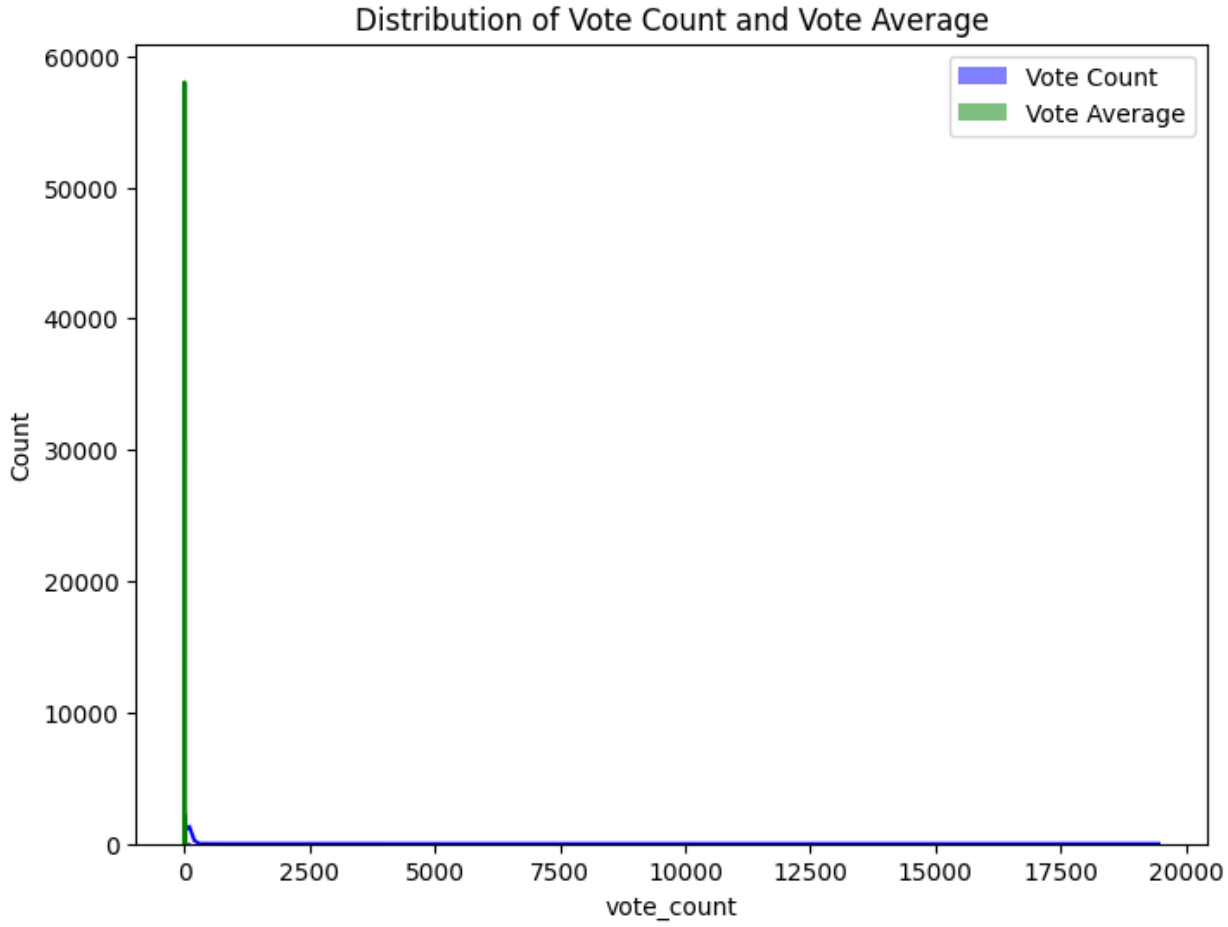
از تحلیل نمودار پراکندگی "بودجه در برابر درآمد" می‌توان نتیجه گرفت که به طور کلی همبستگی مثبتی بین بودجه و درآمد وجود دارد. هرچند بودجه‌های بیشتر معمولاً با درآمدهای بالاتر همراه هستند، اما این تنها عامل تعیین کننده موفقیت نیست. تنوع زیادی در داده‌ها مشاهده می‌شود؛ برخی پروژه‌ها با بودجه کمتر توانسته‌اند درآمدهای بالایی کسب کنند و برخی پروژه‌ها با بودجه بالا درآمدهای نسبتاً کمتری داشته‌اند. این نوسانات نشان می‌دهد که عوامل دیگری همچون کیفیت محتوا، شرایط بازار و زمان‌بندی نیز نقش مهمی ایفا می‌کنند. بنابراین، در برنامه‌ریزی مالی و استراتژی‌گذاری، علاوه بر میزان سرمایه‌گذاری، باید به این عوامل دیگر نیز توجه کرد.

ارتباط بودجه و درآمد:

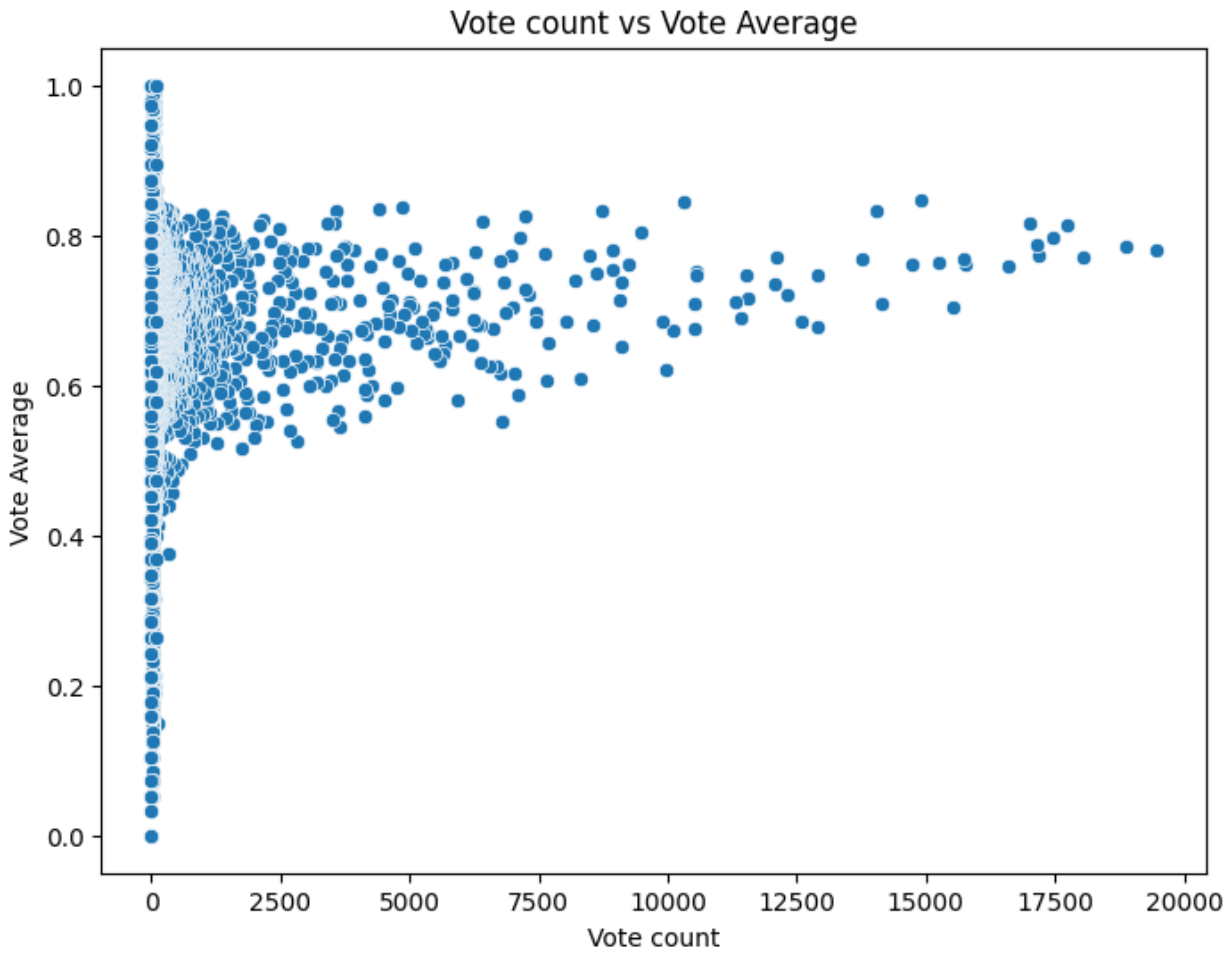
```
correlation = df_clean['budget'].corr(df_clean['revenue'])
print(f'Correlation between Budget and Revenue: {correlation}')
```

✓ 0.0s

```
Correlation between Budget and Revenue: 0.6404117372783544
```

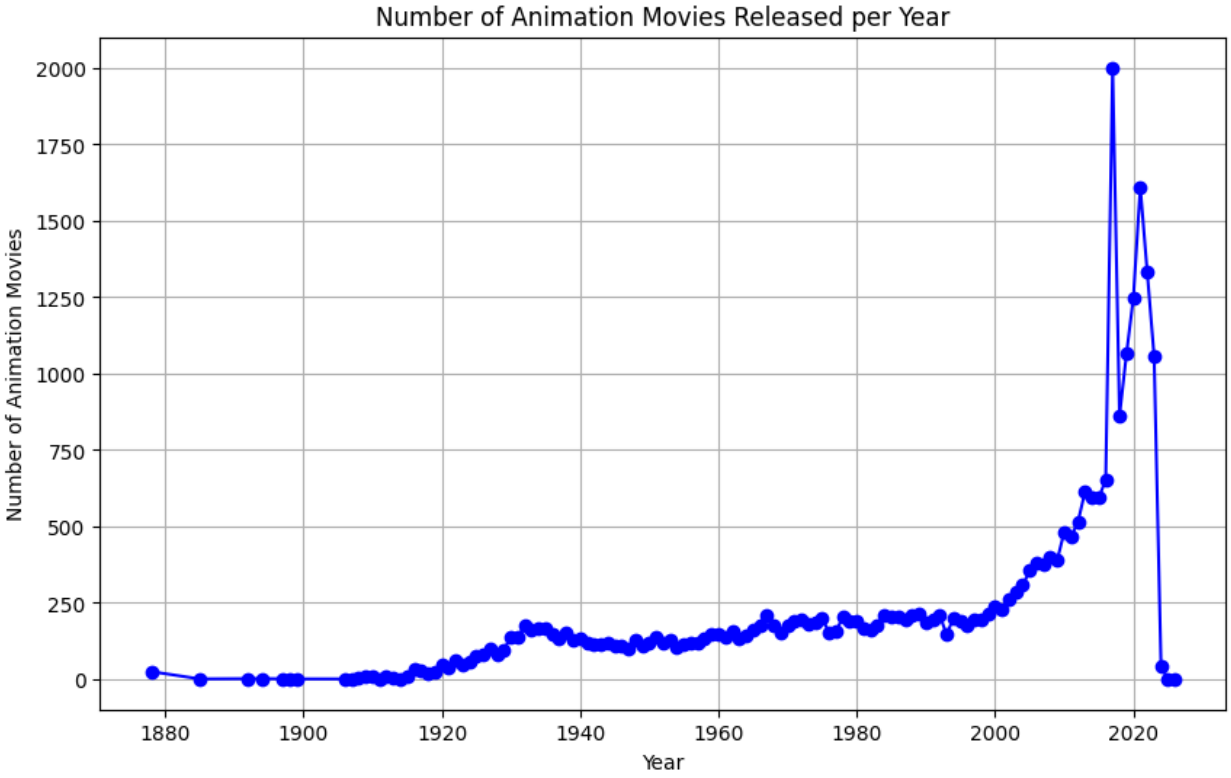


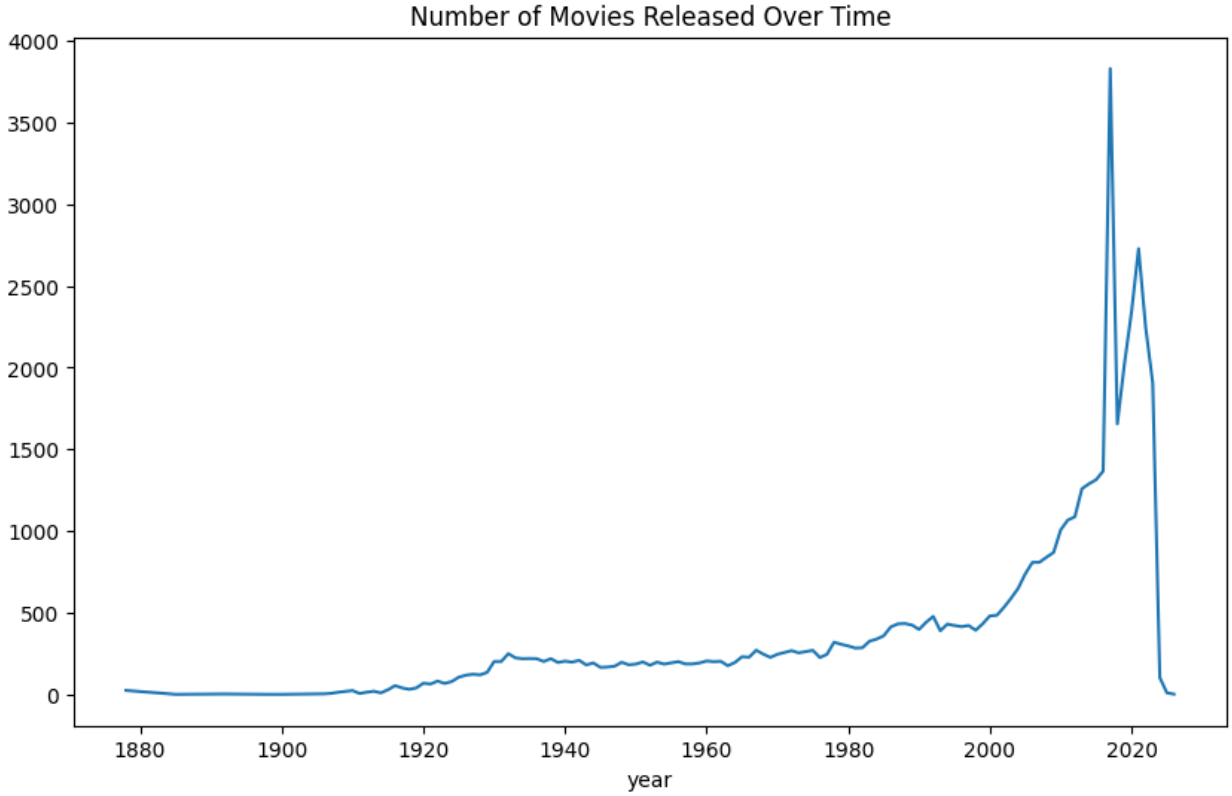
هیستوگرام نشان می‌دهد که اکثر داده‌ها تعداد رای بسیار کمی دارند و یک افزایش قابل توجه در نزدیکی صفر در محور افقی مشاهده می‌شود. این توزیع به سرعت کاهش می‌یابد زیرا تعداد رای‌ها افزایش می‌یابد، که نشان‌دهنده این است که بیشتر آیتم‌ها تعداد رای کمی دارند و تعداد کمی از آیتم‌ها تعداد رای بالایی دارند. این توزیع جالب است زیرا نابرابری در تعداد رای‌ها را برجسته می‌کند و نشان می‌دهد که تعداد کمی از آیتم‌ها بیشتر رای‌ها را دریافت می‌کنند در حالی که بیشتر آیتم‌ها رای‌های بسیار کمی دارند.

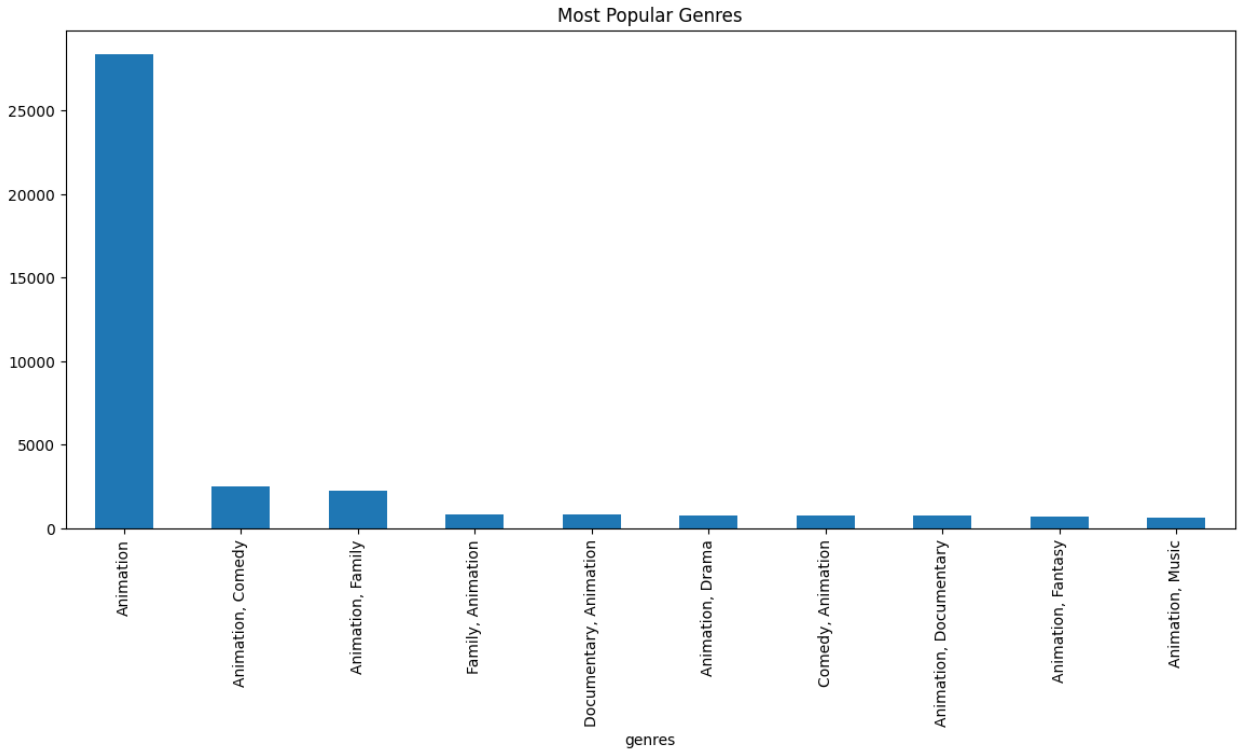


این نمودار نشان می‌دهد که تراکم بالایی از نقاط با تعداد رأی‌های کم (عمدتاً زیر ۲۵۰۰) وجود دارد و میانگین رأی‌ها در این نقاط متغیر است. با افزایش تعداد رأی‌ها، پراکندگی میانگین رأی‌ها کاهش می‌یابد و اکثر نقاط با تعداد رأی‌های بالا میانگین رأی‌های بین ۰.۶ و ۰.۹ دارند.

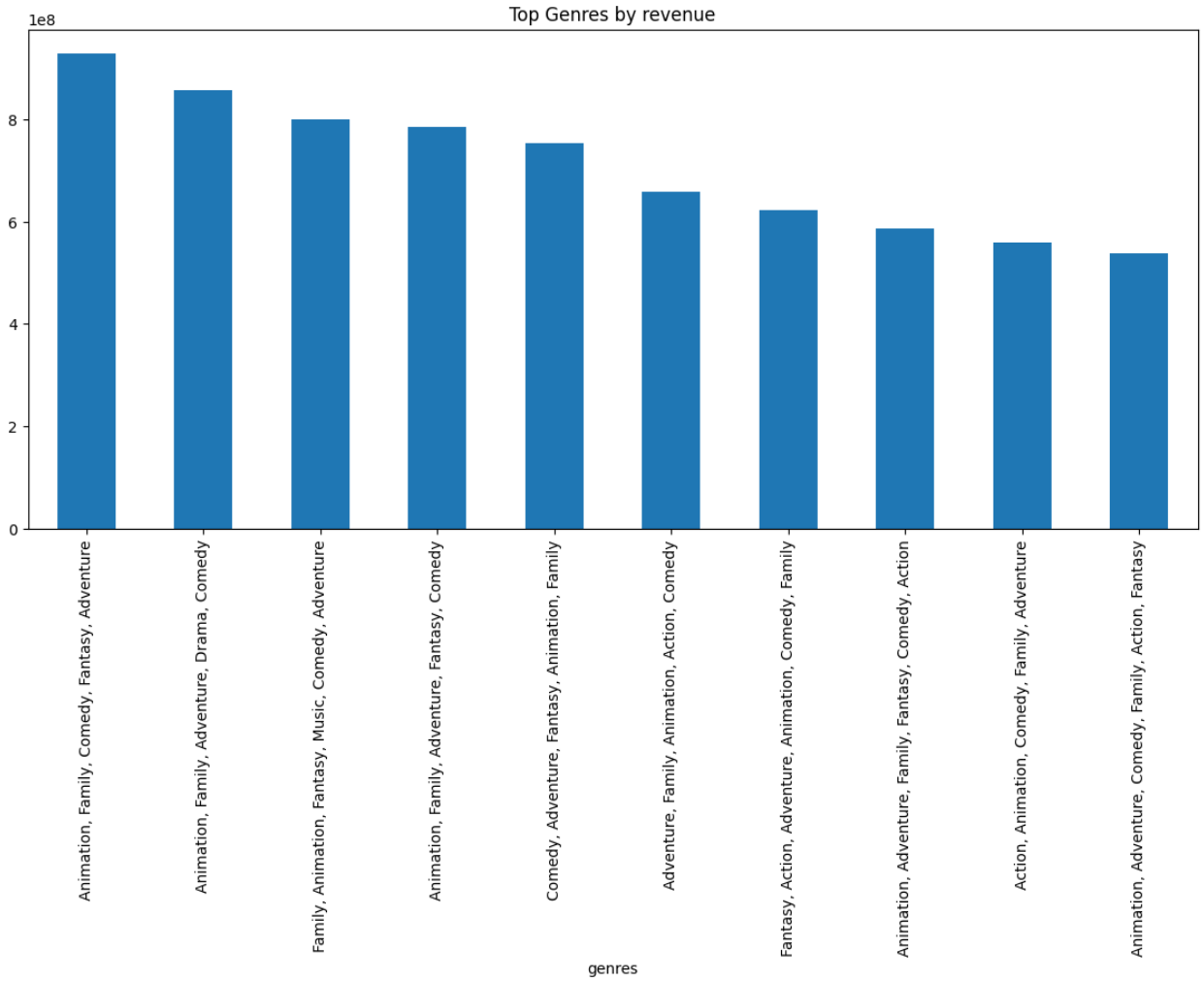
این موضوع نشان می‌دهد که آیتم‌هایی که تعداد رأی بیشتری دارند معمولاً امتیازات بالاتر و پایدارتر دریافت می‌کنند، در حالی که آیتم‌هایی با تعداد رأی کمتر، تنوع بیشتری در امتیازات خود دارند. این الگو ممکن است نشان‌دهنده این باشد که آیتم‌های محبوب‌تر (با تعداد رأی بیشتر) به طور کلی امتیازات مطلوب‌تر و پایدارتر دریافت می‌کنند.

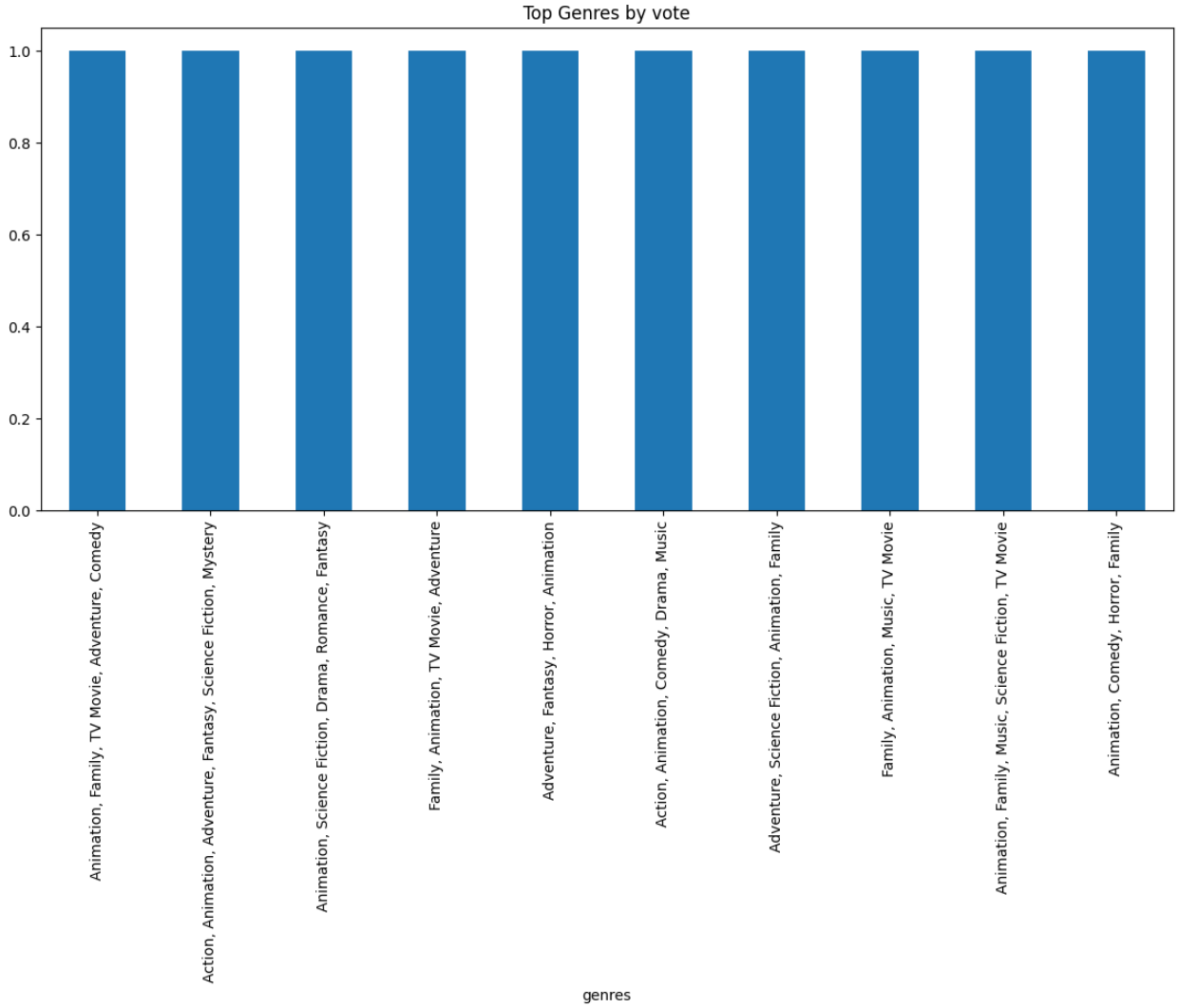

Number of Animation Movies Released per Year
Number of Animation Movies
Year
0
250
500
750
1000
1250
1500
1750
2000
1880
1900
1920
1940
1960
1980
2000
2020

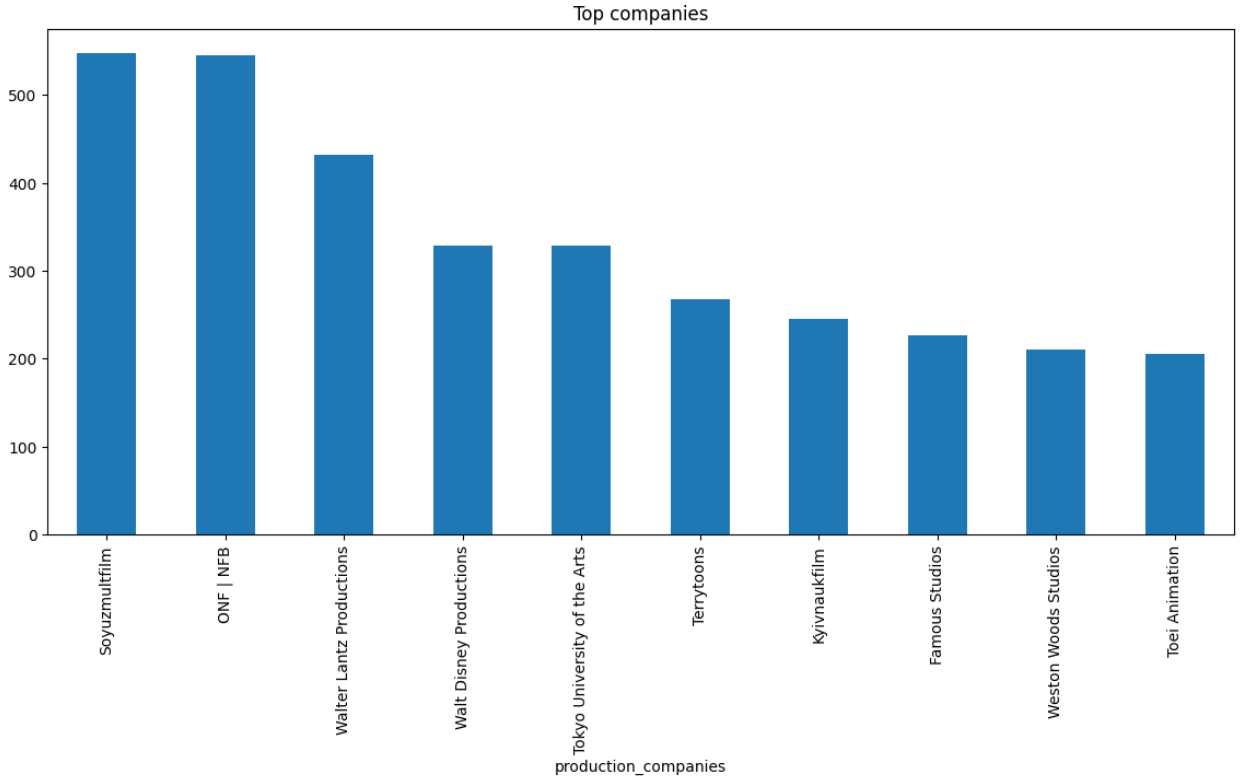

Number of Movies Released Over Time
year
0
500
1000
1500
2000
2500
3000
3500
4000
1880
1900
1920
1940
1960
1980
2000
2020

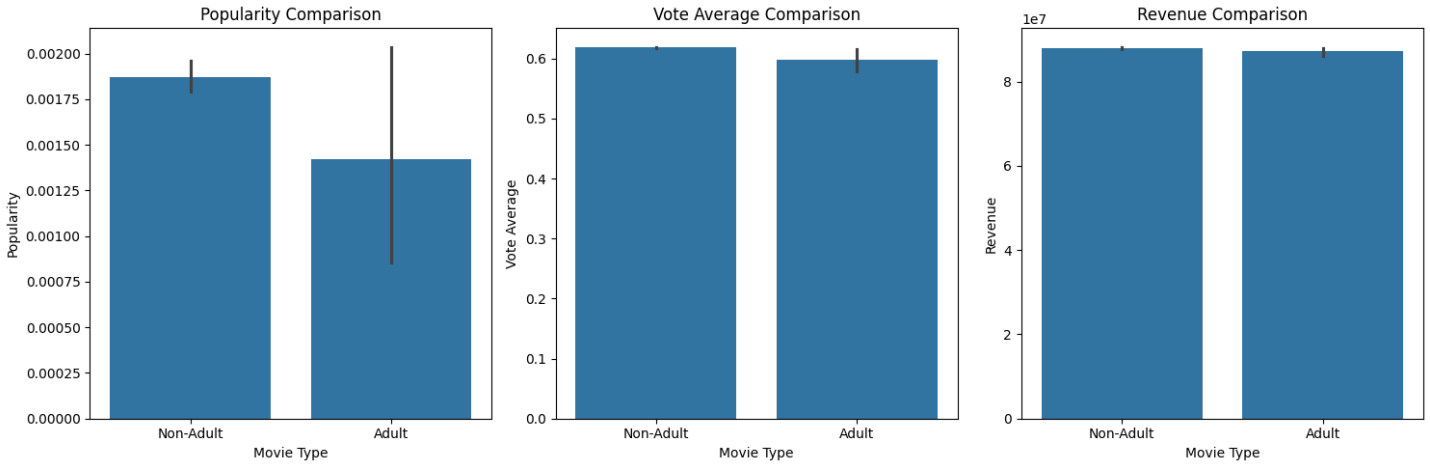


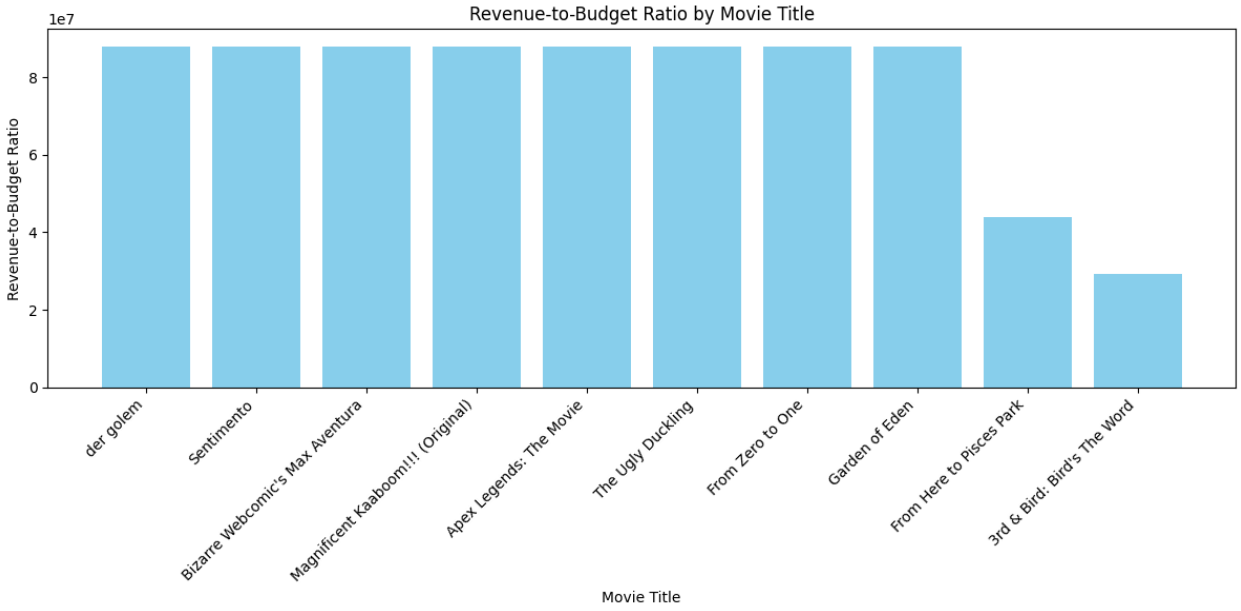
انیمیشن محبوب ترین ژانر

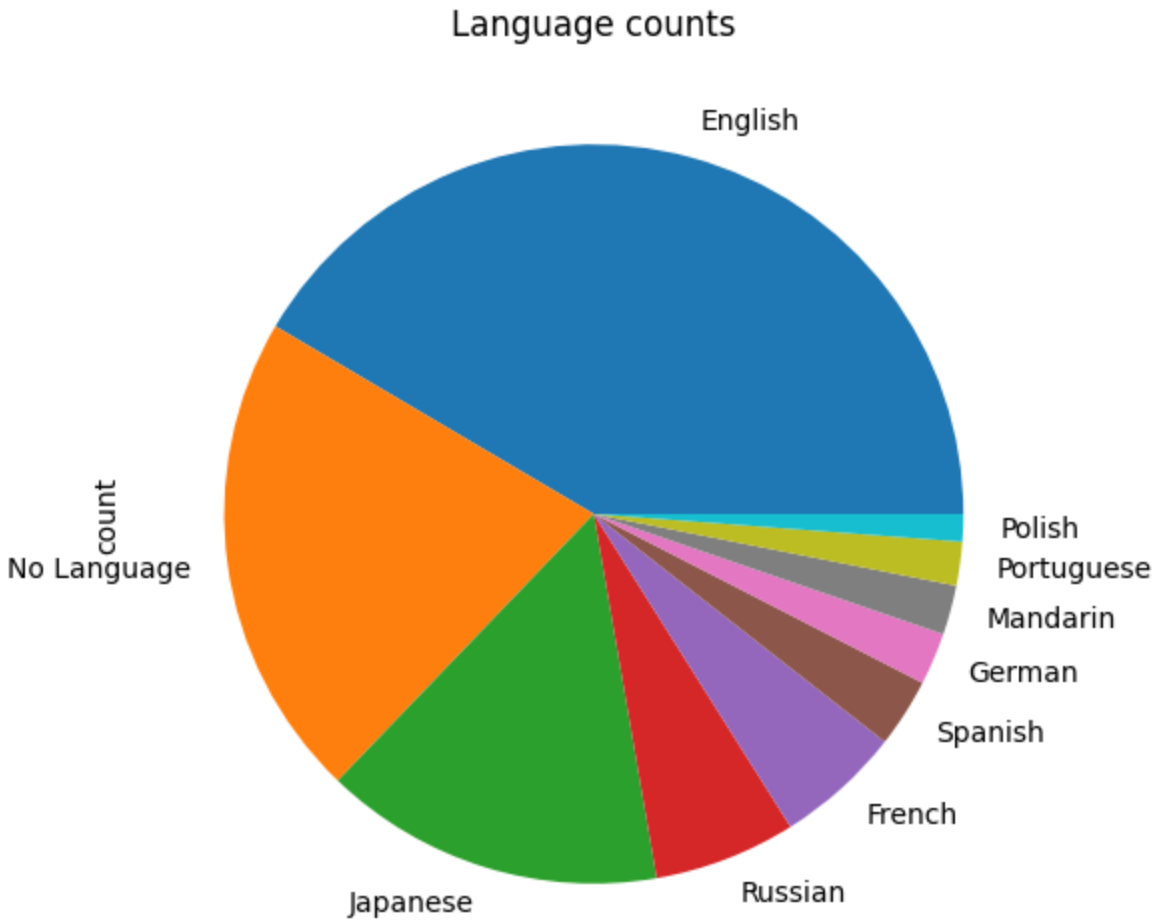

Top Genres by revenue
1e8
0
2
4
6
8
Animation, Family, Comedy, Fantasy, Adventure
Animation, Family, Adventure, Drama, Comedy
Family, Animation, Fantasy, Music, Comedy, Adventure
Animation, Family, Adventure, Fantasy, Comedy
Comedy, Adventure, Fantasy, Animation, Family
Adventure, Family, Animation, Action, Comedy
Fantasy, Action, Adventure, Animation, Comedy, Family
Animation, Adventure, Family, Fantasy, Comedy, Action
Action, Animation, Comedy, Family, Adventure
Animation, Adventure, Comedy, Family, Action, Fantasy
genres


Top Genres by vote
1.0
0.8
0.6
0.4
0.2
0.0
Animation, Family, TV Movie, Adventure, Comedy
Action, Animation, Adventure, Fantasy, Science Fiction, Mystery
Animation, Science Fiction, Drama, Romance, Fantasy
Family, Animation, TV Movie, Adventure
Adventure, Fantasy, Horror, Animation
Action, Animation, Comedy, Drama, Music
Adventure, Science Fiction, Animation, Family
Family, Animation, Music, TV Movie
Animation, Family, Music, Science Fiction, TV Movie
Animation, Comedy, Horror, Family
genres


Top companies
500
400
300
200
100
0
Soyuzmultfilm
ONF | NFB
Walter Lantz Productions
Walt Disney Productions
Tokyo University of the Arts
Terrytoons
Kyivnaukfilm
Famous Studios
Weston Woods Studios
Toei Animation
production_companies


Popularity Comparison
0.00200
0.00175
0.00150
0.00125
0.00100
0.00075
0.00050
0.00025
0.00000
Popularity
Non-Adult
Adult
Movie Type
Vote Average Comparison
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Vote Average
Non-Adult
Adult
Movie Type
Revenue Comparison
1e7
8
6
4
2
0
Revenue
Non-Adult
Adult
Movie Type


Revenue-to-Budget Ratio by Movie Title
1e7
Revenue-to-Budget Ratio
0
2
4
6
8
der golem
Sentimento
Bizarre Webcomic's Max Aventura
Magnificent Kaaboom!!! (Original)
Apex Legends: The Movie
The Ugly Duckling
From Zero to One
Garden of Eden
From Here to Pisces Park
3rd & Bird: Bird's The Word
Movie Title


Language counts
English
count
No Language
Polish
Portuguese
Mandarin
German
Spanish
French
Russian
Japanese

خلاصه خروجی ها :

```
Top Companies with High Rating Movies:

production_companies
Soyuzmultfilm                  548
ONF | NFB                      545
Walter Lantz Productions       432
Walt Disney Productions        329
Tokyo University of the Arts   329
Terrytoons                     268
Kyivnaukfilm                   245
Famous Studios                 226
Weston Woods Studios           211
Toei Animation                 205
Name: count, dtype: int64


Success Metrics for High Rating Movies:
```

	title	popularity	vote_average	revenue
18695	Paw Patrol: Dino Rescue: Roar To The Rescue	0.001556	1.0	8.799786e+07
15626	PAW Patrol - Robo Dog Rescues!	0.002036	1.0	8.799786e+07
15612	Paper Birds	0.000595	1.0	8.799786e+07
18167	The Griffgons: In The Bakehouse	0.000595	1.0	8.799786e+07
18169	Merkabah: Voyage of a Star Seed	0.000595	1.0	8.799786e+07
18172	Daniel	0.001018	1.0	8.799786e+07
18174	Maturation	0.000000	1.0	1.000000e+00
18178	The Aalto Natives	0.000595	1.0	8.799786e+07
18179	Unicorn Boy	0.001679	1.0	8.799786e+07
15605	CarOuzel	0.000595	1.0	8.799786e+07

```
Most Common Languages in Movies:

spoken_languages
English        12005
No Language     6160
Japanese        4308
Russian         1797
French          1588
Spanish          872
German           660
Mandarin         625
Portuguese       553
Polish           346
Name: count, dtype: int64
```