Overview

This analysis aims to guide Trupress's entry into film production by examining data from the film industry to identify the optimal director, release month, and genres for a high Return on Investment film.

Business Problem

Trupress plans to start a movie studio to create original content. Using data from IMDb and The Numbers, I will analyze various films to determine the best directors, release months, and genres for achieving the highest Return on Investment.

Data Understanding

The data sources include:

-`IMDB`

-`The Numbers`

These datasets provide information on film titles, release dates, genres, gross profits, and production budgets. Combining this data will help identify the most profitable options for Trupress's new movie studio.

```
# Import libraries
import pandas as pd
import sqlite3
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Data Cleaning

To get our Numbers dataset in shape for analysis, I tackled a few key steps:

1. **Renaming Columns**: First things first, I renamed the columns to make them more readable and consistent.
2. **Converting Financial Columns**: I converted the financial columns to floats to ensure all calculations run smoothly.
3. **Reformatting Foreign Gross**: I gave the foreign gross figures a makeover, making them easier on the eyes.
4. **Cleaning Up**: Any records missing domestic or foreign gross profit were shown the door to avoid any gaps in our analysis.

```
# Rename the movie column
tn_mb.rename(columns={'title': 'movie_title'}, inplace=True)

# Display the columns
tn_mb.columns
```

```
Index(['movie_title', 'studio', 'domestic_gross', 'foreign_gross',
'year'], dtype='object')
```

```
# Extract the release month from the release date
tn_mb['year'] = tn_mb['year'].astype(str)

# Convert financial columns to float
tn_mb['domestic_gross'] = tn_mb['domestic_gross'].replace('[\$,]', '',
regex=True).astype(float)
tn_mb['foreign_gross'] = tn_mb['foreign_gross'].replace('[\$,]', '',
regex=True).astype(float)

# Remove records with both domestic and worldwide gross equal to 0
tn_mb = tn_mb[(tn_mb['domestic_gross'] != 0) & (tn_mb['foreign_gross']
!= 0)]
```

Merging Datasets

Bringing together the Numbers and IMDB datasets was like assembling the ultimate team for feature engineering and analysis. Here's how I did it:

1. **Merging on 'movie_title': I merged the two datasets on the 'movie_title' column, using an inner join to make sure we only get matched records, keeping things clean and precise.**
2. **Creating ROI Column: I added a new column for Return on Investment (ROI) by subtracting the domestic gross from the foreign gross—this helps us see which movies really paid off.**

3. **Streamlining the Dataset: Finally, I reordered the columns and dropped any extras, leaving us with a neat dataset featuring 'movie_title', 'year', 'genres', 'director_name', and 'roi'.**

```
tn_mb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3387 entries, 0 to 3386
 #   Column          Non-Null Count  Dtype
---  -------         --------------  ---------

 0   movie_title     3387 non-null   object
```

```
 1   studio           3382 non-null   object
 2   domestic_gross   3359 non-null   float64
 3   foreign gross    2037 non-null   float64
 4   year             3387 non-null   object
dtypes: float64(2), object(3)
memory usage: 158.8+ KB
```

```
tn_mb.head()
```

```
                                    movie_title studio  domestic_gross
\
0                                   Toy Story 3     BV     415000000.0

1                     Alice in Wonderland (2010)     BV     334200000.0

0  Harry Potter and the Deathly Hallows Part 1     WB     296000000.0

3                                     Inception     WB     292600000.0

4                           Shrek Forever After   P/DW     238700000.0


   foreign_gross  year
0    652000000.0  2010
1    691300000.0  2010
2    664300000.0  2010
3    535700000.0  2010
4    513900000.0  2010
```

```
# Merge datasets on the 'movie_title' column
movie_data = pd.merge(tn_mb, imdb, on='movie_title', how='inner')

# Create ROI column
movie_data['roi'] = movie_data['foreign_gross'] -
movie_data['domestic_gross']

# Reorder and drop unnecessary columns
movie_data = movie_data[['movie_title', 'year', 'genres',
'director_name', 'roi']]

movie_data.head(5)
```

```
       director_name           roi
0  Christopher Nolan  243100000.0
```

```
1      Mike Mitchell   275200000.0
2        David Slade    97500000.0
3       Byron Howard   190200000.0
4       Chris Renaud    40100000.0
```

Analysis

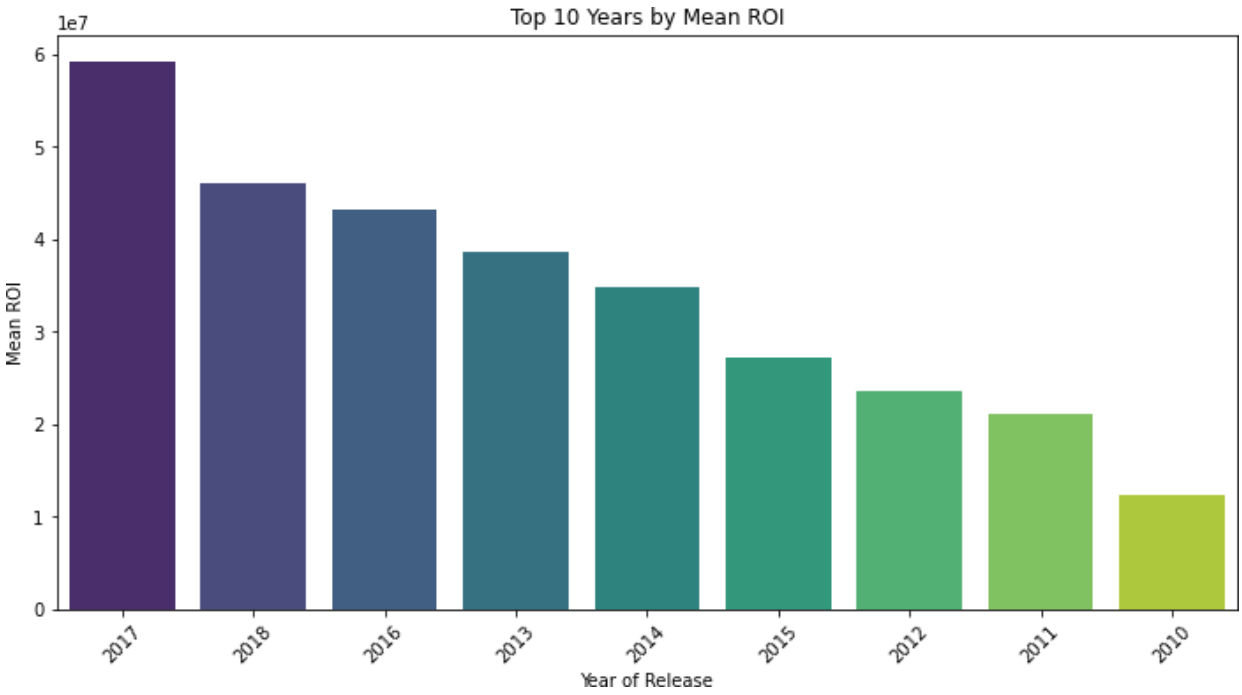
Most Profitable Year of Release

When it comes to releasing a film, timing is everything. Based on my analysis, the golden years for movie releases are 2017, 2018, and 2016. Films from 2017, in particular, boast an impressive average ROI of nearly $59 million! If you ever face delays, consider aiming for a November release—it's a strong secondary option for maximizing returns.

```
# Group data by release year and calculate count, mean, and median of
ROI
profit_years = movie_data.groupby('year')['roi'].agg(['count', 'mean',
'median'])

profit_years_mean = profit_years.sort_values(by='mean',
ascending=False).head(10)
profit_years_mean
```

year	count	mean	median
2017	128	5.913238e+07	10850000.0
2018	123	4.601018e+07	2900000.0
2016	144	4.306255e+07	6293250.0
2013	151	3.858987e+07	6353000.0
2014	161	3.476179e+07	3526000.0
2015	140	2.713382e+07	2531000.0
2012	178	2.348356e+07	1564000.0
2011	218	2.110560e+07	2800000.0
2010	175	1.228090e+07	16900.0

```
# Reset the index to have 'year' as a column
profit_years_mean = profit_years_mean.reset_index()

# Plot the bar chart
plt.figure(figsize=(12, 6))
sns.barplot(x='year', y='mean', data=profit_years_mean,
palette='viridis')

# Add labels and title
plt.xlabel('Year of Release')
plt.ylabel('Mean  ROI')
plt.title('Top 10 Years by Mean ROI')
plt.xticks(rotation=45)
plt.show()
```



Director Most Likely to Create a Film with a High Return on Investment

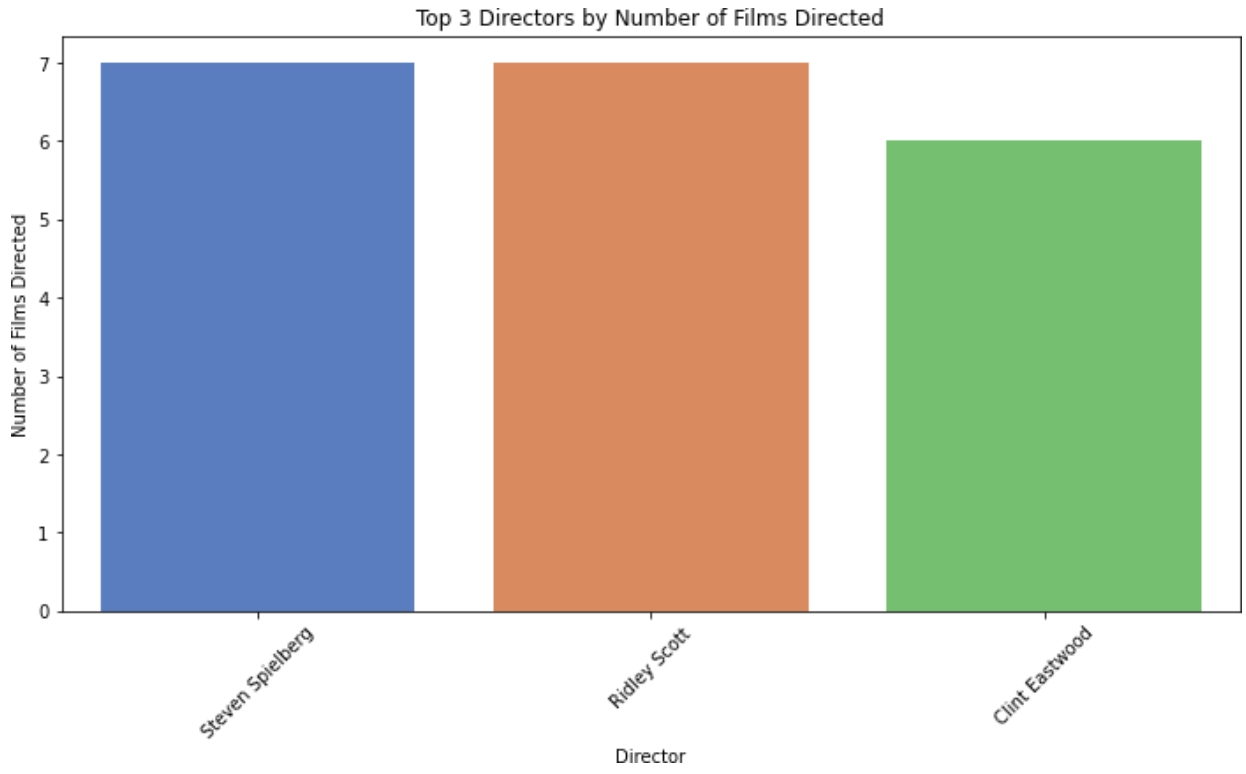
If you're looking to invest in a director, keep an eye on Steven Spielberg, Ridley Scott, and Clint Eastwood. Spielberg consistently delivers with an average ROI of $80 million, while Ridley Scott takes the lead with a staggering $101 million. Interestingly, Clint Eastwood's films show a more volatile return, sometimes even negative, proving that high-risk can also mean high-reward–or the opposite

```
# Group data by director and calculate count, mean, and median of ROI
profit_directors_avg = movie_data.groupby('director_name')
['roi'].agg(['count', 'mean', 'median'])

# Sort by the number of films directed and display top 5 directors
top_directors = profit_directors_avg.sort_values(by='count',
ascending=False).head(3)
top_directors
```

director_name	count	mean	median
Steven Spielberg	7	8.061429e+07	20900000.0
Ridley Scott	7	1.013000e+08	111100000.0
Clint Eastwood	6	-2.566667e+07	-12450000.0

```
# Reset the index to have 'director_name' as a column
top_directors = top_directors.reset_index()

# Plot the number of films directed by top directors
plt.figure(figsize=(12, 6))
```

```
sns.barplot(x='director_name', y='count', data=top_directors,
palette='muted')
plt.xlabel('Director')
plt.ylabel('Number of Films Directed')
plt.title('Top 3 Directors by Number of Films Directed')
plt.xticks(rotation=45)
plt.show()
```



Return on Investment Based on Genre

Genre-wise, if you want a blockbuster ROI, think Action, Adventure, and Sci-Fi. The winning combinations are:

1. **Action, Comedy, Mystery**:
2. **Adventure, Drama, Sport**:
3. **Adventure, Fantasy**:

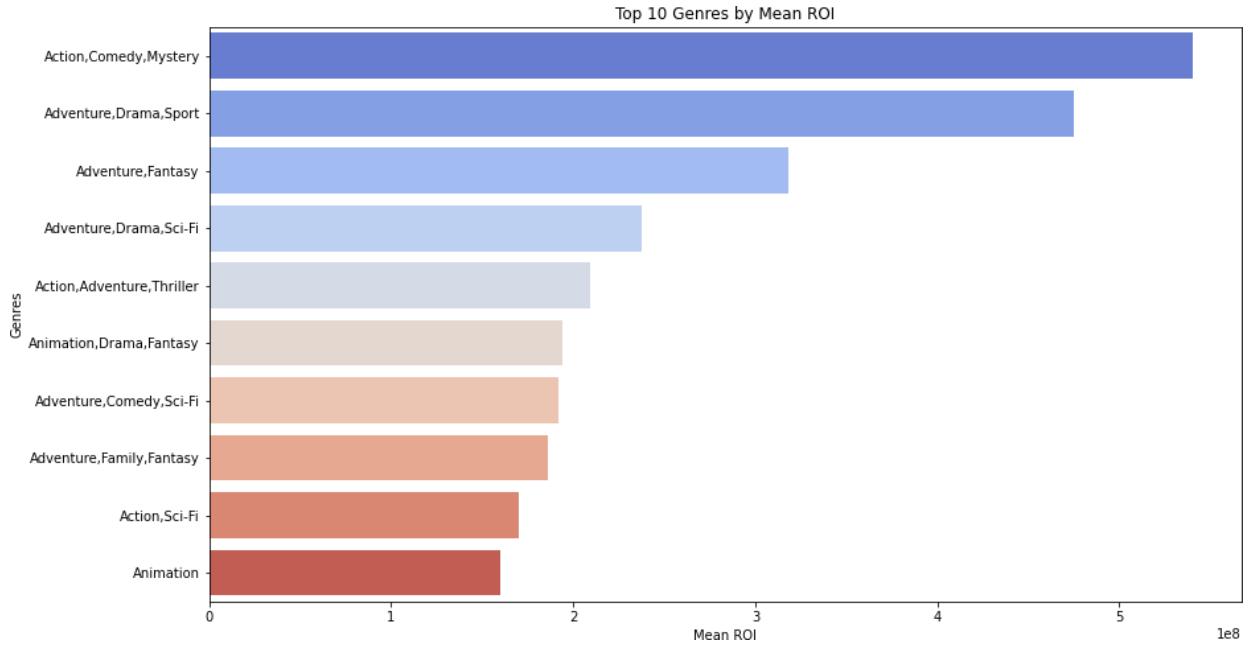
These genres, especially those mixing adventure and fantasy elements, have proven to captivate audiences and deliver strong financial returns.

```
# Group data by genres and calculate count, mean, and median of Return
on Investment
profit_genre_avg = movie_data.groupby('genres')
['roi'].agg(['count','mean', 'median'])

# Sort by mean Return on Investment and display top 10 genres
top_genres_mean = profit_genre_avg.sort_values(by='mean',
ascending=False).head(10)
top_genres_mean
```

	count	mean	median
genres			
Action,Comedy,Mystery	1	5.401000e+08	540100000.0

```
Adventure,Drama,Sport            1  4.750000e+08  475000000.0
Adventure,Fantasy                3  3.182333e+08  441600001.0
Adventure,Drama,Sci-Fi           2  2.373500e+08  237350000.0
Action,Adventure,Thriller       13  2.095154e+08  144900000.0
Animation,Drama,Fantasy          2  1.939500e+08  193950000.0
Adventure,Comedy,Sci-Fi          2  1.917470e+08  191747000.0
Adventure,Family,Fantasy         7  1.858857e+08  145400000.0
Action,Sci-Fi                    1  1.701000e+08  170100000.0
Animation                        1  1.602000e+08  160199999.0
```

```
# Reset the index to have 'genres' as a column
top_genres_mean = top_genres_mean.reset_index()

# Plot the bar chart
plt.figure(figsize=(14, 8))
sns.barplot(x='mean', y='genres', data=top_genres_mean,
palette='coolwarm')

# Add labels and title
plt.xlabel('Mean ROI')
plt.ylabel('Genres')
plt.title('Top 10 Genres by Mean ROI')
plt.show()
```



Conclusions and Recommendations

1. Release Timing: Aim for film releases in `2017`, `2018`, `2016` offer.
2. Director Selection: Focus on directors like Steven `Spielberg`, `Ridley Scott`, and `Clint Eastwood`.

3. Genre Selection: Prioritize films with genres such as `Action, Comedy, Mystery` for the best Return on Investment.