Final Project Submission

Please fill out:

- Student name: **MAK TRNKA**
- Student pace: self paced / part time / full time: **Self Paced**
- Scheduled project review date/time: **2/14/2025**
- Instructor name: **Praveen**

Tools

In [428]:
```
import itertools
import numpy as np
import pandas as pd
from numbers import Number
import sqlite3
from scipy import stats
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

import pickle
```

Other Information

The Numbers (Budget) (*tn.movie_budgets.csv.gz*)

- By comparing production_budget and worldwide_gross, you can identify which movies have made a lot of money but with not such a high production budget

In [442]:
```
import sqlite3
import pandas as pd

Budget_DF = pd.read_csv("zippedData/tn.movie_budgets.csv.gz", encoding='ISO-8

# Clean the 'worldwide_gross' column by removing dollar signs and commas
Budget_DF['worldwide_gross'] = Budget_DF['worldwide_gross'].replace({'\$': ''
Budget_DF['worldwide_gross'] = pd.to_numeric(Budget_DF['worldwide_gross'], er

# Fill missing 'worldwide_gross' values with the mean
mean_value = Budget_DF['worldwide_gross'].mean()
Budget_DF['worldwide_gross'].fillna(mean_value, inplace=True)

# Clean the 'production_budget' column similarly
Budget_DF['production_budget'] = Budget_DF['production_budget'].replace({'\$'
Budget_DF['production_budget'] = pd.to_numeric(Budget_DF['production_budget']

# Calculate Return of Investment (ROI)
Budget_DF['Return_of_Investment'] = Budget_DF['worldwide_gross'] / Budget_DF[

# Create or connect to an SQLite database (use .db extension)
conn = sqlite3.connect("movie_budgets.db")

# Export the DataFrame to an SQLite table with the new 'Return_of_Investment'
Budget_DF.to_sql('movie_budgets', conn, if_exists='replace', index=False)

# Print the first few rows to check the added column
Budget_DF.head(2)
```

Out[442]:

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross	Return_of
0	1	Dec 18, 2009	Avatar	425000000	$760,507,625	2776345279	
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	410600000	$241,063,875	1045663875	

1. Top Movies by Return of Investment *(Worldwide Gross % of Production Budget)*

ANALYSIS

1. **High ROI Films (Low Budget, High Earnings)**

- Movies like *"Paranormal Activity"* (ROI: 431.52) and *"The Gallows"* (ROI: 416.56) show that low-budget films can achieve huge financial success. **Horror** films, in particular, often offer high ROI, making them a great choice for small businesses.

2. **Genre Flexibility with Modest Investment**

- Films like *"Once"* (ROI: 155.49) and *"Napoleon Dynamite"* (ROI: 115.31) show that indie films in genres like **drama**, **comedy**, or **musical** can succeed on modest budgets. Small businesses should consider creating unique, character-driven films to appeal to niche audiences.

3. **Targeted Content for Specific Audiences**

- Movies like *"My Big Fat Greek Wedding"* (ROI: 74.98) and *"Fireproof"* (ROI: 66.95) show that targeted films for specific audiences (e.g., **family** or **faith-based**) can still be successful even with higher budgets. Small businesses can create content for local or niche communities.

DATA CLEANING

The data was cleaned by removing rows with zero worldwide gross, standardizing release dates to the YYYY-MM-DD format, and filtering for movies released after January 1, 2000,

In [443]:
```
# Create a cursor object
cursor = conn.cursor()

# Execute the PRAGMA command
cursor.execute("PRAGMA table_info(movie_budgets);")

# Fetch all results
columns = cursor.fetchall()


q = """
SELECT release_date, movie, production_budget, worldwide_gross, Return_of_Inve

FROM movie_budgets
WHERE worldwide_gross IS NOT '0'

AND date(substr(release_date, -4) || '-' ||
              CASE substr(release_date, 1, 3)
                  WHEN 'Jan' THEN '01'
                  WHEN 'Feb' THEN '02'
                  WHEN 'Mar' THEN '03'
                  WHEN 'Apr' THEN '04'
                  WHEN 'May' THEN '05'
                  WHEN 'Jun' THEN '06'
                  WHEN 'Jul' THEN '07'
                  WHEN 'Aug' THEN '08'
                  WHEN 'Sep' THEN '09'
                  WHEN 'Oct' THEN '10'
                  WHEN 'Nov' THEN '11'
                  WHEN 'Dec' THEN '12'
              END || '-' || substr(release_date, 5, 2)) >= '2000-01-01'

ORDER BY Return_of_Investment DESC

LIMIT 20;

"""

# Run the query and fetch the results
pd.read_sql(q, conn)
```

Out[443]:

	release_date	movie	production_budget	worldwide_gross	Return_of_Investment
0	Sep 25, 2009	Paranormal Activity	450000	194183034	431.517853
1	Jul 10, 2015	The Gallows	100000	41656474	416.564740
2	May 16, 2007	Once	150000	23323631	155.490873
3	Jun 11, 2004	Napoleon Dynamite	400000	46122713	115.306782
4	Sep 29, 2006	Facing the Giants	100000	10243159	102.431590
5	Apr 23, 2009	Home	500000	44793168	89.586336
6	Oct 29, 2004	Saw	1200000	103880027	86.566689
7	Apr 19, 2002	My Big Fat Greek Wedding	5000000	374890034	74.978007
8	Sep 26, 2008	Fireproof	500000	33473297	66.946594
9	Apr 17, 2015	Unfriended	1000000	64364198	64.364198
10	Oct 20, 2010	Paranormal Activity 2	3000000	177512032	59.170677
11	Jan 20, 2017	Split	5000000	278964806	55.792961
12	Mar 21, 2014	GodÃ¢Â□Â□s Not Dead	1150000	63777092	55.458341
13	May 24, 2006	An Inconvenient Truth	1000000	53365925	53.365925
14	Feb 24, 2017	Get Out	5000000	255367951	51.073590
15	Mar 16, 2001	Gabriela	50000	2335352	46.707040
16	May 25, 2012	Les Intouchables	10800000	484873045	44.895652
17	Oct 21, 2016	Moonlight	1500000	65245512	43.497008
18	May 25, 2012	Chernobyl Diaries	1000000	42411721	42.411721
19	Oct 21, 2011	Paranormal Activity 3	5000000	207039844	41.407969

ANALYSIS:

1. **High ROI in Summer (6.71):**

- Summer offers the highest ROI, making it the best season for releasing a movie, maximizing profitability potential.

2. **Stable ROI in Autumn and Spring (4.74, 4.73):**

- Both autumn and spring show solid ROI, suggesting these seasons are reliable for movie releases, offering balanced returns without extreme fluctuations.

3. **Lower ROI in Winter (4.30):**

- Winter has the lowest ROI, indicating it may be a less profitable season for releases, making it a secondary option for smaller studios.

DATA CLEANING The query calculates the average ROI of movies grouped by release season (Winter, Spring, Summer, Autumn), excluding films with zero worldwide gross, and sorts the results by highest average ROI.

In [446]:
```
q = """
SELECT

    AVG(Return_of_Investment) as AVG_ROI,
    CASE
        WHEN release_date LIKE '%Jan%' THEN 'Winter'
        WHEN release_date LIKE '%Feb%' THEN 'Winter'
        WHEN release_date LIKE '%Mar%' THEN 'Spring'
        WHEN release_date LIKE '%Apr%' THEN 'Spring'
        WHEN release_date LIKE '%May%' THEN 'Spring'
        WHEN release_date LIKE '%Jun%' THEN 'Summer'
        WHEN release_date LIKE '%Jul%' THEN 'Summer'
        WHEN release_date LIKE '%Aug%' THEN 'Summer'
        WHEN release_date LIKE '%Sep%' THEN 'Autumn'
        WHEN release_date LIKE '%Oct%' THEN 'Autumn'
        WHEN release_date LIKE '%Nov%' THEN 'Autumn'
        WHEN release_date LIKE '%Dec%' THEN 'Winter'
        ELSE 'Other'
    END AS release_seasons

FROM movie_budgets
WHERE worldwide_gross != 0
GROUP BY release_seasons
ORDER BY AVG_ROI DESC

"""

# Run the query and fetch the results
pd.read_sql(q, conn)
```

Out[446]:

	AVG_ROI	release_seasons
0	6.714963	Summer
1	4.742396	Autumn
2	4.726569	Spring
3	4.302438	Winter

IMDB

1. Combining the 'movie_akas' and 'movie_basics' table

In [456]:
```
q = """

SELECT DISTINCT primary_title, original_title, start_year, region, ordering,
FROM movie_basics b
JOIN movie_akas a USING (movie_id)

WHERE

b.primary_title = b.original_title
AND b.start_year <= 2024
AND b.primary_title NOT LIKE '%untitled%'
AND b.original_title NOT LIKE '%untitled%'
AND region IS NOT null
AND region = "US"
GROUP BY primary_title
ORDER BY ordering DESC

"""
pd.read_sql(q, conn)
```

Out[456]:

	primary_title	original_title	start_year	region	ordering	language
0	The Hobbit: The Battle of the Five Armies	The Hobbit: The Battle of the Five Armies	2014	US	49	None
1	Skyfall	Skyfall	2012	US	43	None
2	The Dark Tower	The Dark Tower	2017	US	40	None
3	Shaun the Sheep Movie	Shaun the Sheep Movie	2015	US	40	None
4	The Hobbit: An Unexpected Journey	The Hobbit: An Unexpected Journey	2012	US	39	None
...	...	...	...	...	...	...
42787	#Captured	#Captured	2017	US	1	None
42788	#BreakfastInBedTour	#BreakfastInBedTour	2013	US	1	None
42789	#BeRobin the Movie	#BeRobin the Movie	2015	US	1	None
42790	#50Fathers	#50Fathers	2015	US	1	None
42791	#5	#5	2013	US	1	None

42792 rows × 6 columns

3. Average Rating by Movie

Data Cleaning

Involved selecting distinct English-language movie titles with at least 1,000 votes, removing duplicates, and grouping by title. The result was sorted by vote count in descending order.

In [457]:
```
q = """

SELECT DISTINCT primary_title, averagerating, numvotes, genres
FROM movie_ratings
JOIN movie_basics USING(movie_id)
JOIN movie_akas USING(movie_id)
WHERE numvotes >= 1000
AND Language = 'en'
GROUP BY primary_title
ORDER BY numvotes DESC
LIMIT 10
"""
pd.read_sql(q, conn)
```

Out[457]:

	primary_title	averagerating	numvotes	genres
0	Inception	8.8	1841066	Action,Adventure,Sci-Fi
1	The Dark Knight Rises	8.4	1387769	Action,Thriller
2	Interstellar	8.6	1299334	Adventure,Drama,Sci-Fi
3	The Avengers	8.1	1183655	Action,Adventure,Sci-Fi
4	The Wolf of Wall Street	8.2	1035358	Biography,Crime,Drama
5	Shutter Island	8.1	1005960	Mystery,Thriller
6	Guardians of the Galaxy	8.1	948394	Action,Adventure,Comedy
7	Deadpool	8.0	820847	Action,Adventure,Comedy
8	The Hunger Games	7.2	795227	Action,Adventure,Sci-Fi
9	Star Wars: Episode VII - The Force Awakens	8.0	784780	Action,Adventure,Fantasy

Action/Adventure = High Popularity and Strong Audience Appeal

ANALYSIS

1. ***High Popularity:***

- Action and Adventure films dominate in popularity, with movies like The Avengers (popularity: 50.289) and Guardians of the Galaxy (popularity: 49.606) standing out as clear favorites among audiences.
- This indicates a strong market demand for these genres.

2. ***Strong Critical and Audience Ratings:***

- Films such as Inception (vote average: 8.3) and Interstellar (vote average: 8.2) are highly rated by both critics and viewers,
- This shows that Action/Adventure films not only attract large audiences but also maintain high levels of engagement.

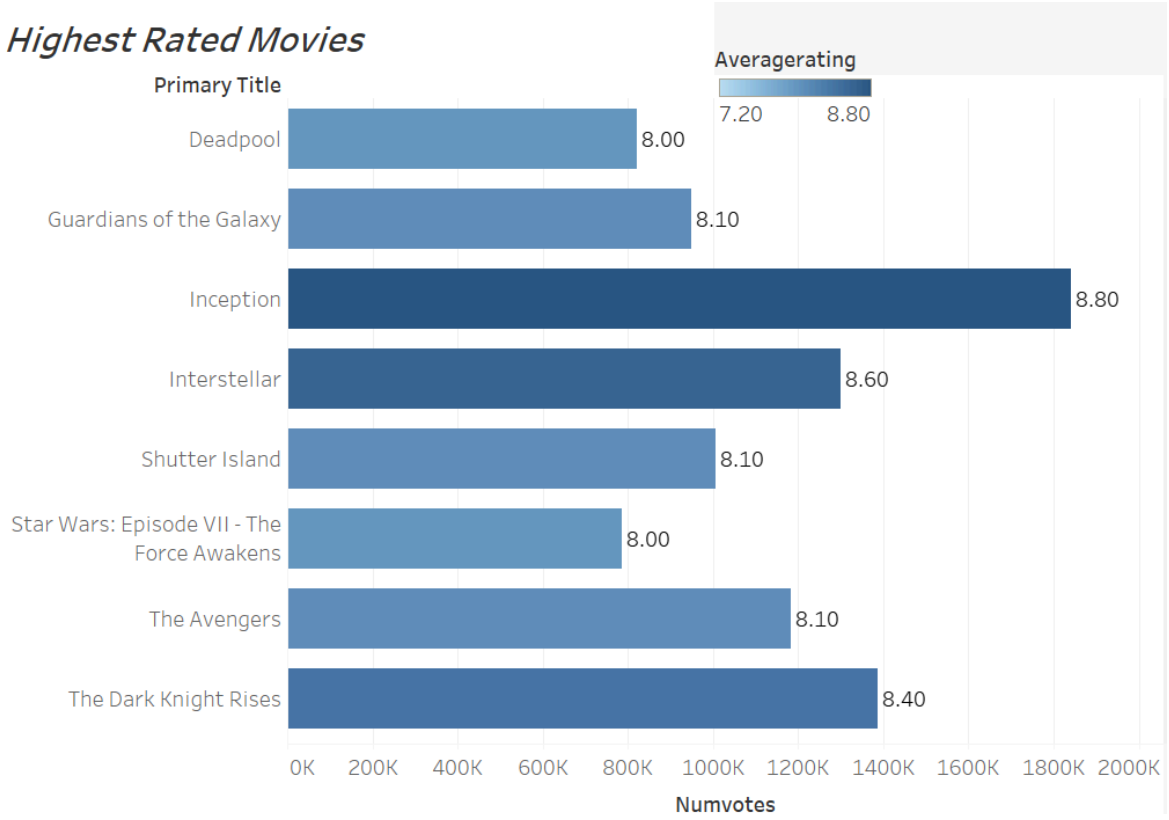
3. ***Large and Active Fan Base:***

- The high vote counts for these titles (Inception: 22,186 votes, Deadpool: 20,175 votes) suggest these films have strong fan communities and significant word-of-mouth

promotion, meaning it can can drive long-term success.

4. ***Consistent Blockbuster Potential:***

   - These genres have a proven track record of producing widely successful, culturally impactful films that often lead to sequels, spin-offs, and expanded universes.



4. Average Rating by Genre

Data Cleaning

Split multiple genre values from a single column into separate individual genre columns, ensuring each genre (e.g., Drama, Thriller, Horror) is uniquely represented for cleaner, more structured data analysis.

In [458]:
```
q = """
SELECT
    genre,
    AVG(numvotes),
    AVG(averagerating) AS Average_Rating
FROM (
    SELECT
        movie_id,
        r.averagerating,
        r.numvotes,
        CASE
        WHEN b.genres LIKE '%Action%' THEN 'Action'
            WHEN b.genres LIKE '%Adventure%' THEN 'Adventure'
            WHEN b.genres LIKE '%Animation%' THEN 'Animation'
            WHEN b.genres LIKE '%Biography%' THEN 'Biography'
            WHEN b.genres LIKE '%Comedy%' THEN 'Comedy'
            WHEN b.genres LIKE '%Crime%' THEN 'Crime'
            WHEN b.genres LIKE '%Documentary%' THEN 'Documentary'
            WHEN b.genres LIKE '%Drama%' THEN 'Drama'
            WHEN b.genres LIKE '%Family%' THEN 'Family'
            WHEN b.genres LIKE '%Fantasy%' THEN 'Fantasy'
            WHEN b.genres LIKE '%History%' THEN 'History'
            WHEN b.genres LIKE '%Horror%' THEN 'Horror'
            WHEN b.genres LIKE '%Musical%' THEN 'Musical'
            WHEN b.genres LIKE '%Romance%' THEN 'Romance'
            WHEN b.genres LIKE '%Sci-Fi%' THEN 'Sci-Fi'
            WHEN b.genres LIKE '%Thriller%' THEN 'Thriller'
            WHEN b.genres LIKE '%War%' THEN 'War'
            WHEN b.genres LIKE '%Western%' THEN 'Western'
            ELSE 'Other'
        END AS genre
    FROM movie_basics b
    JOIN movie_ratings r USING (movie_id)
    WHERE numvotes >= 1000
    AND start_year >= 2000
    AND genres NOT IN ('Talk-Show', 'Music', 'Game-Show', 'NEWS')
) grouped
GROUP BY genre
ORDER BY AVG(numvotes) DESC
;

"""

pd.read_sql(q, conn)
```

Out[458]:

	genre	AVG(numvotes)	Average_Rating
0	Adventure	54649.093878	6.404082
1	Action	50214.528028	6.011261
2	Family	35829.857143	6.628571
3	Biography	28428.251623	6.960877
4	Sci-Fi	23895.521739	5.556522
5	Crime	22988.062389	6.350089
6	Animation	18179.372549	6.835294
7	Drama	17990.333024	6.489410
8	Comedy	16824.906443	6.111827
9	Horror	15634.380805	5.102632
10	Thriller	12266.992701	6.004380
11	Fantasy	11614.083333	5.600000
12	Romance	9941.978723	6.153191
13	Documentary	4308.177686	7.202066
14	War	2437.000000	8.200000
15	Western	1883.666667	5.200000
16	Musical	1723.166667	7.700000
17	History	1540.500000	6.400000

ANALYSIS

1. ***High Ratings vs. Popularity***

- War (8.2), Musical (7.7), and Documentary (7.2) have high ratings but fewer viewers. Adventure (54.6k), Action (50.2k), and Family (35.8k) draw bigger audiences.
- Combine emotional storytelling with popular genres for a balance of acclaim and commercial success.

2. ***Best Genre Balancing Popularity and Rating***

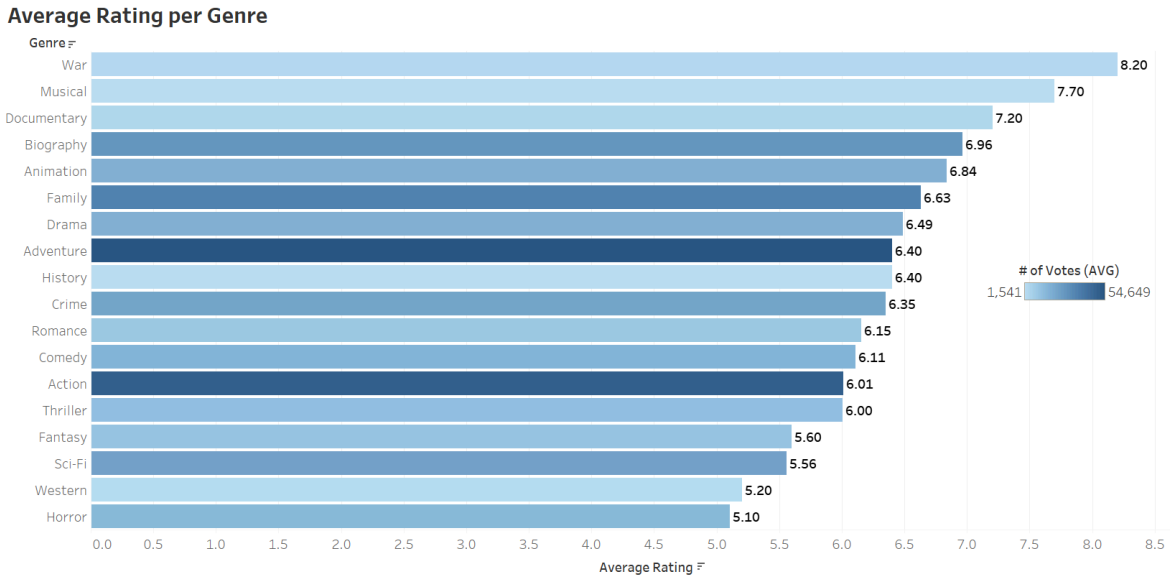
- Animation (6.83, 18.1k) and Biography (6.96, 28.4k) perform well with both high ratings and strong engagement.
- Focus on creating animated films with captivating stories or real-life dramas to appeal to a broad range of viewers.

3. ***Genres with Room for Improvement***

- Sci-Fi (5.55, 23.8k), Fantasy (5.60, 11.6k), and Horror (5.10, 15.6k) are popular but underperform in ratings.
- A thoughtfully crafted sci-fi or fantasy movie could easily stand out and attract more positive attention.

4. ***Focused vs. Wide Appeal***

- War (2.4k), Musical (1.7k), and Documentary (4.3k) attract loyal audiences but smaller numbers. Adventure (54.6k), Action (50.2k), and Comedy (16.8k) have broad appeal.
- Blend focused themes with widely appealing elements to broaden reach and impact.



CLEANED & FINAL USED DATASET

What Metric are we looking at here?

What is Return of Investment (ROI)? Here we focus on *RETURN_OF_INVESTMENT (ROI)*, which is calcuating ratio of a movie's worldwide gross to its production budget, colunm to the Table.

- A **Higher ROI** means the movie has generated more revenue compared to its production cost, indicating a more profitable film.
- A **Lower ROI** suggests the movie did not perform as well financially in comparison to its cost.

Why would we use this metric to resolve the issue?

Return of Investment (ROI) is a useful metric for small business studies to evaluate the financial success. Key Reasons:

- **Risk Mitigation:** By focusing on ROI, small studios minimize financial risks by producing low-cost, high-return films that cater to niche markets.
- **Market Demand Insight:** High ROI reveals market demand, guiding studios to produce movies in popular genres, know when could be best time to release, and styles that resonate with audiences.

Key Filters?

- **Why we chose starting in 2000:** Here we are analyizing movies that made best in Return of Investment since 2000. We chose 2000 and beyond, since it may be the most relevent

to Modern Audiences & Industry Trends.

- **Why the Production Budget is limited to $20m : * * In addition we chose to a limit of f$ 20m dollars since that seems like the average amount of budget for a small studio.

In [470]:
```
import pandas as pd

# Load the CSV files
budget_data = pd.read_csv("zippedData/tn.movie_budgets.csv.gz")
tmdb_data = pd.read_csv("zippedData/tmdb.movies.csv.gz")

# Convert both Release_Date columns to datetime format
budget_data['release_date'] = pd.to_datetime(budget_data['release_date'])
tmdb_data['release_date'] = pd.to_datetime(tmdb_data['release_date'])

# Merge the two dataframes on Release_Date
combined_df = pd.merge(budget_data, tmdb_data, left_on="movie", right_on="ori

# Clean and convert 'production_budget' to numeric
combined_df['production_budget'] = combined_df['production_budget'].replace({
combined_df['production_budget'] = pd.to_numeric(combined_df['production_budg

# Clean and convert 'worldwide_gross' to numeric
combined_df['worldwide_gross'] = combined_df['worldwide_gross'].replace({'\$'
combined_df['worldwide_gross'] = pd.to_numeric(combined_df['worldwide_gross']

# Adding a New Column Return on Investment (ROI)
combined_df['Return_of_Investment'] = (
    combined_df['worldwide_gross'] / combined_df['production_budget']
)


combined_df = combined_df.drop(['domestic_gross', 'id_y', 'release_date_y', '


# Save the cleaned and merged dataframe to a new CSV
combined_df.to_csv("merged_and_cleaned_data.csv", index=False)

# Connect to SQLite database (it will create the database file if it doesn't e
conn = sqlite3.connect('Final_Data.db')

# Create a cursor object
cursor = conn.cursor()

# Convert the DataFrame to SQL and create a table in SQLite
combined_df.to_sql('Final_Data', conn, if_exists='replace', index=False)

# Commit and close the connection
conn.commit()

combined_df.head()
```

Out[470]:

	id_x	release_date_x	movie	production_budget	worldwide_gross	genre_ids	original_lan
0	1	2009-12-18	Avatar	425000000	2776345279	[28, 12, 14, 878]	
1	2	2011-05-20	Pirates of the Caribbean: On Stranger Tides	410600000	1045663875	[12, 28, 14]	
2	4	2015-05-01	Avengers: Age of Ultron	330600000	1403013963	[28, 12, 878]	
3	7	2018-04-27	Avengers: Infinity War	300000000	2048134200	[12, 28, 14]	
4	9	2017-11-17	Justice League	300000000	655945209	[28, 12, 14, 878]	

In [471]:
```
q = """
SELECT
    COUNT(DISTINCT MOVIE) AS movie_count,
    genre,
    AVG(Return_of_Investment) AS AVG_ROI,
    AVG(production_budget) AS AVG_PD,
    AVG(worldwide_gross) AS AVG_WG
FROM (
    SELECT
        CASE
            WHEN genre_ids LIKE '%28%' THEN 'Action'
            WHEN genre_ids LIKE '%12%' THEN 'Adventure'
            WHEN genre_ids LIKE '%16%' THEN 'Animation'
            WHEN genre_ids LIKE '%35%' THEN 'Comedy'
            WHEN genre_ids LIKE '%80%' THEN 'Crime'
            WHEN genre_ids LIKE '%99%' THEN 'Documentary'
            WHEN genre_ids LIKE '%18%' THEN 'Drama'
            WHEN genre_ids LIKE '%10751%' THEN 'Family'
            WHEN genre_ids LIKE '%14%' THEN 'Fantasy'
            WHEN genre_ids LIKE '%36%' THEN 'History'
            WHEN genre_ids LIKE '%27%' THEN 'Horror'
            WHEN genre_ids LIKE '%10402%' THEN 'Music'
            WHEN genre_ids LIKE '%9648%' THEN 'Mystery'
            WHEN genre_ids LIKE '%10749%' THEN 'Romance'
            WHEN genre_ids LIKE '%878%' THEN 'Science Fiction'
            WHEN genre_ids LIKE '%10770%' THEN 'TV Movie'
            WHEN genre_ids LIKE '%53%' THEN 'Thriller'
            WHEN genre_ids LIKE '%10752%' THEN 'War'
            WHEN genre_ids LIKE '%37%' THEN 'Western'
            ELSE 'Unknown'
        END AS genre,
        Return_of_Investment,
        production_budget,
        worldwide_gross,
        genre_ids,
        movie
    FROM Final_Data
    WHERE release_date_x >= '2000-01-01'
    AND genre != 'Unknown'
    AND genre != 'Drama'
    AND production_budget <= 20000000
) AS genre_data
GROUP BY genre
HAVING COUNT(DISTINCT MOVIE) >= 20
ORDER BY AVG_ROI DESC;
"""
# Run the query and fetch the results
print('Average Return of Investmnet per Genre')
pd.read_sql(q, conn)
```

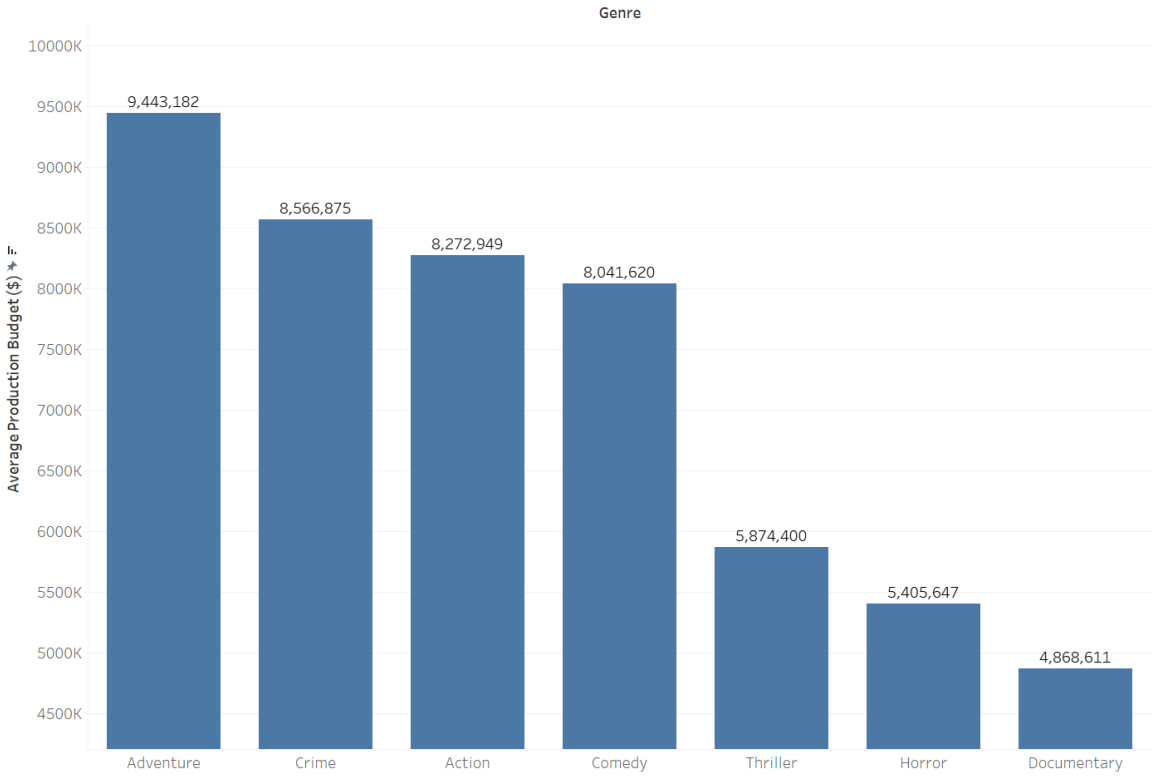
Average Return of Investmnet per Genre

Out[471]:

	movie_count	genre	AVG_ROI	AVG_PD	AVG_WG
0	125	Horror	12.861837	5.405647e+06	4.588531e+07
1	251	Comedy	3.150457	8.041620e+06	2.939913e+07
2	65	Crime	3.145318	8.566875e+06	2.123242e+07
3	32	Adventure	2.299831	9.443182e+06	2.128475e+07
4	110	Action	1.950712	8.272949e+06	1.386958e+07
5	51	Documentary	1.504004	4.868611e+06	9.596667e+06
6	21	Thriller	1.138024	5.874400e+06	7.842042e+06

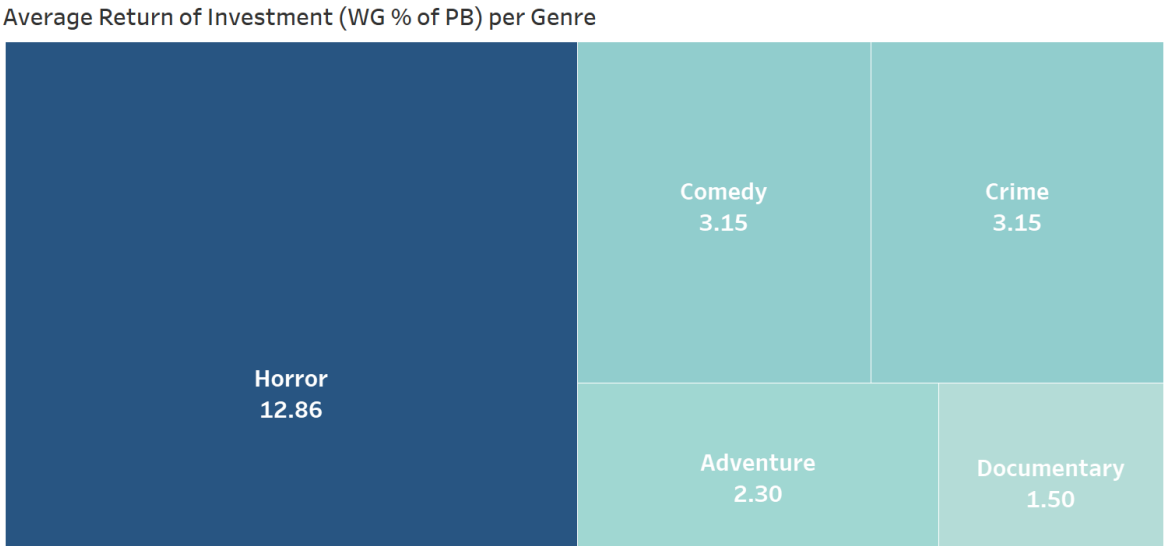
From Most Expensive to Cheapest Movie Genres

Documentaries and thrillers also present low production budgets, with documentaries offering a modest ROI, and thrillers, though lower in ROI than horror, still provide manageable costs for small studios to profit.



Horror is basically the monopoly of Return of Investment (ROI)

Horror movies have the highest average ROI at 12.86, indicating that for every dollar spent, horror movies tend to generate significantly more revenue compared to other genres. This makes it a financially promising genre to explore.



In [472]:
```
q= '''
SELECT strftime('%m', release_date_x) AS Month, AVG(Return_of_Investment) AS A
FROM Final_Data
WHERE genre_ids != 'Unknown'
AND genre_ids != 'Drama'
GROUP BY month
HAVING COUNT(DISTINCT MOVIE) >= 20
ORDER BY AVG_ROI DESC
'''
pd.read_sql(q, conn)
```

Out[472]:

	Month	AVG_ROI	AVG_PD	AVG_WG
0	07	6.357361	4.917224e+07	1.705610e+08
1	10	6.101707	2.578260e+07	7.333551e+07
2	04	6.034681	2.644670e+07	8.435555e+07
3	02	5.302624	3.328077e+07	1.044737e+08
4	01	5.088311	2.486851e+07	6.398888e+07
5	11	4.311775	5.276787e+07	1.852377e+08
6	06	3.770249	5.253662e+07	2.004883e+08
7	05	3.534998	5.816665e+07	1.776495e+08
8	09	3.118161	2.532320e+07	6.423093e+07
9	08	2.993802	2.876839e+07	7.460357e+07
10	12	2.637853	3.760895e+07	1.259407e+08
11	03	2.545894	4.463653e+07	1.271046e+08

ANALYSIS

High ROI in July (6.36):

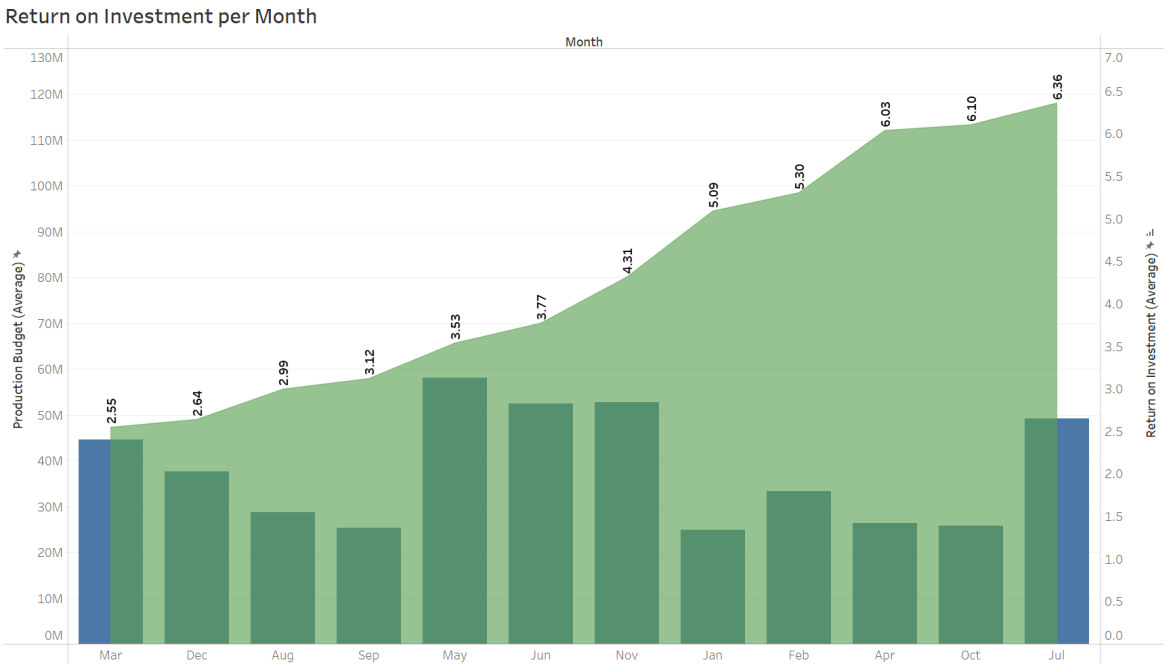
- July offers the highest ROI, suggesting it's the most profitable month for releases, making it an ideal time to launch a movie for maximum returns.

Low Production Budget in October (2.58M):

- October shows one of the lowest average production budgets, allowing studios to release films at lower costs while still maintaining solid profitability potential.

High Marketing Reach in June (2.00B):

- June has the highest average marketing spend, providing an extensive platform for reaching larger audiences, which can support a movie's success in terms of visibility and ROI.



Horror = Most Profitable & Cost-Effective Genre

- **Highest Return on Investment (ROI):** Horror stands out with an average ROI of 12.86x, which is significantly higher than any other genre. This means that for every dollar spent on production, horror movies generate almost 13 times that amount in revenue.
- **Low Average Production Budget:** While Horror holds second lowest in low average production budget of $5.41M, italsooutperformothergenresintermsofprofitability.Thiscombinationoflowcostandhighreturnsmakesthemanidealinvestment.-**EfficientUseofResources:**Thechartshowsthatgenreslike ActionandAdventurehavehigherbudgets(around$ 8–9M), but their ROI is much lower (below 2.5x). Horror offers a more efficient use of resources for a higher return.
- **Consistent Performance:** Even though the budget for Horror is modest compared to other genres, it consistently delivers high returns, indicating strong audience demand and a reliable market.


Average Return of Investment per Genre (Worldwide Gross % of Production Budget)
Genre
Production Budget (AVG $)
70M
60M
50M
40M
30M
14
13
12
11
10
9
8
7
6
5
12.86