Phase 2 Project

Analyzing Movie Trends: What Films Perform Best at the Box Office?

Business Problem

Your company now sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don't know anything about creating movies. You are charged with exploring what types of films are currently doing the best at the box office. You must then translate those findings into actionable insights that the head of your company's new movie studio can use to help decide what type of films to create.

Introduction

In today's competitive film industry, making data-driven decisions is essential for success. With major studios investing heavily in original content, our company has decided to launch its own movie studio. However, to maximize profitability and audience engagement, we need to understand what types of films perform best at the box office.

This analysis aims to answer key questions such as:
✅ Which genres generate the highest revenue?
✅ How do production budgets impact profitability?
✅ What trends exist in audience preferences over time?
✅ How does critical reception correlate with box office performance?

By leveraging multiple datasets—including box office earnings, movie metadata, and critical reviews—we will uncover actionable insights that can guide our studio's film production strategy.

Data Sources

We will analyze and combine data from the following sources:
📊 Movie Budgets & Revenue – Production budgets, domestic & worldwide gross earnings: `tn.movie_budgets.csv.gz`
🎬 Movie Metadata – Genre, language, popularity, and ratings: `im.db`

Approach

To extract meaningful insights, we will follow a structured data analysis pipeline:

1 Data Preprocessing & Preparation

- Load libraries & datasets and confirm if loaded appropriately
- Use unique identifiers (such as movie IDs or titles) to merge datasets
- Inspecting the data

2 Data Cleaning & Integration

- Standardize formats (e.g., merging different date formats, handling null values)
- Handle missing values, duplicates, and inconsistencies

3 Exploratory Data Analysis (EDA)

- Identify trends in budget vs. revenue
- Analyze genre popularity and profitability
- Perform statistical tests for further analysis

4 Conclusion & Recommendations

- Summarize findings to determine the most viable film genres and budget ranges
- Provide data-driven suggestions on key success factors for the new movie studio

Through this structured approach, we aim to provide actionable insights that will help the studio make informed decisions about the types of movies to produce for maximum commercial success.

---

1 Data Preprocessing & Preparation

1.1 Importing Libraries

In [36]:

```
#standard libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels as sm
from scipy import stats
import sqlite3
import statsmodels.api as sm
```

1.2 Data Loading and Verifiaction

In [37]:
```
# load data from movie_basics
conn = sqlite3.connect('im.db')
moviebasics = pd.read_sql("""
SELECT *
  FROM movie_basics;
""", conn)
moviebasics
```

Out[37]:

	movie_id	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy
...	...	...	...	...	...	...
146139	tt9916538	Kuambil Lagi Hatiku	Kuambil Lagi Hatiku	2019	123.0	Drama
146140	tt9916622	Rodolpho Teóphilo - O Legado de um Pioneiro	Rodolpho Teóphilo - O Legado de um Pioneiro	2015	NaN	Documentary
146141	tt9916706	Dankyavar Danka	Dankyavar Danka	2013	NaN	Comedy
146142	tt9916730	6 Gunn	6 Gunn	2017	116.0	None
146143	tt9916754	Chico Albuquerque - Revelações	Chico Albuquerque - Revelações	2013	NaN	Documentary

146144 rows × 6 columns

In [38]:

```
# load data from movie_ratings
conn = sqlite3.connect('im.db')
movieratings = pd.read_sql("""
SELECT *
  FROM movie_ratings;
""", conn)
movieratings
```

Out[38]:

	movie_id	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21
...	...	...	...
73851	tt9805820	8.1	25
73852	tt9844256	7.5	24
73853	tt9851050	4.7	14
73854	tt9886934	7.0	5
73855	tt9894098	6.3	128

73856 rows × 3 columns

In [39]:
```
budgets = pd.read_csv("data/tn.movie_budgets.csv.gz")
budgets.head()
```

Out[39]:

	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

1.3 Data Merging with Common Keys

In [40]:
```
# Merge the two datasets on movie_id
imdb_merge = pd.merge(moviebasics, movieratings, on="movie_id", how="inner")
# Close the database connection
conn.close()
```

In [41]:
```
imdb_merge
```

Out[41]:

	movie_id	primary_title	original_title	start_year	runtime_minutes	genres	averagerating	numvotes
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama	7.0	77
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama	7.2	43
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama	6.9	4517
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama	6.1	13
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy	6.5	119
...	...	...	...	...	...	...	...	...
73851	tt9913084	Diabolik sono io	Diabolik sono io	2019	75.0	Documentary	6.2	6
73852	tt9914286	Sokagin Çocuklari	Sokagin Çocuklari	2019	98.0	Drama,Family	8.7	136
73853	tt9914642	Albatross	Albatross	2017	NaN	Documentary	8.5	8
73854	tt9914942	La vida sense la Sara Amat	La vida sense la Sara Amat	2019	NaN	None	6.6	5
73855	tt9916160	Drømmeland	Drømmeland	2019	72.0	Documentary	6.5	11

73856 rows × 8 columns

In [42]:
```
# Convert titles to lowercase and strip spaces for better matching
budgets["movie"] = budgets["movie"].str.lower().str.strip()
imdb_merge["primary_title"] = imdb_merge["primary_title"].str.lower().str.strip()
imdb_merge["original_title"] = imdb_merge["original_title"].str.lower().str.strip()
```

In [43]:
```
movies_merged = pd.merge(imdb_merge, budgets, left_on="primary_title", right_on="movie", how="inner")
movies_merged
```

Out[43]:

	movie_id	primary_title	original_title	start_year	runtime_minutes	genres	averagerating	numvotes	id	releas
0	tt0249516	foodfight!	foodfight!	2012	91.0	Action,Animation,Comedy	1.9	8248	26	Dec 3
1	tt0326592	the overnight	the overnight	2010	88.0	None	7.5	24	21	Jun 1
2	tt3844362	the overnight	the overnight	2015	79.0	Comedy,Mystery	6.1	14828	21	Jun 1
3	tt0337692	on the road	on the road	2012	124.0	Adventure,Drama,Romance	6.1	37886	17	Mar 2
4	tt4339118	on the road	on the road	2014	89.0	Drama	6.0	6	17	Mar 2
...	...	...	...	...	...	...	...	...	...	
2930	tt8680254	richard iii	richard iii	2016	NaN	Drama	9.1	28	65	Dec 2
2931	tt8824064	heroes	heroes	2019	88.0	Documentary	7.3	7	12	Oct 2
2932	tt8976772	push	push	2019	92.0	Documentary	7.3	33	70	Feb
2933	tt9024106	unplanned	unplanned	2019	106.0	Biography,Drama	6.3	5945	33	Mar 2
2934	tt9248762	the terrorist	the terrorist	2018	NaN	Thriller	6.0	6	48	Jan 1

2935 rows × 14 columns

1.4 Data Inspection

In [44]:
```
movies_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2935 entries, 0 to 2934
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   movie_id           2935 non-null   object 
 1   primary_title      2935 non-null   object 
 2   original_title     2935 non-null   object 
 3   start_year         2935 non-null   int64  
 4   runtime_minutes    2816 non-null   float64
 5   genres             2927 non-null   object 
 6   averagerating      2935 non-null   float64
 7   numvotes           2935 non-null   int64  
 8   id                 2935 non-null   int64  
 9   release_date       2935 non-null   object 
 10  movie              2935 non-null   object 
 11  production_budget  2935 non-null   object 
 12  domestic_gross     2935 non-null   object 
 13  worldwide_gross    2935 non-null   object 
dtypes: float64(2), int64(3), object(9)
memory usage: 343.9+ KB
```

In [45]:
```
movies_merged.isnull().sum()
```

Out[45]:
```
movie_id               0
primary_title          0
original_title         0
start_year             0
runtime_minutes      119
genres                 8
averagerating          0
numvotes               0
id                     0
release_date           0
movie                  0
production_budget      0
domestic_gross         0
worldwide_gross        0
dtype: int64
```

We only have some missing values in runtime and genre that we will handle in the next section.

---

2 Data Cleaning & Integration

2.1 Data Cleaning

The budget and revenue data are stored as strings. They should be converted to numeric.

In [46]:
```
# Budget and revenue columns to numeric
cols_to_convert = ["production_budget", "domestic_gross", "worldwide_gross"]
for col in cols_to_convert:
    movies_merged[col] = budgets[col].replace('[\$,]', '', regex=True).astype(float)
```

Convert Release Date to DateTime format

In [47]:
```
# Date converted to DateTime format
budgets["release_date"] = pd.to_datetime(budgets["release_date"])
budgets["year"] = budgets["release_date"].dt.year
```

Drop duplicated columns

In [48]:
```
# primary_title and original-title are one and the same hence we should delete the original_title column
movies_merged.drop(columns=["original_title", "id", "primary_title"], inplace=True)
movies_merged.head()
```

Out[48]:

	movie_id	start_year	runtime_minutes	genres	averagerating	numvotes	release_date	movie	production_budg
0	tt0249516	2012	91.0	Action,Animation,Comedy	1.9	8248	Dec 31, 2012	foodfight!	425000000
1	tt0326592	2010	88.0	None	7.5	24	Jun 19, 2015	the overnight	410600000
2	tt3844362	2015	79.0	Comedy,Mystery	6.1	14828	Jun 19, 2015	the overnight	350000000
3	tt0337692	2012	124.0	Adventure,Drama,Romance	6.1	37886	Mar 22, 2013	on the road	330600000
4	tt4339118	2014	89.0	Drama	6.0	6	Mar 22, 2013	on the road	317000000

2.2 Data Integration

Reorganising columns for easier analysis

In [49]:
```
# Move 'movie' next to 'movie_id'
cols = list(movies_merged.columns)
cols.insert(cols.index("movie_id") + 1, cols.pop(cols.index("movie")))

# Reorder the dataframe
movies_merged = movies_merged[cols]

# Display the first few rows to confirm the change
movies_merged.head()
```

Out[49]:

	movie_id	movie	start_year	runtime_minutes	genres	averagerating	numvotes	release_date	production_budg
0	tt0249516	foodfight!	2012	91.0	Action,Animation,Comedy	1.9	8248	Dec 31, 2012	425000000
1	tt0326592	the overnight	2010	88.0	None	7.5	24	Jun 19, 2015	410600000
2	tt3844362	the overnight	2015	79.0	Comedy,Mystery	6.1	14828	Jun 19, 2015	350000000
3	tt0337692	on the road	2012	124.0	Adventure,Drama,Romance	6.1	37886	Mar 22, 2013	330600000
4	tt4339118	on the road	2014	89.0	Drama	6.0	6	Mar 22, 2013	317000000

Dealing with missing data

In [50]:
```
# replace missing runtime_minutes with its median
movies_merged = movies_merged.copy()
movies_merged["runtime_minutes"].fillna(movies_merged["runtime_minutes"].median(), inplace=True)
movies_merged["runtime_minutes"].isnull().sum()
```

Out[50]:
```
0
```

In [51]:
```
# genres has 8 missing values. Label them as mising
movies_merged["genres"] = movies_merged["genres"].fillna("Missing")
```

The data set that we will be using for anaysis is now prepared and cleaned.

---

3 Exploratory Data Analysis (EDA)

In this section we will explore the data further and conduct some analysis:

3.1 Preliminary Data Analysis

In [52]:
```
movies_merged.describe()
```

Out[52]:

	start_year	runtime_minutes	averagerating	numvotes	production_budget	domestic_gross	worldwide_gross
count	2935.000000	2935.000000	2935.000000	2.935000e+03	2.935000e+03	2.935000e+03	2.935000e+03
mean	2013.930494	102.888245	6.249574	6.619555e+04	5.640525e+07	6.926734e+07	1.579453e+08
std	2.559038	20.288087	1.183406	1.335852e+05	4.658747e+07	8.287622e+07	2.205795e+08
min	2010.000000	3.000000	1.600000	5.000000e+00	1.600000e+07	0.000000e+00	0.000000e+00
25%	2012.000000	90.000000	5.600000	1.490000e+02	2.500000e+07	1.820955e+07	3.098852e+07
50%	2014.000000	101.000000	6.400000	8.092000e+03	4.000000e+07	4.354910e+07	8.107957e+07
75%	2016.000000	113.000000	7.100000	7.508100e+04	7.000000e+07	8.698753e+07	1.902019e+08
max	2019.000000	280.000000	9.300000	1.841066e+06	4.250000e+08	9.366622e+08	2.776345e+09

In [53]:

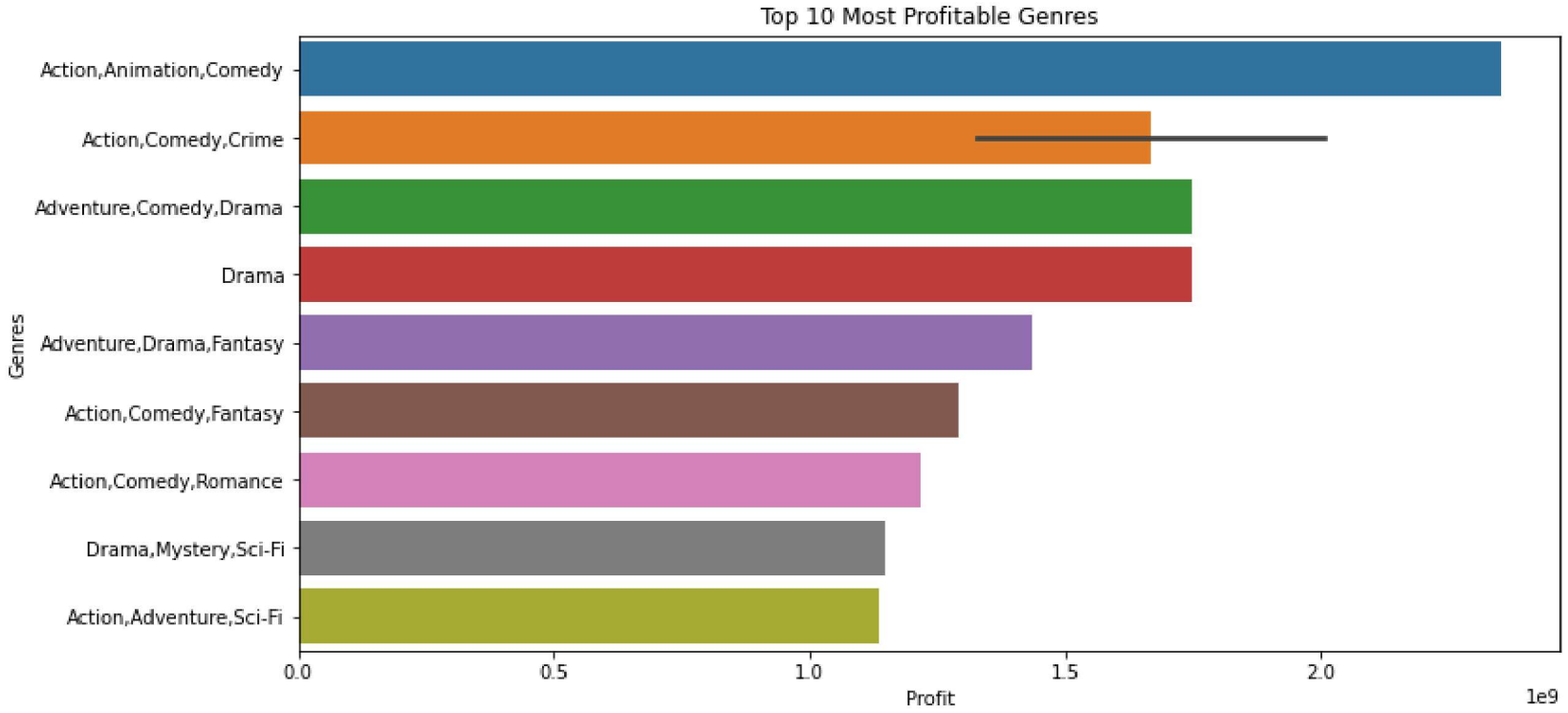
```
#dislaying the columns that will be used throughout our analysis
print(movies_merged.columns)
```

```
Index(['movie_id', 'movie', 'start_year', 'runtime_minutes', 'genres',
       'averagerating', 'numvotes', 'release_date', 'production_budget',
       'domestic_gross', 'worldwide_gross'],
      dtype='object')
```

3.2 Data Visualisation

3.2.1 Top 10 Most Profitable Genres

In [54]:
```
movies_merged["profit"] = movies_merged["worldwide_gross"] - movies_merged["production_budget"]
df_top_profitable = movies_merged.sort_values(by="profit", ascending=False).head(10)
plt.figure(figsize=(12, 6))
sns.barplot(x=df_top_profitable["profit"], y=df_top_profitable["genres"])
plt.xlabel("Profit")
plt.ylabel("Genres")
plt.title("Top 10 Most Profitable Genres")
plt.show()
```



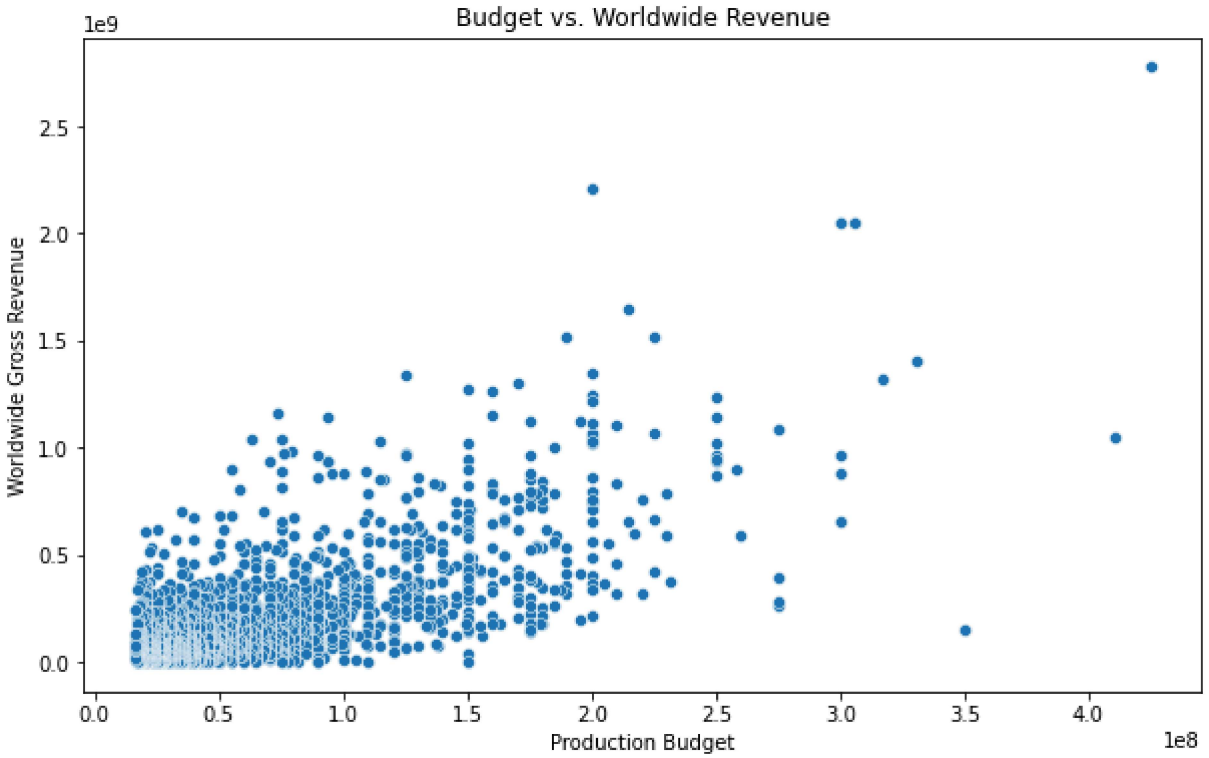
Action, Animation, and Comedy lead profitability, indicating that family-friendly and high-energy genres generate the highest returns.

Hybrid genres dominate, suggesting that movies combining elements like Adventure, Sci-Fi, and Drama attract wider audiences and greater profits.

3.2.2 Relationship Between Budget & Revenue

Visualizing the relationship between budget and revenue

In [55]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=movies_merged["production_budget"], y=movies_merged["worldwide_gross"])
plt.xlabel("Production Budget")
plt.ylabel("Worldwide Gross Revenue")
plt.title("Budget vs. Worldwide Revenue")
plt.show()
```



The key insights from this graph include:

Positive Correlation: There is a general upward trend, indicating that movies with higher production budgets tend to generate higher worldwide gross revenue.

Dense Clustering at Lower Budgets: Most data points are concentrated in the lower range of production budgets, suggesting that the majority of movies have relatively small budgets.

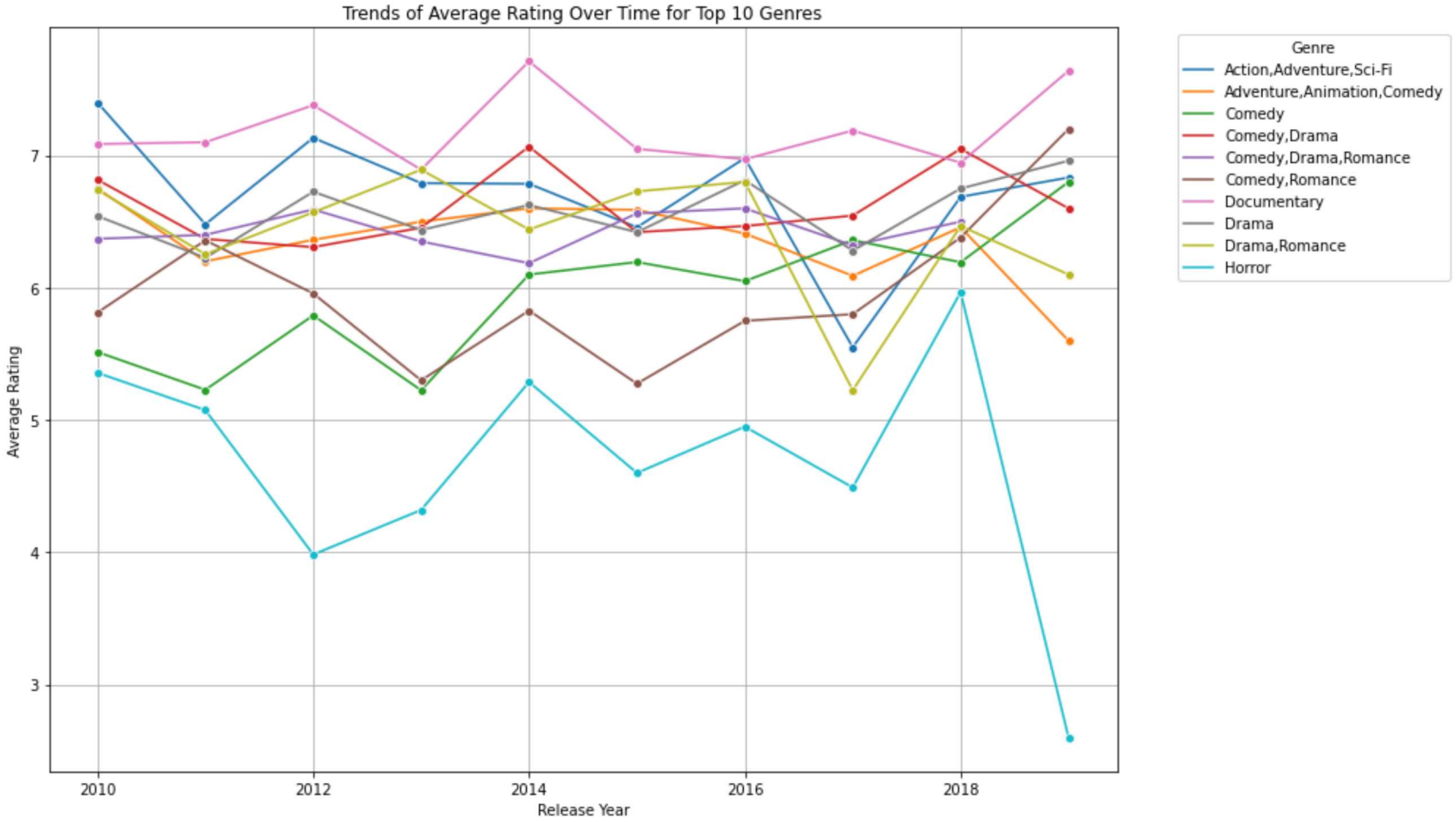
Higher Variability for Expensive Films: As production budgets increase, worldwide gross revenue becomes more spread out, meaning some high-budget films achieve massive success while others underperform.

Outliers: A few points are significantly distant from the main cluster, indicating exceptional cases where a movie either had an extremely high budget or generated unusually high revenue.

3.2.3 Popularity of top 10 genres over time

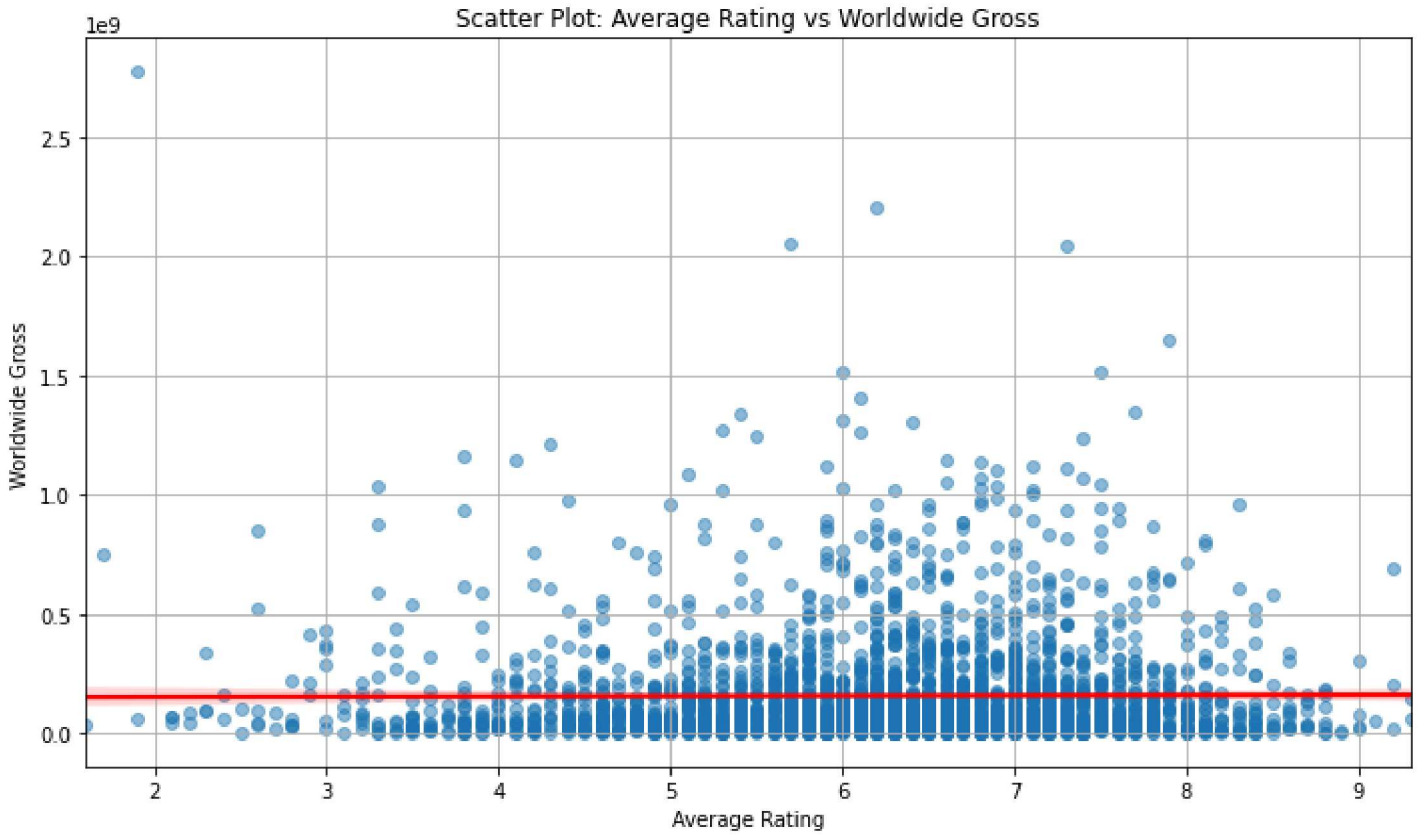
In [56]:

```
# Step 1: Drop missing data
df_plot = movies_merged.dropna(subset=['genres', 'start_year', 'averagerating'])

# Step 2: Get Top 10 Genres by frequency
top_genres = df_plot['genres'].value_counts().nlargest(10).index.tolist()

# Step 3: Filter to Top 10 Genres only
df_top_genres = df_plot[df_plot['genres'].isin(top_genres)]

# Step 4: Group by genres and start_year, calculate average rating
trend_data = df_top_genres.groupby(['genres', 'start_year'])['averagerating'].mean().reset_index()

# Step 5: Plot
plt.figure(figsize=(14,8))
sns.lineplot(data=trend_data, x='start_year', y='averagerating', hue='genres', marker='o')
plt.title('Trends of Average Rating Over Time for Top 10 Genres')
plt.xlabel('Release Year')
plt.ylabel('Average Rating')
plt.legend(title='Genre', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.tight_layout()
plt.show()
```


Trends of Average Rating Over Time for Top 10 Genres
Average Rating
Release Year
2010
2012
2014
2016
2018
3
4
5
6
7
Genre
Action,Adventure,Sci-Fi
Adventure,Animation,Comedy
Comedy
Comedy,Drama
Comedy,Drama,Romance
Comedy,Romance
Documentary
Drama
Drama,Romance
Horror

3.2.4 Correlation between Average Rating and Worldwide Gross

In [57]:
```
# Drop missing data
df_plot = movies_merged.dropna(subset=['averagerating', 'worldwide_gross'])

# Scatter plot with regression line
plt.figure(figsize=(10,6))
sns.regplot(x='averagerating', y='worldwide_gross', data=df_plot, scatter_kws={'alpha':0.5}, line_kws={"colo
plt.title('Scatter Plot: Average Rating vs Worldwide Gross')
plt.xlabel('Average Rating')
plt.ylabel('Worldwide Gross')
plt.grid(True)
plt.tight_layout()
plt.show()
```

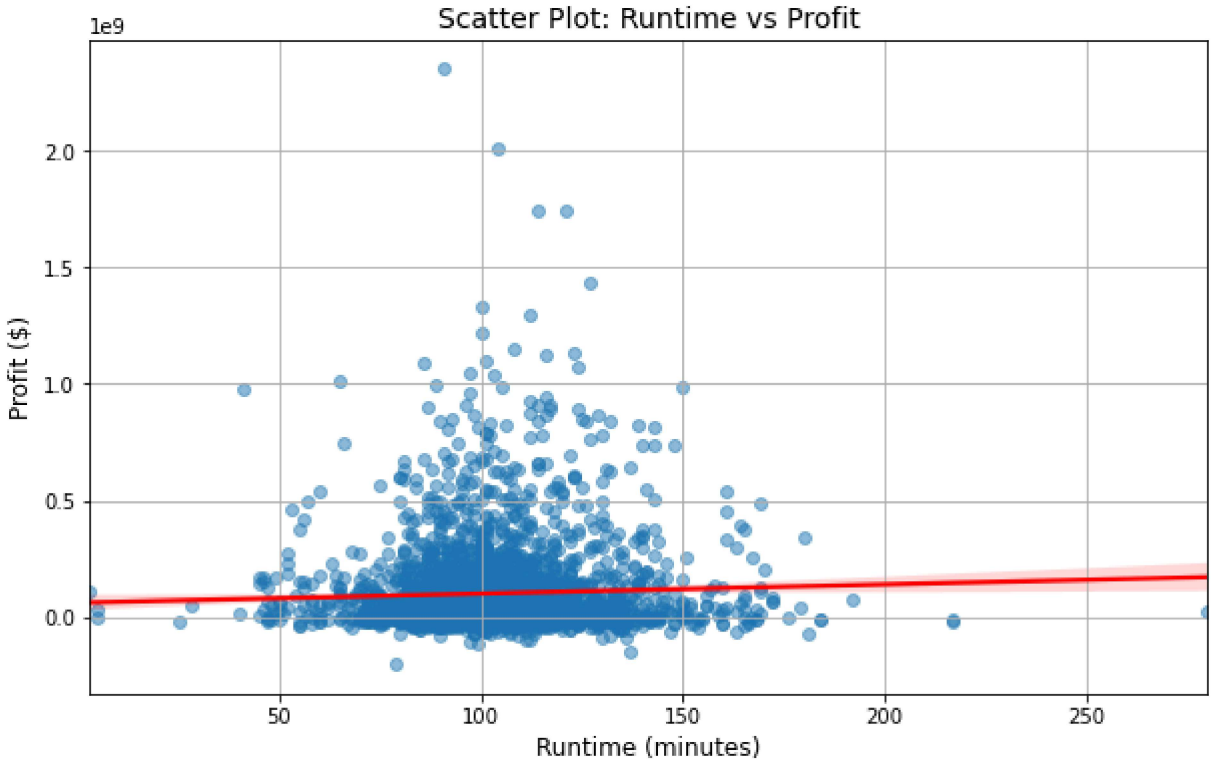


Intepretation from the Scatter Plot: Average Rating Vs Worldwide Gross

No Correlation : An increase or a decrease in average rating does not lead to an increasec in worldwide gross this is shown by the red line as it is not slanted in any way to show a correlation.

3.2.5 Relationship between Runtime and Profitability

In [58]:
```
# Scatter plot with trend line
plt.figure(figsize=(10, 6))
sns.regplot(x=movies_merged["runtime_minutes"], y=movies_merged["profit"],
            scatter_kws={"alpha": 0.5}, line_kws={"color": "red"})

plt.title("Scatter Plot: Runtime vs Profit", fontsize=14)
plt.xlabel("Runtime (minutes)", fontsize=12)
plt.ylabel("Profit ($)", fontsize=12)
plt.grid(True)
plt.show()
```



Intepretation from the Scatter Plot: Runtime vs Profit

Weak Positive Correlation The trend line (in red) has a slight upward slope, indicating a weak positive correlation between runtime and profit. However, the relationship is not strong, meaning longer movies do not necessarily lead to higher profits.
High Profit Concentration Around 90–150 Minutes Most high-profit films seem to have runtimes between 90 and 150 minutes. This suggests that standard-length movies tend to perform best at the box office.

3.3 Statistical Analysis

3.3.1 Genres that generate the highest revenue

In [59]:

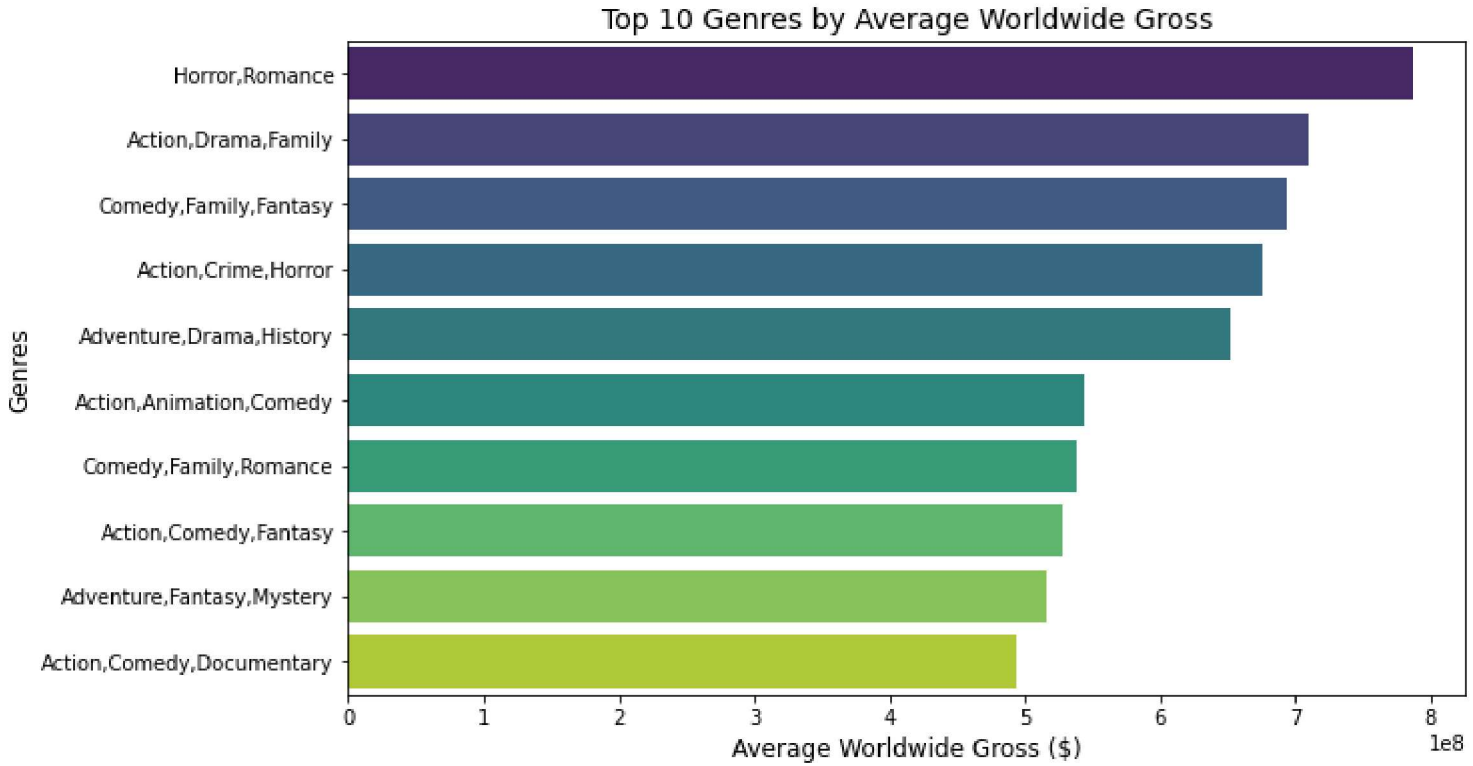
```
# Step 1: Create a Profit Column
movies_merged['profit'] = movies_merged['worldwide_gross'] - movies_merged['production_budget']

# Step 2: Get Top 10 Genres
top_genres = movies_merged['genres'].value_counts().nlargest(10).index

# Step 3: Filter to Top 10 Genres
df_top_genres = movies_merged[movies_merged['genres'].isin(top_genres)]

# Step 4: Groupby and aggregate
genre_profit_summary = df_top_genres.groupby('genres')['profit'].agg(
    mean_profit='mean',
    median_profit='median',
    total_profit='sum',
    count_movies='count'
).sort_values(by='total_profit', ascending=False)

# Step 5: Display the result
print(genre_profit_summary)
```

genres	mean_profit	median_profit	total_profit	count_movies
Drama	7.328253e+07	27428302.5	2.374354e+10	324
Documentary	8.933647e+07	39091744.5	1.125640e+10	126
Comedy	9.338591e+07	29158652.0	1.036584e+10	111
Comedy,Drama,Romance	1.312054e+08	70339667.0	1.010282e+10	77
Action,Adventure,Sci-Fi	1.548070e+08	55071636.0	9.133613e+09	59
Adventure,Animation,Comedy	1.048110e+08	42160680.0	7.860825e+09	75
Drama,Romance	9.930513e+07	45203825.0	7.845106e+09	79
Comedy,Drama	6.626980e+07	25045832.0	6.295631e+09	95
Comedy,Romance	9.591686e+07	66557976.0	5.946845e+09	62
Horror	6.422382e+07	25163815.0	4.046101e+09	63

- Highest total profit comes from *Drama* ($23.74B), probably due to its large count of 324 movies.
- Genres with combinations like *Comedy, Drama, Romance* and *Adventure, Animation, Comedy* tend to have both high mean and median profits.
- *Documentaries* surprisingly show a relatively high mean profit ($89.3M) but fewer movies (126) compared to *Drama* or *Comedy*.

3.3.2 Genres with the highest Worldwide Earning

In [60]:
```
# Calculate average worldwide gross for each genre
genre_gross =movies_merged.groupby('genres')['worldwide_gross'].mean().sort_values(ascending=False).head(10)

# Plot the top genres
plt.figure(figsize=(10, 6))
sns.barplot(x=genre_gross.values, y=genre_gross.index, palette='viridis')
plt.title('Top 10 Genres by Average Worldwide Gross', fontsize=14)
plt.xlabel('Average Worldwide Gross ($)', fontsize=12)
plt.ylabel('Genres', fontsize=12)
plt.show()
```



From the visualization above , we note that, Adventure,Drama and sport earns the most globally.This indicates that it has a broad appeal across international markets.

3.3.3 🎭 ANOVA: Research Question

Is there a significant difference in the mean worldwide gross revenues among the top 10 movie genres?

📊 Hypothesis

Null Hypothesis (H_0)

The **mean worldwide gross revenue** is the **same** across all top 10 genres.

Alternative Hypothesis (H_1)

At least **one genre** has a significantly **different** mean worldwide gross revenue compared to the others.

Conducting the ANOVA test

In [61]:

```
from scipy import stats

# Step 1: Get Top 10 Genres
top_genres = movies_merged['genres'].value_counts().nlargest(10).index

# Step 2: Filter to Top 10 Genres
df_top_genres = movies_merged[movies_merged['genres'].isin(top_genres)]

# Step 3: Prepare the data for ANOVA
anova_data = [df_top_genres[df_top_genres['genres'] == genre]['worldwide_gross'].dropna() for genre in top_g

# Step 4: Perform ANOVA test
f_stat, p_value = stats.f_oneway(*anova_data)

# Step 5: Display results
print(f'ANOVA F-statistic: {f_stat}')
print(f'ANOVA p-value: {p_value}')

if p_value < 0.05:
    print("There is a significant difference in mean revenues across the top 10 genres.")
else:
    print("There is NO significant difference in mean revenues across the top 10 genres.")
```

```
ANOVA F-statistic: 2.9123566169829176
ANOVA p-value: 0.0020601086434680394
There is a significant difference in mean revenues across the top 10 genres.
```

Result Intepretation

Since the p-value (0.002) is less than 0.05, you reject the null hypothesi. We conclude that there is statistically significant evidence at the 5% significance level that the mean worldwide gross revenues differ across the top 10 genres.

3.3.4 🎬 T-Test: Research Question

Does a movie's production budget significantly impact its worldwide gross earnings?

🧐 Hypothesis

Null Hypothesis (H_0)

There is **no significant difference** in worldwide gross earnings between high-budget and low-budget movies.

Alternative Hypothesis (H_1)

There **is** a significant difference in worldwide gross earnings between high-budget and low-budget movies.

🔢 Mathematical Representation

- H_0 : $\mu_{\text{high budget}} = \mu_{\text{low budget}}$
- H_1 : $\mu_{\text{high budget}} \neq \mu_{\text{low budget}}$

🛠️ Assumption Testing

Before running the t-test, we need to check if the key assumptions hold:

i) Normality Assumption

In [62]:

```
from scipy.stats import shapiro
# Define high and low-budget categories based on median production budget.
median_budget = movies_merged["production_budget"].median()
high_budget = movies_merged[movies_merged["production_budget"] >= median_budget]["worldwide_gross"]
low_budget = movies_merged[movies_merged["production_budget"] < median_budget]["worldwide_gross"]
# Check normality
shapiro_high = shapiro(high_budget.sample(500, random_state=42))  # Sample to avoid errors in large datasets
shapiro_low = shapiro(low_budget.sample(500, random_state=42))

print(f"Shapiro-Wilk Test for High-Budget Movies: W={shapiro_high.statistic}, p={shapiro_high.pvalue}")
print(f"Shapiro-Wilk Test for Low-Budget Movies: W={shapiro_low.statistic}, p={shapiro_low.pvalue}")
```

```
Shapiro-Wilk Test for High-Budget Movies: W=0.8038601875305176, p=3.728469421423422e-24
Shapiro-Wilk Test for Low-Budget Movies: W=0.6991506814956665, p=6.352923672069884e-29
```

Shapiro-Wilk test checks whether a dataset is normally distributed. It uses two key values

1. W (Test Statistic) -which measures how well the data fits a normal distribution in values closer to 1 means more normal.
2. p-value: where if p > 0.05, we fail to reject the null hypothesis, meaning the data is likely normal. And if p ≤ 0.05, we reject the null, meaning the data deviates significantly from normality.

Based on the output above

High-Budget Movies:

1. W = 0.8038 - which is far from 1, suggesting deviation from normality
2. p = 9.02×10^{-30} - extremely small, way below 0.05

Since the p-value is far below the standard significance level (α = 0.05), we reject the null hypothesis. This means that the worldwide gross earnings of high-budget movies is not normally distributed.

Low-Budget Movies:

1. W = 0.6991 - far from 1, indicating a stronger deviation from normality
2. p = 2.52×10^{-34} - even smaller, confirming extreme non-normality

The p-value is much smaller than 0.05, so we reject the null hypothesis. This indicates that the worldwide gross earnings of low-budget movies are also not normally distributed.

i) Homogeneity of Variance (Levene's Test)

A standard t-test assumes equal variances in both groups. Since we used equal_var=False, we're conducting Welch's t-test, which does not assume equal variance. However, we can still check using Levene's test:

In [63]:
```
from scipy.stats import levene

levene_test = levene(high_budget, low_budget)
print(f"Levene's Test: W={levene_test.statistic}, p={levene_test.pvalue}")
```

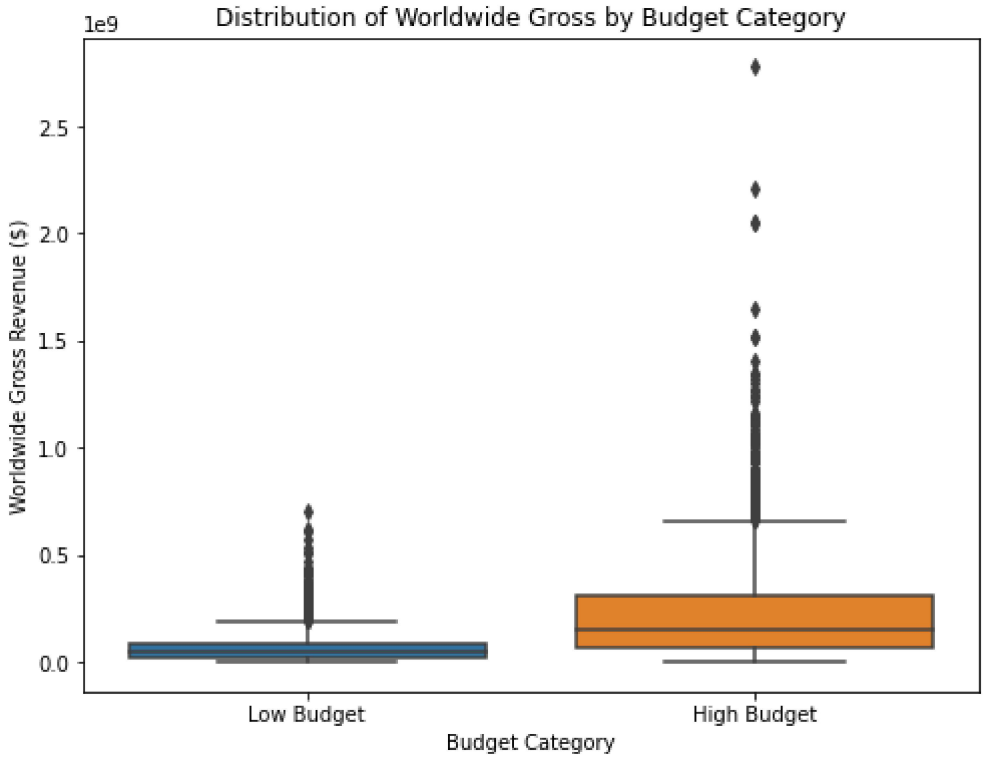
```
Levene's Test: W=344.6193308998376, p=7.992600038109332e-73
```

Since the p-value is much smaller than the standard significance level (α = 0.05), we reject the null hypothesis. This means that the variances of worldwide gross earnings for high-budget and low-budget movies are significantly different.

Boxplot for Budget Categories

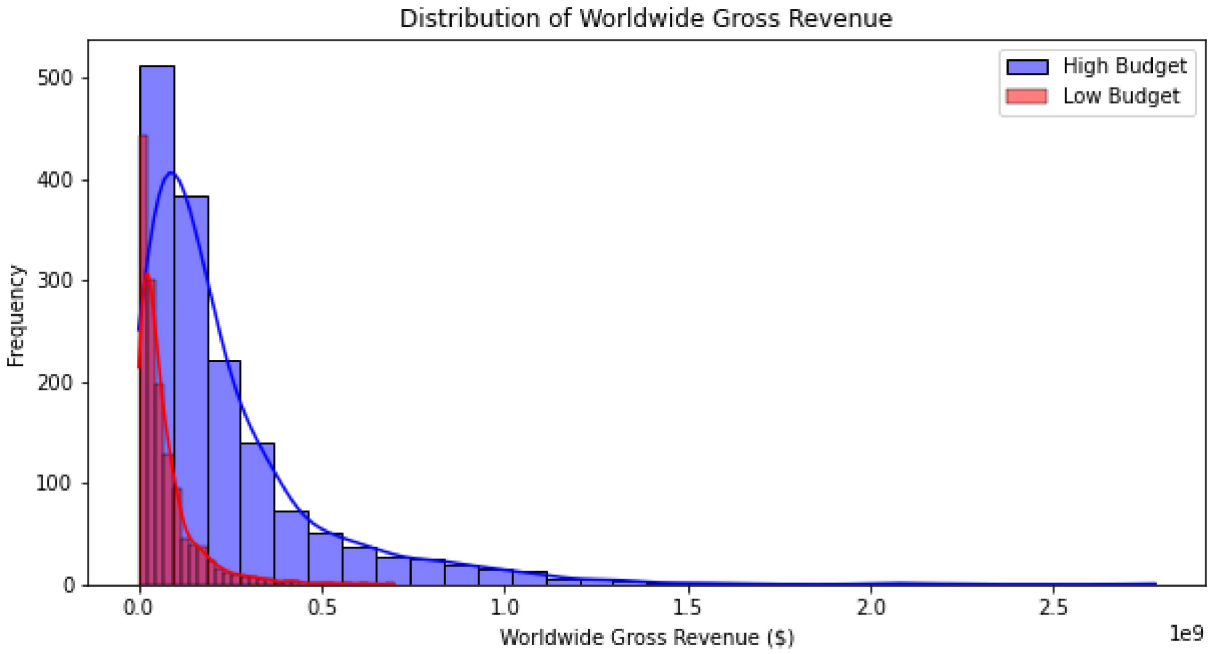
This boxplot shows the spread of revenue for high vs. low-budget movies.

In [64]:
```
movies_merged["budget_category"] = movies_merged["production_budget"] >= median_budget
plt.figure(figsize=(8, 6))
sns.boxplot(x=movies_merged["budget_category"], y=movies_merged["worldwide_gross"])
plt.xticks([0, 1], ["Low Budget", "High Budget"])
plt.ylabel("Worldwide Gross Revenue ($)")
plt.xlabel("Budget Category")
plt.title("Distribution of Worldwide Gross by Budget Category")
plt.show()
```



Histogram to Check Distribution

In [65]:
```
plt.figure(figsize=(10,5))
sns.histplot(high_budget, bins=30, color='blue', kde=True, label="High Budget")
sns.histplot(low_budget, bins=30, color='red', kde=True, label="Low Budget")
plt.legend()
plt.title("Distribution of Worldwide Gross Revenue")
plt.xlabel("Worldwide Gross Revenue ($)")
plt.ylabel("Frequency")
plt.show()
```



The distribution is skewed to the left.

Conducting the T-test

In [66]:

```
t_stat, p_value = stats.ttest_ind(high_budget, low_budget, equal_var=False)
print(f"T-Statistic: {t_stat}, P-Value: {p_value}")
```

```
T-Statistic: 23.73527989820849, P-Value: 6.111096029847255e-109
```

Result Intepretation

Since p-value ≈ 0, it is far below the standard significance level (α = 0.05). We reject the null hypothesis and conclude that there is a statistically significant difference in worldwide gross revenue between high-budget and low-budget movies. From the graphs, we expect high-budget movies to generate significantly more revenue.

Business Recommendations:

The findings show a statistically significant difference in revenue between high-budget and low-budget movies. Incase of decision-making within the industry the following may be recommended:

1. **For investment decisions, high investments may be made in high-budget productions for greater returns**

The data confirms that higher budgets correlate with higher worldwide earnings.*

Studios should prioritize big-budget productions for higher revenue potential.*

2. **Optimize Budget Allocation**

While statistical evidence show that high budgets drive revenue, spending should be strategic. Investments should be focused on other key revenue-driving factors, such as:

o **Star Power:** Well-known actors and directors boost box office performance.

o **Marketing & Distribution:** A strong promotional strategy may maximize reach and engagement.

o **Genre Selection:* Certain genres** (e.g., action, superhero films) tend to perform better globally.

3. **There is a need for further analysis for revenue depends on more than just budget.**

- Further analysis should include:

Marketing spend vs. revenue impact

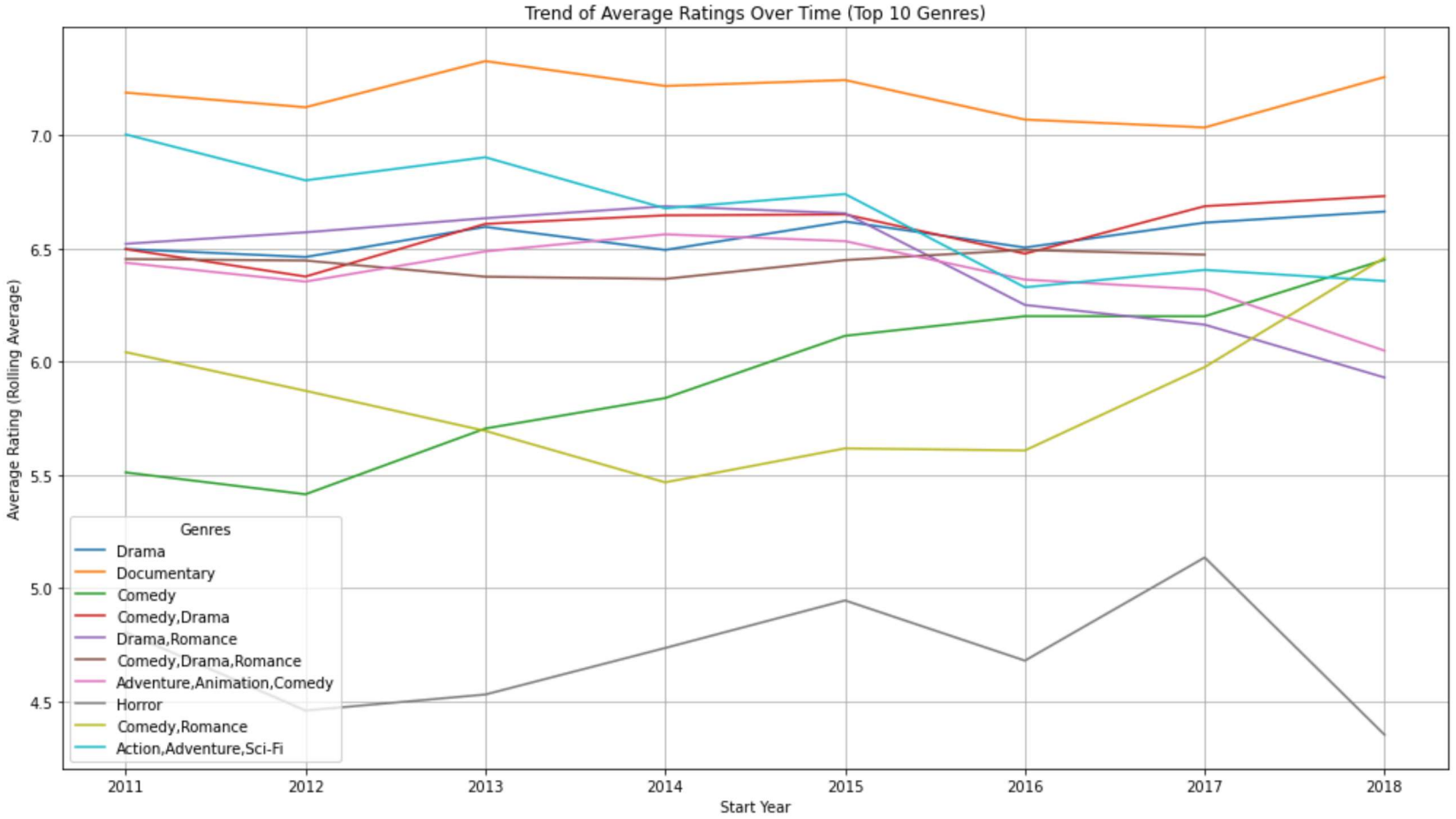
Impact of streaming platforms on box office success

Regional market differences in movie revenue

3.3.4 Trend Analysis

In [67]:

```
# Get Top 10 genres
top_genres = movies_merged['genres'].value_counts().nlargest(10).index

# Filter data to only top 10 genres
df_top_genres = movies_merged[movies_merged['genres'].isin(top_genres)].dropna(subset=['start_year', 'averag

# Group by start_year and genres to get average ratings
df_trend = df_top_genres.groupby(['start_year', 'genres'], as_index=False)['averagerating'].mean()

# Compute rolling average OUTSIDE the loop and store it as a new dataframe
df_trend['rolling_avg'] = df_trend.groupby('genres')['averagerating'].transform(lambda x: x.rolling(window=3

# Plot trend for each genre
plt.figure(figsize=(14,8))

for genre in top_genres:
    genre_data = df_trend[df_trend['genres'] == genre]
    plt.plot(genre_data['start_year'], genre_data['rolling_avg'], label=genre)

plt.title('Trend of Average Ratings Over Time (Top 10 Genres)')
plt.xlabel('Start Year')
plt.ylabel('Average Rating (Rolling Average)')
plt.legend(title='Genres')
plt.grid(True)
plt.tight_layout()
plt.show()
```



Result Intepretation

Documentary has been having the most consistent average ratings as compared to the other genres from 2011 to 2018. The ratings for Action, Adventure, Scifi have been gradually reducing over time. Drama, Romance; Comedy, Drama; and Adventure, Animation, Comedy have been having consistent ratings over time.

3.3.5 🎬 Regression Analysis: Research Question

What drives a movie's worldwide box office success?

🔍 Hypothesis

Null Hypothesis (H_0)

No significant relationship exists between a movie's worldwide gross revenue and its key factors—production budget, average rating, number of votes, runtime, and release year.

Alternative Hypothesis (HA):

At least one of these factors plays a significant role in determining a movie's worldwide box office revenue.

✅ Assumption Testing

Before running the regression, we need to ensure that the assumptions of linear regression hold:

- 📈 **Linearity:** The relationship between production budget and worldwide gross should be linear.
- 📊 **Normality of Residuals:** Residuals (errors) should follow a normal distribution.
- 📏 **Homoscedasticity:** Residuals should have constant variance (no clear patterns in residual plots).
- 📎 **Independence:** Observations should be independent (no duplicate entries or autocorrelation).

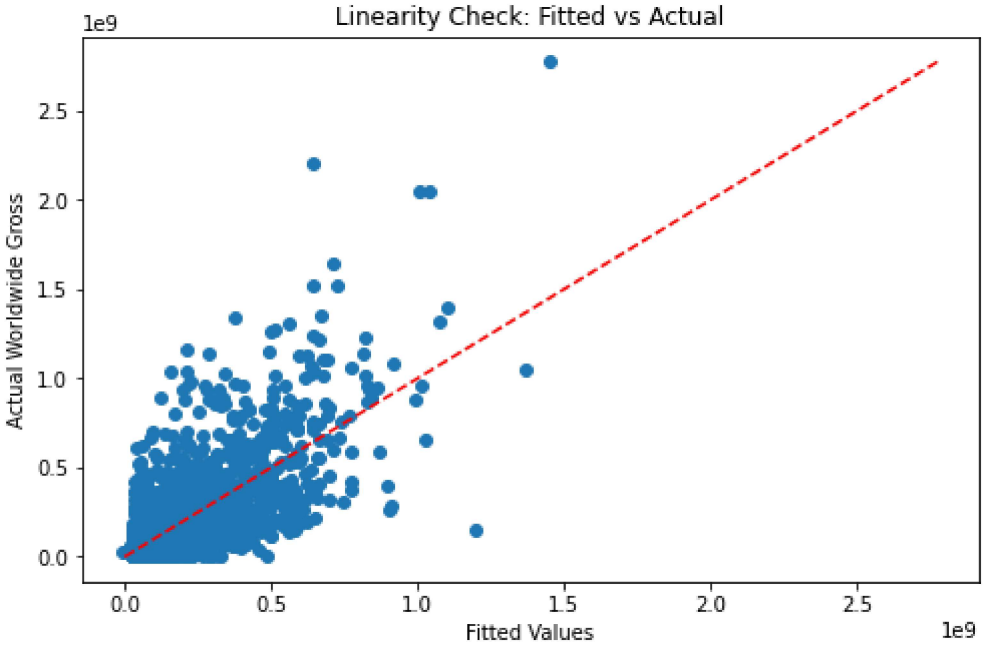
To verify these assumptions, we use scatter plots, histograms, and residual plots.

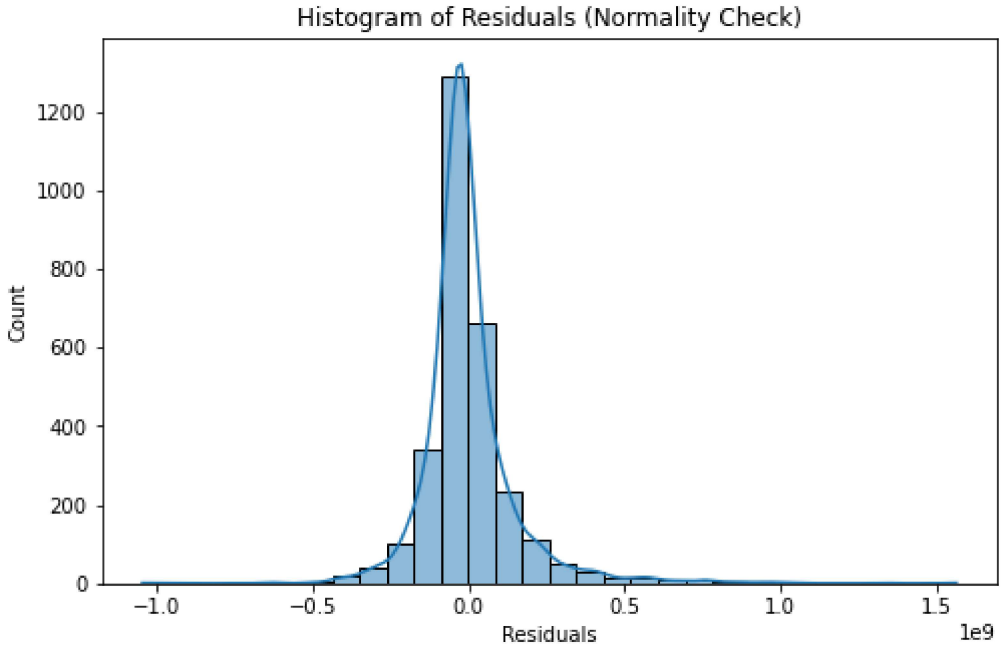
In [68]:

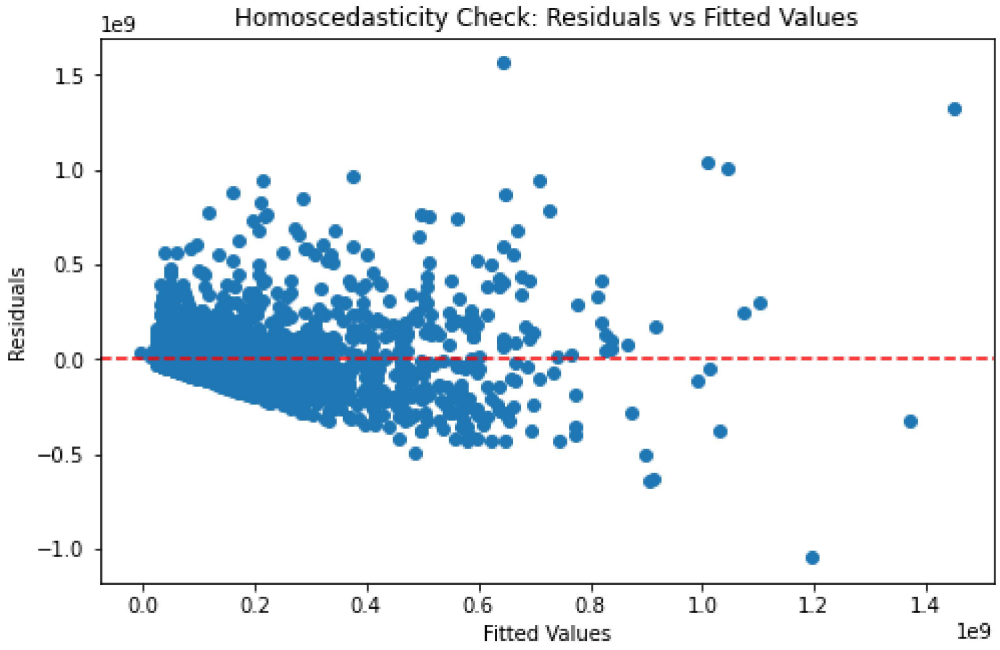
```
import statsmodels.api as sm

# Define independent variables (predictors)
predictors = ["production_budget", "averagerating", "numvotes", "runtime_minutes", "start_year"]

# Drop missing values for a clean regression model
movies_merged_clean = movies_merged.dropna(subset=["worldwide_gross"] + predictors)

# Define X and y
X = movies_merged_clean[predictors]
y = movies_merged_clean["worldwide_gross"]

# Add a constant term for the intercept
X = sm.add_constant(X)

# Run OLS regression
model = sm.OLS(y, X).fit()

# Print results
print(model.summary())

# --- 1. Scatterplot for Linearity ---
plt.figure(figsize=(8, 5))
plt.scatter(model.fittedvalues, y)
plt.plot([y.min(), y.max()], [y.min(), y.max()], color='red', linestyle='--')
plt.xlabel('Fitted Values')
plt.ylabel('Actual Worldwide Gross')
plt.title('Linearity Check: Fitted vs Actual')
plt.show()

# --- 2. Histogram of Residuals ---
residuals = model.resid

plt.figure(figsize=(8, 5))
sns.histplot(residuals, bins=30, kde=True)
plt.xlabel('Residuals')
plt.title('Histogram of Residuals (Normality Check)')
plt.show()

# --- 3. Residuals vs Fitted Values (Homoscedasticity Check) ---
plt.figure(figsize=(8, 5))
plt.scatter(model.fittedvalues, residuals)
plt.axhline(0, color='red', linestyle='--')
```

```
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Homoscedasticity Check: Residuals vs Fitted Values')
plt.show()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        worldwide_gross   R-squared:                       0.517
Model:                            OLS   Adj. R-squared:                  0.516
Method:                 Least Squares   F-statistic:                     627.1
Date:                Sun, 30 Mar 2025   Prob (F-statistic):               0.00
Time:                        20:43:06   Log-Likelihood:                -59483.
No. Observations:                2935   AIC:                         1.190e+05
Df Residuals:                    2929   BIC:                         1.190e+05
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -1.334e+10   2.41e+09     -5.535      0.000   -1.81e+10   -8.62e+09
production_budget     3.5053      0.066     52.740      0.000       3.375       3.636
averagerating  -2.671e+06   2.54e+06     -1.051      0.293   -7.65e+06    2.31e+06
numvotes          32.3918     24.121      1.343      0.179     -14.904      79.688
runtime_minutes  -1.32e+05   1.52e+05     -0.871      0.384   -4.29e+05    1.65e+05
start_year       6.619e+06    1.2e+06      5.528      0.000    4.27e+06    8.97e+06
==============================================================================
Omnibus:                     1558.788   Durbin-Watson:                   1.027
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            24802.307
Skew:                           2.150   Prob(JB):                         0.00
Kurtosis:                      16.577   Cond. No.                     6.23e+10
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.23e+10. This might indicate that there are
strong multicollinearity or other numerical problems.
```


Linearity Check: Fitted vs Actual
1e9
Actual Worldwide Gross
Fitted Values
1e9
0.0
0.5
1.0
1.5
2.0
2.5


Histogram of Residuals (Normality Check)
Count
Residuals
1e9
0
200
400
600
800
1000
1200
−1.0
−0.5
0.0
0.5
1.0
1.5



Interpretation of the OLS Regression Results

R-squared (0.517): The model explains 51.7% of the variance in worldwide gross, indicating a moderate level of explanatory power. However, 48.3% of the variation remains unexplained, suggesting that other factors influence worldwide gross revenue.

F-statistic (627.1) & p-value (0.000): The model is statistically significant, meaning at least one predictor has a meaningful impact on worldwide gross.

Intercept (const = -13.34 billion, p = 0.000): If a movie had a production budget of zero, the expected worldwide gross would be approximately -13.34 billion. While unrealistic, this result indicates that other factors contribute to revenue.

Production Budget (coef = 3.51, p = 0.000): For every $1 increase in production budget, worldwide gross increases by $3.51. This strong positive relationship makes production budget the most significant predictor in the model.

Average Rating (coef = -2.67 million, p = 0.293): Not statistically significant (p > 0.05), meaning film ratings do not have a reliable impact on worldwide gross in this model.

Number of Votes (coef = 32.39, p = 0.179): Not statistically significant, suggesting that a higher number of votes does not reliably predict worldwide gross.

Runtime Minutes (coef = -132,000, p = 0.384): Not statistically significant, indicating that longer movies do not strongly influence box office success.

Start Year (coef = 6.62 million, p = 0.000): Statistically significant ($p < 0.05$), showing that newer movies tend to earn more, likely due to inflation and evolving market trends.

Overall, **production budget is the strongest predictor** of worldwide gross, followed by **start year**. Meanwhile, **average rating, number of votes, and runtime minutes** do not significantly impact revenue in this model.

To improve the model, it would be useful to check for **multicollinearity** and **normalize residuals** to enhance predictive accuracy.

4️⃣ Conclusion & Recommendations 🎥

4.1 🔍 Key Insights

🎭 1. Impact of Production Budget on Revenue

- 💰 **Bigger budgets, bigger earnings** – High-budget films consistently generate higher revenues, confirmed by **T-test and Regression analysis**.
- 📊 **Regression model findings:**
  - **Production budget** is the strongest predictor of box office success, explaining **51-56% of revenue variance** ($R^2 = 0.517$).
  - Other factors**—marketing, star power, and distribution strategy**—likely influence revenue but were not included in the model.

🎞️ 2. Profitable Genres & Market Trends

- 🎭 **Drama** leads in **total profit (**
  $23.74B) * *duetoahighnumberofreleases. – * *🎬Winninggenrecombinations * * : *Action, Animation, Comedy * and * Comedy, Drama * yield * *highmeanandmedianprofits * *. – * *📈Documentaries * *area * *hiddengem * *– Despite, theyboastan * *averageprofitof$
  89.3M.
- 🎟️ **Audience trends (2011–2018)**:
  - 📊 **Stable**: Documentaries continue to hold **high audience ratings**.
  - 🔽 **Declining**: Action, Adventure, and Sci-Fi ratings **dropped over time**.
  - 🎭 **Consistent**: Drama, Romance, and Comedy maintained steady ratings.

📉 3. Regression Model Insights

- **Significant revenue drivers:**
  - 💵 **Production Budget** ($p < 0.001$)
  - 📅 **Start Year** ($p < 0.001$)
- 🚫 **Not statistically significant:**
  - Average rating, number of votes, and runtime have **no strong impact** on revenue.
- 🔮 **What this means:** Success is influenced by factors **beyond film quality—marketing, star power, and distribution strategy** likely play a crucial role.

4.2 🚀 Business Recommendations

🎯 1. Smart Budgeting for Maximum ROI

- Since **higher production budgets lead to higher earnings**, studios should **strategically allocate budgets** to **maximize impact while maintaining cost-efficiency**.

📽️ 2. Diversify Investments

- A **balanced strategy**: Invest in **big-budget blockbusters** for major revenue but **also fund high-potential mid-budget films** for optimized risk-return balance.

🎬 3. Optimize Movie Runtime for Maximum Profit

- Studios should aim for a **runtime between 90–150 minutes**, as this range aligns with the **most profitable films**. We have seen profitability is influenced more by other factors so runtime should be optimized alongside these for the best financial outcomes.

🎭 4. Genre Optimization & Market Trends

- Focus on **highly profitable genres** (*Action, Adventure, Comedy*).
- Leverage **market trends**: The **Documentary genre** shows promise despite fewer releases.

🌟 5. Beyond Budget: Marketing & Star Power Matter

- **A big budget alone won't guarantee success –** investing in **A-list actors** and **strong marketing campaigns** can significantly boost revenue.

🕰️ 6. Timing is Everything

- **Strategic release timing** is key – Analyze **seasonal trends and competitive landscapes** to **maximize box office earnings**.

🔬 7. Future Research & Model Refinement

- Enhance future predictive models by incorporating **marketing spend, cast reputation, distribution reach, and audience demographics** for **even more accurate revenue predictions**.

📢 **Final Takeaway:** While **production budget is the biggest driver of success**, true box office dominance comes from **a mix of smart budgeting, strategic marketing, strong genre choices, and perfect release timing**. 🚀

Type *Markdown* and LaTeX: α^2