Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
```

Loading the Dataset

```
df = pd.read_csv('E:/Projects/ratings.csv')

# Displaying the first few rows of the dataset
df.head()
```

	Game Name	Developer	Genre	Rating
0	Candy Crush Saga	King	Puzzle	4.6
1	Clash of Clans	Supercell	Strategy	4.5
2	Among Us	InnerSloth	Party	4.4
3	Pokémon GO	Niantic	Augmented Reality	4.3
4	PUBG Mobile	Tencent Games	Battle Royale	4.2

Examining the Dataset

Examine the dataset to understand its structure and look for any immediate issues

```
# Displaying basic information about the dataset
df.info()

# Displaying summary statistics
df.describe(include='all')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Game Name  101 non-null    object 
 1   Developer  101 non-null    object 
 2   Genre      101 non-null    object 
 3   Rating     101 non-null    float64
```

```
dtypes: float64(1), object(3)
memory usage: 3.3+ KB
```

	Game Name	Developer	Genre	Rating
count	101	101	101	101.000000
unique	101	81	24	NaN
top	Candy Crush Saga	Supercell	Strategy	NaN
freq	1	5	17	NaN
mean	NaN	NaN	NaN	4.398020
std	NaN	NaN	NaN	0.175488
min	NaN	NaN	NaN	4.100000
25%	NaN	NaN	NaN	4.300000
50%	NaN	NaN	NaN	4.400000
75%	NaN	NaN	NaN	4.500000
max	NaN	NaN	NaN	4.900000

From the basic dataset information and summary statistics, we can conclude that the dataset is well-structured and doesn't have any missing values.

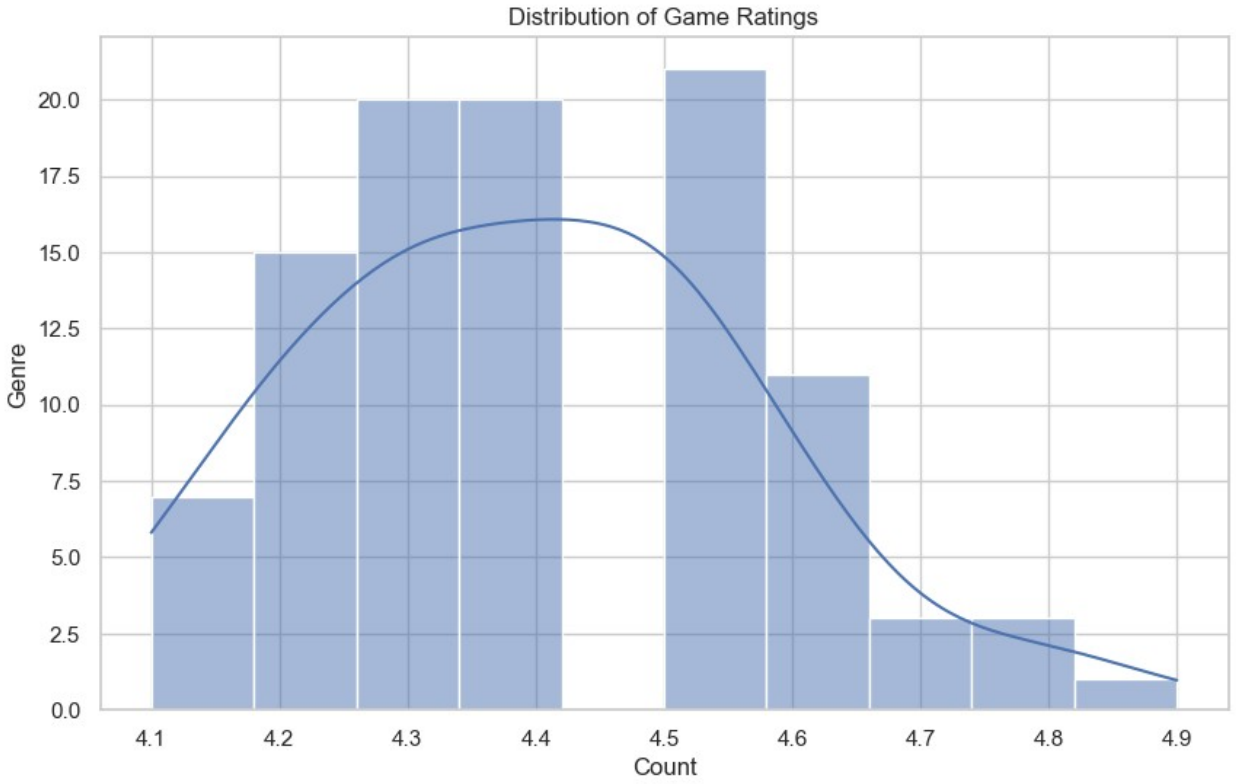
Exploratory Data Analysis

This stage visualizes some key aspects of the dataset

i.) Distribution of Game Ratings

Created a histogram to visualize the distribution of game ratings.

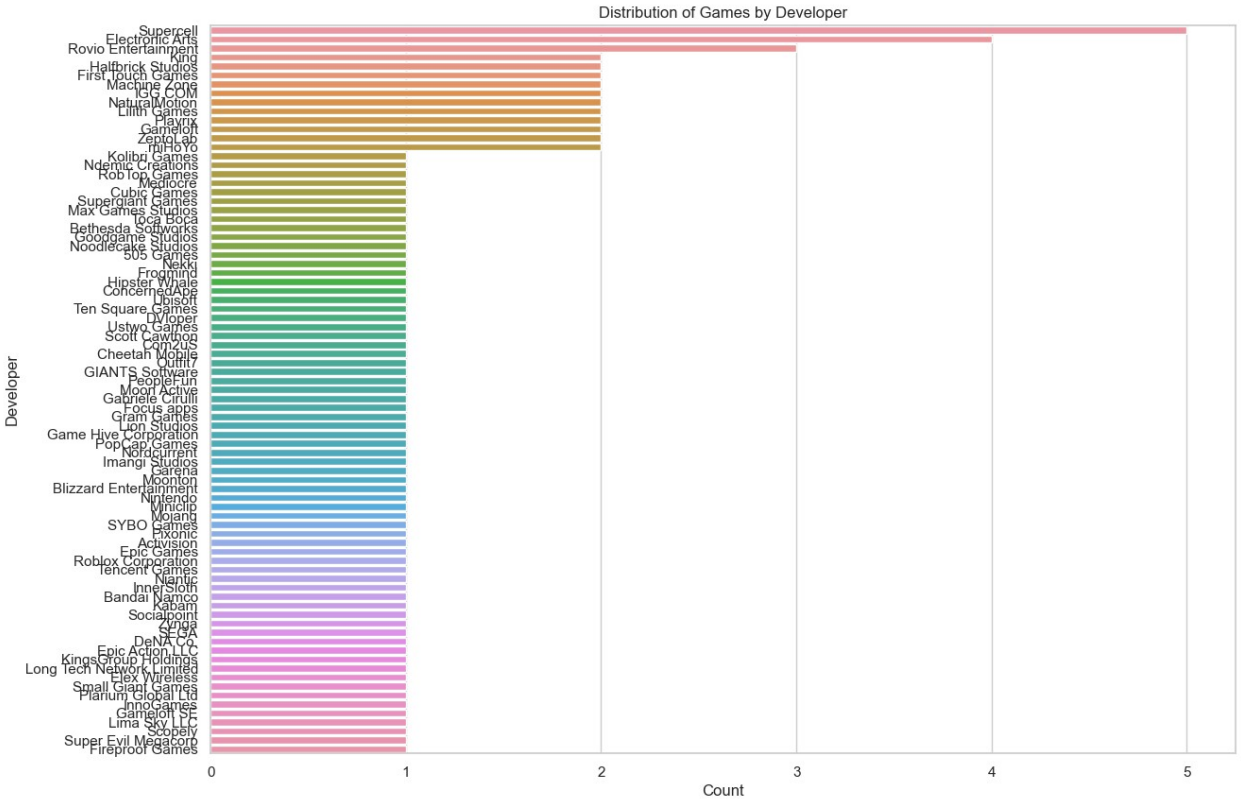
```
# Setting the style
sns.set(style='whitegrid')

plt.figure(figsize=(10, 6))
sns.histplot(df['Rating'], bins=10, kde=True)
plt.title('Distribution of Game Ratings')
plt.xlabel('Count')
plt.ylabel('Genre')
plt.show()
```



ii.) Distribution of Developers

Created a bar plot to visualize the number of games developed by each developer.

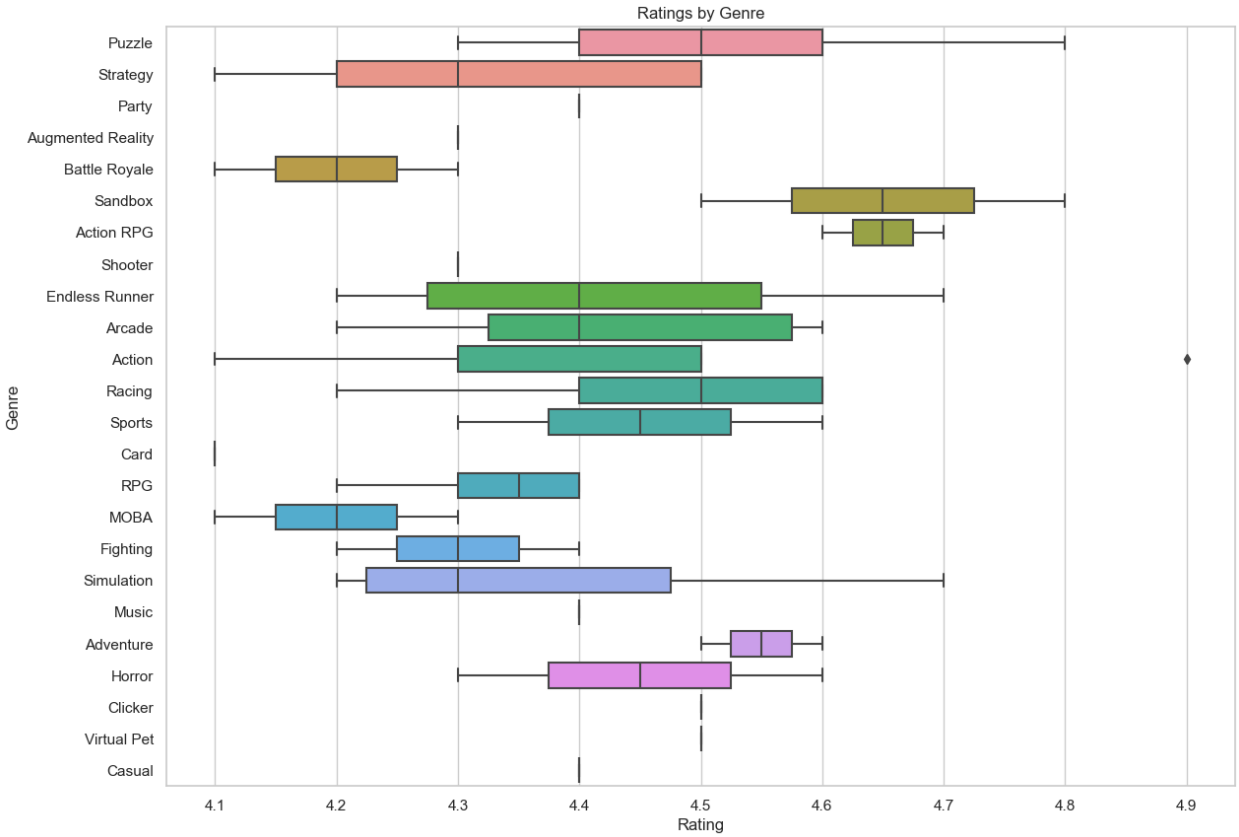
```
plt.figure(figsize=(14, 10))
sns.countplot(y='Developer', data=df,
order=df['Developer'].value_counts().index)
plt.title('Distribution of Games by Developer')
plt.xlabel('Count')
plt.ylabel('Developer')
plt.show()
```



iii.) Ratings by Genre

Created a box plot to visualize the distribution of ratings for each genre.

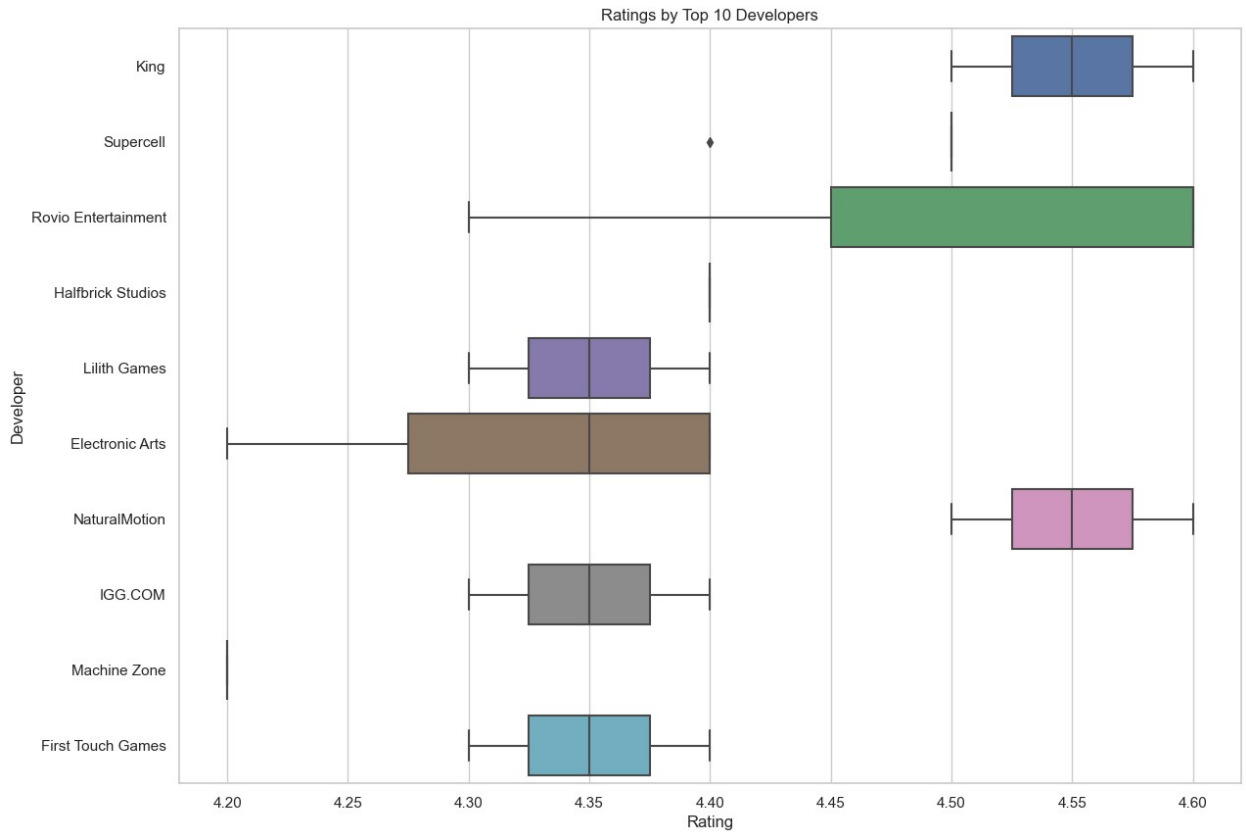
```
plt.figure(figsize=(14, 10))
sns.boxplot(x='Rating', y='Genre', data=df)
plt.title('Ratings by Genre')
plt.xlabel('Rating')
plt.ylabel('Genre')
plt.show()
```



iv.) Ratings by Developer

Created a box plot to visualize the distribution of ratings for each developer.

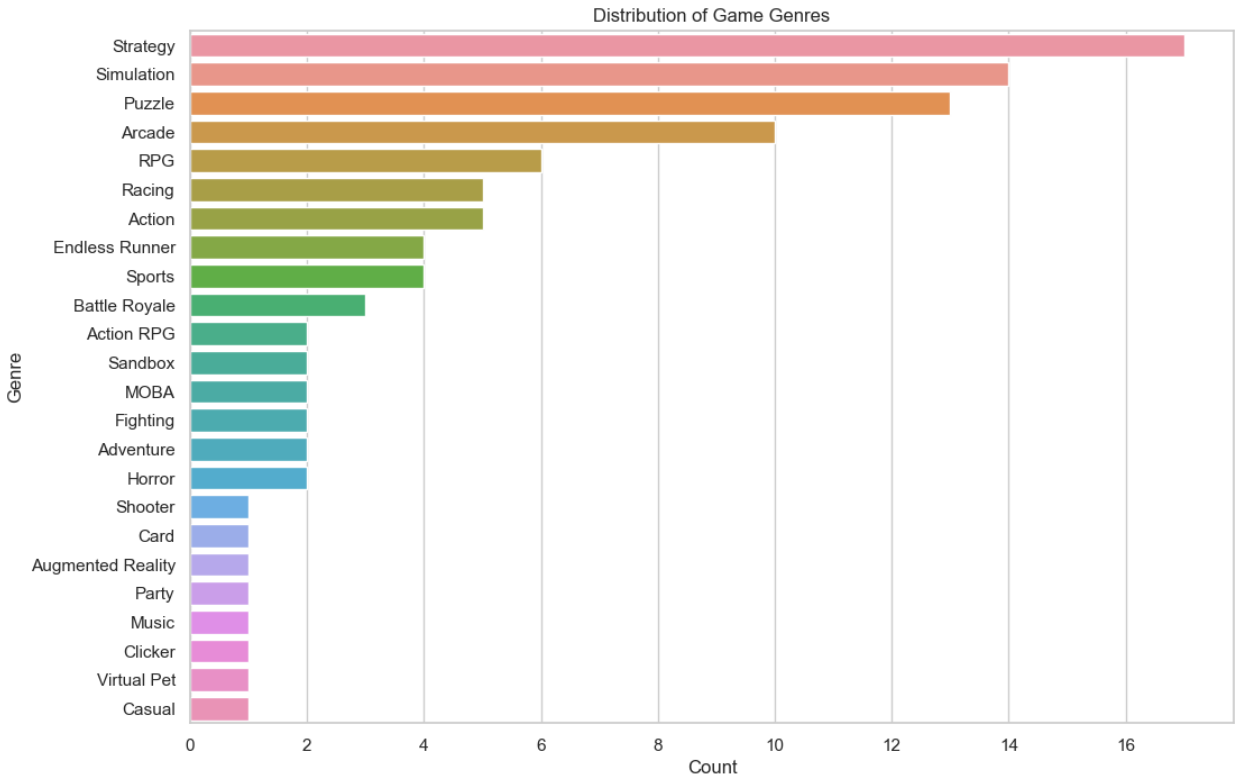
```
# Selecting top 10 developers
top_developers = df['Developer'].value_counts().index[:10]
df_top_developers = df[df['Developer'].isin(top_developers)]
plt.figure(figsize=(14, 10))
sns.boxplot(x='Rating', y='Developer', data=df_top_developers)
plt.title('Ratings by Top 10 Developers')
plt.xlabel('Rating')
plt.ylabel('Developer')
plt.show()
```



v.) Distribution of Genres

Created a bar plot to visualize the distribution of game genres.

```
plt.figure(figsize=(12, 8))
sns.countplot(y='Genre', data=df,
order=df['Genre'].value_counts().index)
plt.title('Distribution of Game Genres')
plt.xlabel('Count')
plt.ylabel('Genre')
plt.show()
```



Summary of Exploratory Data Analysis (EDA)

1. Distribution of Game Ratings:

The histogram of game ratings shows that the majority of games have ratings around 4.3 to 4.5, indicating that most games are well-received by users.

2. Distribution of Developers:

The count plot of developers illustrates the distribution of games by each developer. Some developers such as Supercell, Electronic Arts and Rovio Entertainment have a larger portfolio of games compared to others.

3. Ratings by Genre:

The box plot of ratings by genre displays the distribution of ratings for each genre. It shows variations in ratings across different genres, with some genres having higher median ratings than others.

4. Ratings by Developer:

The box plot of ratings by developer focuses on the top 10 developers with the highest number of games. It reveals differences in ratings among these developers, indicating variations in the quality of games produced by different developers.

5. Distribution of Genres:

The bar plot of game genres reveals that certain genres, such asStrategy, Simulation and Puzzle are more common compared to others like Casual or Virtual Pet.

Insights and Implications:

Game Quality:

Most games in the dataset have ratings above 4.0, indicating that the majority of mobile games are well-received by users.

Genre Preferences:

Certain genres like Strategy, Simulation and Puzzle are more popular among developers and players, suggesting potential areas for game development or investment.

Developer Performance:

There are variations in the quality of games produced by different developers. Some developers consistently produce highly-rated games, while others may need to focus on improving the quality of their games.

Strategic Decisions:

Game developers can use these insights to make strategic decisions about game development, genre selection, and quality improvement efforts.

Correlation Analysis

1. Correlation between Rating and Genre

Grouping the data by genre to calculate the average rating for each genre. Then examine if there's a correlation between the average ratings and the genres.

```
# Importing the label encoder
from sklearn.preprocessing import LabelEncoder

# Encoding the 'Genre' column
label_encoder = LabelEncoder()
df['Genre_encoded'] = label_encoder.fit_transform(df['Genre'])

# Grouping the data by encoded genre and calculating the average
rating for each genre
genre_avg_rating = df.groupby('Genre_encoded')
['Rating'].mean().reset_index()

# Calculating the correlation coefficient between average ratings and
```

```
genres
correlation_genre_rating = genre_avg_rating.corr().loc['Rating',
'Genre_encoded']
print(f"Correlation between Rating and Genre:
{correlation_genre_rating}")
```

```
Correlation between Rating and Genre: 0.023430088867641623
```

2. Correlation between Rating and Developer

Grouping the data by developer and calculate the average rating for each developer. Then examine if there's a correlation between the average ratings and the developers.

```
# Encoding the 'Developer' column
df['Developer_encoded'] = label_encoder.fit_transform(df['Developer'])

# Grouping the data by encoded developer and calculating the average
rating for each developer
developer_avg_rating = df.groupby('Developer_encoded')
['Rating'].mean().reset_index()

# Calculating the correlation coefficient between average ratings and
developers
correlation_developer_rating =
developer_avg_rating.corr().loc['Rating', 'Developer_encoded']
print(f"Correlation between Rating and Developer:
{correlation_developer_rating}")
```

```
Correlation between Rating and Developer: 0.2270012288130561
```

Summary of the Correlation Analysis

i.) Correlation between Rating and Genre:

Correlation Coefficient: 0.023

Interpretation:

There is a very weak positive correlation between game ratings and genres. This suggests that the genre of a game has little to no influence on its rating.

ii.) Correlation between Rating and Developer:

Correlation Coefficient: 0.227

Interpretation:

There is a weak positive correlation between game ratings and developers. This indicates that developers with higher average ratings tend to produce games with higher ratings overall.

Statistical Analysis

1. Hypothesis Testing

i.) Hypothesis Testing for Genres

Using ANOVA to assess whether there are statistically significant differences in ratings between different genres.

```
# Importing necessary library for ANOVA
from scipy.stats import f_oneway

# Grouping the data by genre
genre_groups = df.groupby('Genre')['Rating'].apply(list)

# Performing ANOVA
f_statistic, p_value = f_oneway(*genre_groups)

# Printing the results
print("Results of ANOVA for Genres:")
print(f"F-statistic: {f_statistic}")
print(f"P-value: {p_value}")
```

```
Results of ANOVA for Genres:
F-statistic: 1.7167871950489102
P-value: 0.0415594922256249
```

ii.) Hypothesis Testing for Developers

Using ANOVA to test for statistically significant differences in ratings between different developers.

```
# Grouping the data by developer
developer_groups = df.groupby('Developer')['Rating'].apply(list)

# Performing ANOVA
f_statistic_dev, p_value_dev = f_oneway(*developer_groups)

# Printing the results
print("\nResults of ANOVA for Developers:")
print(f"F-statistic: {f_statistic_dev}")
print(f"P-value: {p_value_dev}")
```

```

Results of ANOVA for Developers:
F-statistic: 3.3226264041717535
P-value: 0.0018405592492906026
```

2. Regression Analysis

Building a regression model to analyze how various independent variables collectively influence game ratings. This step will involve using multiple linear regression since there are multiple independent variables.

```
# Encoding categorical variables ('Developer' and 'Genre')
df['Developer_encoded'] = label_encoder.fit_transform(df['Developer'])
df['Genre_encoded'] = label_encoder.fit_transform(df['Genre'])

# Selecting independent variables and dependent variable (ratings)
X = df[['Developer_encoded', 'Genre_encoded']]
y = df['Rating']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Building and training the multiple linear regression model
regression_model = LinearRegression()
regression_model.fit(X_train, y_train)

# Evaluating the model's performance
train_score = regression_model.score(X_train, y_train)
test_score = regression_model.score(X_test, y_test)

print(f"Training R^2 Score: {train_score}")
print(f"Testing R^2 Score: {test_score}")
```

```
Training R^2 Score: 0.12183668048461438
Testing R^2 Score: -0.10464038981450452
```

Summary of Statistical Analysis

Hypothesis Testing

Genres:

F-statistic: 1.717

P-value: 0.042

Interpretation:

The p-value of 0.042 is less than the significance level of 0.05, indicating statistically significant evidence to reject the null hypothesis. Therefore, there are significant differences in ratings between different genres.

Developers:

F-statistic: 3.323

P-value: 0.002

Interpretation

The p-value of 0.002 is less than the significance level of 0.05, indicating statistically significant evidence to reject the null hypothesis. Therefore, there are significant differences in ratings between different developers.

Regression Analysis:

Model Performance:

Training R^2 Score: 0.122

Testing R^2 Score: -0.105

Interpretation:

The R^2 score measures the proportion of the variance in the dependent variable (ratings) that is predictable from the independent variables (developers and genres) in the model. The low R^2 scores suggest that the model does not explain much of the variability in the ratings. The negative testing R^2 score indicates poor performance of the model on unseen data, suggesting potential overfitting or lack of generalization.

Conclusion

This comprehensive analysis reveals that both the genre and the developer significantly influence mobile game ratings. However, these factors alone do not fully determine a game's success. To achieve higher ratings, developers must look beyond just the type of game or the developer's reputation.

Recommendations for Stakeholders

i.) Focus on High-Rating Genres and Developers:

Prioritize investing in and developing games within genres like Action RPG and Adventure, which have shown higher average ratings. Partner with or learn from top developers known for their high-rated games.

ii.) Expand Data Utilization:

Incorporate additional data such as user reviews, game mechanics, and marketing strategies into your analysis. This holistic approach will provide deeper insights into what truly drives game ratings and player satisfaction.

iii.) Innovate Beyond Basics:

While genre and developer reputation are important, innovation in gameplay, storylines, and user engagement is crucial. Focus on unique and engaging game features that differentiate your product in the crowded market.

iv.) Monitor Market Trends:

Stay ahead of market trends by continuously monitoring shifts in genre popularity and emerging successful developers. Adapting to these trends can help maintain competitiveness and relevance in the dynamic mobile gaming industry.

By adopting these recommendations, stakeholders can strategically enhance their game development process, leading to higher-rated and more successful mobile games.