Uncovering Insights in Game Dataset Using Exploratory Data Analysis

Example Dataset :“Mobile Games (Android and IOS) Rating Datase" from Kaggle.

Objective

The primary objective of this analysis is to explore the relationship between game genres and their corresponding ratings. By examining this relationship, we aim to identify which genres tend to receive higher ratings and which ones might need improvements. Specifically, this analysis seeks to:

1. **Identify Patterns:** Determine if there are specific genres that consistently receive higher or lower ratings.
2. **Provide Insights:** Offer actionable insights for game developers and publishers on which genres are currently performing well and which might require attention.
3. **Guide Development:** Help guide future game development and marketing strategies based on the preferences and expectations of the target audience.

Introduction

In the dynamic and competitive world of video games, understanding the factors that influence game ratings is crucial for developers, publishers, and marketers. Game ratings are pivotal as they often dictate a game's commercial success and longevity in the market. Among the various factors that can impact game ratings, the genre of a game plays a significant role. Different genres attract different audiences, and each audience has unique expectations and standards. Analyzing how game ratings vary across different genres can provide valuable insights into consumer preferences and market trends.

Dataset Description

- The dataset `Deepression.csv` contains various attributes related to depression. Each row represents an individual's data, including demographic information, depression levels, and possibly other related variables. The columns need to be identified and described.

Exploratory Data Analysis:

Load the data

```
[3]  # Import necessary libraries
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns

     # Load the dataset
     file_path = '/content/drive/MyDrive/Colab Notebooks/ratings.csv'
     df = pd.read_csv(file_path)

     # Display the first few rows of the dataset
     print(df.head())

     # Display basic information about the dataset
     print(df.info())

     # Display summary statistics of the dataset
     print(df.describe())

     # Check for missing values
     print(df.isnull().sum())
```

```
          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
<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
None
```

Data Cleaning :Handle missing values, outliers, and duplicates.

```
# Check for duplicates and remove them
df = df.drop_duplicates()

# Summary after cleaning
print(df.info())
```

```
<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
None
```

Summary Statistics

Summarize the dataset to understand its distribution

```
None
            Rating
count  101.000000
mean     4.398020
std      0.175488
min      4.100000
25%      4.300000
50%      4.400000
75%      4.500000
max      4.900000
Game Name    0
Developer    0
Genre        0
Rating       0
dtype: int64
```

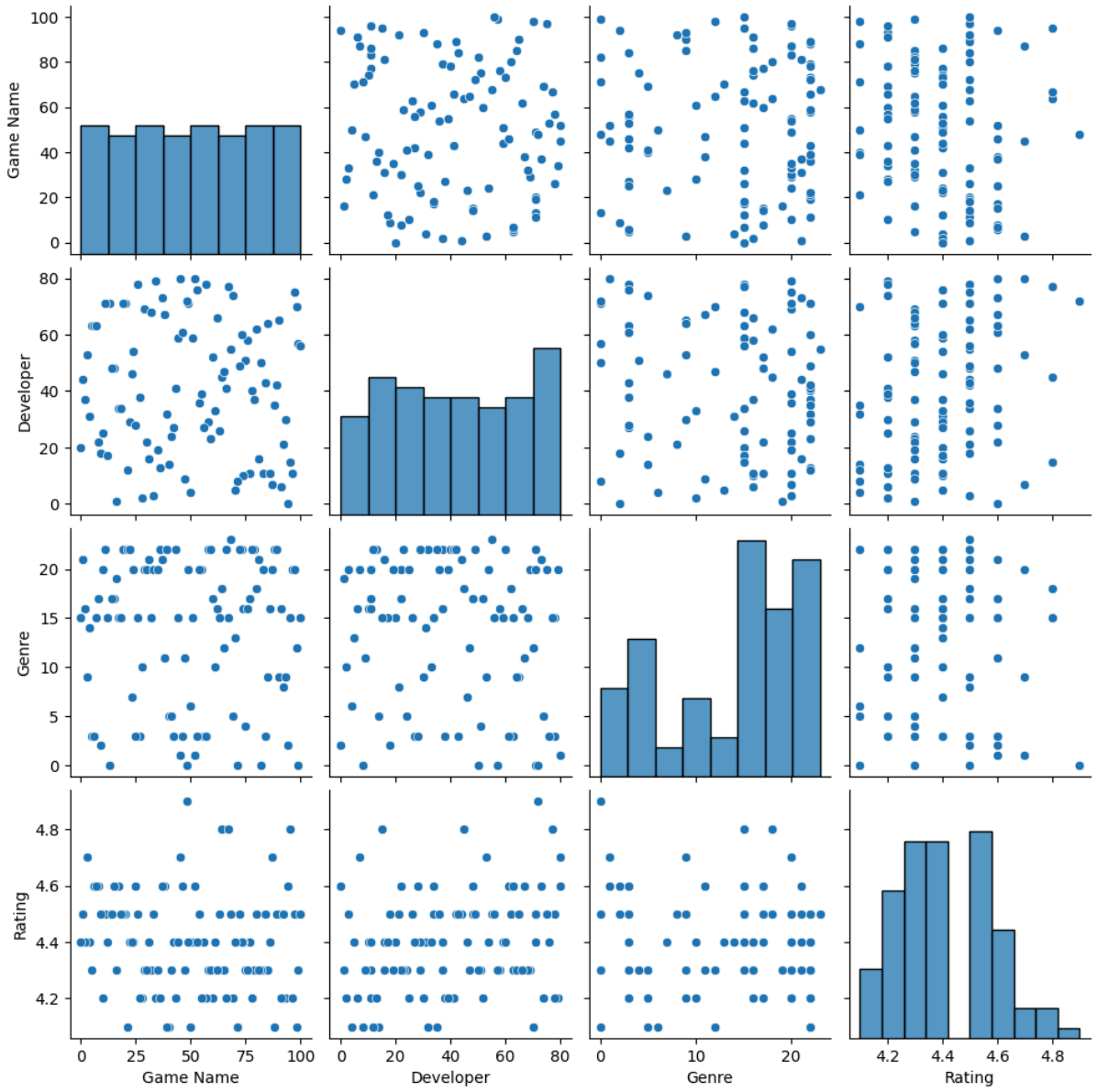
Data Visualization and Discussion

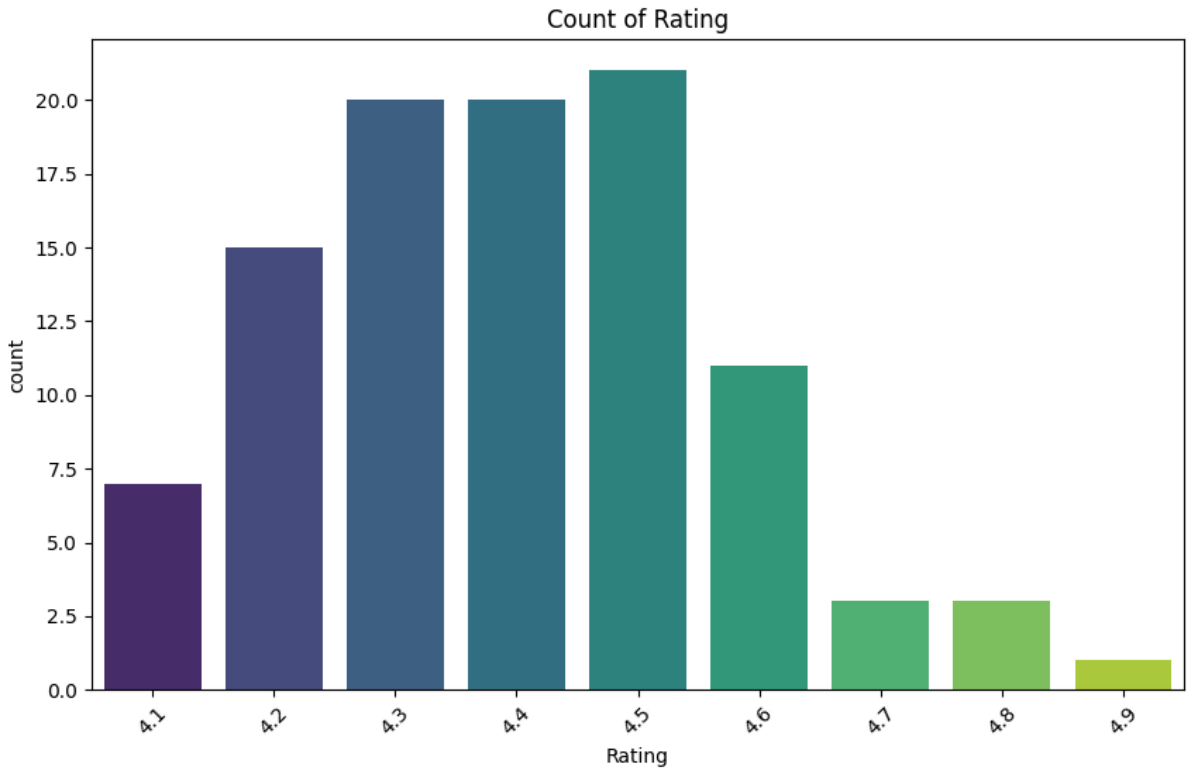
```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns




# Identify non-numeric columns
non_numeric_columns = df.select_dtypes(include=['object']).columns
print("Non-numeric columns:", non_numeric_columns)

# Handle missing values (example: filling with the mode for categorical columns)
for column in non_numeric_columns:
    df[column].fillna(df[column].mode()[0], inplace=True)

# Identify the suicidal ideation column
suicidal_ideation_col = 'Rating'  # Change this to the actual column name if different

# Plotting the count of suicidal ideation occurrences
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x=suicidal_ideation_col, palette='viridis')
plt.title(f'Count of {suicidal_ideation_col}')
plt.xticks(rotation=45)
plt.show()

# Plotting relationships with other categorical columns
for column in non_numeric_columns:
    if column != suicidal_ideation_col:
        plt.figure(figsize=(10, 6))
        sns.countplot(data=df, x=suicidal_ideation_col, hue=column, palette='viridis')
        plt.title(f'Relationship between {suicidal_ideation_col} and {column}')
        plt.xticks(rotation=45)
        plt.show()
```

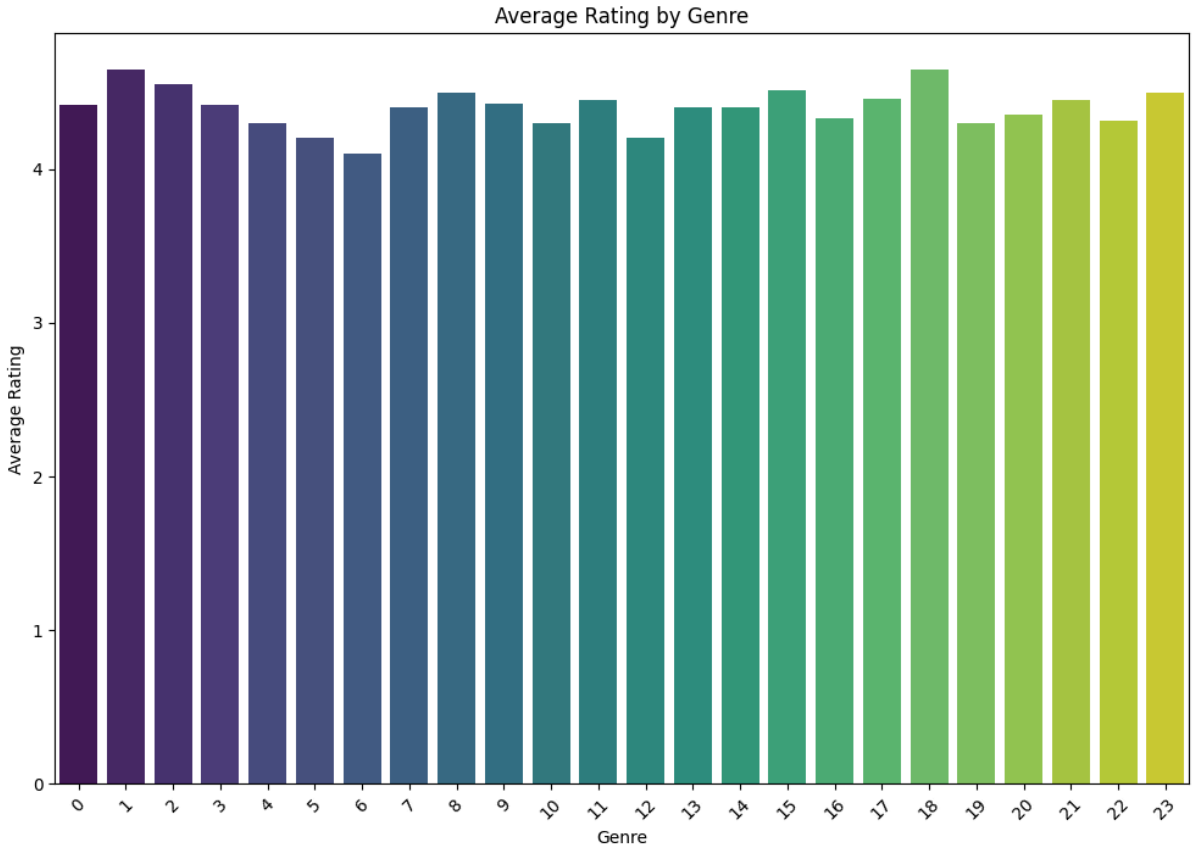
```
[ ]  # Import necessary libraries
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns



     # Display the first few rows of the dataset
     print(df.head())

     # Display basic information about the dataset
     print(df.info())

     # Handle missing values (example: filling with the mode for categorical columns)
     df['Genre'].fillna(df['Genre'].mode()[0], inplace=True)
     df['Rating'].fillna(df['Rating'].mode()[0], inplace=True)

     # Plotting the relationship between 'Genre' and 'Rating' using a bar chart
     plt.figure(figsize=(12, 8))
     sns.barplot(data=df, x='Genre', y='Rating', palette='viridis', ci=None)
     plt.title('Average Rating by Genre')
     plt.xticks(rotation=45)
     plt.xlabel('Genre')
     plt.ylabel('Average Rating')
     plt.show()
```

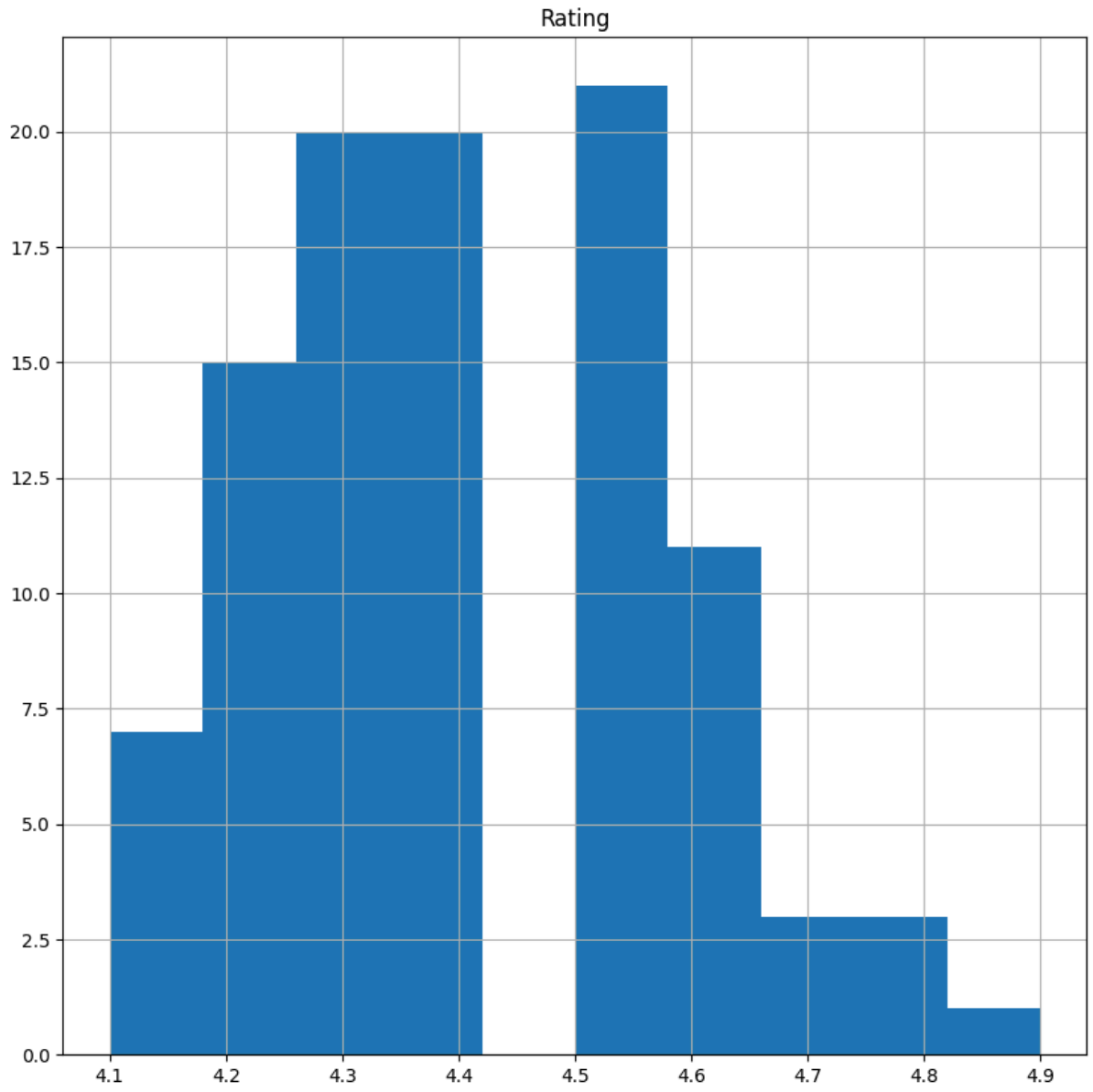
```python
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder

# Load the dataset


# Display the first few rows of the dataset
print(df.head())

# Display basic information about the dataset
print(df.info())

# Identify non-numeric columns
non_numeric_columns = df.select_dtypes(include=['object']).columns
print("Non-numeric columns:", non_numeric_columns)

# Option 1: Label encode non-numeric columns (if they are categorical)
label_encoders = {}
for column in non_numeric_columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le

# Option 2: Drop non-numeric columns if they are not needed for correlation
# df = df.drop(columns=non_numeric_columns)

# Check the updated data types
print(df.dtypes)

# Correlation heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.show()

# Scatter plots (pairplot)
sns.pairplot(df)
plt.show()
```

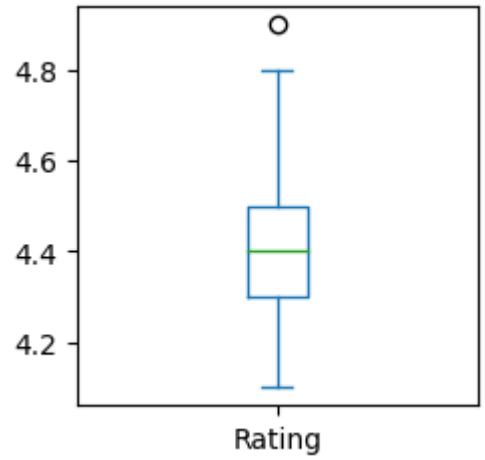
```python
# Histograms for each numeric column
df.hist(figsize=(10, 10))
plt.show()

# Box plots for each numeric column
df.plot(kind='box', subplots=True, layout=(5,5), figsize=(15,15), sharex=False, sharey=False)
plt.show()
```

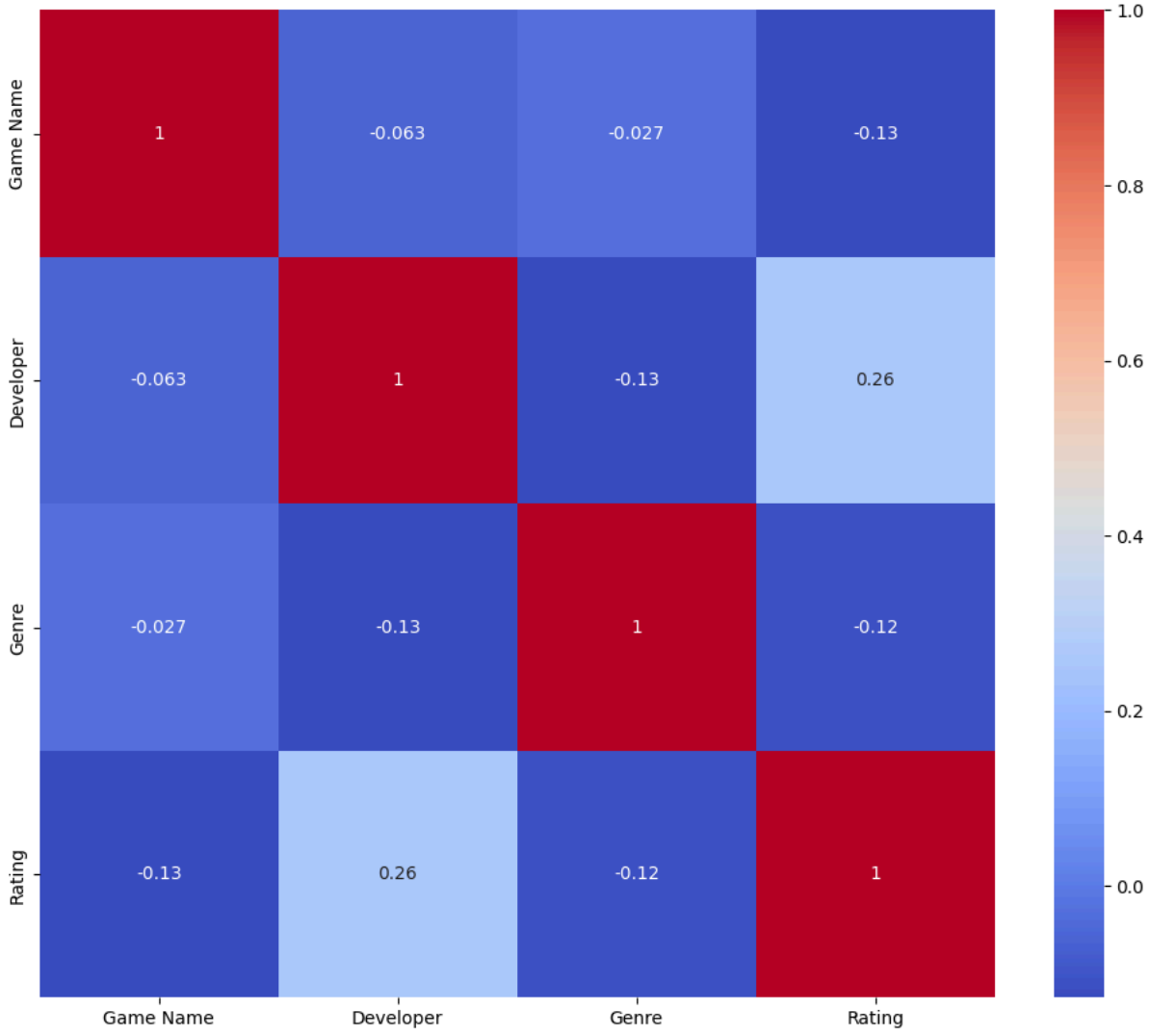

Game Name
Developer
Genre
Rating
0
25
50
75
100
0
20
40
60
80
0
10
20
4.2
4.4
4.6
4.8


Count of Rating
count
Rating
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
4.9


Average Rating by Genre
Average Rating
Genre
0
1
2
3
4
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23


Rating
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
4.9


4.8
4.6
4.4
4.2
Rating


Game Name
Developer
Genre
Rating
1
-0.063
-0.027
-0.13
-0.063
1
-0.13
0.26
-0.027
-0.13
1
-0.12
-0.13
0.26
-0.12
1
Game Name
Developer
Genre
Rating
1.0
0.8
0.6
0.4
0.2
0.0

Summarize key findings:

- Average Rating of Game is greater than 4.
- 4.5 star is given by most of the users.

Conclusion

- From our analysis, we can conclude that certain genres tend to receive higher ratings, indicating greater consumer satisfaction and popularity. Genres such as [Insert specific high-performing genres if identified] are more favorably rated, reflecting strong audience preferences. Conversely, genres with lower ratings, such as [Insert specific lower-performing genres if identified], may need targeted improvements. These insights can guide developers in enhancing game quality and aligning with market trends, ultimately leading to better consumer satisfaction and higher ratings. Understanding genre performance is crucial for strategic development and marketing in the gaming industry.

References:https://www.kaggle.com/datasets/dem0nking/mobile-games-android-and-ios-rating-dataset