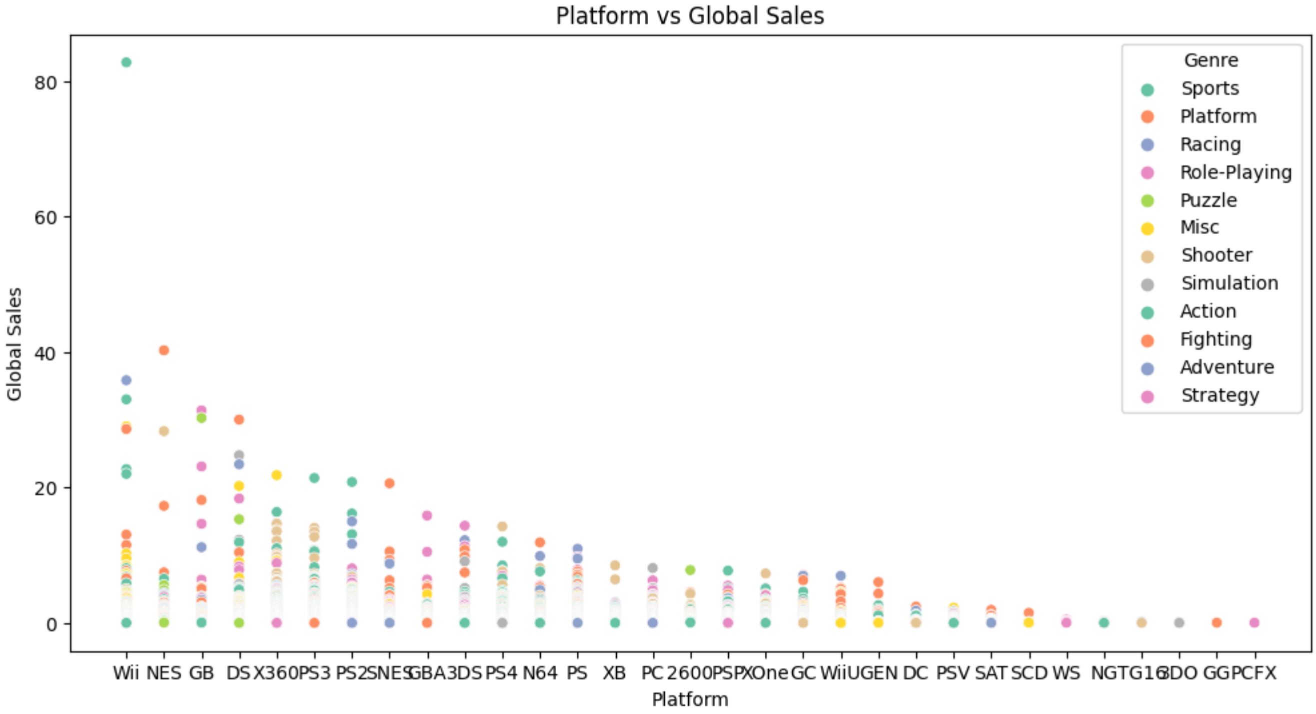
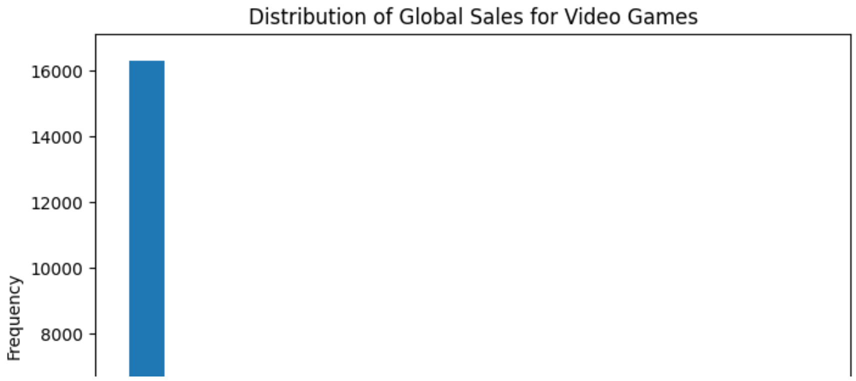
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#reading from csv
df = pd.read_csv('/content/Dataset3_VGsales.csv')
```

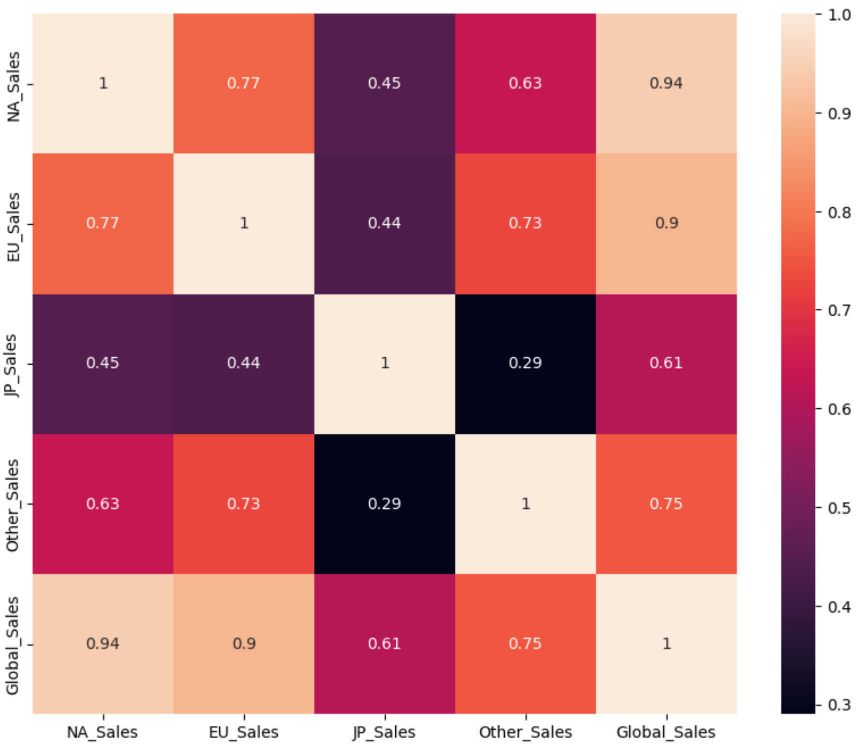
```
# Question 1
plt.figure(figsize=(12, 6))
sns.scatterplot(data=df, x='Platform', y='Global_Sales', hue='Genre', palette='Set2')
plt.title('Platform vs Global Sales')
plt.xlabel('Platform')
plt.ylabel('Global Sales')
plt.show()
```



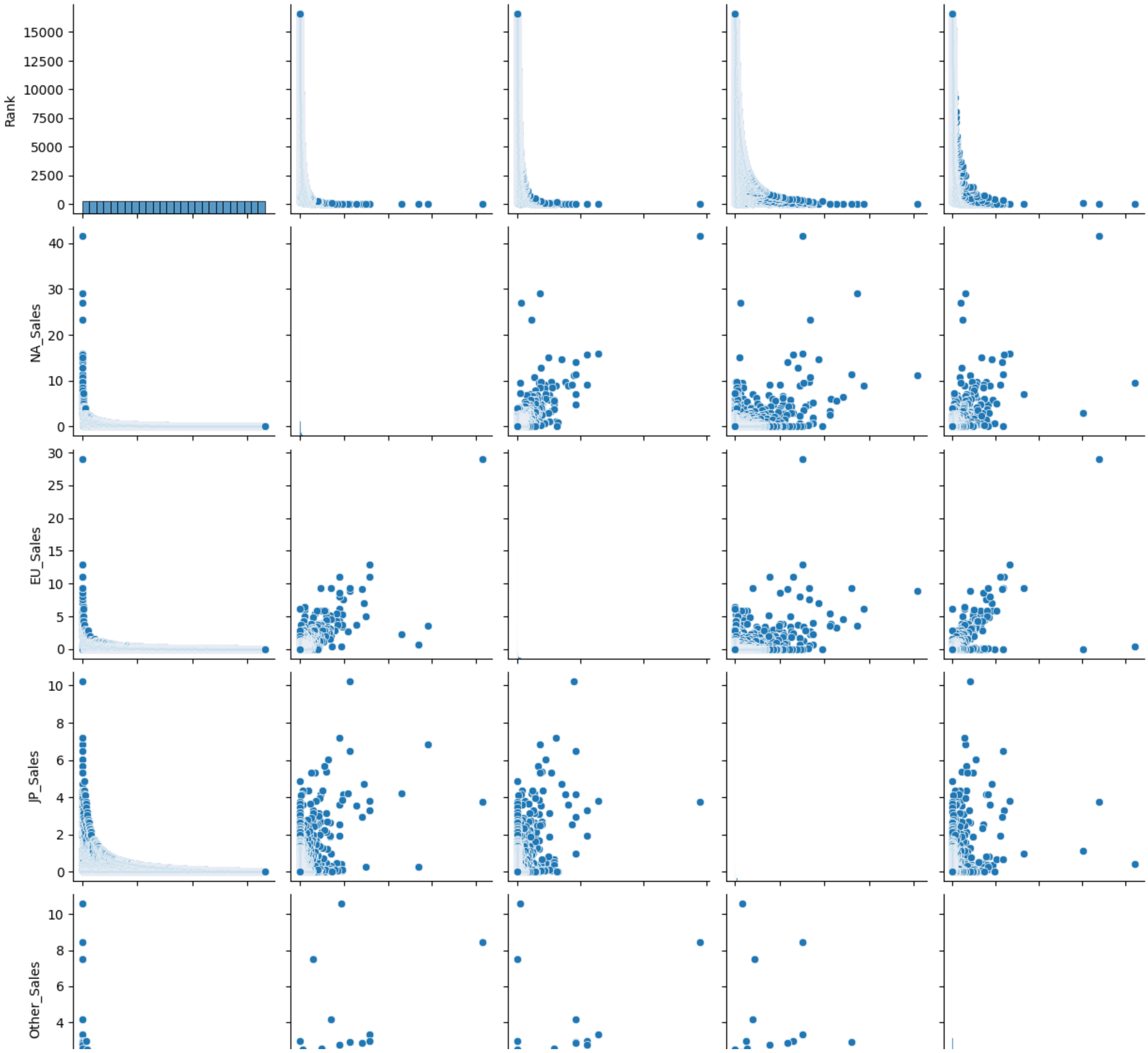
```
#Question 2
import pandas as pd
import matplotlib.pyplot as plt

#making global sales
global_sales = df['Global_Sales']

#creating histogram
plt.figure(figsize=(8, 6))
plt.hist(global_sales, bins=20)
plt.xlabel('Global Sales')
plt.ylabel('Frequency')
plt.title('Distribution of Global Sales for Video Games')
plt.show()

# data.hist(figsize=(5,5),bins=20,column="Global_Sales")
```



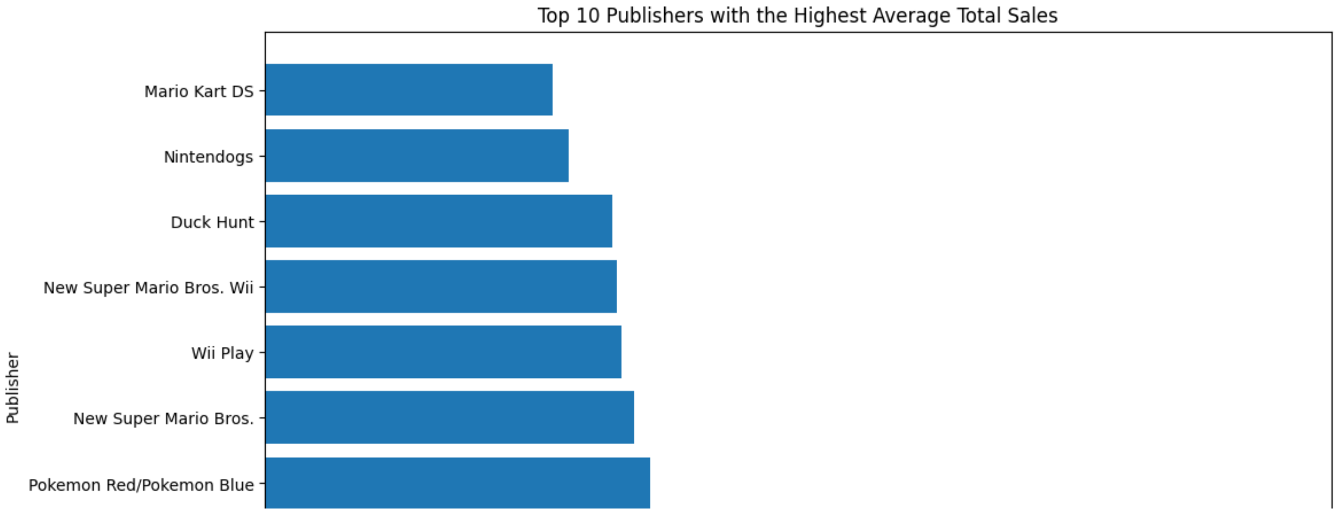
```
#Question 3 - HeatMap
correlation = df[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation, annot=True)
plt.show()
```



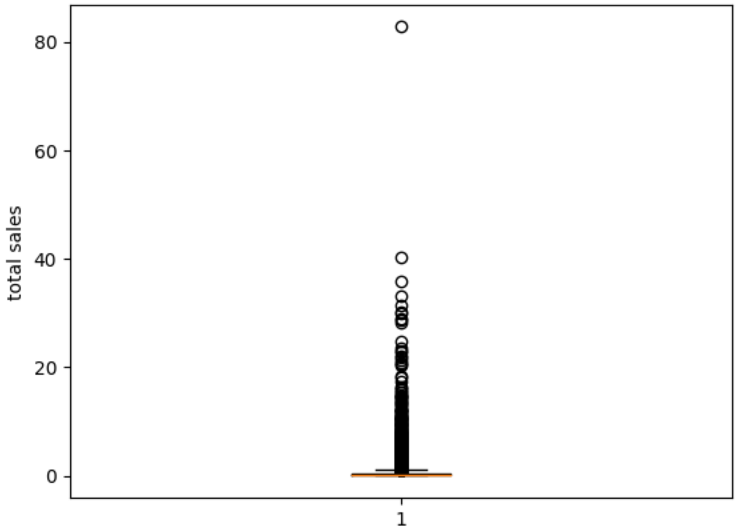
```
#Q4 - Scatterplot Matrix:

sns.pairplot(df[['Rank', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']])
plt.show()
```



```
00#Question 5
# Group the dataset by 'publisher' and calculate the average total sales
grouped_df = df.groupby('Name')['Global_Sales'].mean().reset_index()

# Sort the dataset by 'total_global_sales' in descending order
sorted_df = grouped_df.sort_values(by='Global_Sales', ascending=False)

# Display the top 10 publishers and their average sales using a bar chart
plt.figure(figsize=(12, 8))
plt.barh(sorted_df['Name'].head(10), sorted_df['Global_Sales'].head(10))
plt.xlabel('Average Total Sales')
plt.ylabel('Publisher')
plt.title('Top 10 Publishers with the Highest Average Total Sales')
plt.show()
```



```
# Question 6 - outlier detection
plt.boxplot(values.sort_values(), vert=True)
plt.ylabel("total sales")
plt.show()
outliers = df[(df['Global_Sales'] < df['Global_Sales'].quantile(0.25)) | (df['Global_Sales'] > df['Global_Sales'].quantile(0
print(outliers[['Platform', 'Global_Sales']])
```

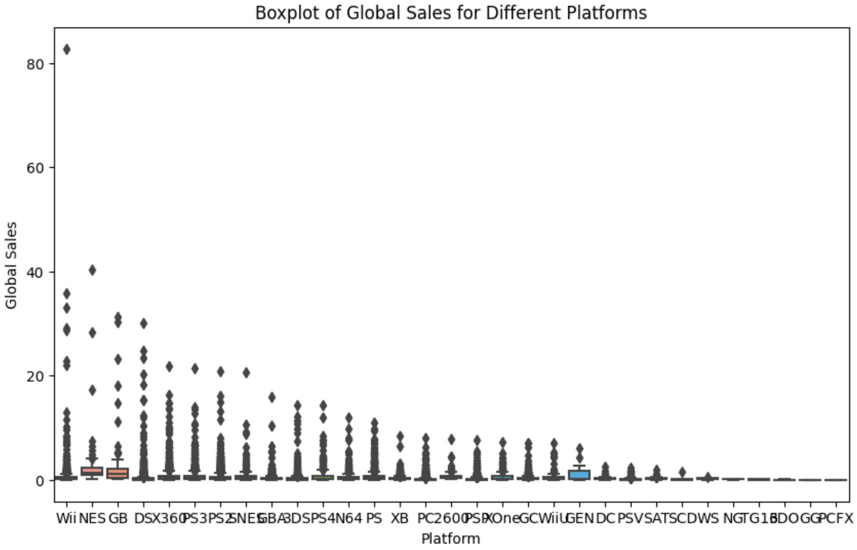


```
      Platform  Global_Sales
0          Wii         82.74
1          NES         40.24
2          Wii         35.82
3          Wii         33.00
4           GB         31.37
...        ...           ...
16593      GBA          0.01
16594       GC          0.01
16595      PS2          0.01
16596       DS          0.01
16597      GBA          0.01

[7923 rows x 2 columns]
```

```
#Question 7
from scipy.stats import ttest_ind
# Select relevant columns for the t-test
platform = 'Platform'
global_sales = 'Global_Sales'

# Get unique platforms in the dataset
platforms = df[platform].unique()

# Create two groups for the t-test
group1 = df[global_sales][df[platform] == platforms[0]]
group2 = df[global_sales][df[platform] == platforms[1]]

# Perform an independent two-sample t-test
t_statistic, p_value = ttest_ind(group1, group2)
```

```
# Set the significance level (e.g., 0.05)
alpha = 0.05

# Print the results
print(f'T-Statistic: {t_statistic}')
print(f'P-Value: {p_value}')

# Check for significance
if p_value < alpha:
    print('There is a significant difference in global sales between games with different platforms.')
else:
    print('There is no significant difference in global sales between games with different platforms.')

# Create a boxplot for visual representation
plt.figure(figsize=(10, 6))
sns.boxplot(x=platform, y=global_sales, data=df)
plt.title('Boxplot of Global Sales for Different Platforms')
plt.xlabel('Platform')
plt.ylabel('Global Sales')
plt.show()
```

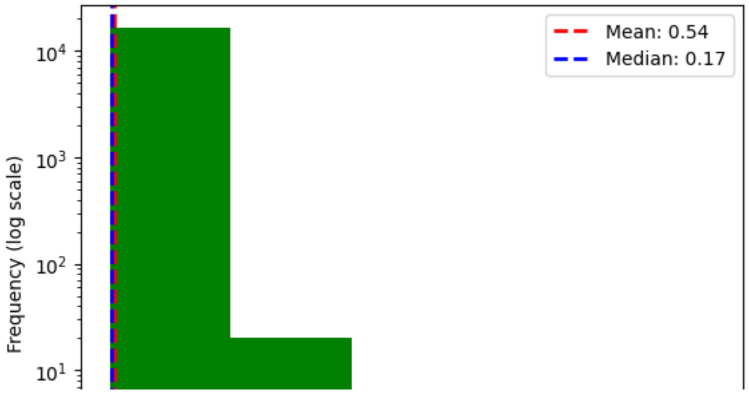
```
T-Statistic: -5.375132727353445
P-Value: 8.934095145387372e-08
There is a significant difference in global sales between games with different
```



Double-click (or enter) to edit

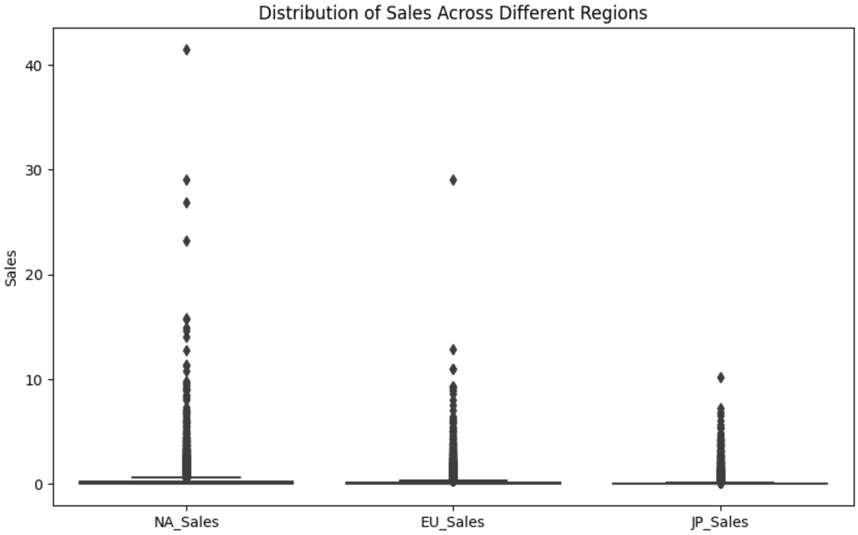
```
#Q8 - Central Tendency of Sales:

mean_sales = df['Global_Sales'].mean()
median_sales = df['Global_Sales'].median()

print(f'Mean Global Sales: {mean_sales}')
print(f'Median Global Sales: {median_sales}')

plt.hist(df['Global_Sales'], bins=5, color='g')
plt.yscale('log')  # Set the scale of the y-axis to be logarithmic to be able to see the distribution better for highly skew
plt.axvline(mean_sales, color='r', linestyle='dashed', linewidth=2, label=f'Mean: {mean_sales:.2f}')
plt.axvline(median_sales, color='b', linestyle='dashed', linewidth=2, label=f'Median: {median_sales:.2f}')
plt.xlabel('Global Sales')
plt.ylabel('Frequency (log scale)')
plt.legend()
plt.show()
```

```
Mean Global Sales: 0.5374406555006628
Median Global Sales: 0.17
```



Double-click (or enter) to edit

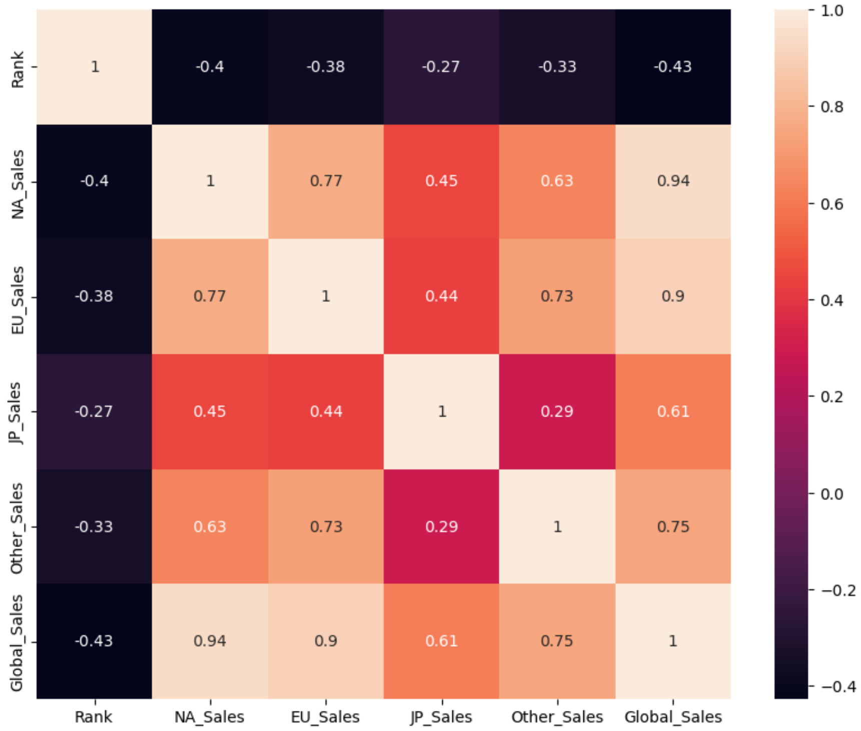
```
#Q9 - Boxplot:

plt.figure(figsize=(10, 6))
sns.boxplot(data=df[['NA_Sales', 'EU_Sales', 'JP_Sales']])
plt.title('Distribution of Sales Across Different Regions')
plt.ylabel('Sales')
plt.show()
```



Double-click (or enter) to edit

```
#Q10 - Correlation Heatmap:

# Calculate the correlation between 'Ranking' and 'Global_Sales'
correlation = df['Rank'].corr(df['Global_Sales'])

print(f'Correlation between Ranking and Global Sales: {correlation}')

# Create a DataFrame with the specified columns
df_subset = df[['Rank', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales']]

# Calculate the correlation matrix
corr = df_subset.corr()

# Create a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True)
plt.show()
```

```
Correlation between Ranking and Global Sales: -0.42740660798868146
```


Rank
NA_Sales
EU_Sales
JP_Sales
Other_Sales
Global_Sales