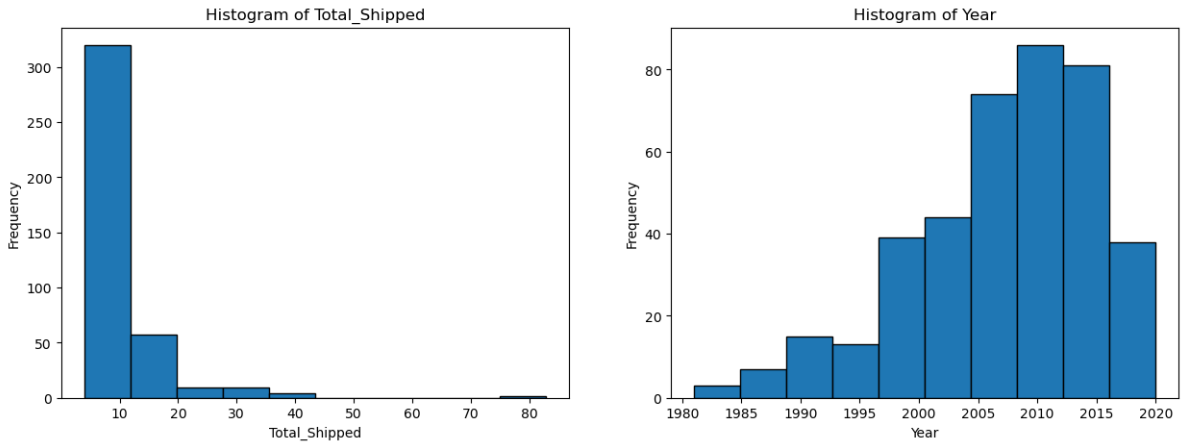
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/ga
```

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Name           400 non-null    object 
 1   Platform       400 non-null    object 
 2   Publisher      400 non-null    object 
 3   Developer      400 non-null    object 
 4   Total_Shipped  400 non-null    float64
 5   Year           400 non-null    int64  
dtypes: float64(1), int64(1), object(4)
memory usage: 18.9+ KB
```

In [4]:
```
# Select numerical columns for the histograms
numerical_columns = df.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```



In [5]:
```
df1=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/g
```

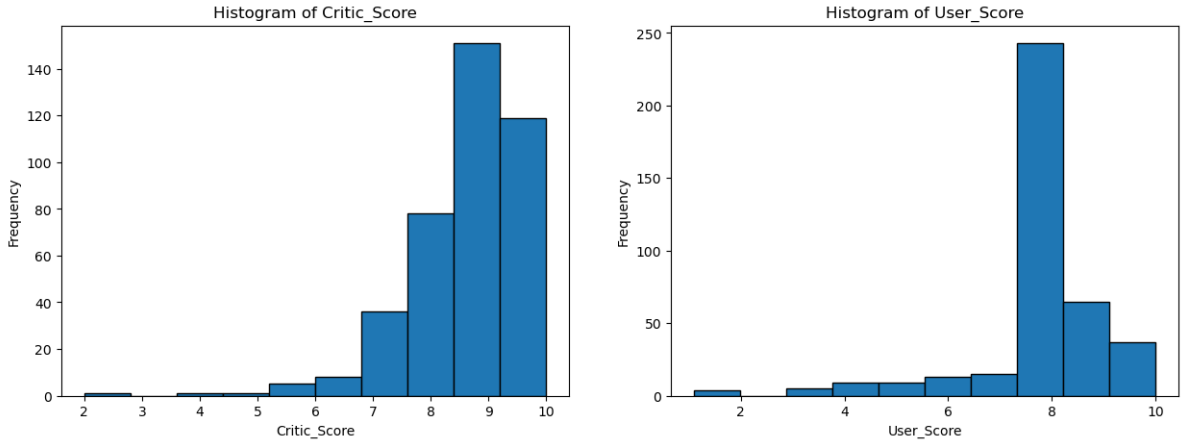
In [6]:
```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Name          400 non-null    object 
 1   Critic_Score  399 non-null    float64
 2   User_Score    188 non-null    float64
dtypes: float64(2), object(1)
memory usage: 9.5+ KB
```

In [7]:
```
df1['Critic_Score']=df1['Critic_Score'].fillna(df1['Critic_Score'].mean())
df1['User_Score']=df1['User_Score'].fillna(df1['User_Score'].mean())
```

In [8]:
```
df1.to_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/game_r
```

In [9]:
```
# Select numerical columns for the histograms
numerical_columns = df1.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df1[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```

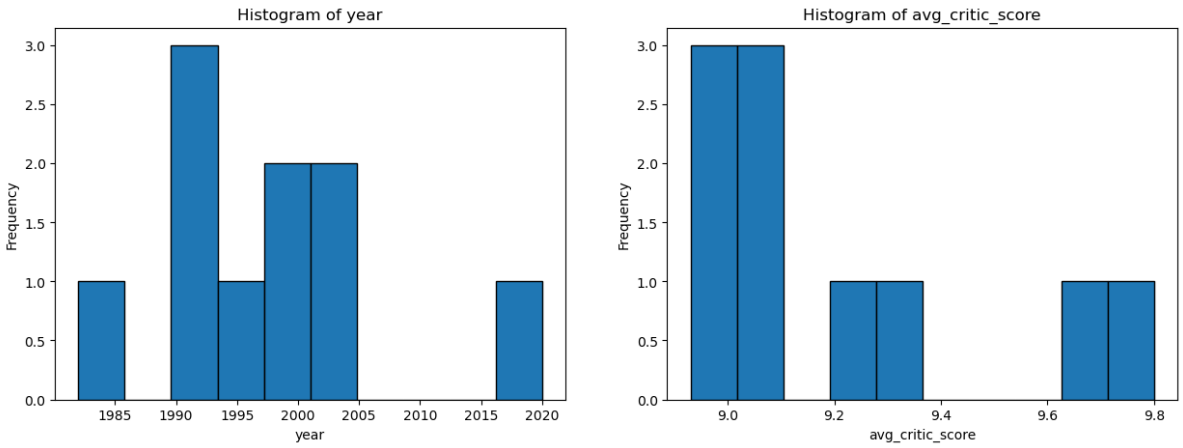


In [10]:
```
df2=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/t
```

In [11]:
```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   year              10 non-null     int64
 1   avg_critic_score  10 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 288.0 bytes
```

In [12]:
```
# Select numerical columns for the histograms
numerical_columns = df2.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df2[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```

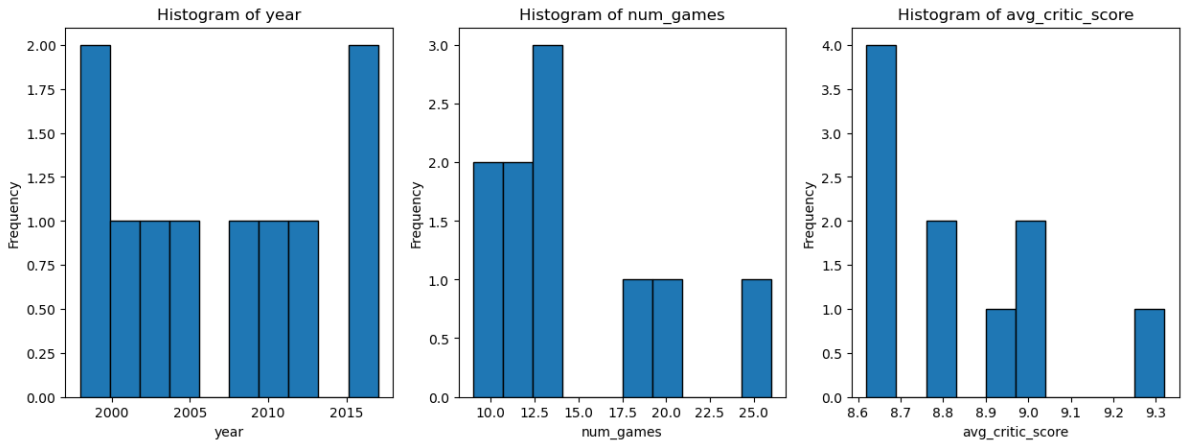


In [13]:
```
df3=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/t
```

In [14]:
```
df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   year             10 non-null     int64  
 1   num_games        10 non-null     int64  
 2   avg_critic_score  10 non-null     float64
dtypes: float64(1), int64(2)
memory usage: 368.0 bytes
```

In [15]:
```
# Select numerical columns for the histograms
numerical_columns = df3.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df3[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```

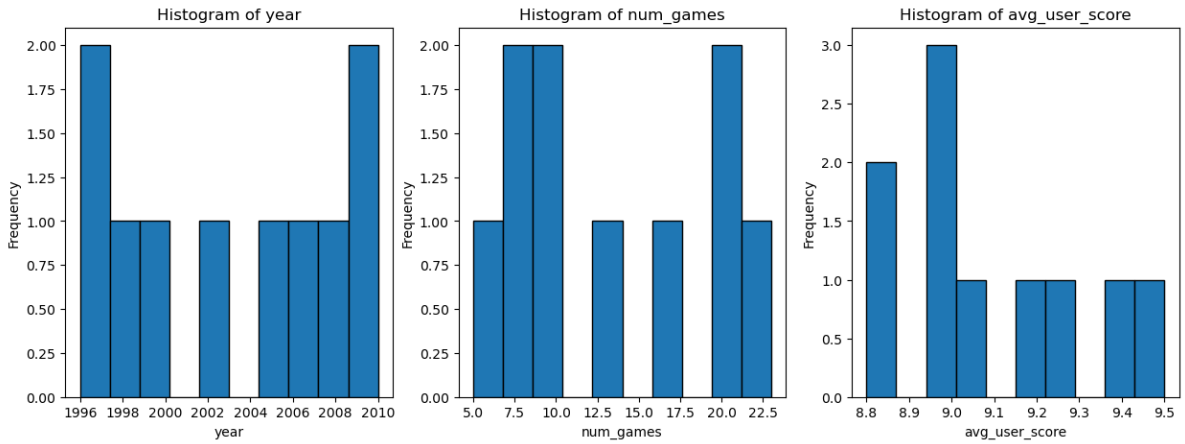


In [16]:
```
df4=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/t
```

In [17]:
```
df4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   year            10 non-null     int64  
 1   num_games       10 non-null     int64  
 2   avg_user_score  10 non-null     float64
dtypes: float64(1), int64(2)
memory usage: 368.0 bytes
```

In [18]:
```
# Select numerical columns for the histograms
numerical_columns = df4.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df4[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```



In [19]:
```
df5=pd.read_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/v
```

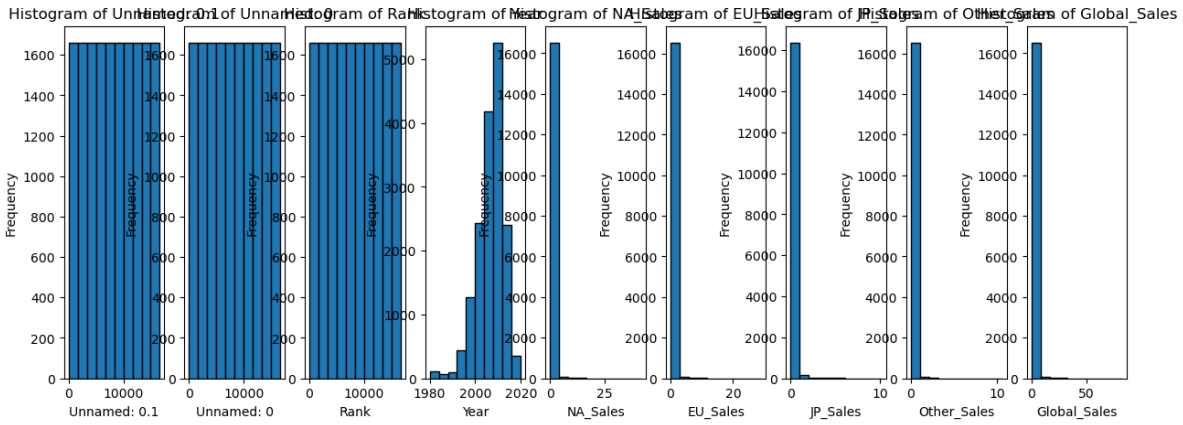
In [20]:
```
df5.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Unnamed: 0.1   16598 non-null  int64  
 1   Unnamed: 0     16598 non-null  int64  
 2   Rank           16598 non-null  int64  
 3   Game           16598 non-null  object 
 4   Platform       16598 non-null  object 
 5   Year           16598 non-null  float64
 6   Genre          16598 non-null  object 
 7   Publisher      16540 non-null  object 
 8   NA_Sales       16598 non-null  float64
 9   EU_Sales       16598 non-null  float64
 10  JP_Sales       16598 non-null  float64
 11  Other_Sales    16598 non-null  float64
 12  Global_Sales   16598 non-null  float64
dtypes: float64(6), int64(3), object(4)
memory usage: 1.6+ MB
```

In [21]:
```
df5['Year']=df5['Year'].fillna(df5['Year'].mean())
df5['Genre']=df5['Genre'].fillna(df5['Genre'].mode())
df5['Publisher']=df5['Publisher'].fillna(df5['Publisher'].mode())
```

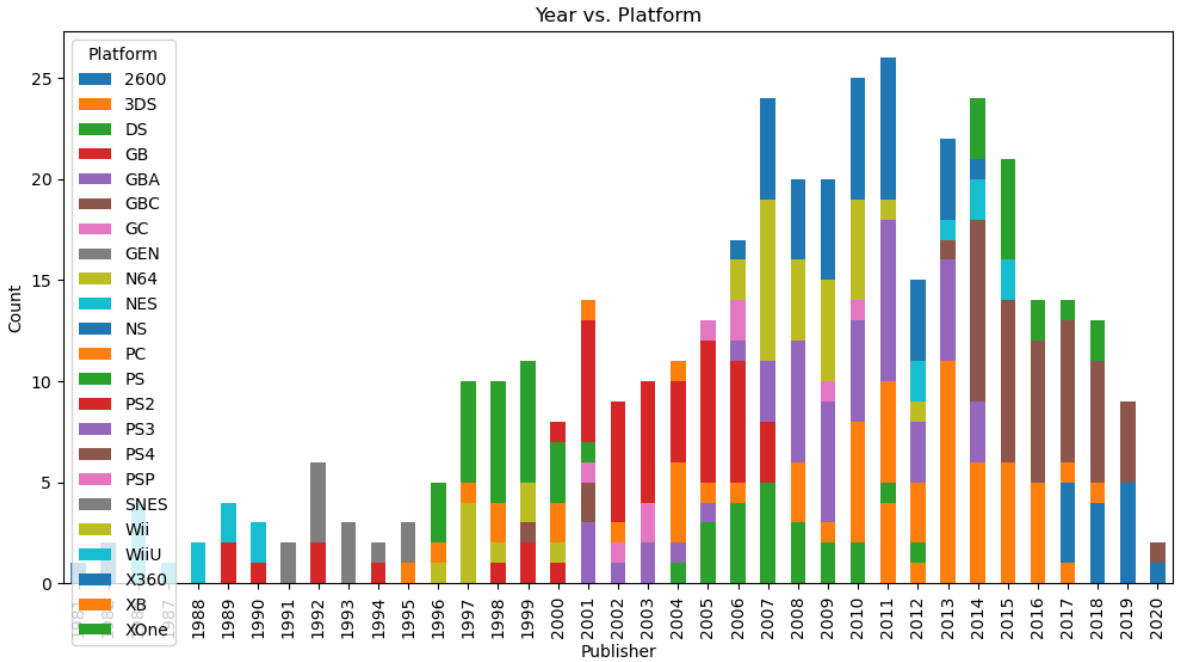
In [22]:
```
df5.to_csv("C:/Users/suman/OneDrive/Desktop/Data warehousing/New folder/vgsale
```

In [23]:
```
# Select numerical columns for the histograms
numerical_columns = df5.select_dtypes(include=['number'])

# Set the number of bins for the histograms
num_bins = 10

# Create subplots to display histograms side by side
fig, axes = plt.subplots(1, len(numerical_columns.columns), figsize=(15, 5))

# Loop through numerical columns and create histograms
for i, column in enumerate(numerical_columns.columns):
    axes[i].hist(df5[column], bins=num_bins, edgecolor='k')
    axes[i].set_title(f'Histogram of {column}')
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Frequency')

plt.show()
```



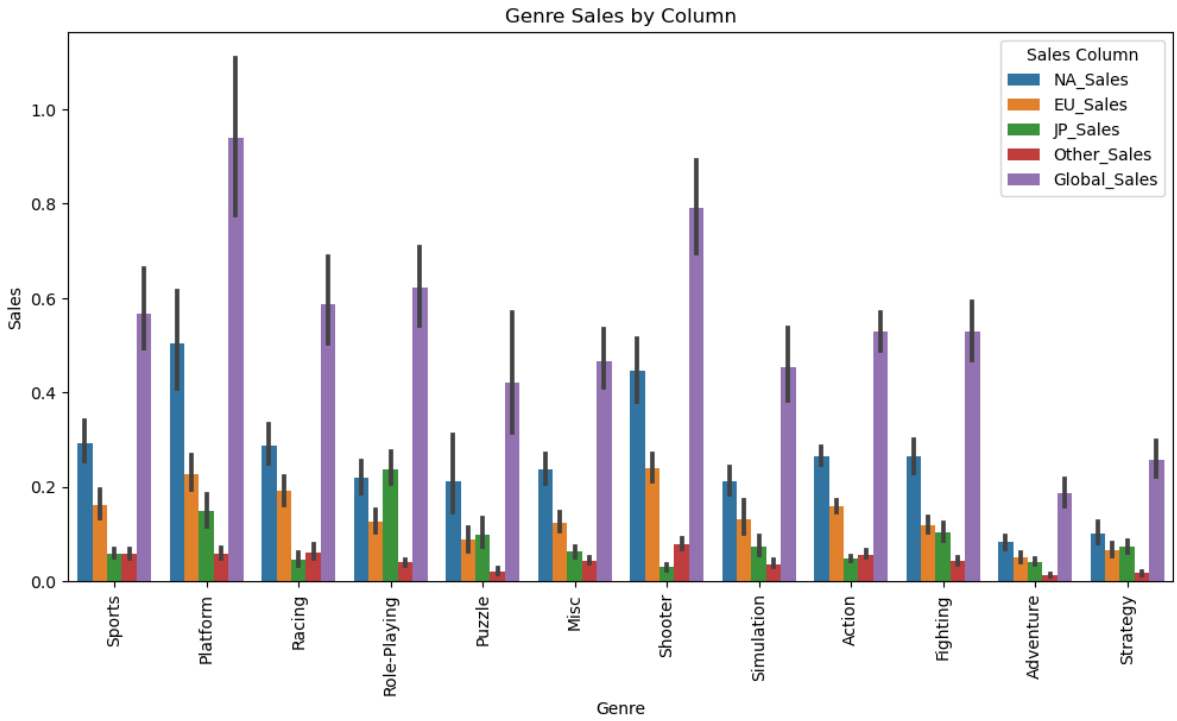
In [24]:

```
# Assuming you have columns 'Genre' and 'Platform' in your dataset
genre_platform_count = df.groupby(['Year', 'Platform']).size().unstack(fill_va

# Create a bar chart
genre_platform_count.plot(kind='bar', stacked=True, figsize=(12, 6))

# Customize the chart
plt.title('Year vs. Platform')
plt.xlabel('Publisher')
plt.ylabel('Count')

# Display the legend
plt.legend(title='Platform', loc='upper left')

# Show the chart
plt.show()
```

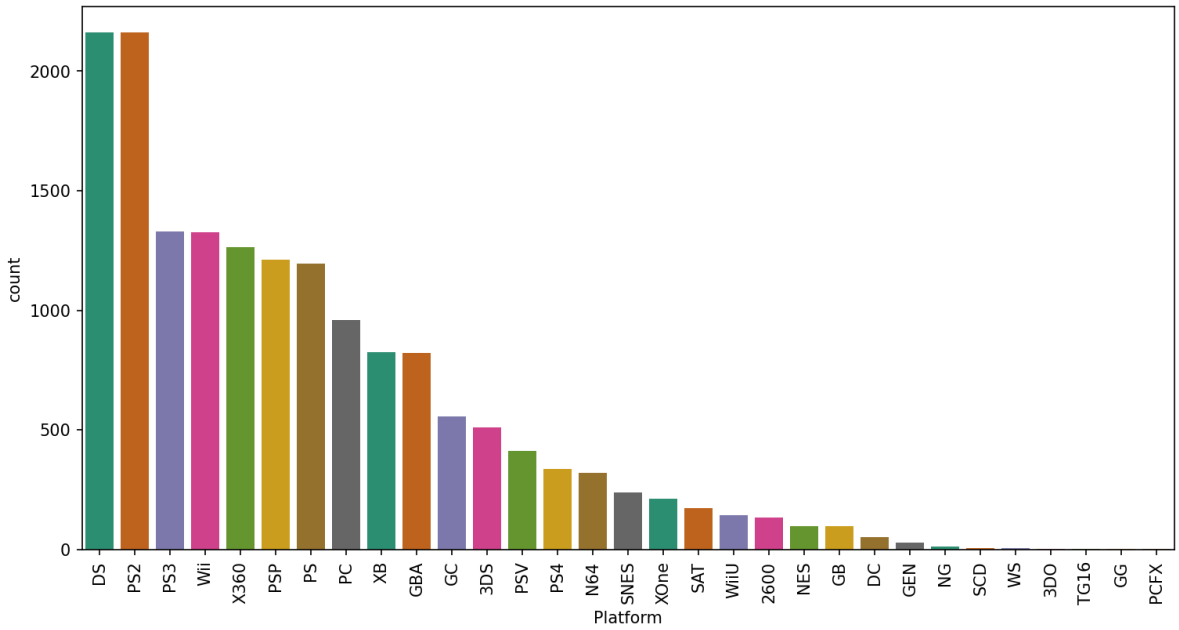


In [29]:

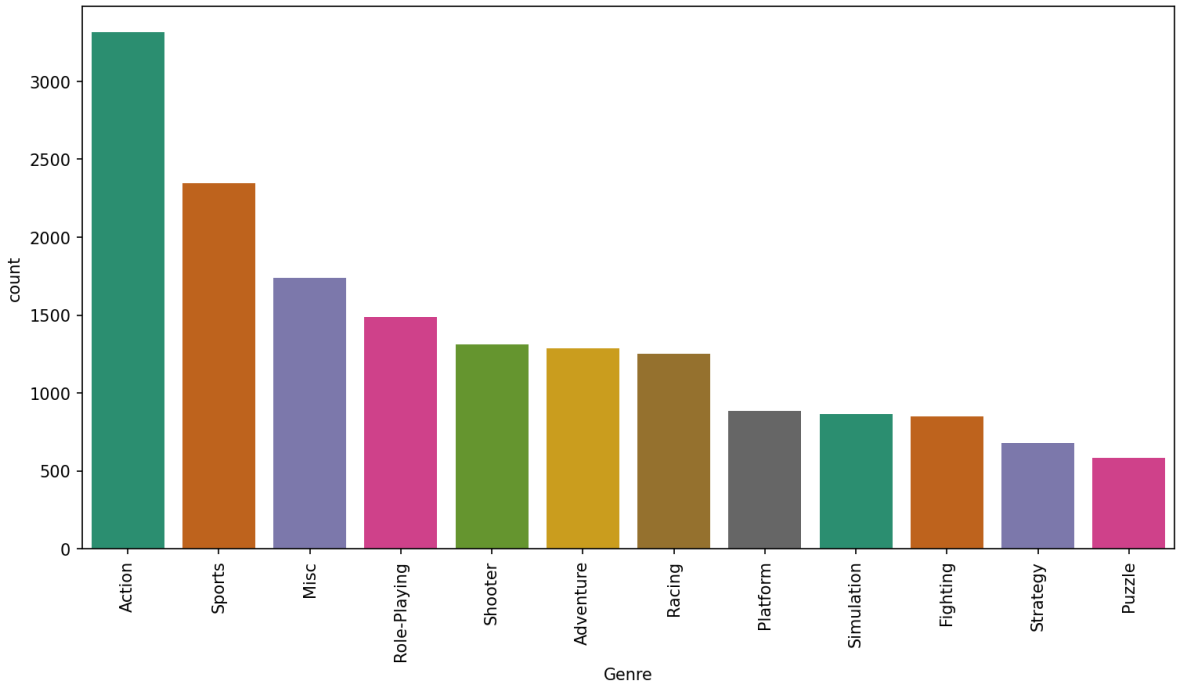
```
# Select the relevant columns
sales_data = df5[['Genre', 'NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales',

# Melt the data to make it suitable for plotting
melted_data = sales_data.melt(id_vars='Genre', var_name='Sales_Column', value_

# Create a bar chart
plt.figure(figsize=(12, 6))
sns.barplot(data=melted_data, x='Genre', y='Sales', hue='Sales_Column')

# Customize the chart
plt.title('Genre Sales by Column')
plt.xlabel('Genre')
plt.ylabel('Sales')
plt.xticks(rotation=90)
plt.legend(title='Sales Column', loc='upper right')

# Show the chart
plt.show()
```



In [32]:
```
plt.figure(figsize=(12,6), dpi=150)
sns.countplot(data=df5,
              x="Platform",
              order = df5["Platform"].value_counts().index,
              palette="Dark2")
plt.xticks(rotation=90)
plt.show()
```



In [27]:
```
plt.figure(figsize=(12,6), dpi=150)
sns.countplot(data=df5,
              x="Genre",
              order = df5["Genre"].value_counts().index,
              palette="Dark2")
plt.xticks(rotation=90)
plt.show()
```



In [28]:

```
# Group the data by 'Publisher' and sum the 'Global_Sales' for each publisher
publisher_sales = df5.groupby('Publisher')['Global_Sales'].sum().reset_index()

# Sort the data by global sales in descending order
publisher_sales = publisher_sales.sort_values(by='Global_Sales', ascending=Fal

# Select the top 5 publishers
top_publishers = publisher_sales.head(5)

# Create a bar plot
plt.figure(figsize=(12, 6))
sns.barplot(data=top_publishers, x='Publisher', y='Global_Sales', palette='vir

# Customize the chart
plt.title('Top 5 Publishers by Global Sales')
plt.xlabel('Publisher')
plt.ylabel('Global Sales (in millions)')

# Show the chart
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```