

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

```
In [2]: 1 df = pd.read_csv('Video_Games.csv')
```

```
In [3]: 1 df.head(20)
```

Out[3]:

| | Name | Platform | Year_of_Release | Genre | Publisher | NA_Sales | EU_Sales | JP_Sales | Othe |
|----|--|----------|-----------------|--------------|------------------------|----------|----------|----------|------|
| 0 | Wii Sports | Wii | 2006.0 | Sports | Nintendo | 41.36 | 28.96 | 3.77 | |
| 1 | Super Mario Bros. | NES | 1985.0 | Platform | Nintendo | 29.08 | 3.58 | 6.81 | |
| 2 | Mario Kart Wii | Wii | 2008.0 | Racing | Nintendo | 15.68 | 12.76 | 3.79 | |
| 3 | Wii Sports Resort | Wii | 2009.0 | Sports | Nintendo | 15.61 | 10.93 | 3.28 | |
| 4 | Pokemon Red/Pokemon Blue | GB | 1996.0 | Role-Playing | Nintendo | 11.27 | 8.89 | 10.22 | |
| 5 | Tetris | GB | 1989.0 | Puzzle | Nintendo | 23.20 | 2.26 | 4.22 | |
| 6 | New Super Mario Bros. | DS | 2006.0 | Platform | Nintendo | 11.28 | 9.14 | 6.50 | |
| 7 | Wii Play | Wii | 2006.0 | Misc | Nintendo | 13.96 | 9.18 | 2.93 | |
| 8 | New Super Mario Bros. Wii | Wii | 2009.0 | Platform | Nintendo | 14.44 | 6.94 | 4.70 | |
| 9 | Duck Hunt | NES | 1984.0 | Shooter | Nintendo | 26.93 | 0.63 | 0.28 | |
| 10 | Nintendogs | DS | 2005.0 | Simulation | Nintendo | 9.05 | 10.95 | 1.93 | |
| 11 | Mario Kart DS | DS | 2005.0 | Racing | Nintendo | 9.71 | 7.47 | 4.13 | |
| 12 | Pokemon Gold/Pokemon Silver | GB | 1999.0 | Role-Playing | Nintendo | 9.00 | 6.18 | 7.20 | |
| 13 | Wii Fit | Wii | 2007.0 | Sports | Nintendo | 8.92 | 8.03 | 3.60 | |
| 14 | Kinect Adventures! | X360 | 2010.0 | Misc | Microsoft Game Studios | 15.00 | 4.89 | 0.24 | |
| 15 | Wii Fit Plus | Wii | 2009.0 | Sports | Nintendo | 9.01 | 8.49 | 2.53 | |
| 16 | Grand Theft Auto V | PS3 | 2013.0 | Action | Take-Two Interactive | 7.02 | 9.09 | 0.98 | |
| 17 | Grand Theft Auto: San Andreas | PS2 | 2004.0 | Action | Take-Two Interactive | 9.43 | 0.40 | 0.41 | |
| 18 | Super Mario World | SNES | 1990.0 | Platform | Nintendo | 12.78 | 3.75 | 3.54 | |
| 19 | Brain Age: Train Your Brain in Minutes a Day | DS | 2005.0 | Misc | Nintendo | 4.74 | 9.20 | 4.16 | |

```
In [4]: 1 df.shape
```

```
Out[4]: (16719, 16)
```

```
In [5]: 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16719 entries, 0 to 16718
Data columns (total 16 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                   16717 non-null  object
1   Platform               16719 non-null  object
2   Year_of_Release        16450 non-null  float64
3   Genre                  16717 non-null  object
4   Publisher               16665 non-null  object
5   NA_Sales                16719 non-null  float64
6   EU_Sales                16719 non-null  float64
7   JP_Sales                16719 non-null  float64
8   Other_Sales             16719 non-null  float64
9   Global_Sales            16719 non-null  float64
10  Critic_Score            8137 non-null   float64
11  Critic_Count            8137 non-null   float64
12  User_Score              10015 non-null  object
13  User_Count              7590 non-null   float64
14  Developer               10096 non-null  object
15  Rating                  9950 non-null   object
dtypes: float64(9), object(7)
memory usage: 2.0+ MB
```

```
In [6]: 1 df.isna().sum()
```

```
Out[6]: Name                2
Platform              0
Year_of_Release       269
Genre                  2
Publisher              54
NA_Sales               0
EU_Sales               0
JP_Sales               0
Other_Sales            0
Global_Sales           0
Critic_Score          8582
Critic_Count          8582
User_Score             6704
User_Count            9129
Developer              6623
Rating                 6769
dtype: int64
```

```
In [7]: 1 df = df.dropna(axis=0, how='any')
```

```
In [8]: 1 df.isna().sum()
```

```
Out[8]: Name          0
Platform        0
Year_of_Release  0
Genre           0
Publisher        0
NA_Sales         0
EU_Sales         0
JP_Sales         0
Other_Sales      0
Global_Sales     0
Critic_Score     0
Critic_Count     0
User_Score       0
User_Count       0
Developer        0
Rating           0
dtype: int64
```

Null values cleared

```
In [9]: 1 df.duplicated().sum()
```

```
Out[9]: 0
```

There are no duplicate values

```
In [10]: 1 df.nunique()
```

```
Out[10]: Name          4377
Platform         17
Year_of_Release  25
Genre           12
Publisher        262
NA_Sales         351
EU_Sales         273
JP_Sales         157
Other_Sales      144
Global_Sales     536
Critic_Score     81
Critic_Count     106
User_Score       89
User_Count       875
Developer        1289
Rating           7
dtype: int64
```

```
In [11]: 1 df.index = df.index + 1
```

reseting index to start from 1

In [12]:

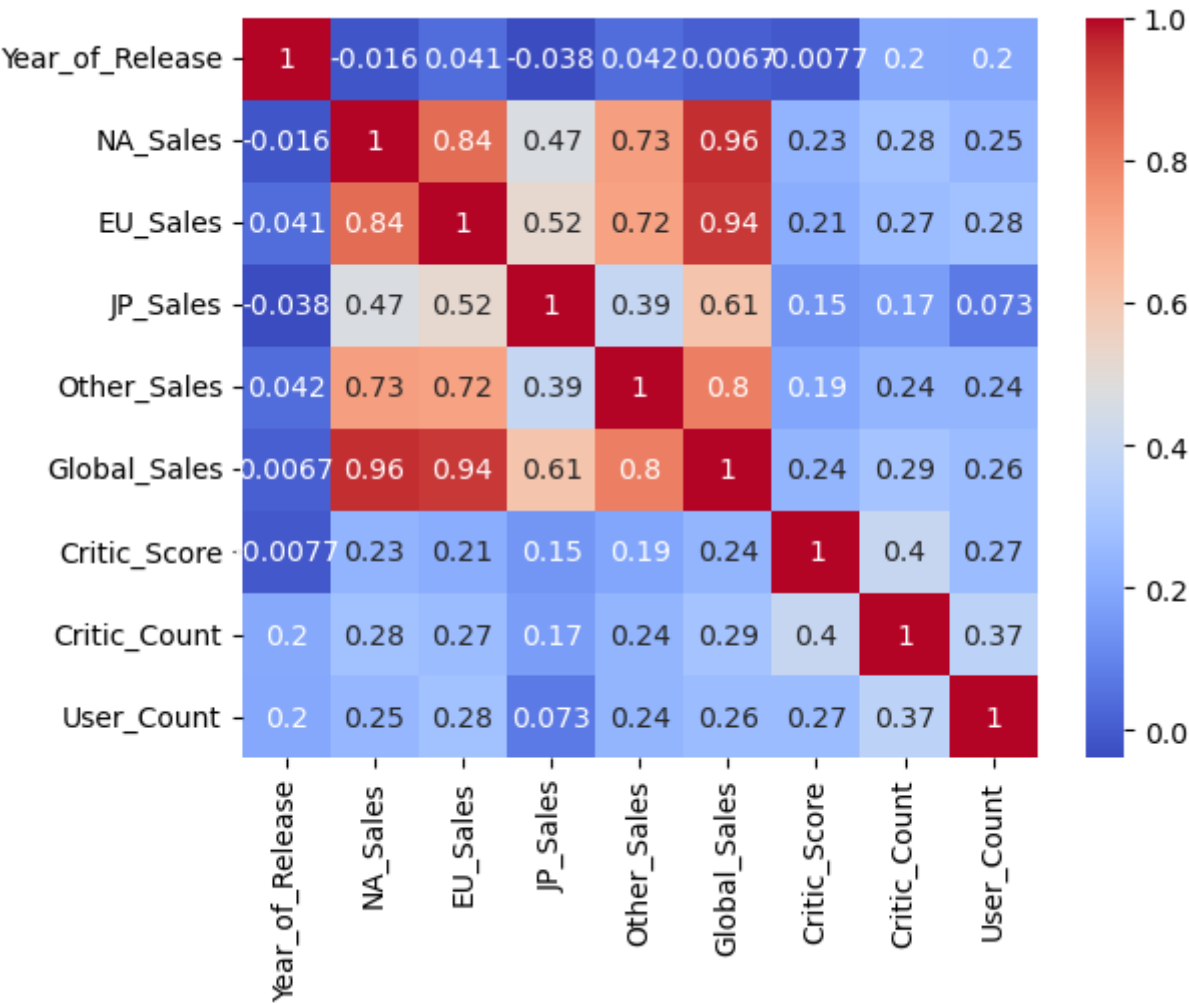
1 df.head()

Out[12]:

| | Name | Platform | Year_of_Release | Genre | Publisher | NA_Sales | EU_Sales | JP_Sales | Other_Sales | G |
|---|-----------------------|----------|-----------------|----------|-----------|----------|----------|----------|-------------|---|
| 1 | Wii Sports | Wii | 2006.0 | Sports | Nintendo | 41.36 | 28.96 | 3.77 | 8.45 | |
| 3 | Mario Kart Wii | Wii | 2008.0 | Racing | Nintendo | 15.68 | 12.76 | 3.79 | 3.29 | |
| 4 | Wii Sports Resort | Wii | 2009.0 | Sports | Nintendo | 15.61 | 10.93 | 3.28 | 2.95 | |
| 7 | New Super Mario Bros. | DS | 2006.0 | Platform | Nintendo | 11.28 | 9.14 | 6.50 | 2.88 | |
| 8 | Wii Play | Wii | 2006.0 | Misc | Nintendo | 13.96 | 9.18 | 2.93 | 2.84 | |

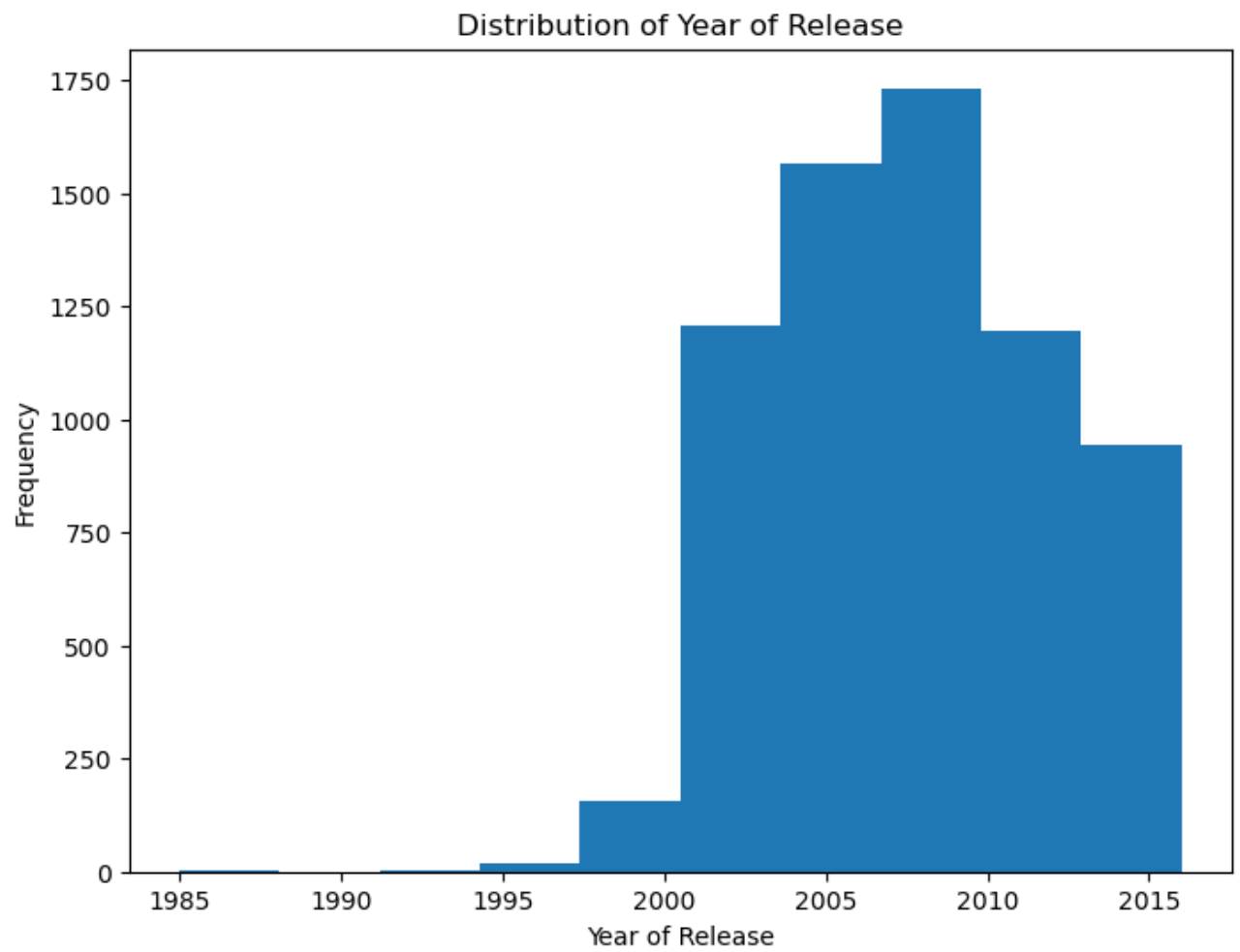
In [13]:

1 corr_matrix = df.corr()
2 sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
3 plt.show()



In [14]:

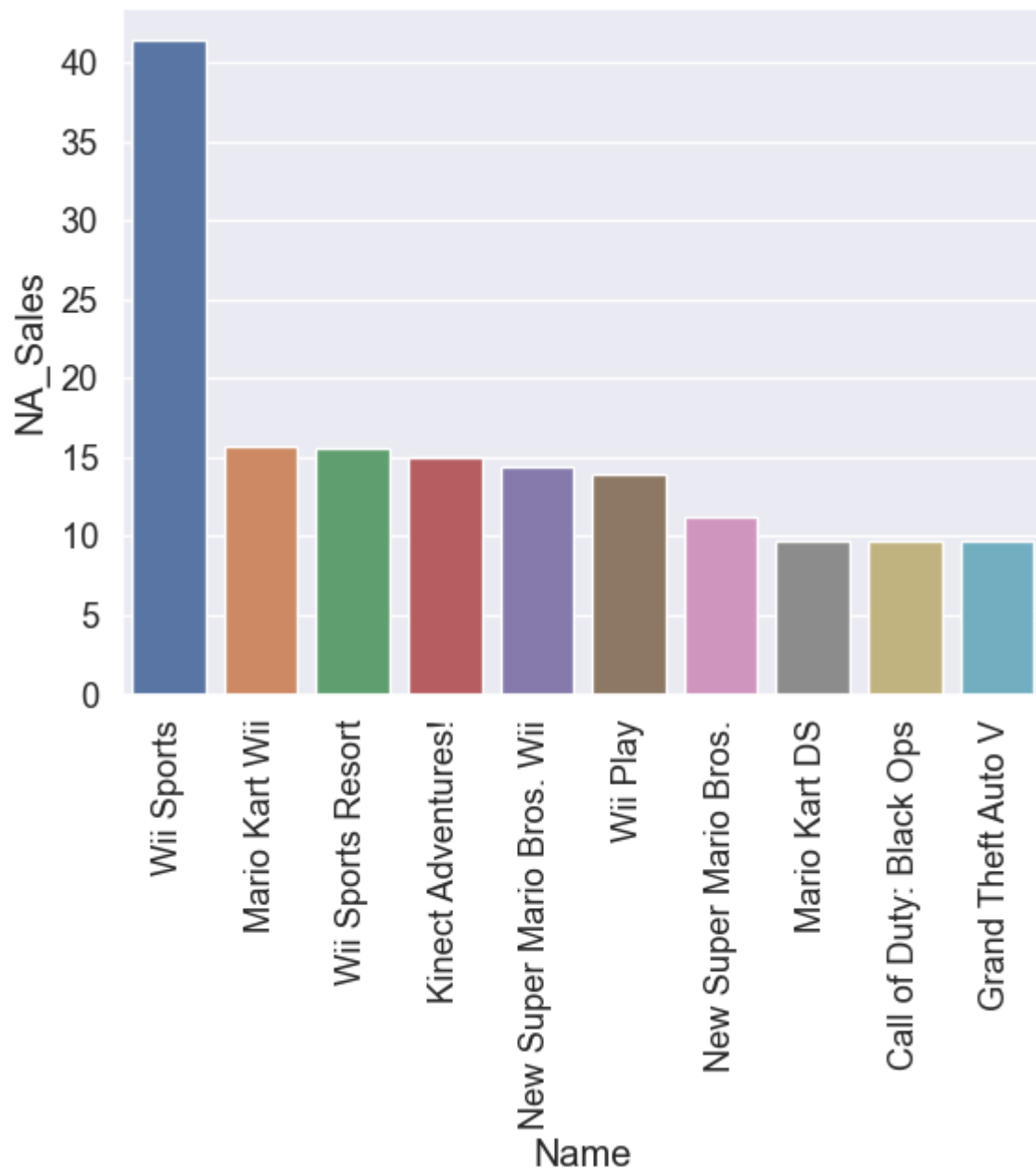
```
1 fig, ax = plt.subplots(figsize=(8, 6))
2 ax.hist(df["Year_of_Release"].dropna())
3 ax.set_xlabel("Year of Release")
4 ax.set_ylabel("Frequency")
5 ax.set_title("Distribution of Year of Release")
6 plt.show()
```



In [27]:

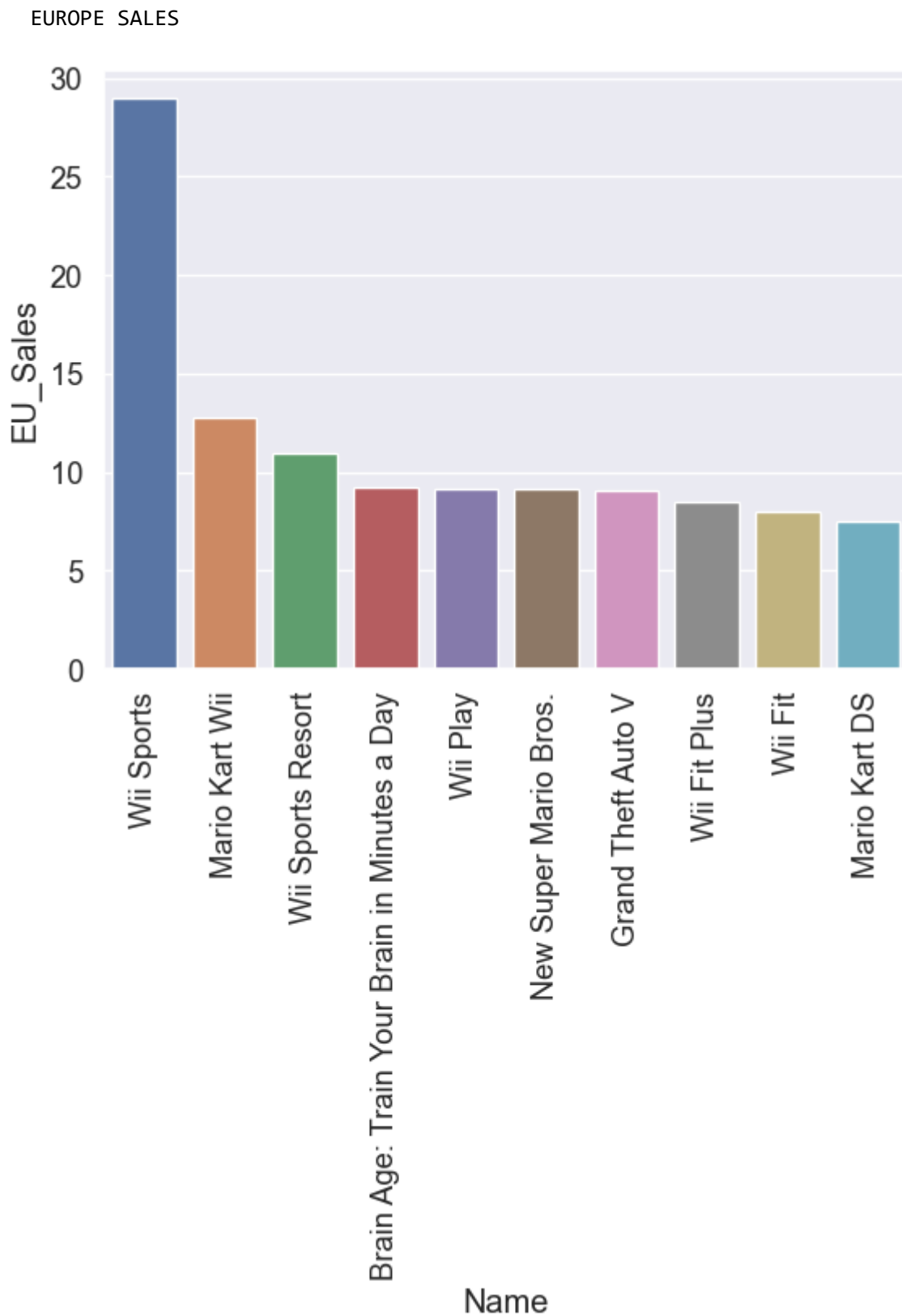
```
1 # barplot for north america sales
2
3 top_na = df.sort_values(by="NA_Sales", ascending=False).head(10)
4 print("    NORTH AMERICA SALES    ")
5 sns.barplot(x="Name", y="NA_Sales", data=top_na)
6 plt.xticks(rotation=90)
7 plt.show()
```

NORTH AMERICA SALES



In [26]:

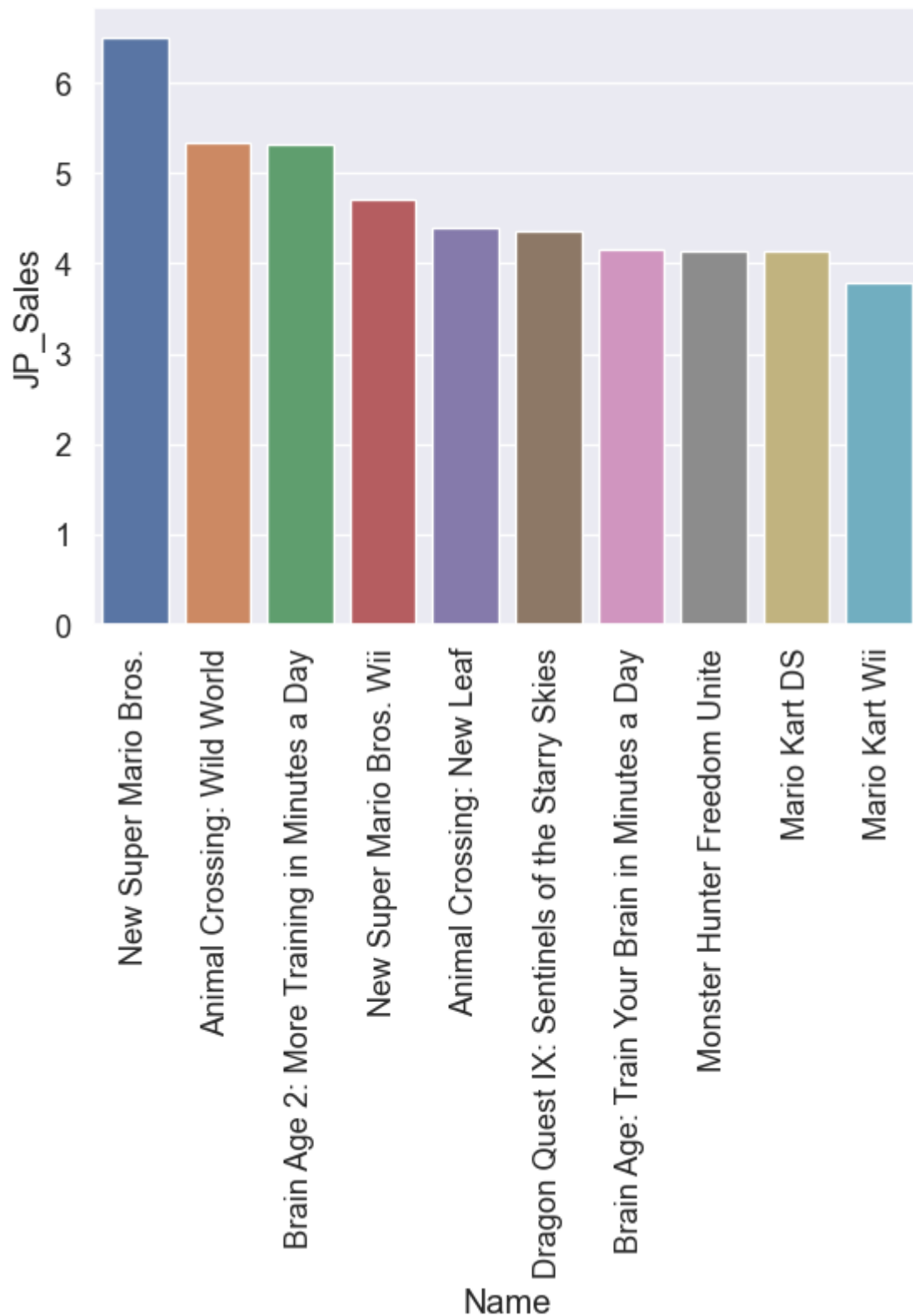
```
1 #barlplot for europe sales
2
3 print("    EUROPE SALES    ")
4 top_eu = df.sort_values(by="EU_Sales", ascending=False).head(10)
5 sns.barplot(x="Name", y="EU_Sales", data=top_eu)
6 plt.xticks(rotation=90)
7 plt.show()
```



In [24]:

```
1 #barplot for japans sales
2
3 print("    JAPAN SALES    ")
4 top_jp = df.sort_values(by="JP_Sales", ascending=False).head(10)
5 sns.barplot(x="Name", y="JP_Sales", data=top_jp)
6 plt.xticks(rotation=90)
7 plt.show()
```

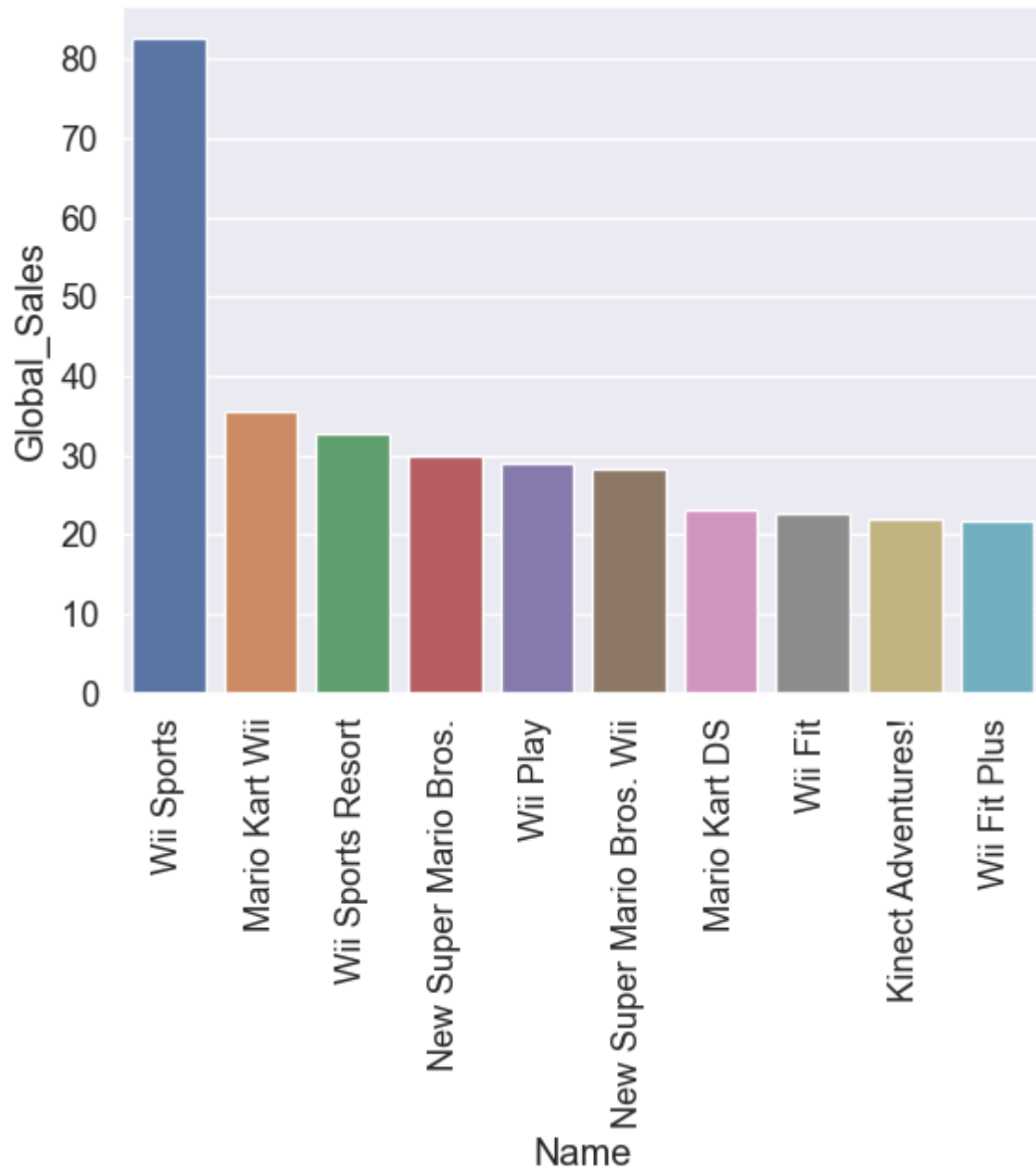
JAPAN SALES



In [25]:

```
1 #barplot for global sales
2
3 print(" GLOBALSALES ")
4 top_global = df.sort_values(by="Global_Sales", ascending=False).head(10)
5 sns.barplot(x="Name", y="Global_Sales", data=top_global)
6 plt.xticks(rotation=90)
7 plt.show()
```

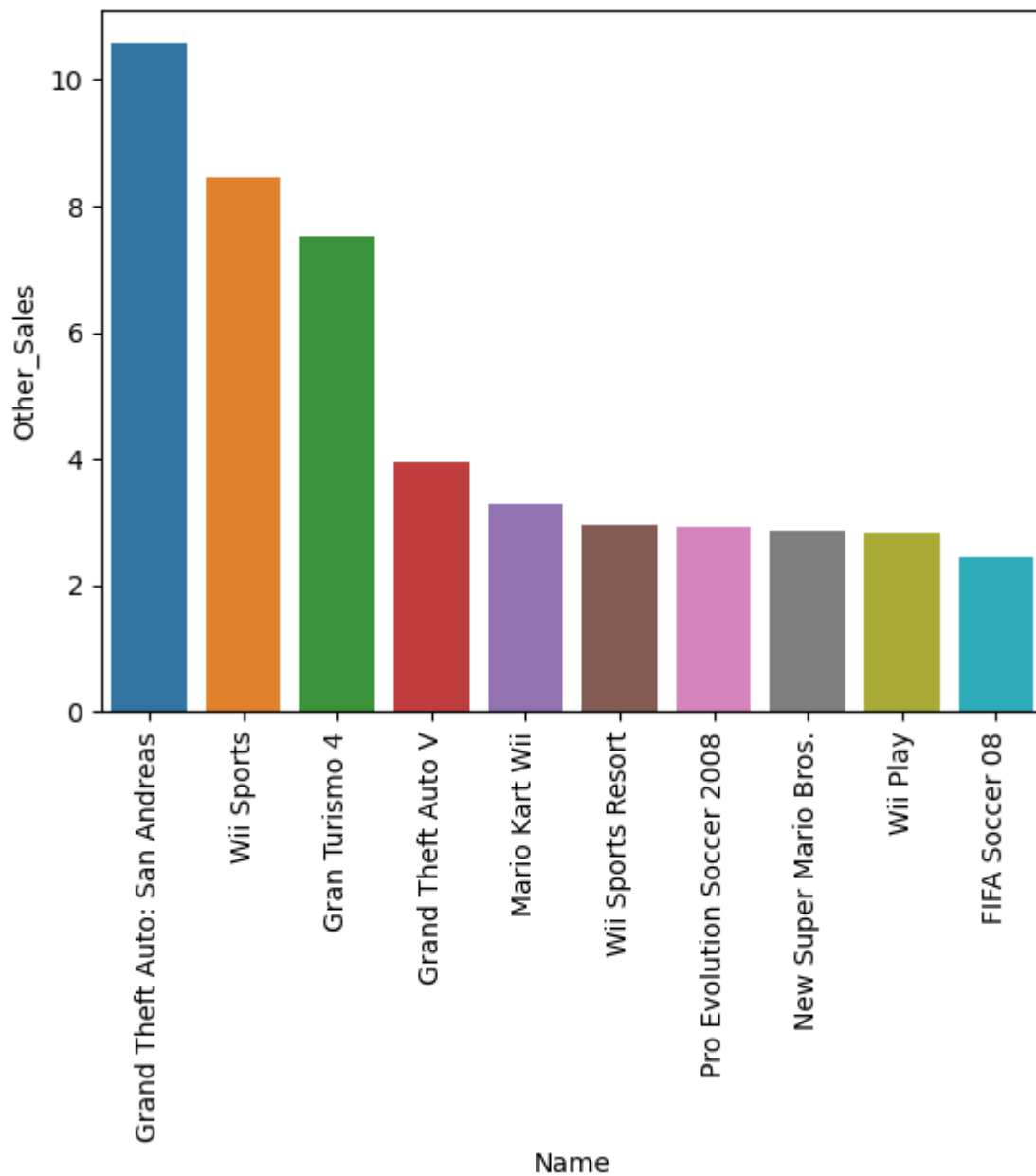
GLOBALSALES



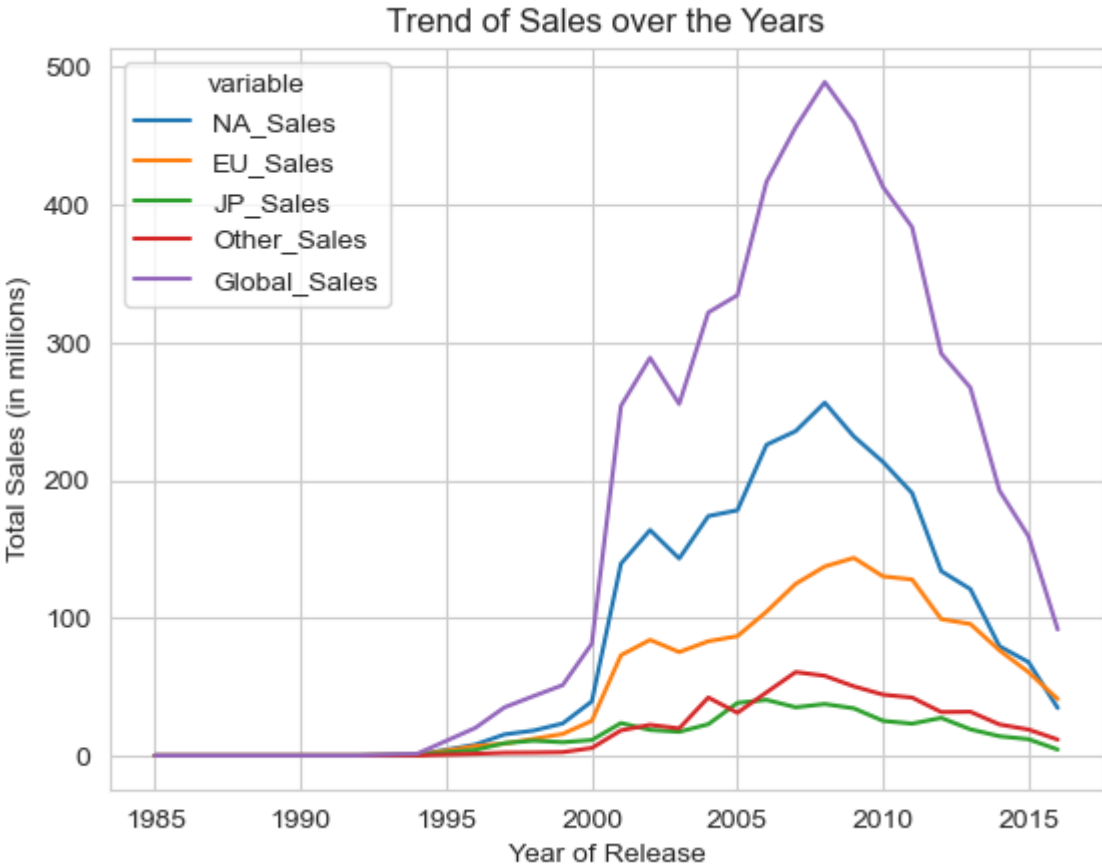
In [17]:

```
1 # barplot for other sales
2
3 print("  OTHERSALES  ")
4 top_other = df.sort_values(by="Other_Sales", ascending=False).head(10)
5 sns.barplot(x="Name", y="Other_Sales", data=top_other,)
6 plt.xticks(rotation=90)
7 plt.show()
```

OTHERSALES



```
In [18]: 1 yearly_sales = df.groupby("Year_of_Release")[["NA_Sales", "EU_Sales", "JP_Sales"]
2
3 sns.set_style("whitegrid")
4 sns.lineplot(x="Year_of_Release", y="value", hue="variable", data=pd.melt(yearly_
5 id_vars="Year_of_Release", var_name="variable", value_name="value"))
6 plt.title("Trend of Sales over the Years")
7 plt.xlabel("Year of Release")
8 plt.ylabel("Total Sales (in millions)")
9 plt.show()
```



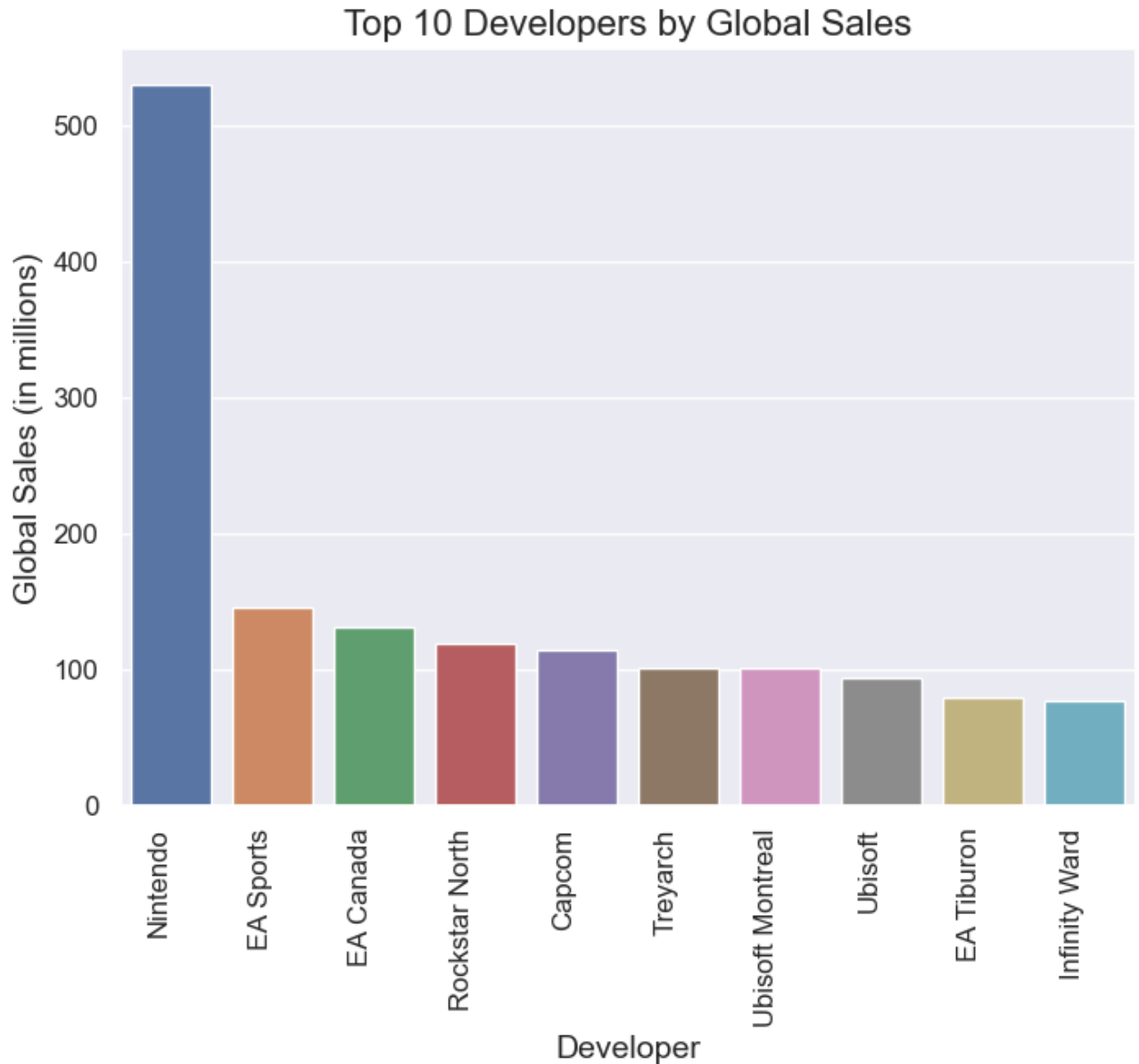
```
In [19]: 1 top_dev_sales = df.groupby("Developer")["Global_Sales"].sum().reset_index().sort_
2 top_dev_sales
```

Out[19]:

| | Developer | Global_Sales |
|------|------------------|--------------|
| 771 | Nintendo | 529.90 |
| 345 | EA Sports | 145.93 |
| 328 | EA Canada | 131.46 |
| 933 | Rockstar North | 119.47 |
| 204 | Capcom | 114.52 |
| 1165 | Treyarch | 101.37 |
| 1189 | Ubisoft Montreal | 101.24 |
| 1181 | Ubisoft | 94.53 |
| 349 | EA Tiburon | 79.77 |
| 553 | Infinity Ward | 77.56 |

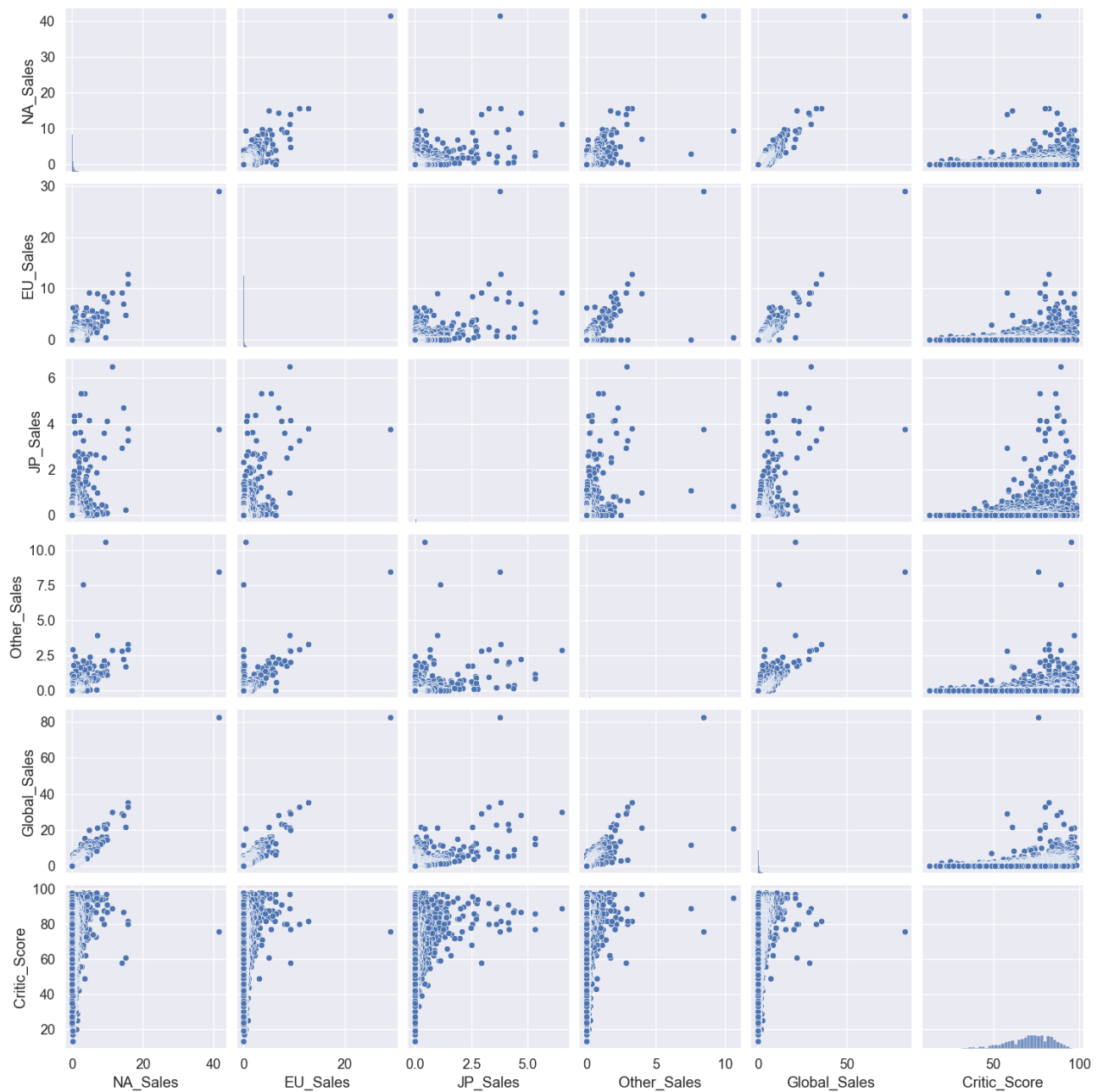
In [53]:

```
1 top_sales = developer_stats.sort_values(by="Global_Sales", ascending=False).head
2
3 fig, ax = plt.subplots(figsize=(8, 6))
4 sns.barplot(x="Developer", y="Global_Sales", data=top_sales, ax=ax)
5 ax.set_title("Top 10 Developers by Global Sales", fontsize=16)
6 ax.set_xlabel("Developer", fontsize=14)
7 ax.set_ylabel("Global Sales (in millions)", fontsize=14)
8 ax.tick_params(axis='both', labelsizes=12)
9 ax.set_xticklabels(ax.get_xticklabels(), rotation=90, ha="right")
10 plt.show()
```



```
In [21]: 1 cols = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales', 'Crit
2 sns.set(font_scale=1.2)
3 # Create the pairplot
4 sns.pairplot(df[cols].dropna(), diag_kind='hist')
```

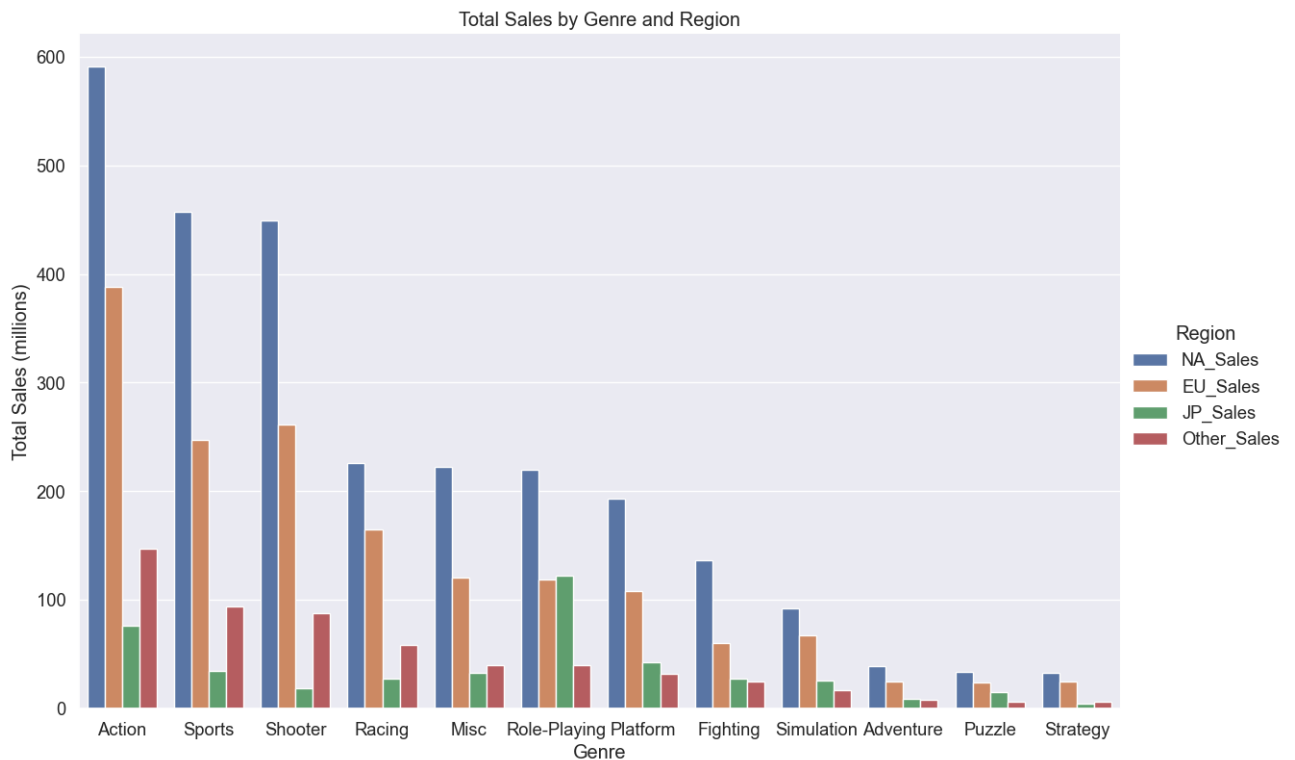
Out[21]: <seaborn.axisgrid.PairGrid at 0x2677d440eb0>



1. There appears to be a positive correlation between critic scores and sales, with higher scores generally corresponding to higher sales.

In [22]:

```
1 sales_by_genre = df.groupby('Genre')[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
2 sales_by_genre = sales_by_genre.sort_values(by='NA_Sales', ascending=False)
3
4 sales_by_genre = sales_by_genre.reset_index()
5 sales_by_genre = pd.melt(sales_by_genre, id_vars=['Genre'], var_name='Region', value_vars=['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales'])
6
7 sns.catplot(x='Genre', y='Sales', hue='Region', data=sales_by_genre, kind='bar',
8
9 plt.title('Total Sales by Genre and Region')
10 plt.xlabel('Genre')
11 plt.ylabel('Total Sales (millions)')
12 plt.show()
```

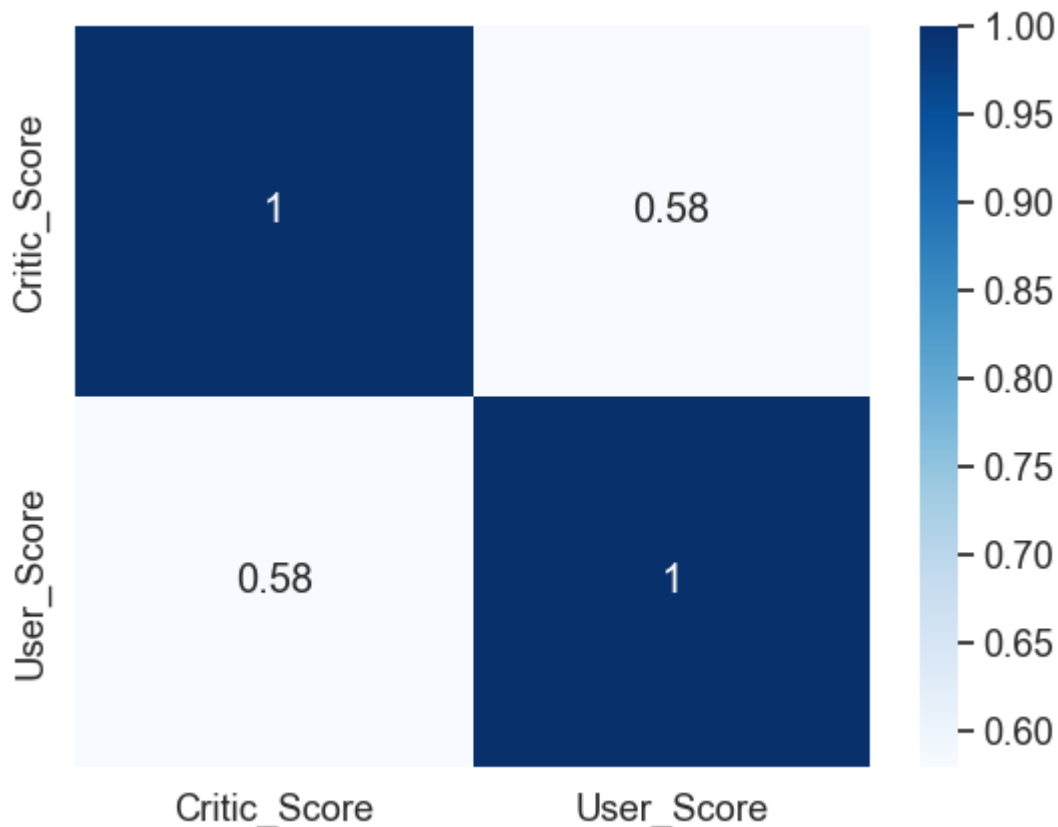


1.Action games are the most popular genre, followed by sports and shooters.

2.Adventure and puzzle games are the least popular genres.

```
In [23]: 1 sns.scatterplot(x='Critic_Score', y='User_Score', data=df, alpha=0.5)
2
3 # Create a heatmap showing the correlation between critic and user scores
4 sns.heatmap(df[['Critic_Score', 'User_Score']].corr(), cmap='Blues', annot=True)
```

Out[23]: <AxesSubplot:>



1. There is a weak positive correlation between critic scores and user scores.
2. Critics tend to give higher scores than users do.

FINDINGS

1. The top-selling platform is the PS2, followed by the X360 and the PS3.
2. The most common game genre is Action, followed by Sports and Shooter.
3. The most prolific publisher is Electronic Arts, followed by Activision and Namco Bandai Games.
4. The top-selling game of all time is Wii Sports, with over 80 million copies sold globally.
5. Sales of video games have generally increased over time, peaking in 2008 and declining since then.
6. The top developers by global sales are Nintendo, Electronic Arts, and Activision, while the top developers by rating score are Valve Corporation, Rockstar North, and Bethesda Game Studios.
7. There is a weak positive correlation between critic score and global sales, but no significant correlation between user score and global sales.
8. Video games with higher ratings tend to have more sales in all regions, with Japan having the weakest correlation.
9. The dataset covers video games released from 1980 to 2016, with a concentration of games released in the 2000s and 2010s.

