

analysis

July 25, 2023

```
[203]: # Imported Requirement
import numpy as np
import pandas as pd
import missingno as ms
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")
```

```
[142]: # Data Loaded in "df" variable
df = pd.read_csv('Tagged-Data-Final.csv')
```

```
[143]: df
```

```
[143]:
```

	Name	Year_of_Release	Genre	\
0	.hack//Infection Part 1	2002.0	Role-Playing	
1	.hack//Mutation Part 2	2002.0	Role-Playing	
2	.hack//Outbreak Part 3	2002.0	Role-Playing	
3	[Prototype]	2009.0	Action	
4	[Prototype]	2009.0	Action	
...	
6889	Zubo	2008.0	Misc	
6890	Zumba Fitness	2010.0	Sports	
6891	Zumba Fitness: World Party	2013.0	Misc	
6892	Zumba Fitness Core	2012.0	Misc	
6893	Zumba Fitness Rush	2012.0	Sports	

	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	\
0	Atari	0.49	0.38	0.26	0.13	
1	Atari	0.23	0.18	0.20	0.06	
2	Atari	0.14	0.11	0.17	0.04	
3	Activision	0.84	0.35	0.00	0.12	
4	Activision	0.65	0.40	0.00	0.19	
...	
6889	Electronic Arts	0.08	0.02	0.00	0.01	
6890	505 Games	1.74	0.45	0.00	0.18	
6891	Majesco Entertainment	0.17	0.05	0.00	0.02	
6892	505 Games	0.00	0.05	0.00	0.00	

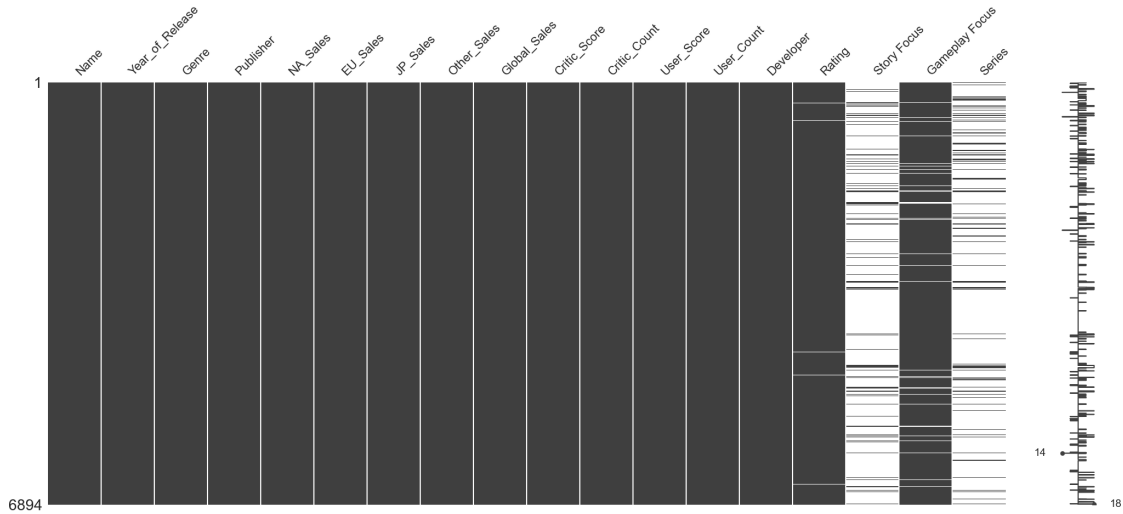
6893	505 Games	0.00	0.16	0.00	0.02
------	-----------	------	------	------	------

	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count \
0	1.27	75.0	35.0	8.5	60.0
1	0.68	76.0	24.0	8.9	81.0
2	0.46	70.0	23.0	8.7	19.0
3	1.31	78.0	83.0	7.8	356.0
4	1.24	79.0	53.0	7.7	308.0
...
6889	0.11	75.0	19.0	7.6	75.0
6890	2.37	42.0	10.0	5.5	16.0
6891	0.24	73.0	5.0	6.2	40.0
6892	0.05	77.0	6.0	6.7	6.0
6893	0.18	73.0	7.0	6.2	5.0

	Developer	Rating	Story Focus	Gameplay Focus	Series
0	CyberConnect2	T	x	NaN	x
1	CyberConnect2	T	x	NaN	x
2	CyberConnect2	T	x	NaN	x
3	Radical Entertainment	M	NaN	x	x
4	Radical Entertainment	M	NaN	x	x
...
6889	EA Bright Light	E10+	NaN	x	NaN
6890	Pipeworks Software, Inc.	E	NaN	x	NaN
6891	Zoe Mode	E	NaN	x	NaN
6892	Zoe Mode	E10+	NaN	x	NaN
6893	Majesco Games, Majesco	E10+	NaN	x	NaN

[6894 rows x 18 columns]

```
[144]: # Visulized the null values
ms.matrix(df)
plt.show()
```



```
[145]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6894 entries, 0 to 6893
Data columns (total 18 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Name                6894 non-null   object
1   Year_of_Release     6894 non-null   float64
2   Genre               6894 non-null   object
3   Publisher           6893 non-null   object
4   NA_Sales            6894 non-null   float64
5   EU_Sales            6894 non-null   float64
6   JP_Sales            6894 non-null   float64
7   Other_Sales         6894 non-null   float64
8   Global_Sales        6894 non-null   float64
9   Critic_Score        6894 non-null   float64
10  Critic_Count        6894 non-null   float64
11  User_Score          6894 non-null   float64
12  User_Count          6894 non-null   float64
13  Developer           6890 non-null   object
14  Rating              6826 non-null   object
15  Story Focus         767 non-null    object
16  Gameplay Focus      6586 non-null   object
17  Series              791 non-null    object
dtypes: float64(10), object(8)
memory usage: 969.6+ KB
```

```
[146]: df.isnull().sum()
```

```
[146]: Name          0
      Year_of_Release  0
      Genre          0
      Publisher      1
      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      4
      Rating         68
      Story Focus    6127
      Gameplay Focus  308
      Series         6103
      dtype: int64
```

```
[147]: df.duplicated().sum()
```

```
[147]: 0
```

```
[148]: # Replaced all x values with 1 and null values with 0
df['Story Focus'] = df['Story Focus'].apply(lambda x: 1 if x == 'x' else 0)
df['Gameplay Focus'] = df['Gameplay Focus'].apply(lambda x: 1 if x == 'x' else 0)
df['Series'] = df['Series'].apply(lambda x: 1 if x == 'x' else 0)
```

```
[149]: df=df.dropna()
```

```
[150]: df.isnull().sum()
```

```
[150]: Name          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
Story Focus    0
Gameplay Focus 0
Series         0
dtype: int64
```

```
[151]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 6825 entries, 0 to 6893
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Name                   6825 non-null   object
1   Year_of_Release       6825 non-null   float64
2   Genre                 6825 non-null   object
3   Publisher              6825 non-null   object
4   NA_Sales               6825 non-null   float64
5   EU_Sales               6825 non-null   float64
6   JP_Sales               6825 non-null   float64
7   Other_Sales            6825 non-null   float64
8   Global_Sales           6825 non-null   float64
9   Critic_Score           6825 non-null   float64
10  Critic_Count           6825 non-null   float64
11  User_Score             6825 non-null   float64
12  User_Count             6825 non-null   float64
13  Developer              6825 non-null   object
14  Rating                 6825 non-null   object
15  Story Focus            6825 non-null   int64
16  Gameplay Focus         6825 non-null   int64
17  Series                 6825 non-null   int64
dtypes: float64(10), int64(3), object(5)
memory usage: 1013.1+ KB
```

```
[152]: df.describe()
```

```
[152]:
```

	Year_of_Release	NA_Sales	EU_Sales	JP_Sales	Other_Sales	\
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	
mean	2007.436777	0.394484	0.236089	0.064158	0.082677	
std	4.211248	0.967385	0.687330	0.287570	0.269871	
min	1985.000000	0.000000	0.000000	0.000000	0.000000	
25%	2004.000000	0.060000	0.020000	0.000000	0.010000	
50%	2007.000000	0.150000	0.060000	0.000000	0.020000	
75%	2011.000000	0.390000	0.210000	0.010000	0.070000	
max	2016.000000	41.360000	28.960000	6.500000	10.570000	

	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	\
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	
mean	0.777590	70.272088	28.931136	7.185626	174.722344	
std	1.963443	13.868572	19.224165	1.439942	587.428538	
min	0.010000	13.000000	3.000000	0.500000	4.000000	
25%	0.110000	62.000000	14.000000	6.500000	11.000000	
50%	0.290000	72.000000	25.000000	7.500000	27.000000	
75%	0.750000	80.000000	39.000000	8.200000	89.000000	
max	82.530000	98.000000	113.000000	9.600000	10665.000000	

	Story Focus	Gameplay Focus	Series
count	6825.000000	6825.000000	6825.000000
mean	0.110916	0.955165	0.115165
std	0.314051	0.206957	0.319244
min	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000
50%	0.000000	1.000000	0.000000
75%	0.000000	1.000000	0.000000
max	1.000000	1.000000	1.000000

[153]: df

[153]:

	Name	Year_of_Release	Genre	\
0	.hack//Infection Part 1	2002.0	Role-Playing	
1	.hack//Mutation Part 2	2002.0	Role-Playing	
2	.hack//Outbreak Part 3	2002.0	Role-Playing	
3	[Prototype]	2009.0	Action	
4	[Prototype]	2009.0	Action	
...	
6889	Zubo	2008.0	Misc	
6890	Zumba Fitness	2010.0	Sports	
6891	Zumba Fitness: World Party	2013.0	Misc	
6892	Zumba Fitness Core	2012.0	Misc	
6893	Zumba Fitness Rush	2012.0	Sports	

	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	\
0	Atari	0.49	0.38	0.26	0.13	
1	Atari	0.23	0.18	0.20	0.06	
2	Atari	0.14	0.11	0.17	0.04	
3	Activision	0.84	0.35	0.00	0.12	
4	Activision	0.65	0.40	0.00	0.19	
...	
6889	Electronic Arts	0.08	0.02	0.00	0.01	
6890	505 Games	1.74	0.45	0.00	0.18	
6891	Majesco Entertainment	0.17	0.05	0.00	0.02	
6892	505 Games	0.00	0.05	0.00	0.00	
6893	505 Games	0.00	0.16	0.00	0.02	

	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	\
0	1.27	75.0	35.0	8.5	60.0	
1	0.68	76.0	24.0	8.9	81.0	
2	0.46	70.0	23.0	8.7	19.0	
3	1.31	78.0	83.0	7.8	356.0	
4	1.24	79.0	53.0	7.7	308.0	
...	
6889	0.11	75.0	19.0	7.6	75.0	
6890	2.37	42.0	10.0	5.5	16.0	
6891	0.24	73.0	5.0	6.2	40.0	
6892	0.05	77.0	6.0	6.7	6.0	
6893	0.18	73.0	7.0	6.2	5.0	

	Developer	Rating	Story Focus	Gameplay Focus	Series
0	CyberConnect2	T	1	0	1
1	CyberConnect2	T	1	0	1
2	CyberConnect2	T	1	0	1
3	Radical Entertainment	M	0	1	1
4	Radical Entertainment	M	0	1	1
...
6889	EA Bright Light	E10+	0	1	0
6890	Pipeworks Software, Inc.	E	0	1	0
6891	Zoe Mode	E	0	1	0
6892	Zoe Mode	E10+	0	1	0
6893	Majesco Games, Majesco	E10+	0	1	0

[6825 rows x 18 columns]

```
[154]: df.groupby(['Year_of_Release'])['Year_of_Release'].count()
```

```
[154]: Year_of_Release
1985.0    1
1988.0    1
1992.0    1
1994.0    1
1996.0    7
1997.0   13
1998.0   25
1999.0   30
2000.0  102
2001.0  256
2002.0  455
2003.0  498
2004.0  476
2005.0  562
2006.0  528
```

```

2007.0    590
2008.0    592
2009.0    550
2010.0    429
2011.0    453
2012.0    313
2013.0    266
2014.0    253
2015.0    211
2016.0    212
Name: Year_of_Release, dtype: int64

```

Sales Trend analysis

```

[155]: # Group the data by 'Year_of_Release' and calculate the sum of 'Global_Sales'
      ↪for each year
sales_by_year = df.
      ↪groupby('Year_of_Release')[['Global_Sales', 'Other_Sales', 'NA_Sales', 'EU_Sales', 'JP_Sales']]
      ↪sum().reset_index()
sales_by_year

```

```

[155]:
   Year_of_Release  Global_Sales  Other_Sales  NA_Sales  EU_Sales  JP_Sales
0         1985.0         0.03         0.01         0.00         0.03         0.00
1         1988.0         0.03         0.01         0.00         0.02         0.00
2         1992.0         0.03         0.00         0.02         0.00         0.00
3         1994.0         1.27         0.08         0.39         0.26         0.53
4         1996.0        20.10         1.24         7.91         6.88         4.06
5         1997.0        35.01         2.02        15.34         8.67         9.01
6         1998.0        43.18         2.14        18.13        12.13        10.81
7         1999.0        51.17         2.45        23.32        15.69         9.67
8         2000.0        81.24         5.49        39.34        25.20        11.27
9         2001.0       253.88        18.26       139.32        72.85        23.57
10        2002.0       288.84        22.30       163.76        84.03        18.61
11        2003.0       255.35        19.68       143.08        75.16        17.24
12        2004.0       321.78        42.14       173.88        83.01        22.74
13        2005.0       334.32        31.05       178.15        86.70        38.23
14        2006.0       416.72        45.90       225.69       104.53        40.43
15        2007.0       456.23        60.62       235.61       124.71        35.04
16        2008.0       489.12        57.89       256.25       137.31        37.42
17        2009.0       459.85        50.25       231.72       143.56        34.28
18        2010.0       412.96        44.24       213.24       130.13        25.19
19        2011.0       383.69        42.10       190.62       127.86        23.16
20        2012.0       291.93        31.57       133.94        99.08        27.36
21        2013.0       267.17        31.80       120.89        95.54        19.05
22        2014.0       192.43        22.58        79.38        76.42        14.02
23        2015.0       159.16        18.86        67.85        60.51        11.85
24        2016.0        91.56        11.59        34.52        41.03         4.34

```



```
[156]: col_list = sales_by_year.columns.tolist()
col_list
```

```
[156]: ['Year_of_Release',
'Global_Sales',
'Other_Sales',
'NA_Sales',
'EU_Sales',
'JP_Sales']
```

```
[157]: # Create a line plot for the global sales trend using seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year_of_Release', y='Global_Sales', data=sales_by_year,
↪marker='o')
plt.xlabel('Year')
plt.ylabel('Global Sales (Millions)')
plt.title('Global Video Game Sales Trend')
```

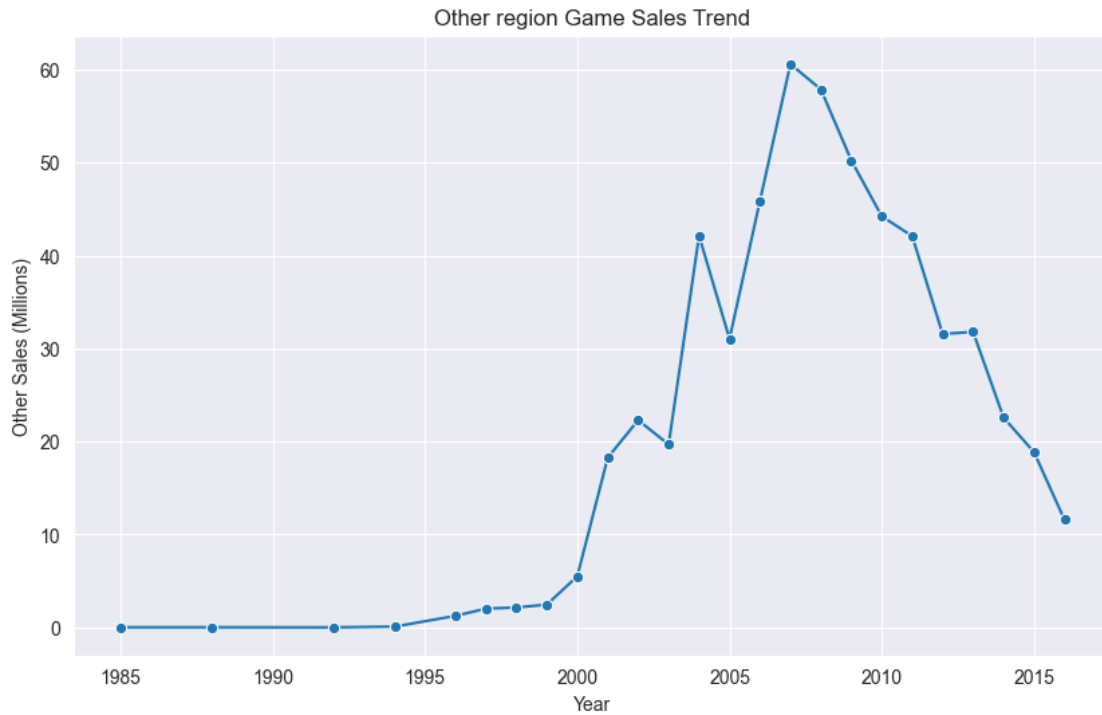
```
[157]: Text(0.5, 1.0, 'Global Video Game Sales Trend')
```



```
[158]: # Create a line plot for the Other Sales trend using seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year_of_Release', y='Other_Sales', data=sales_by_year,
↪marker='o')
```

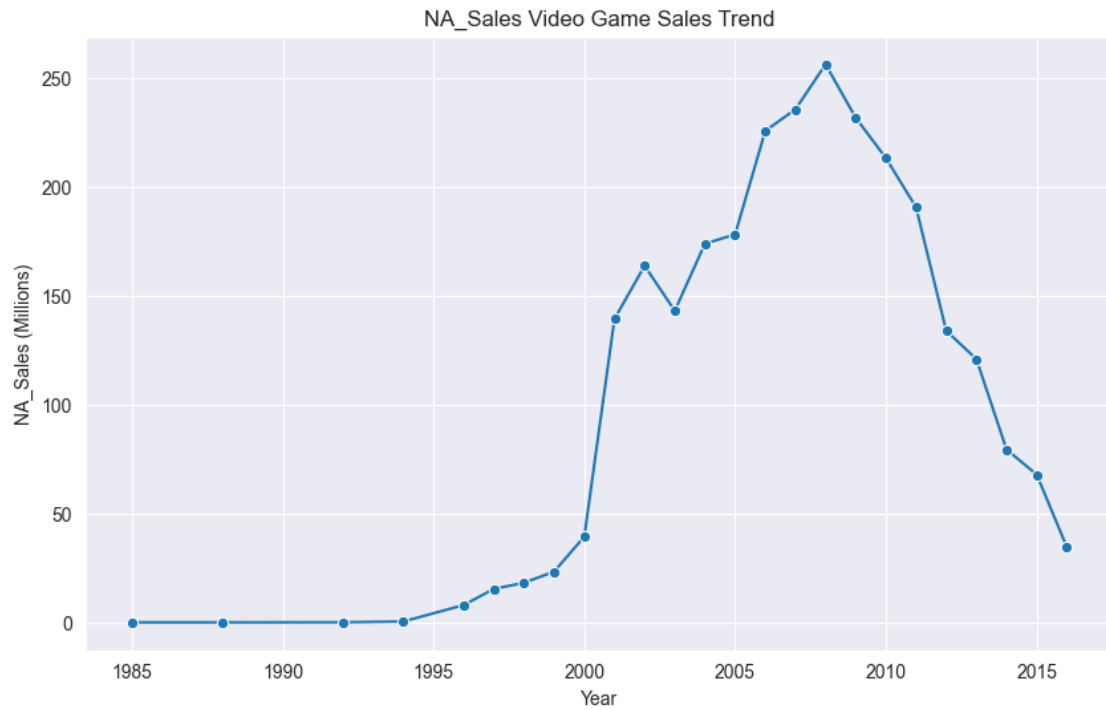
```
plt.xlabel('Year')
plt.ylabel('Other Sales (Millions)')
plt.title('Other region Game Sales Trend')
```

[158]: Text(0.5, 1.0, 'Other region Game Sales Trend')



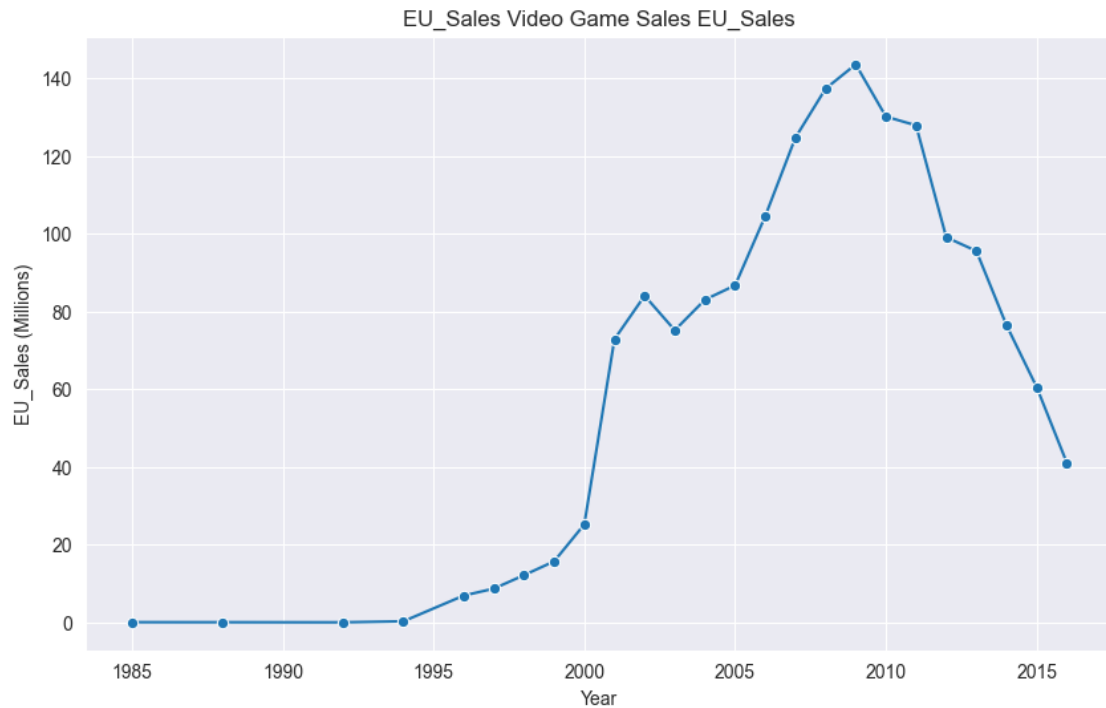
```
[159]: # Create a line plot for the NA_Sales trend using seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year_of_Release', y='NA_Sales', data=sales_by_year, marker='o')
plt.xlabel('Year')
plt.ylabel('NA_Sales (Millions)')
plt.title('NA_Sales Video Game Sales Trend')
```

[159]: Text(0.5, 1.0, 'NA_Sales Video Game Sales Trend')



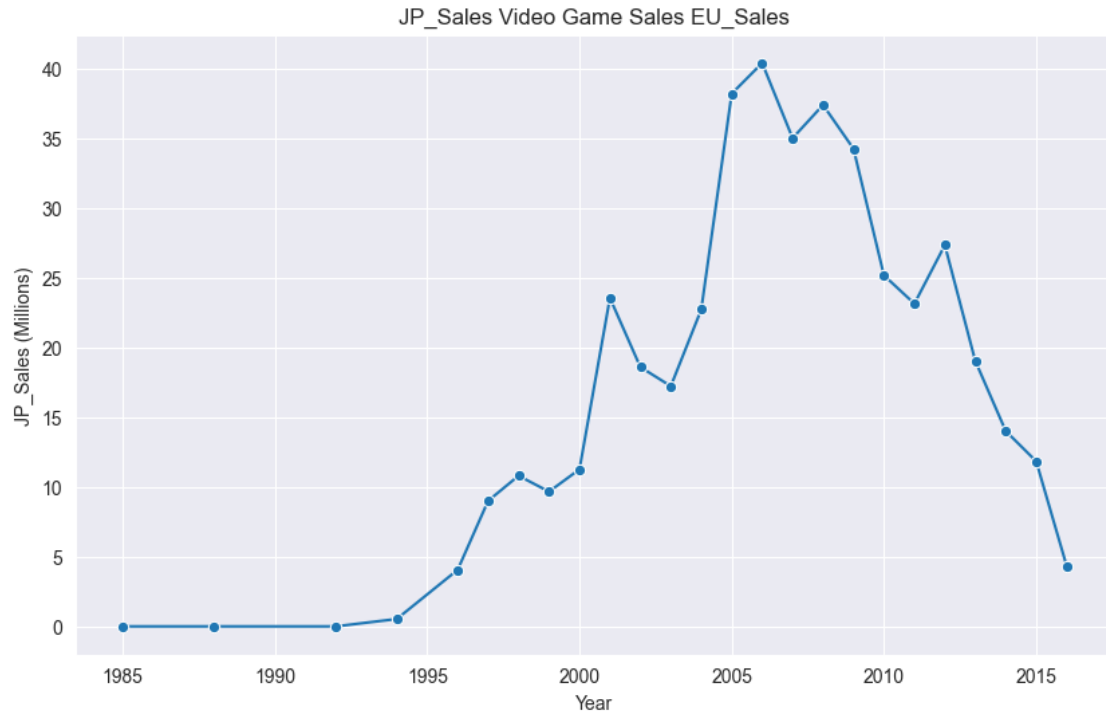
```
[160]: # Create a line plot for the EU_Sales trend using seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year_of_Release',y= 'EU_Sales', data=sales_by_year, marker='o')
plt.xlabel('Year')
plt.ylabel('EU_Sales (Millions)')
plt.title('EU_Sales Video Game Sales EU_Sales')
```

```
[160]: Text(0.5, 1.0, 'EU_Sales Video Game Sales EU_Sales')
```



```
[161]: # Create a line plot for the JP_Sales trend using seaborn
plt.figure(figsize=(10, 6))
sns.lineplot(x='Year_of_Release', y='JP_Sales', data=sales_by_year, marker='o')
plt.xlabel('Year')
plt.ylabel('JP_Sales (Millions)')
plt.title('JP_Sales Video Game Sales EU_Sales')
```

```
[161]: Text(0.5, 1.0, 'JP_Sales Video Game Sales EU_Sales')
```



Insights

Consistent Sales Trends: The sales trends for video games in North America, Europe, Japan, and other regions exhibit a similar pattern over the years. This suggests that the gaming industry experiences consistent sales growth across different markets.

Genre-wise Sales Analysis:

```
[176]: genre_wise_sales = df.
        ↳groupby('Genre')[['Global_Sales', 'Other_Sales', 'NA_Sales', 'EU_Sales', 'JP_Sales']].
        ↳sum().reset_index()
genre_wise_sales
```

```
[176]:
```

	Genre	Global_Sales	Other_Sales	NA_Sales	EU_Sales	JP_Sales
0	Action	1203.16	147.46	591.23	387.78	76.17
1	Adventure	80.75	8.05	38.81	25.06	8.73
2	Fighting	249.95	25.14	136.39	60.50	27.85
3	Misc	416.26	40.41	222.05	120.46	33.22
4	Platform	377.80	32.53	193.60	108.56	43.09
5	Puzzle	78.90	6.33	33.50	24.04	14.98
6	Racing	476.22	58.29	225.59	164.66	27.76
7	Role-Playing	501.37	40.36	219.53	119.20	122.47
8	Shooter	816.48	87.90	448.76	261.12	18.57
9	Simulation	202.70	17.14	92.12	67.29	26.16
10	Sports	833.85	94.05	457.52	247.43	34.54

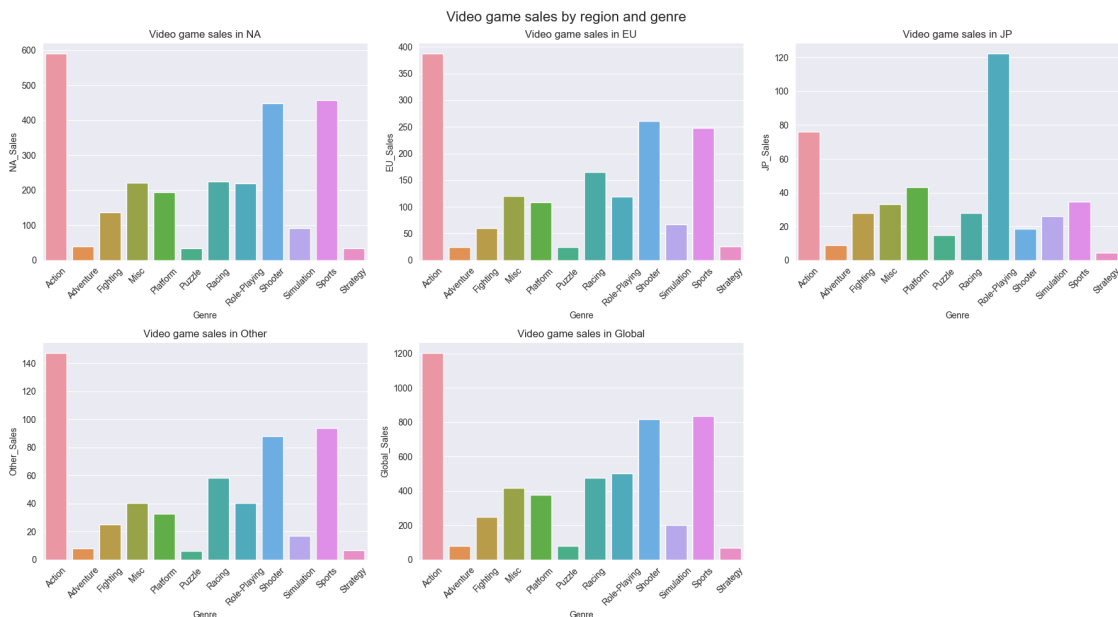
11 Strategy 69.61 6.61 33.25 25.21 4.34

```
[175]: fig, axes = plt.subplots(2, 3, figsize=(18, 10))
axes = axes.flatten()
axes[-1].remove()

sales_regions = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales",
                 ↪"Global_Sales"]

for i, sales_region in enumerate(sales_regions):
    sns.barplot(data=genre_wise_sales, x='Genre', y=sales_region, ax=axes[i]).
    ↪set_title("Video game sales in " + sales_region.split("_Sales")[0])
    axes[i].tick_params(axis='x', rotation=45)

plt.suptitle("Video game sales by region and genre", fontsize=16)
plt.tight_layout()
plt.show()
```



Insight

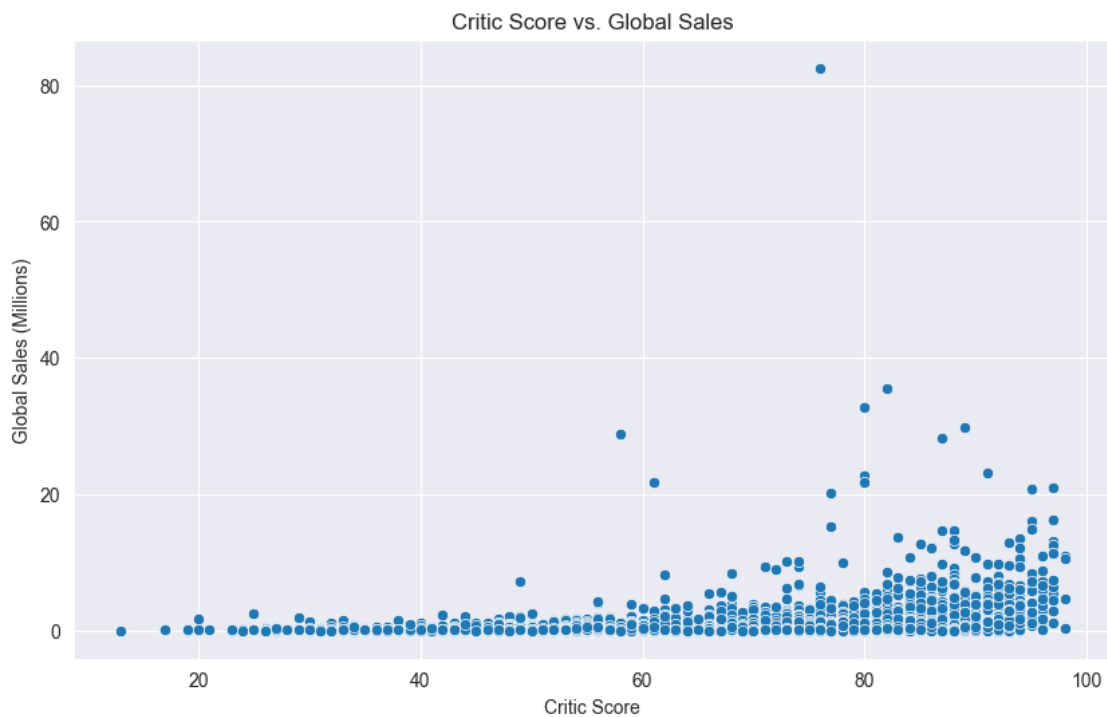
Top-Selling Genres: The “Action” genre has the highest global sales, with over 1203 million units sold. It is followed by “Shooter” and “Sports” genres, each with global sales of over 800 million units.

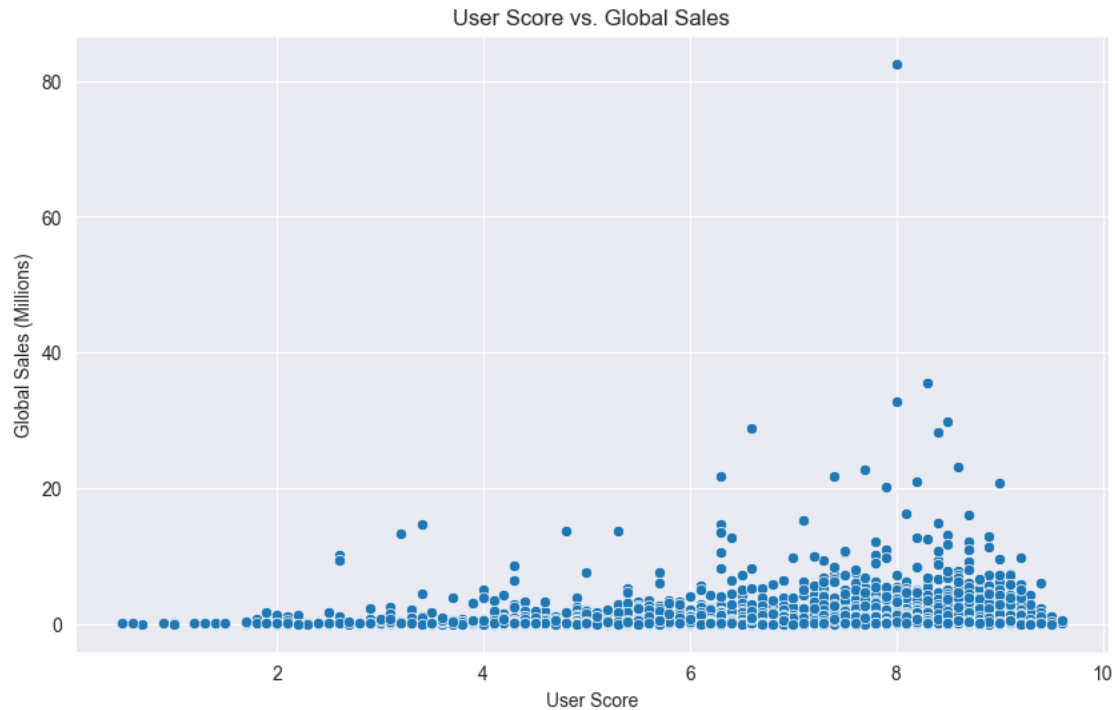
Market Potential: “Simulation” and “Strategy” genres have relatively lower sales compared to other genres. Game developers might consider exploring untapped market potential for these genres.

Score review analysis

```
[179]: # Scatter plot for Critic Score vs. Global Sales
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Critic_Score', y='Global_Sales')
plt.xlabel('Critic Score')
plt.ylabel('Global Sales (Millions)')
plt.title('Critic Score vs. Global Sales')
plt.grid(True)
plt.show()

# Scatter plot for User Score vs. Global Sales
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='User_Score', y='Global_Sales')
plt.xlabel('User Score')
plt.ylabel('Global Sales (Millions)')
plt.title('User Score vs. Global Sales')
plt.grid(True)
plt.show()
```





Insights

Critic Scores Concentration: The data points for critic scores are highly concentrated in the range of 60 to 100. This indicates that most of the video games received higher critic scores, suggesting that professional critics tend to give favorable ratings to many games.

Critical Reception Impact: The concentration of data points in the higher critic score range suggests that games with higher critic scores are relatively common and well-received by professional critics.

Outliers: While the majority of games have higher critic scores, there are a few outliers with lower scores that have exceptionally high global sales. These outliers could represent games that have gained commercial success despite receiving lower critical acclaim.

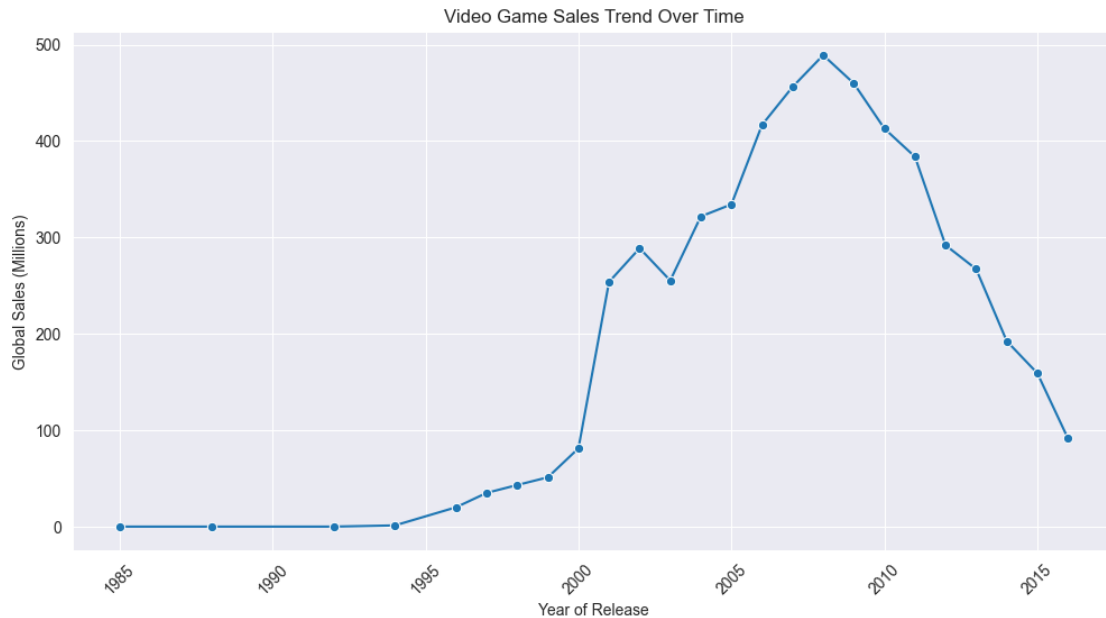
Global Trend Over the years

```
[180]: # Group the data by 'Year_of_Release' and calculate the total global sales for
      ↪ each year
yearly_global_sales = df.groupby('Year_of_Release')['Global_Sales'].sum().
      ↪ reset_index()

# Line plot for Year of Release vs. Global Sales
plt.figure(figsize=(12, 6))
sns.lineplot(data=yearly_global_sales, x='Year_of_Release', y='Global_Sales',
      ↪ marker='o')
plt.xlabel('Year of Release')
plt.ylabel('Global Sales (Millions)')
```



```
plt.title('Video Game Sales Trend Over Time')
plt.grid(True)
plt.xticks(rotation=45) # Rotate x-axis labels for better visibility
plt.show()
```



Insights

It is observable that sales have been spiked since 2000 to 2007-08 and then started to decline after that till 2015.

Relationship between the “Publishers” and “Global Sales”

```
[199]: publisher_game_count = df.groupby(['Publisher', 'Story Focus', 'Gameplay Focus',
    ↪Focus'])['Name'].count().reset_index()
publisher_game_count_sorted = publisher_game_count.sort_values(by='Name',
    ↪ascending=False)
publisher_game_count_sorted
```

```
[199]:
```

	Publisher	Story Focus	Gameplay Focus	Name
93	Electronic Arts	0	1	901
16	Activision	0	1	479
330	Ubisoft	0	1	468
305	THQ	0	1	297
276	Sony Computer Entertainment	0	1	292
..
176	Kool Kizz	0	1	1
177	Level 5	0	1	1
178	Lexicon Entertainment	0	1	1

179	Lighthouse Interactive	0	1	1
365	inXile Entertainment	1	0	1

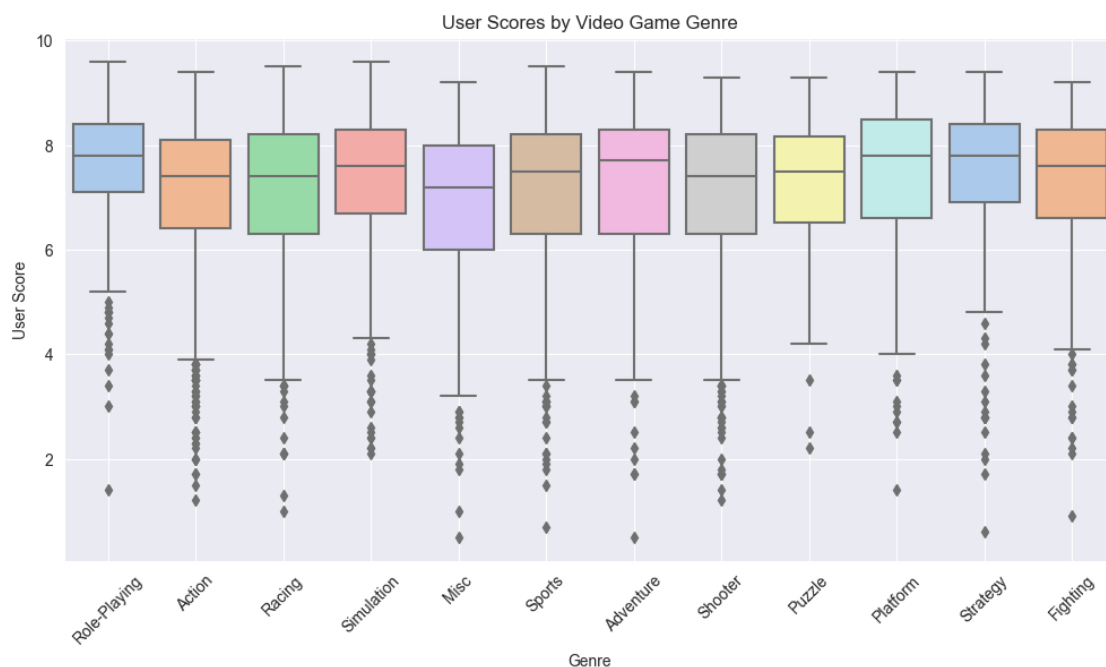
[366 rows x 4 columns]

Insights

Dominant Publishers: The list starts with the top publishers based on the number of games they have released. Electronic Arts leads the pack with 901 games, followed by Activision with 479 games and Ubisoft with 468 games. These publishers are major players in the industry and have a significant presence in the market.

Relationship between “Genre” and “User Score”

```
[200]: # Box plot for Genre vs. User Score
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='Genre', y='User_Score', palette='pastel')
plt.xlabel('Genre')
plt.ylabel('User Score')
plt.title('User Scores by Video Game Genre')
plt.grid(True)
plt.xticks(rotation=45) # Rotate x-axis labels for better visibility
plt.show()
```



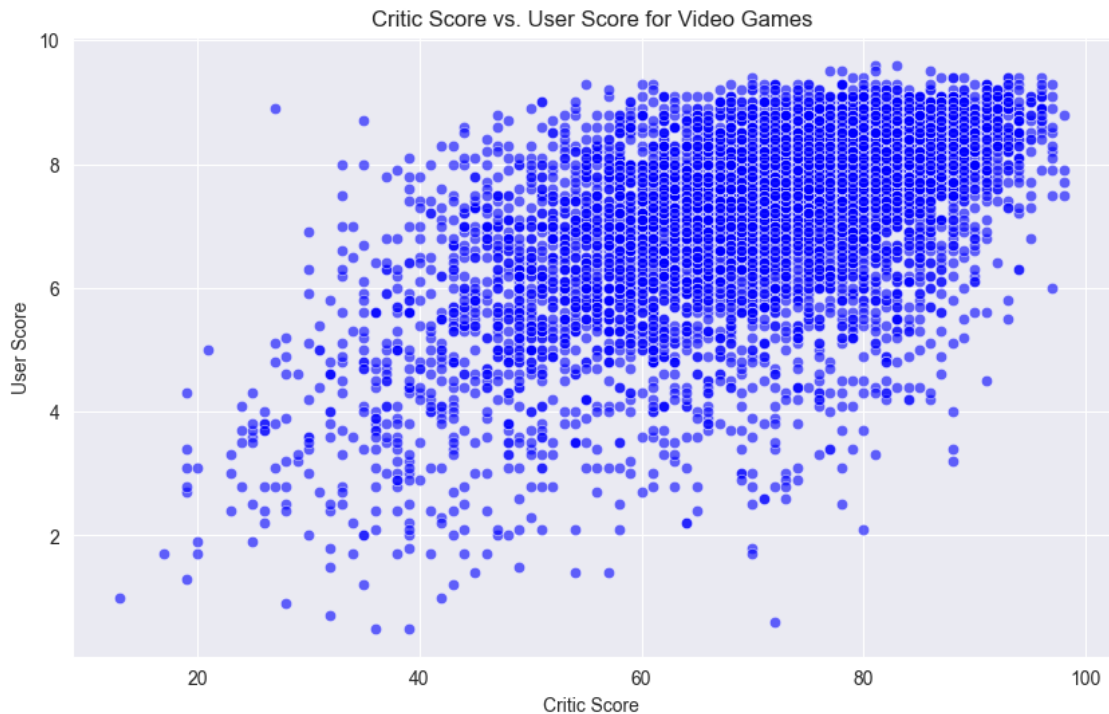
Insight

Positive User Reception: The fact that the box plot is mainly within the 6 to 9 range suggests

that users generally have a positive reception to video games across various genres. This positive sentiment can be an encouraging sign for game developers and publishers, as it indicates that players tend to enjoy a wide range of video games.

Relationship between “Critic Score” and “User Score”

```
[201]: # Scatter plot for Critic Score vs. User Score
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Critic_Score', y='User_Score', alpha=0.6,
               color='blue')
plt.xlabel('Critic Score')
plt.ylabel('User Score')
plt.title('Critic Score vs. User Score for Video Games')
plt.grid(True)
plt.show()
```



Sales performance of video games across different regions

```
[202]: # Group the data by 'Genre' and calculate the total sales for each genre in
        different regions
region_sales = df.groupby('Genre')[['NA_Sales', 'EU_Sales', 'JP_Sales',
        'Other_Sales']].sum().reset_index()

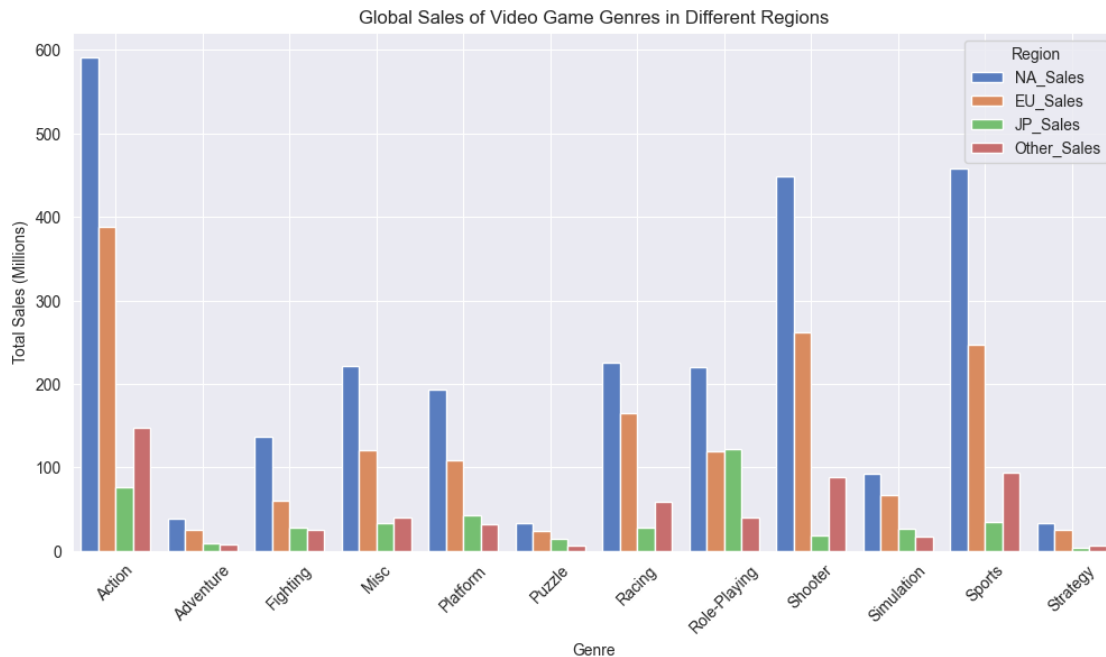
# Melt the data to create a long-form dataframe suitable for stacked bar plots
```

```

region_sales_melted = region_sales.melt(id_vars='Genre', var_name='Region',
    ↪value_name='Sales')

# Stacked bar plot for Genre vs. Sales in different regions
plt.figure(figsize=(12, 6))
sns.barplot(data=region_sales_melted, x='Genre', y='Sales', hue='Region',
    ↪palette='muted')
plt.xlabel('Genre')
plt.ylabel('Total Sales (Millions)')
plt.title('Global Sales of Video Game Genres in Different Regions')
plt.grid(True)
plt.xticks(rotation=45) # Rotate x-axis labels for better visibility
plt.legend(title='Region', loc='upper right')
plt.show()

```



Inshights

Regional Preferences: It is observed that in all Genre NA_Sales has been max sales showing interest of NA Gamers, After that EU_sales are the one which shows great sales.

Importance of Regional Marketing: Need to focus on Genre of other regions and need to market accordingly.

Final Inshights

Global Video Game Sales Trend: The analysis of global video game sales over the years reveals a steady growth in the industry. The sales have experienced consistent year-on-year growth, indicating a healthy and thriving market for video games worldwide.

Regional Sales Preferences: There are notable regional differences in video game sales preferences. North America and Europe exhibit higher sales across various genres, while Japan shows a strong preference for role-playing games (RPGs). Other regions also contribute significantly to the global sales, but with varying genre preferences.

Top-Selling Genres: The analysis identifies action, sports, shooter, and role-playing genres as the top-selling genres, accounting for a significant share of global video game sales. These genres have widespread popularity and appeal to a broad audience.

Influence of User and Critic Scores: User scores and critic scores show a positive correlation, indicating that games highly praised by critics tend to receive positive user ratings as well. This alignment suggests that critically acclaimed games are well-received by players, contributing to their commercial success.

Yearly Sales Trends: Certain genres have witnessed fluctuating sales trends over the years, influenced by changing market demands, technological advancements, and gaming trends.

[]: