Importing Libs and Treating Data

In [ ]:
```
import matplotlib.pyplot as plt # plotting
import numpy as np # linear algebra
import os # accessing directory structure
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

In [ ]:
```
vg_df = pd.read_csv(r'F:\Aprendizado\Projetos\video game\vgsales.csv')
vg_df.head(5)
```

Out[ ]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10

In [ ]:
```
vg_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16598 non-null  int64
 1   Name          16598 non-null  object
 2   Platform      16598 non-null  object
 3   Year          16327 non-null  float64
 4   Genre         16598 non-null  object
 5   Publisher     16540 non-null  object
 6   NA_Sales      16598 non-null  float64
 7   EU_Sales      16598 non-null  float64
 8   JP_Sales      16598 non-null  float64
 9   Other_Sales   16598 non-null  float64
 10  Global_Sales  16598 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

In [ ]:
```
# Fixing Year Column
print(vg_df['Year'].unique())
vg_df['Year'] = vg_df['Year'].fillna(0).astype(int)
```

```
[2006. 1985. 2008. 2009. 1996. 1989. 1984. 2005. 1999. 2007. 2010. 2013.
 2004. 1990. 1988. 2002. 2001. 2011. 1998. 2015. 2012. 2014. 1992. 1997.
 1993. 1994. 1982. 2003. 1986. 2000.   nan 1995. 2016. 1991. 1981. 1987.
 1980. 1983. 2020. 2017.]
```

1. Which Gaming Platform Dominates the Global Market?

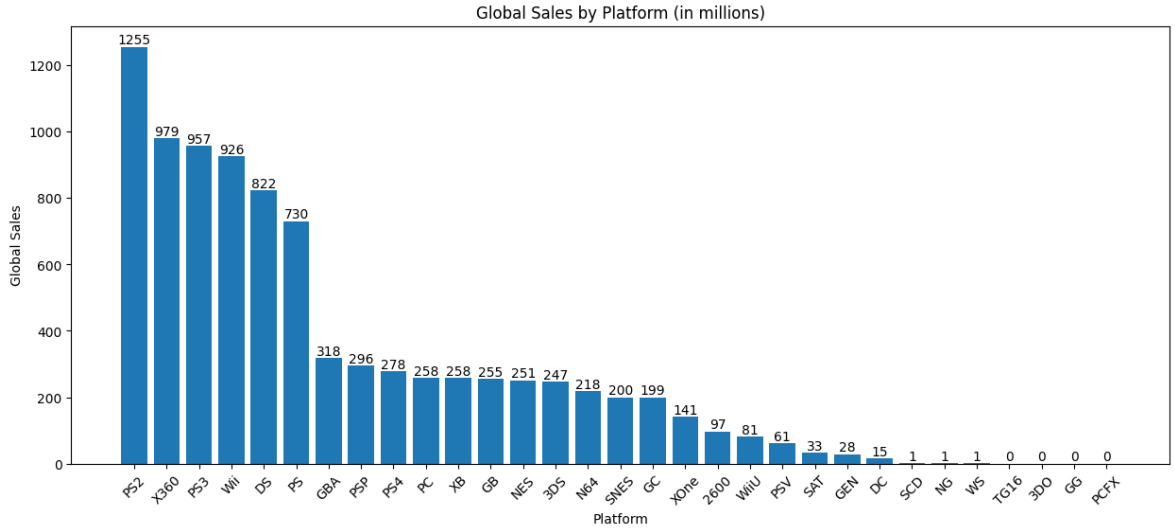
Analyze global sales data to pinpoint the gaming platform with the highest total sales. This analysis will shed light on the most prevalent platform in the market.

In [ ]:
```
platform_sales = vg_df.groupby('Platform')['Global_Sales'].sum()
platform_sales = platform_sales.sort_values(ascending=False)
print(platform_sales)
```

```
Platform
PS2     1255.64
X360     979.96
PS3      957.84
Wii      926.71
DS       822.49
PS       730.66
GBA      318.50
PSP      296.28
PS4      278.10
PC       258.82
XB       258.26
GB       255.45
NES      251.07
3DS      247.46
N64      218.88
SNES     200.05
GC       199.36
XOne     141.06
2600      97.08
WiiU      81.86
PSV       61.93
SAT       33.59
GEN       28.36
DC        15.97
SCD        1.87
NG         1.44
WS         1.42
TG16       0.16
3DO        0.10
GG         0.04
PCFX       0.03
Name: Global_Sales, dtype: float64
```

In [ ]:
```
# Create a bar graph
plt.figure(figsize=(15, 6))

bars = plt.bar(platform_sales.index, platform_sales.values)
plt.xlabel('Platform')
plt.ylabel('Global Sales')
plt.title('Global Sales by Platform (in millions)')

for bar, value in zip(bars, platform_sales.values):
    plt.text(bar.get_x() + bar.get_width() / 2, value, str(int(value)), ha='cent
```

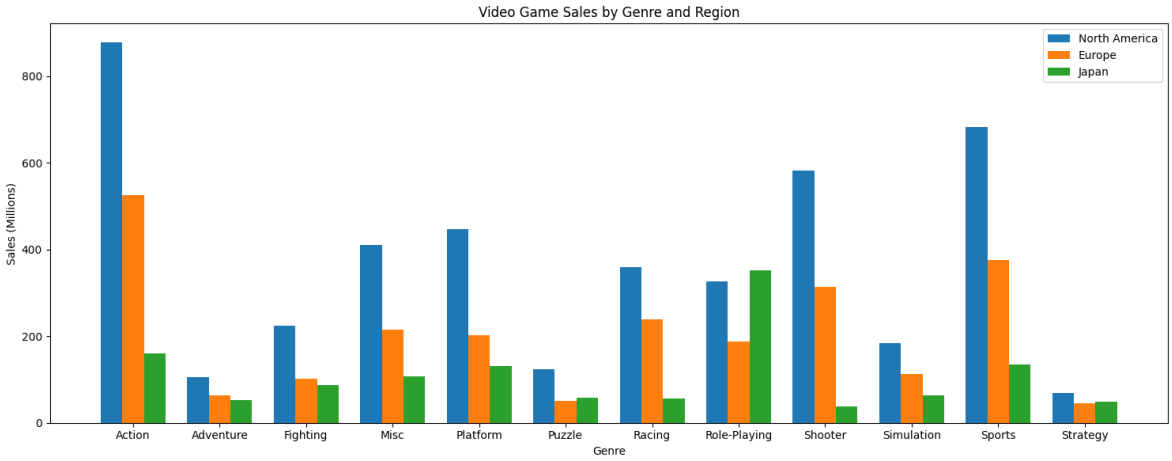
```
# Show the plot
plt.xticks(rotation=45)  # Rotate the platform labels for readability
plt.show()
```



2. What are the Top-selling Genres in North America, Europe, and Japan?

- Break down sales data by region to identify the most popular video game genres in North America, Europe, and Japan. Are there regional variations in gaming preferences?

In [ ]:
```
# Group the data by 'Genre' and calculate the total sales in each region
genre_sales = vg_df.groupby('Genre')[['NA_Sales', 'EU_Sales', 'JP_Sales']].sum()

# Set the genres as the index for the grouped DataFrame
genre_sales.reset_index(inplace=True)

# Create a grouped bar chart
plt.figure(figsize=(15, 6))

x = genre_sales['Genre']
x_indexes = range(len(x))
width = 0.25

plt.bar(x_indexes, genre_sales['NA_Sales'], width=width, label='North America',
plt.bar([i + width for i in x_indexes], genre_sales['EU_Sales'], width=width, la
plt.bar([i + width * 2 for i in x_indexes], genre_sales['JP_Sales'], width=width

plt.xlabel('Genre')
plt.ylabel('Sales (Millions)')
plt.title('Video Game Sales by Genre and Region')
plt.xticks([i + width for i in x_indexes], x)
plt.legend()

# Show the plot
plt.tight_layout()
plt.show()
```



3.Trends in Global Sales Over the Years?

Explore the evolution of global video game sales over time. Are there specific trends or patterns in the industry's growth or decline?

In [ ]:
```
print(vg_df['Year'].unique())
yearly_sales = vg_df.groupby('Year')['Global_Sales'].sum()
display(yearly_sales)
```

```
[2006 1985 2008 2009 1996 1989 1984 2005 1999 2007 2010 2013 2004 1990
 1988 2002 2001 2011 1998 2015 2012 2014 1992 1997 1993 1994 1982 2003
 1986 2000    0 1995 2016 1991 1981 1987 1980 1983 2020 2017]
```

```
Year
0       100.08
1980     11.38
1981     35.77
1982     28.86
1983     16.79
1984     50.36
1985     53.94
1986     37.07
1987     21.74
1988     47.22
1989     73.45
1990     49.39
1991     32.23
1992     76.16
1993     45.98
1994     79.17
1995     88.11
1996    199.15
1997    200.98
1998    256.47
1999    251.27
2000    201.56
2001    331.47
2002    395.52
2003    357.85
2004    419.31
2005    459.94
2006    521.04
2007    611.13
2008    678.90
2009    667.30
2010    600.45
2011    515.99
2012    363.54
2013    368.11
2014    337.05
2015    264.44
2016     70.93
2017      0.05
2020      0.29
Name: Global_Sales, dtype: float64
```

In [ ]:

```
# Convert 'Year' column to integers
vg_df['Year'] = vg_df['Year'].astype(int)

# Deleting rows with no intel of year
filtered_df = vg_df[vg_df['Year'] > 0]

# Group by 'Year' and calculate the sum of 'Global_Sales' for each year
yearly_sales = filtered_df.groupby('Year')['Global_Sales'].sum()

# Find the index of the maximum and minimum values
max_index = yearly_sales.idxmax()
min_index = yearly_sales.idxmin()

# Plotting the line chart
plt.figure(figsize=(18, 6))
plt.plot(yearly_sales.index, yearly_sales.values, marker='o', linestyle='-', col
plt.scatter([max_index, min_index], [yearly_sales[max_index], yearly_sales[min_i
```

```
plt.annotate(f'Max: {yearly_sales[max_index]:.2f} in {max_index}', xy=(max_index
             xytext=(max_index + 1, yearly_sales[max_index] - 10), color='red',
             arrowprops=dict(facecolor='red', arrowstyle='->'))
plt.annotate(f'Min: {yearly_sales[min_index]:.2f} in {min_index}', xy=(min_index
             xytext=(min_index - 5, yearly_sales[min_index] + 10), color='red',
             arrowprops=dict(facecolor='red', arrowstyle='->'))
plt.xlabel('Year')
plt.ylabel('Global Sales (Millions)')
plt.title('Global Game Sales Over Time')
plt.grid(True)
plt.show()
```

```
2008
```