untitled

November 28, 2023

1 Video Games Sales Analysis And Visualization

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

[2]:
```
df = pd.read_csv('vgsales.csv', encoding='latin')
df.head()
```

[2]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37

[3]:
```
df.tail(5)
```

[3]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.0
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.0
16595	16598	SCORE International Baja 1000: The Official Game	PS2						
16596	16599	Know How 2	DS						
16597	16600	Spirits & Spells	GBA						

```
16595  2008.0    Racing  Activision      0.00      0.00       0.0
16596  2010.0    Puzzle    7G//AMES      0.00      0.01       0.0
16597  2003.0  Platform     Wanadoo      0.01      0.00       0.0

       Other_Sales  Global_Sales
16593          0.0          0.01
16594          0.0          0.01
16595          0.0          0.01
16596          0.0          0.01
16597          0.0          0.01
```

[4]:
```
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
```

[5]:
```
df.describe()
```

[5]:
```
                Rank          Year      NA_Sales      EU_Sales      JP_Sales  \
count  16598.000000  16327.000000  16598.000000  16598.000000  16598.000000
mean    8300.605254   2006.406443      0.264667      0.146652      0.077782
std     4791.853933      5.828981      0.816683      0.505351      0.309291
min        1.000000   1980.000000      0.000000      0.000000      0.000000
25%     4151.250000   2003.000000      0.000000      0.000000      0.000000
50%     8300.500000   2007.000000      0.080000      0.020000      0.000000
75%    12449.750000   2010.000000      0.240000      0.110000      0.040000
max    16600.000000   2020.000000     41.490000     29.020000     10.220000

        Other_Sales  Global_Sales
count  16598.000000  16598.000000
mean       0.048063      0.537441
```

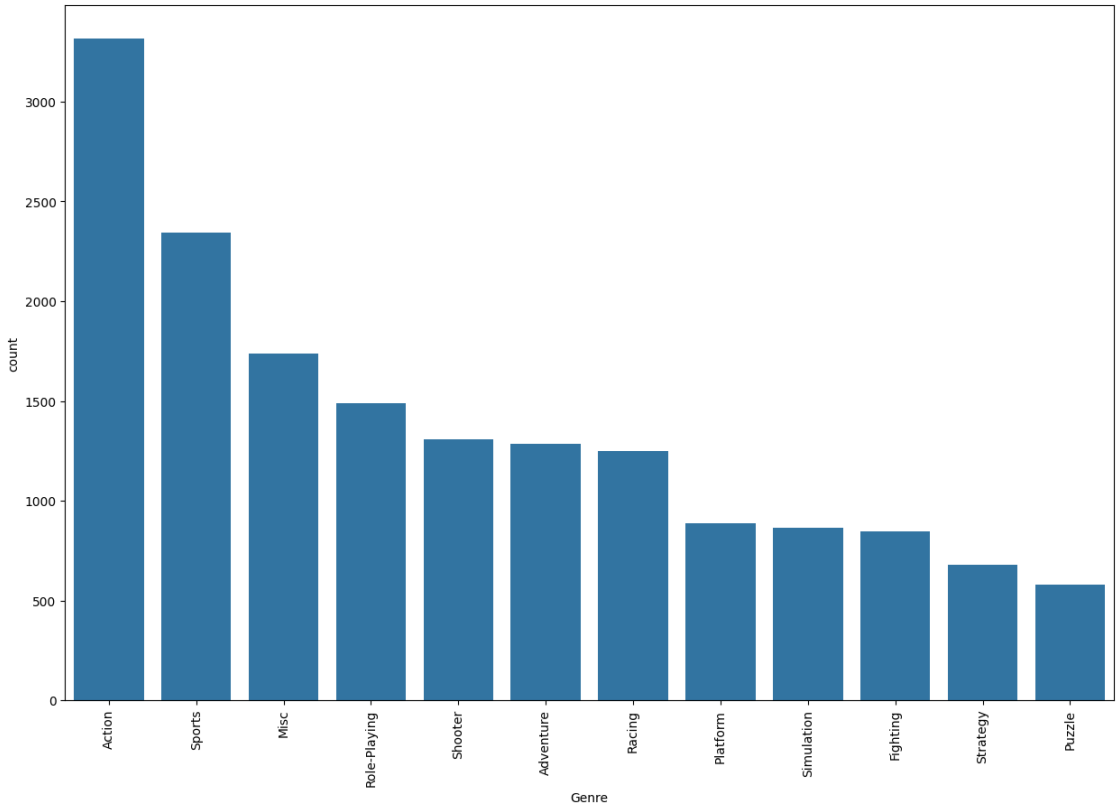
```
std        0.188588      1.555028
min        0.000000      0.010000
25%        0.000000      0.060000
50%        0.010000      0.170000
75%        0.040000      0.470000
max       10.570000     82.740000
```

[7]:
```
df.shape
```

[7]:
```
(16598, 11)
```

[14]:
```
#What genre games have been made the most?
plt.figure(figsize=(15, 10))
sns.countplot(x="Genre", data=df, order = df['Genre'].value_counts().index)
plt.xticks(rotation=90)
```

[14]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
 [Text(0, 0, 'Action'),
  Text(1, 0, 'Sports'),
  Text(2, 0, 'Misc'),
  Text(3, 0, 'Role-Playing'),
  Text(4, 0, 'Shooter'),
  Text(5, 0, 'Adventure'),
  Text(6, 0, 'Racing'),
  Text(7, 0, 'Platform'),
  Text(8, 0, 'Simulation'),
  Text(9, 0, 'Fighting'),
  Text(10, 0, 'Strategy'),
  Text(11, 0, 'Puzzle')])
```

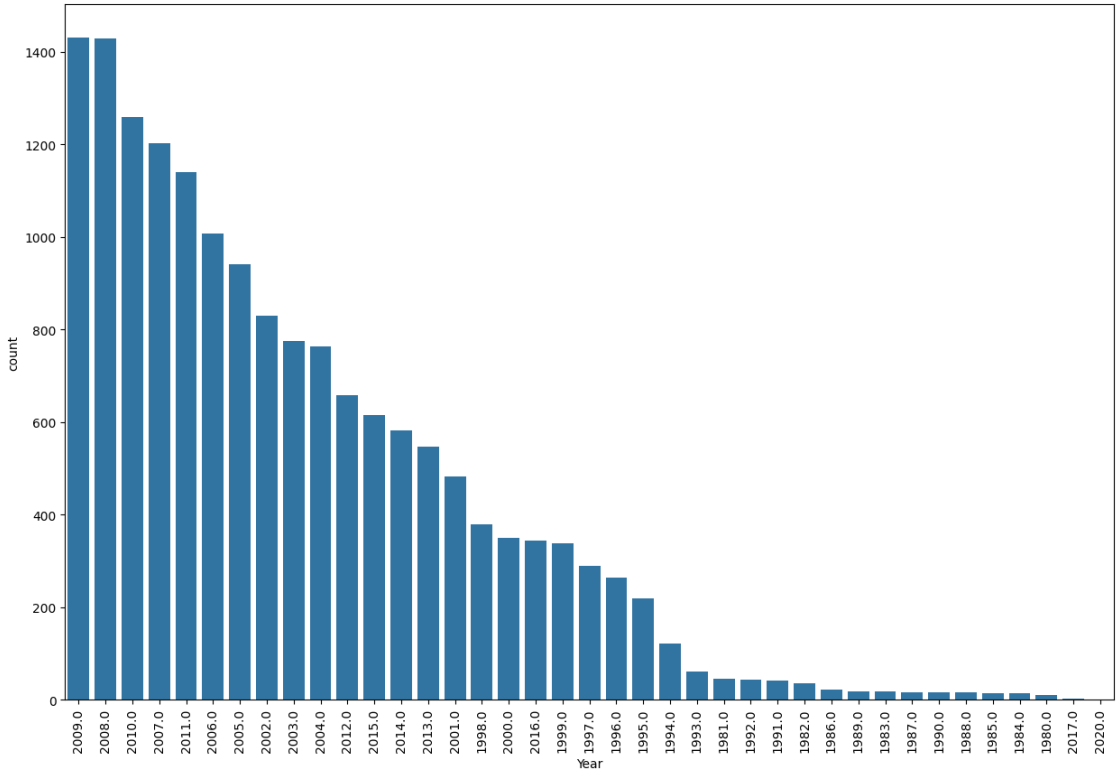


[17]:
```
# #What genre games have been made the most?
plt.figure(figsize=(15, 10))
sns.countplot(x="Year", data=df, order = df.groupby(by=['Year'])['Name'].
 ↪count().sort_values(ascending=False).index)
plt.xticks(rotation=90)
```

[17]:
```
([0,
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13,
  14,
```

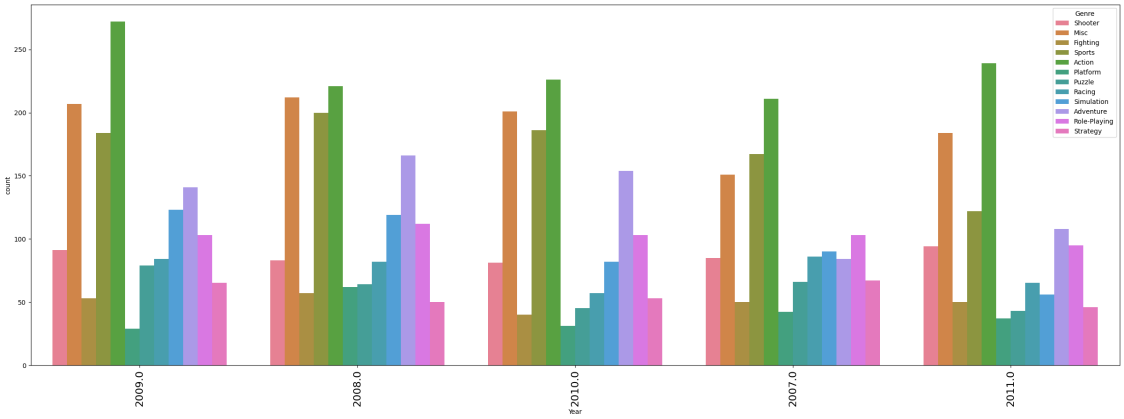
```
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30,
 31,
 32,
 33,
 34,
 35,
 36,
 37,
 38],
[Text(0, 0, '2009.0'),
 Text(1, 0, '2008.0'),
 Text(2, 0, '2010.0'),
 Text(3, 0, '2007.0'),
 Text(4, 0, '2011.0'),
 Text(5, 0, '2006.0'),
 Text(6, 0, '2005.0'),
 Text(7, 0, '2002.0'),
 Text(8, 0, '2003.0'),
 Text(9, 0, '2004.0'),
 Text(10, 0, '2012.0'),
 Text(11, 0, '2015.0'),
 Text(12, 0, '2014.0'),
 Text(13, 0, '2013.0'),
 Text(14, 0, '2001.0'),
 Text(15, 0, '1998.0'),
 Text(16, 0, '2000.0'),
 Text(17, 0, '2016.0'),
 Text(18, 0, '1999.0'),
 Text(19, 0, '1997.0'),
 Text(20, 0, '1996.0'),
 Text(21, 0, '1995.0'),
 Text(22, 0, '1994.0'),
```

```
Text(23, 0, '1993.0'),
Text(24, 0, '1981.0'),
Text(25, 0, '1992.0'),
Text(26, 0, '1991.0'),
Text(27, 0, '1982.0'),
Text(28, 0, '1986.0'),
Text(29, 0, '1989.0'),
Text(30, 0, '1983.0'),
Text(31, 0, '1987.0'),
Text(32, 0, '1990.0'),
Text(33, 0, '1988.0'),
Text(34, 0, '1985.0'),
Text(35, 0, '1984.0'),
Text(36, 0, '1980.0'),
Text(37, 0, '2017.0'),
Text(38, 0, '2020.0')])
```



[19]:
```
# Top 5 years games release by genre.
plt.figure(figsize=(30, 10))
sns.countplot(x="Year", data=df, hue='Genre', order=df.Year.value_counts().
  ↪iloc[:5].index)
plt.xticks(size=16, rotation=90)
```

[19]:
```
([0, 1, 2, 3, 4],
 [Text(0, 0, '2009.0'),
  Text(1, 0, '2008.0'),
  Text(2, 0, '2010.0'),
  Text(3, 0, '2007.0'),
  Text(4, 0, '2011.0')])
```

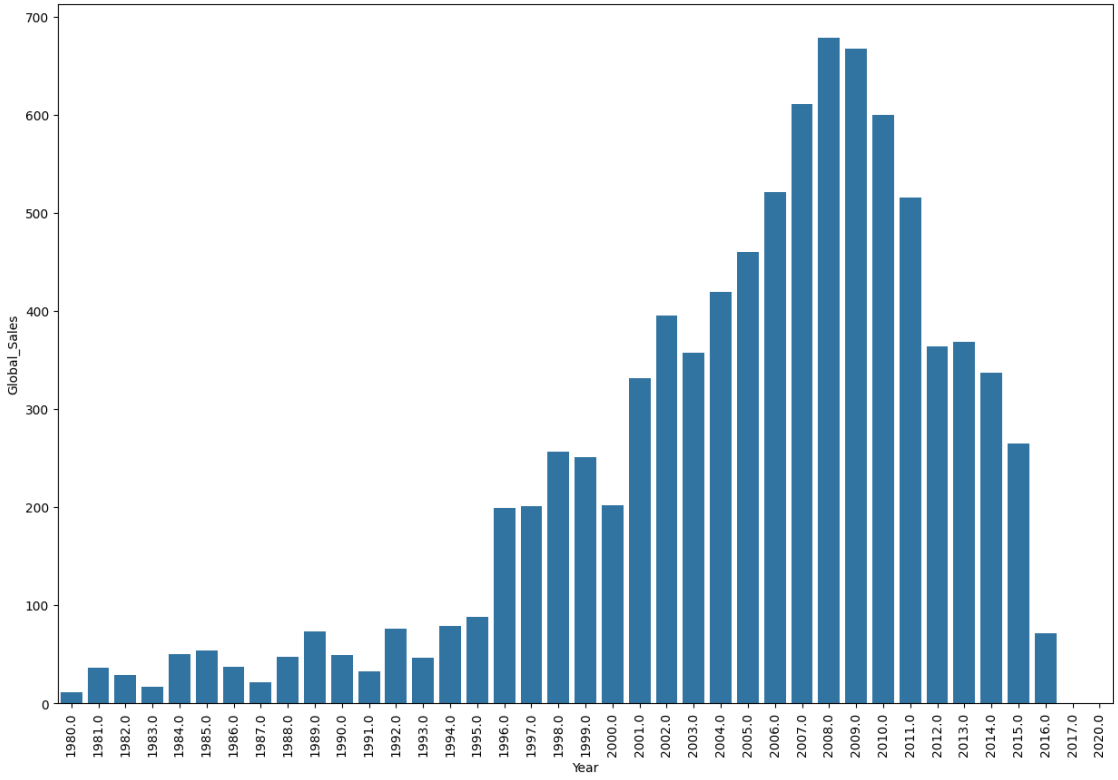


[22]:
```
# Which year had the highest sales worldwide?
plt.figure(figsize=(15, 10))
sns.barplot(x="Year", y="Global_Sales", data=data_year)
plt.xticks(rotation=90)
```

[22]:
```
([0,
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13,
  14,
  15,
  16,
  17,
  18,
  19,
```

```
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30,
 31,
 32,
 33,
 34,
 35,
 36,
 37,
 38],
[Text(0, 0, '1980.0'),
 Text(1, 0, '1981.0'),
 Text(2, 0, '1982.0'),
 Text(3, 0, '1983.0'),
 Text(4, 0, '1984.0'),
 Text(5, 0, '1985.0'),
 Text(6, 0, '1986.0'),
 Text(7, 0, '1987.0'),
 Text(8, 0, '1988.0'),
 Text(9, 0, '1989.0'),
 Text(10, 0, '1990.0'),
 Text(11, 0, '1991.0'),
 Text(12, 0, '1992.0'),
 Text(13, 0, '1993.0'),
 Text(14, 0, '1994.0'),
 Text(15, 0, '1995.0'),
 Text(16, 0, '1996.0'),
 Text(17, 0, '1997.0'),
 Text(18, 0, '1998.0'),
 Text(19, 0, '1999.0'),
 Text(20, 0, '2000.0'),
 Text(21, 0, '2001.0'),
 Text(22, 0, '2002.0'),
 Text(23, 0, '2003.0'),
 Text(24, 0, '2004.0'),
 Text(25, 0, '2005.0'),
 Text(26, 0, '2006.0'),
 Text(27, 0, '2007.0'),
```

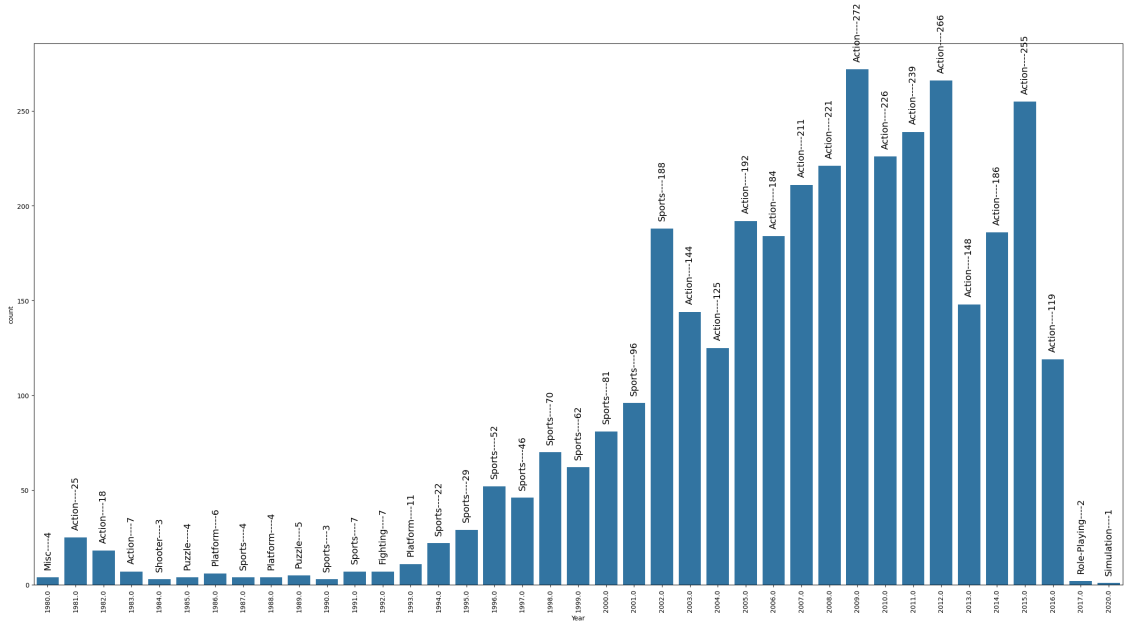
```
Text(28, 0, '2008.0'),
Text(29, 0, '2009.0'),
Text(30, 0, '2010.0'),
Text(31, 0, '2011.0'),
Text(32, 0, '2012.0'),
Text(33, 0, '2013.0'),
Text(34, 0, '2014.0'),
Text(35, 0, '2015.0'),
Text(36, 0, '2016.0'),
Text(37, 0, '2017.0'),
Text(38, 0, '2020.0')])
```



[29]:
```
#Which genre game has been released the most in a single year?

year_max_df = df.groupby(['Year', 'Genre']).size().reset_index(name='count')
year_max_idx = year_max_df.groupby(['Year'])['count'].transform(max) ==␣
 ↪year_max_df['count']
year_max_genre = year_max_df[year_max_idx].reset_index(drop=True)
year_max_genre = year_max_genre.drop_duplicates(subset=["Year", "count"],␣
 ↪keep='last').reset_index(drop=True)

genre = year_max_genre['Genre'].values
```

```
plt.figure(figsize=(30, 15))
g = sns.barplot(x='Year', y='count', data=year_max_genre)
index = 0
for value in year_max_genre['count'].values:
#     print(asd)
    g.text(index, value + 5, str(genre[index] + '----' +str(value)),␣
 ↪color='#000', size=14, rotation= 90, ha="center")
    index += 1

plt.xticks(rotation=90)
plt.show()
```

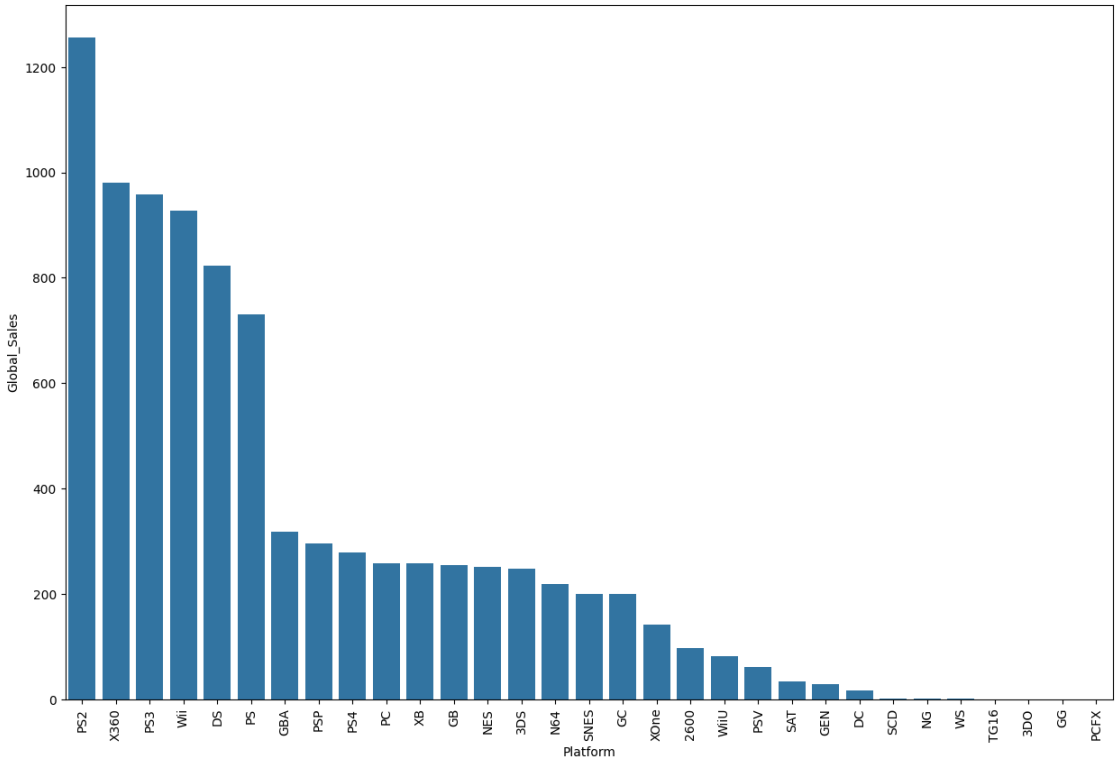
```
C:\Users\yakob\AppData\Local\Temp\ipykernel_31500\502012060.py:4: FutureWarning:
The provided callable <built-in function max> is currently using
SeriesGroupBy.max. In a future version of pandas, the provided callable will be
used directly. To keep current behavior pass the string "max" instead.
  year_max_idx = year_max_df.groupby(['Year'])['count'].transform(max) ==
year_max_df['count']
```



[31]:
```
# Which platfrom have the highest sale price globally
data_platform = df.groupby(by=['Platform'])['Global_Sales'].sum()
data_platform = data_platform.reset_index()
data_platform = data_platform.sort_values(by=['Global_Sales'], ascending=False)
plt.figure(figsize=(15, 10))
sns.barplot(x="Platform", y="Global_Sales", data=data_platform)
plt.xticks(rotation=90)
```

```
[31]: ([0,
        1,
        2,
        3,
        4,
        5,
        6,
        7,
        8,
        9,
        10,
        11,
        12,
        13,
        14,
        15,
        16,
        17,
        18,
        19,
        20,
        21,
        22,
        23,
        24,
        25,
        26,
        27,
        28,
        29,
        30],
       [Text(0, 0, 'PS2'),
        Text(1, 0, 'X360'),
        Text(2, 0, 'PS3'),
        Text(3, 0, 'Wii'),
        Text(4, 0, 'DS'),
        Text(5, 0, 'PS'),
        Text(6, 0, 'GBA'),
        Text(7, 0, 'PSP'),
        Text(8, 0, 'PS4'),
        Text(9, 0, 'PC'),
        Text(10, 0, 'XB'),
        Text(11, 0, 'GB'),
        Text(12, 0, 'NES'),
        Text(13, 0, '3DS'),
        Text(14, 0, 'N64'),
        Text(15, 0, 'SNES'),
```

```
Text(16, 0, 'GC'),
Text(17, 0, 'XOne'),
Text(18, 0, '2600'),
Text(19, 0, 'WiiU'),
Text(20, 0, 'PSV'),
Text(21, 0, 'SAT'),
Text(22, 0, 'GEN'),
Text(23, 0, 'DC'),
Text(24, 0, 'SCD'),
Text(25, 0, 'NG'),
Text(26, 0, 'WS'),
Text(27, 0, 'TG16'),
Text(28, 0, '3DO'),
Text(29, 0, 'GG'),
Text(30, 0, 'PCFX')])
```



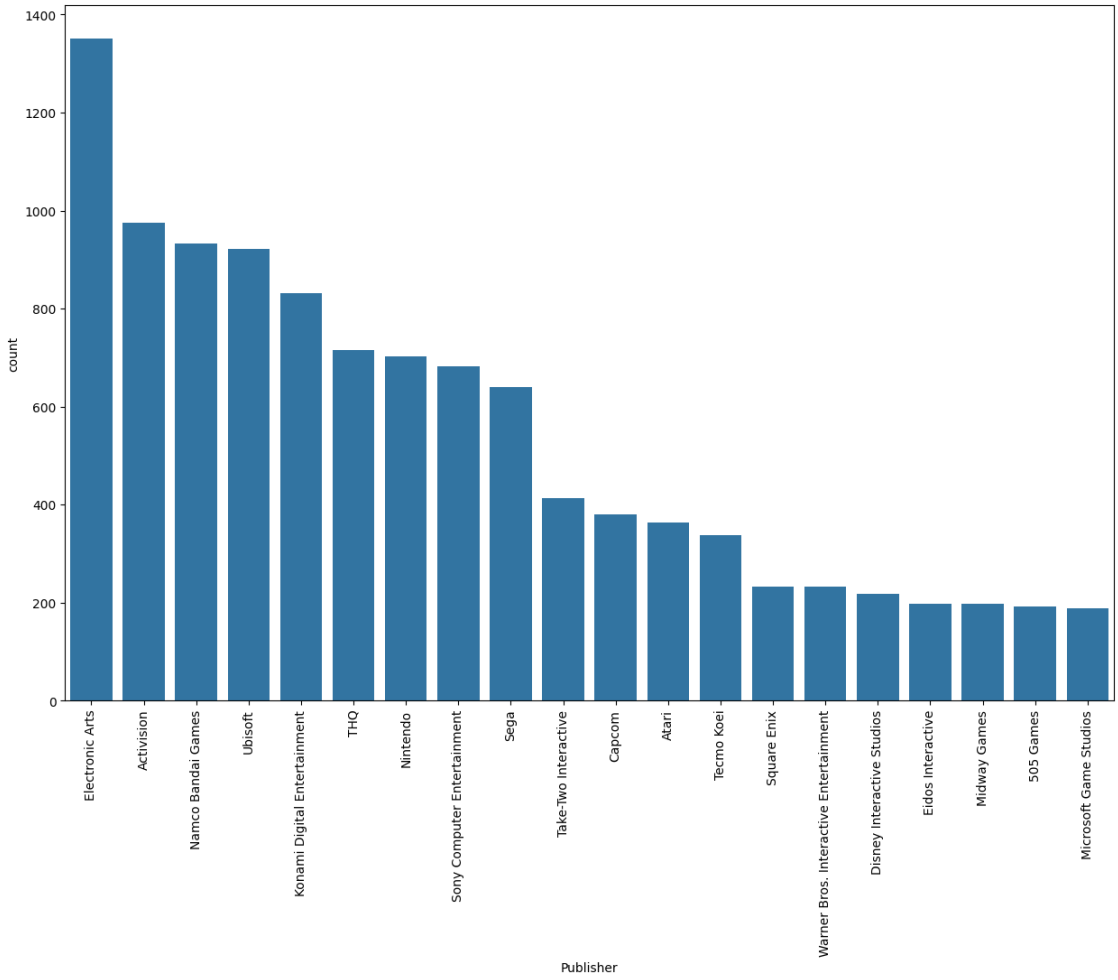
[34]:
```
#Sales compearison by platform
comp_platform = df[['Platform', 'NA_Sales', 'EU_Sales', 'JP_Sales',␣
 ↪'Other_Sales']]
comp_platform.head()
```

[34]:

	Platform	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	Wii	41.49	29.02	3.77	8.46
1	NES	29.08	3.58	6.81	0.77
2	Wii	15.85	12.88	3.79	3.31
3	Wii	15.75	11.01	3.28	2.96
4	GB	11.27	8.89	10.22	1.00

[37]:
```
# Top 20 Publisher
top_publisher = df.groupby(by=['Publisher'])['Year'].count().
 ↪sort_values(ascending=False).head(20)
top_publisher = pd.DataFrame(top_publisher).reset_index()
plt.figure(figsize=(15, 10))
sns.countplot(x="Publisher", data=df, order = df.
 ↪groupby(by=['Publisher'])['Year'].count().sort_values(ascending=False).iloc[:
 ↪20].index)
plt.xticks(rotation=90)
```

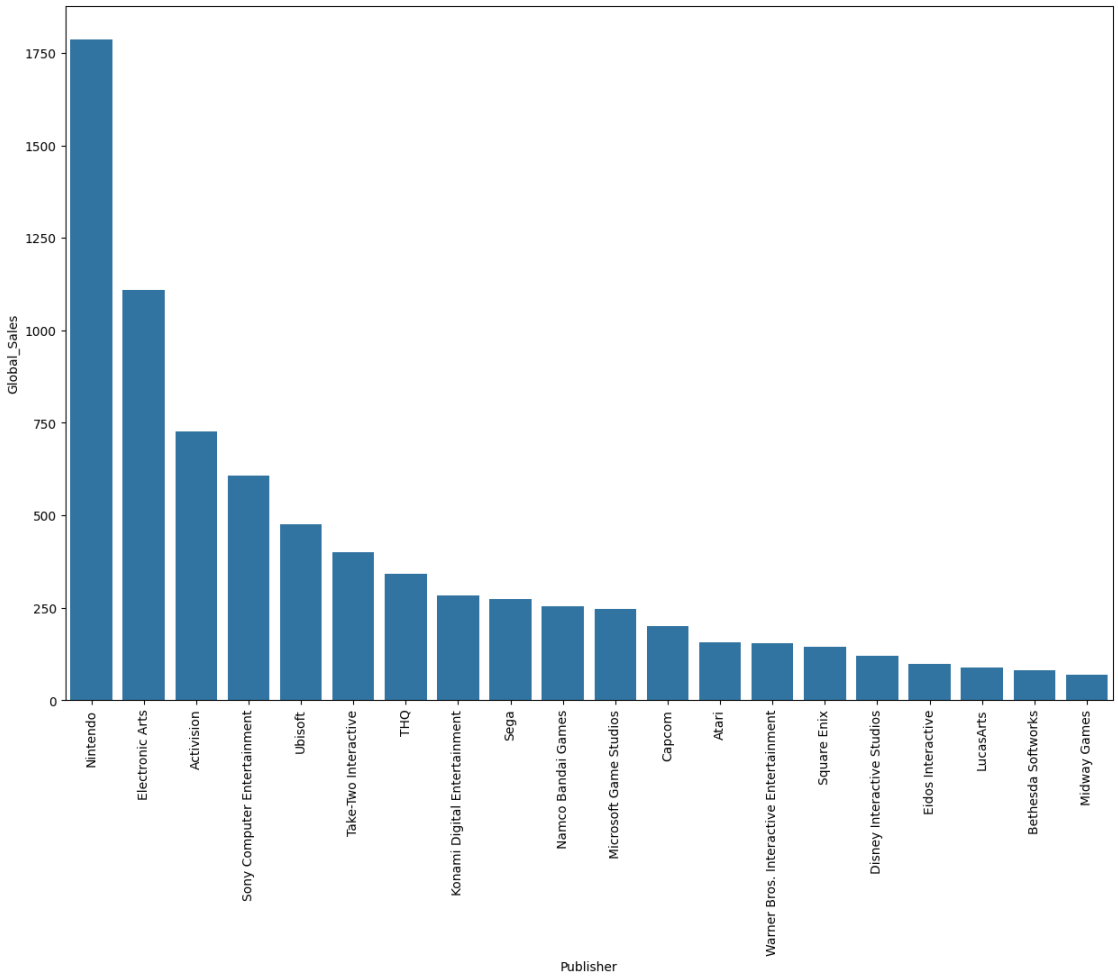
[37]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
 [Text(0, 0, 'Electronic Arts'),
  Text(1, 0, 'Activision'),
  Text(2, 0, 'Namco Bandai Games'),
  Text(3, 0, 'Ubisoft'),
  Text(4, 0, 'Konami Digital Entertainment'),
  Text(5, 0, 'THQ'),
  Text(6, 0, 'Nintendo'),
  Text(7, 0, 'Sony Computer Entertainment'),
  Text(8, 0, 'Sega'),
  Text(9, 0, 'Take-Two Interactive'),
  Text(10, 0, 'Capcom'),
  Text(11, 0, 'Atari'),
  Text(12, 0, 'Tecmo Koei'),
  Text(13, 0, 'Square Enix'),
  Text(14, 0, 'Warner Bros. Interactive Entertainment'),
  Text(15, 0, 'Disney Interactive Studios'),
  Text(16, 0, 'Eidos Interactive'),
  Text(17, 0, 'Midway Games'),
  Text(18, 0, '505 Games'),
  Text(19, 0, 'Microsoft Game Studios')])
```



[39]:
```
#Top global sales by publisher
sale_pbl = df[['Publisher', 'Global_Sales']]
sale_pbl = sale_pbl.groupby('Publisher')['Global_Sales'].sum().
 ↪sort_values(ascending=False).head(20)
sale_pbl = pd.DataFrame(sale_pbl).reset_index()
plt.figure(figsize=(15, 10))
sns.barplot(x='Publisher', y='Global_Sales', data=sale_pbl)
plt.xticks(rotation=90)
```

[39]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
 [Text(0, 0, 'Nintendo'),
  Text(1, 0, 'Electronic Arts'),
  Text(2, 0, 'Activision'),
  Text(3, 0, 'Sony Computer Entertainment'),
  Text(4, 0, 'Ubisoft'),
  Text(5, 0, 'Take-Two Interactive'),
  Text(6, 0, 'THQ'),
```

```
Text(7, 0, 'Konami Digital Entertainment'),
Text(8, 0, 'Sega'),
Text(9, 0, 'Namco Bandai Games'),
Text(10, 0, 'Microsoft Game Studios'),
Text(11, 0, 'Capcom'),
Text(12, 0, 'Atari'),
Text(13, 0, 'Warner Bros. Interactive Entertainment'),
Text(14, 0, 'Square Enix'),
Text(15, 0, 'Disney Interactive Studios'),
Text(16, 0, 'Eidos Interactive'),
Text(17, 0, 'LucasArts'),
Text(18, 0, 'Bethesda Softworks'),
Text(19, 0, 'Midway Games')])
```



[41]:
```
comp_publisher = df[['Publisher', 'NA_Sales', 'EU_Sales', 'JP_Sales',␣
↪'Other_Sales', 'Global_Sales']]
comp_publisher.head()
```

[41]:

	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	Nintendo	41.49	29.02	3.77	8.46	82.74
1	Nintendo	29.08	3.58	6.81	0.77	40.24
2	Nintendo	15.85	12.88	3.79	3.31	35.82
3	Nintendo	15.75	11.01	3.28	2.96	33.00
4	Nintendo	11.27	8.89	10.22	1.00	31.37

[43]:
```
# Total revenue by region
top_sale_reg = df[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
top_sale_reg = top_sale_reg.sum().reset_index()
top_sale_reg = top_sale_reg.rename(columns={"index": "region", 0: "sale"})
top_sale_reg
```

[43]:

	region	sale
0	NA_Sales	4392.95
1	EU_Sales	2434.13
2	JP_Sales	1291.02
3	Other_Sales	797.75

[ ]: