In [17]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import style
import seaborn as sns
```

In [18]:
```
#read the csv file
df = pd.read_csv(r"C:\\Users\HP\Downloads\vgsales.csv")
df
```

Out[18]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sale
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.7
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.8
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.7
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.2
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.2
...	...	...	...	...	...	...	...	...	
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	2008.0	Racing	Activision	0.00	0.00	0.0
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.0
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.0

16598 rows × 11 columns

In [19]:
```
#1 it shows the data type with count the number of rows
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 [20]:
```
#2 describe the columns which have numerical value.
df.describe()
```

Out[20]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Gl
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	165
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	

In [21]:
```
#3 it shows the null values in the data
df[df.isnull().any(axis=1)]
```

Out[21]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sal
179	180	Madden NFL 2004	PS2	NaN	Sports	Electronic Arts	4.26	0.26	0.(
377	378	FIFA Soccer 2004	PS2	NaN	Sports	Electronic Arts	0.59	2.36	0.(
431	432	LEGO Batman: The Videogame	Wii	NaN	Action	Warner Bros. Interactive Entertainment	1.86	1.02	0.(
470	471	wwe Smackdown vs. Raw 2006	PS2	NaN	Fighting	NaN	1.57	1.02	0.(
607	608	Space Invaders	2600	NaN	Shooter	Atari	2.36	0.14	0.(
...	...	...	...	...	...	...	...	...	
16427	16430	Virtua Quest	GC	NaN	Role-Playing	Unknown	0.01	0.00	0.(
16493	16496	The Smurfs	3DS	NaN	Action	Unknown	0.00	0.01	0.(
16494	16497	Legends of Oz: Dorothy's Return	3DS	2014.0	Puzzle	NaN	0.00	0.01	0.(
16543	16546	Driving Simulator 2011	PC	2011.0	Racing	NaN	0.00	0.01	0.(
16553	16556	Bound By Flame	X360	2014.0	Role-Playing	NaN	0.00	0.01	0.(

307 rows × 11 columns

In [22]:
```
#4 in this we find the genre wise sales
genre_wise_sales = df.groupby('Genre')['Global_Sales'].sum().sort_values(ascen
print('    Genre_Wise_Sales',)
print(genre_wise_sales)
```

```
    Genre_Wise_Sales
Genre
Action          1751.18
Sports          1330.93
Shooter         1037.37
Role-Playing     927.37
Platform         831.37
Misc             809.96
Racing           732.04
Fighting         448.91
Simulation       392.20
Puzzle           244.95
Adventure        239.04
Strategy         175.12
Name: Global_Sales, dtype: float64
```

In [23]:
```
#5 Number of Games per Platform
platform_counts = df['Platform'].value_counts()
print(platform_counts)
```

```
DS      2163
PS2     2161
PS3     1329
Wii     1325
X360    1265
PSP     1213
PS      1196
PC       960
XB       824
GBA      822
GC       556
3DS      509
PSV      413
PS4      336
N64      319
SNES     239
XOne     213
SAT      173
WiiU     143
2600     133
GB        98
NES       98
DC        52
GEN       27
NG        12
SCD        6
WS         6
3DO        3
TG16       2
PCFX       1
GG         1
Name: Platform, dtype: int64
```

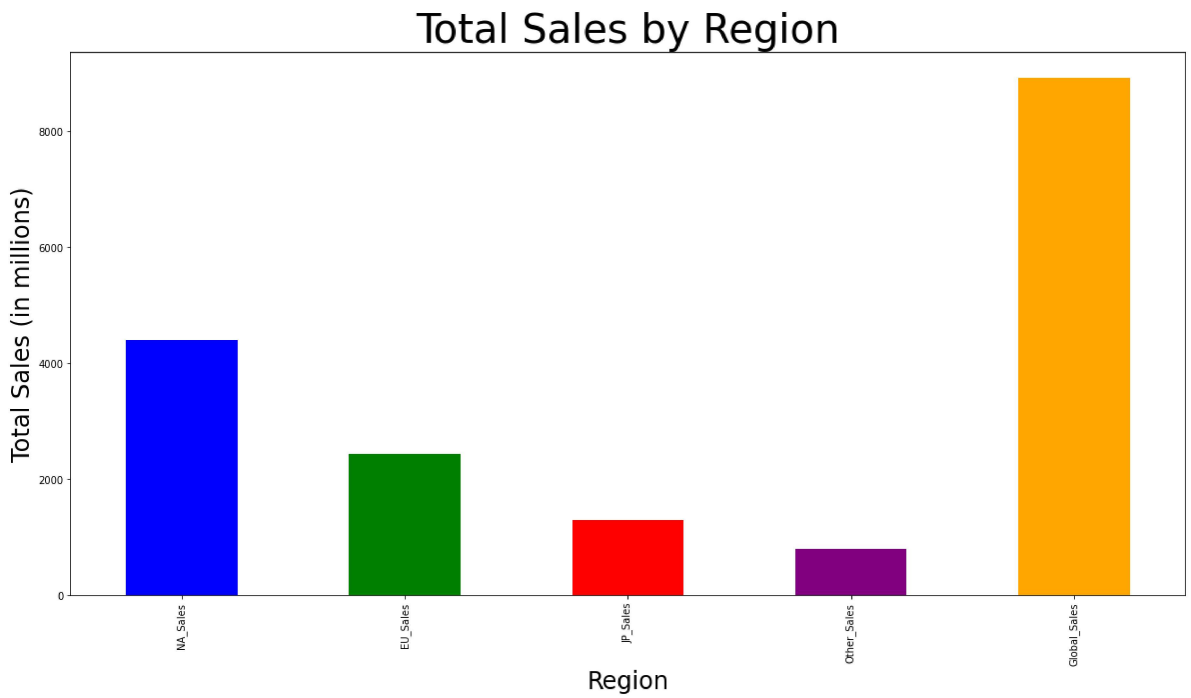
In [24]:
```
#8 Number of Unique Values in Each Column
df.nunique()
```

Out[24]:
```
Rank            16598
Name            11493
Platform           31
Year               39
Genre              12
Publisher         578
NA_Sales          409
EU_Sales          305
JP_Sales          244
Other_Sales       157
Global_Sales      623
dtype: int64
```

In [25]:
```
#9 Total Sales by Region
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
sales_columns = ['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sa
total_sales = df[sales_columns].sum()
print(total_sales)
plt.figure(figsize=(20, 10))
total_sales.plot(kind='bar', color=['blue', 'green', 'red', 'purple', 'orange'
plt.title('Total Sales by Region',fontsize=40)
plt.xlabel('Region',fontsize=24)
plt.ylabel('Total Sales (in millions)',fontsize=24)
```

```
NA_Sales        4392.95
EU_Sales        2434.13
JP_Sales        1291.02
Other_Sales      797.75
Global_Sales    8920.44
dtype: float64
```

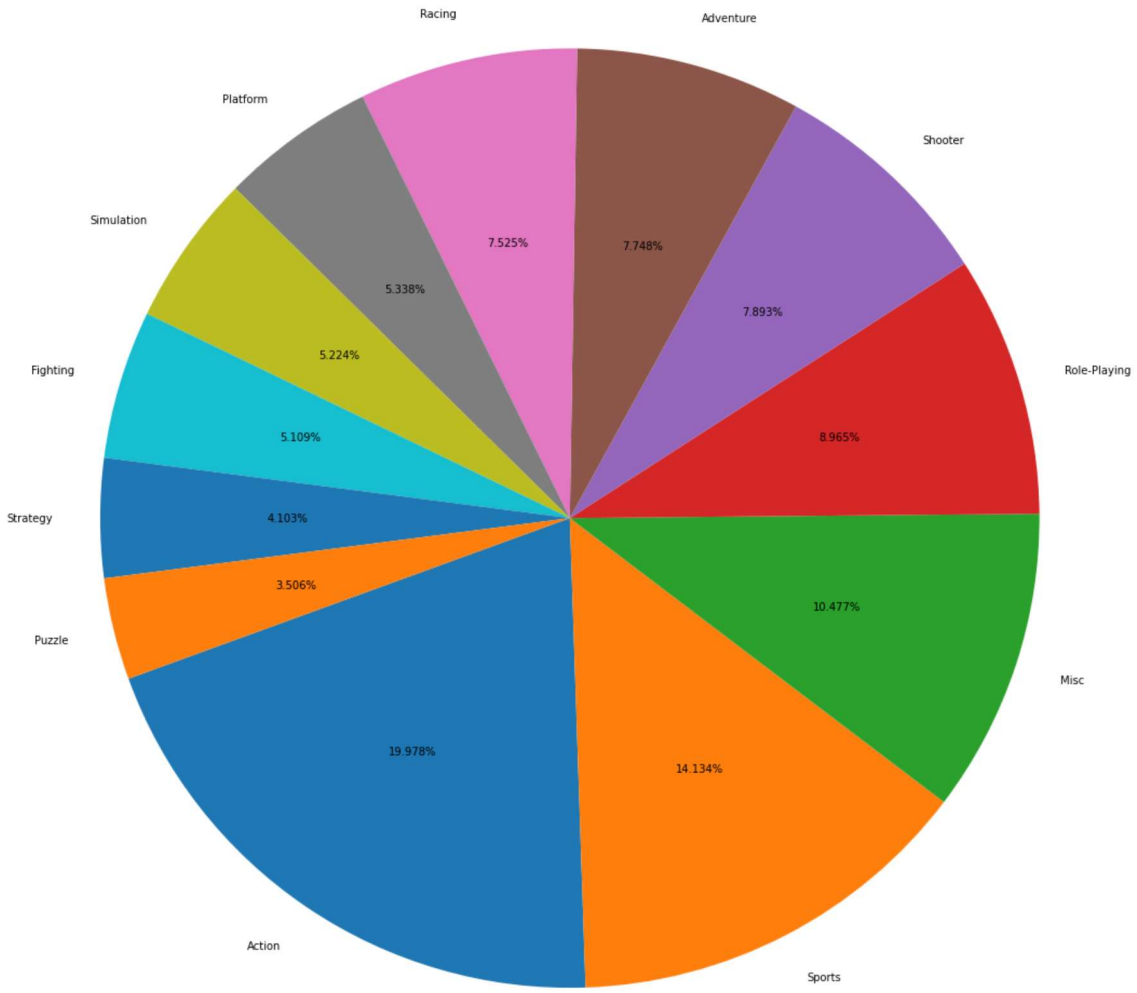
Out[25]: Text(0, 0.5, 'Total Sales (in millions)')



In [26]:
```
#10 No of games in each
genre_distribution = df['Genre'].value_counts()
print(genre_distribution)
plt.figure(figsize=(30, 20))
plt.pie(genre_distribution, labels=genre_distribution.index, autopct='%1.3f%%'
plt.title('Genre Distribution', fontsize=50)
plt.show()
```

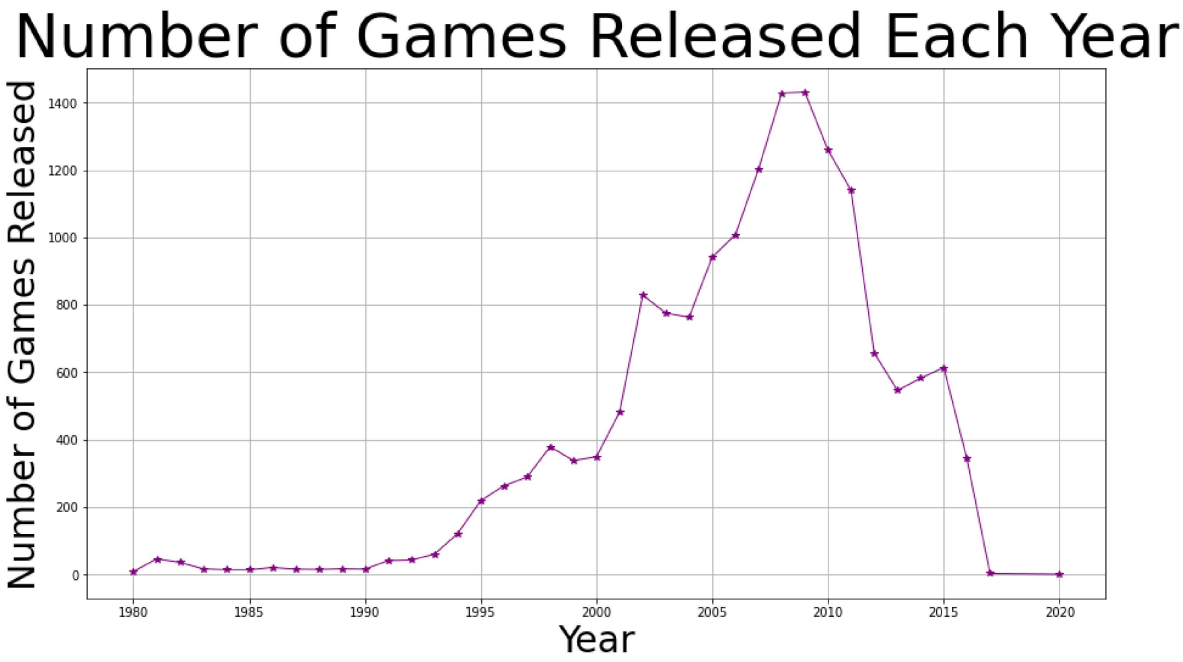
```
Action          3316
Sports          2346
Misc            1739
Role-Playing    1488
Shooter         1310
Adventure       1286
Racing          1249
Platform         886
Simulation       867
Fighting         848
Strategy         681
Puzzle           582
Name: Genre, dtype: int64
```

Genre Distribution



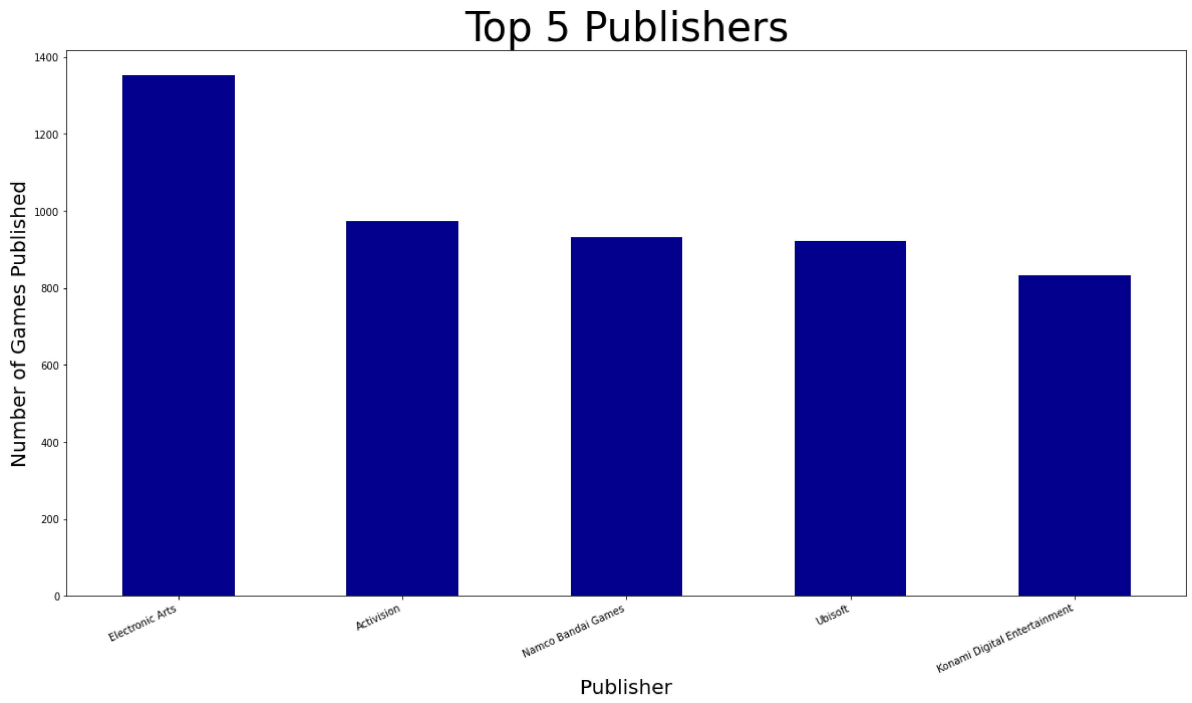
In [27]:
```
#11 Games Released Each Year
games_per_year = df.groupby('Year')['Name'].count()
print(games_per_year)
plt.figure(figsize=(15, 8))
games_per_year.plot(kind='line', marker='*', color='purple', linestyle='-', li
plt.title('Number of Games Released Each Year',fontsize=50)
plt.xlabel('Year',fontsize=30)
plt.ylabel('Number of Games Released',fontsize=30)
plt.grid(True)
plt.show()
```

```
Year
1980.0       9
1981.0      46
1982.0      36
1983.0      17
1984.0      14
1985.0      14
1986.0      21
1987.0      16
1988.0      15
1989.0      17
1990.0      16
1991.0      41
1992.0      43
1993.0      60
1994.0     121
1995.0     219
1996.0     263
1997.0     289
1998.0     379
1999.0     338
2000.0     349
2001.0     482
2002.0     829
2003.0     775
2004.0     763
2005.0     941
2006.0    1008
2007.0    1202
2008.0    1428
2009.0    1431
2010.0    1259
2011.0    1139
2012.0     657
2013.0     546
2014.0     582
2015.0     614
2016.0     344
2017.0       3
2020.0       1
Name: Name, dtype: int64
```


Number of Games Released Each Year
Number of Games Released
0
200
400
600
800
1000
1200
1400
1980
1985
1990
1995
2000
2005
2010
2015
2020
Year

In [28]:
```
#12 Top 5 Publishers
top_publishers = df['Publisher'].value_counts().head(5)
print(top_publishers)
plt.figure(figsize=(20, 10))
top_publishers.plot(kind='bar', color='darkblue')
plt.title('Top 5 Publishers',fontsize=40)
plt.xlabel('Publisher',fontsize=20)
plt.ylabel('Number of Games Published',fontsize=20)
plt.xticks(rotation=25, ha='right')
plt.show()
```

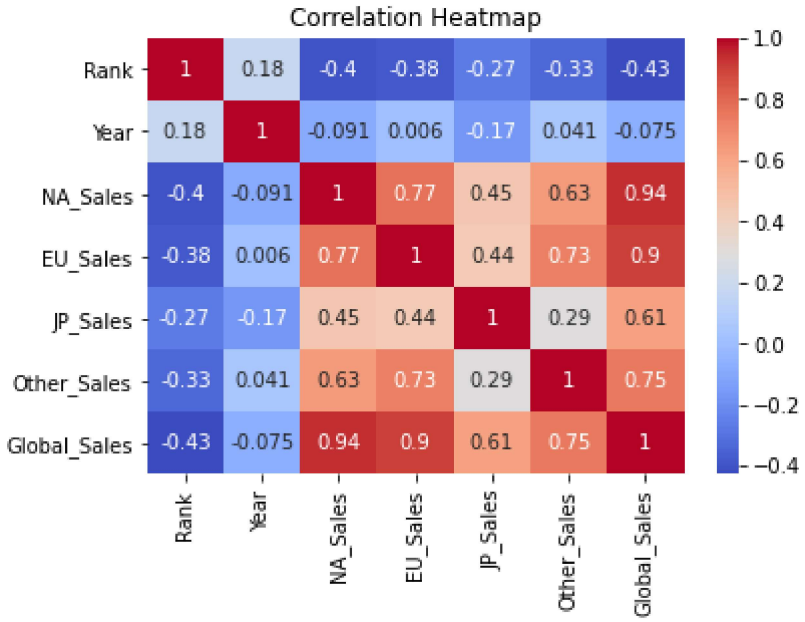
```
Electronic Arts                 1351
Activision                       975
Namco Bandai Games               932
Ubisoft                          921
Konami Digital Entertainment     832
Name: Publisher, dtype: int64
```



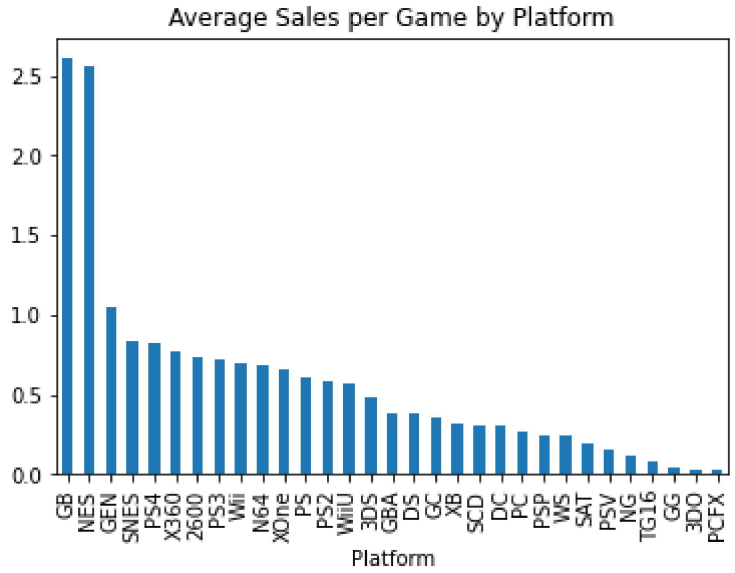
In [29]:
```
#13 Yearly Sales Trend
yearly_sales = df.groupby('Year')['Global_Sales'].sum()
print(yearly_sales)
plt.figure(figsize=(15, 8))
yearly_sales.plot(kind='line', marker='o', color='green', linestyle='-', linew
plt.title('Yearly Global Sales',fontsize=40,color='red')
plt.xlabel('Year',fontsize=20)
plt.ylabel('Global Sales (in millions)',fontsize=20)
plt.show()
```

```
Year
1980.0     11.38
1981.0     35.77
1982.0     28.86
1983.0     16.79
1984.0     50.36
1985.0     53.94
1986.0     37.07
1987.0     21.74
1988.0     47.22
1989.0     73.45
1990.0     49.39
1991.0     32.23
1992.0     76.16
1993.0     45.98
1994.0     79.17
1995.0     88.11
1996.0    199.15
1997.0    200.98
1998.0    256.47
1999.0    251.27
2000.0    201.56
2001.0    331.47
2002.0    395.52
2003.0    357.85
2004.0    419.31
2005.0    459.94
2006.0    521.04
2007.0    611.13
2008.0    678.90
2009.0    667.30
2010.0    600.45
2011.0    515.99
2012.0    363.54
2013.0    368.11
2014.0    337.05
2015.0    264.44
2016.0     70.93
2017.0      0.05
2020.0      0.29
Name: Global_Sales, dtype: float64
```

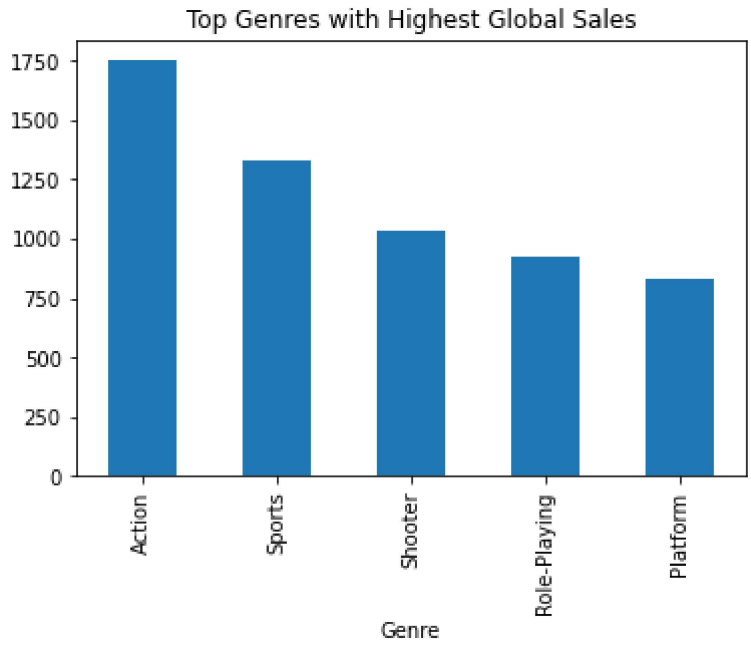
In [30]:
```
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



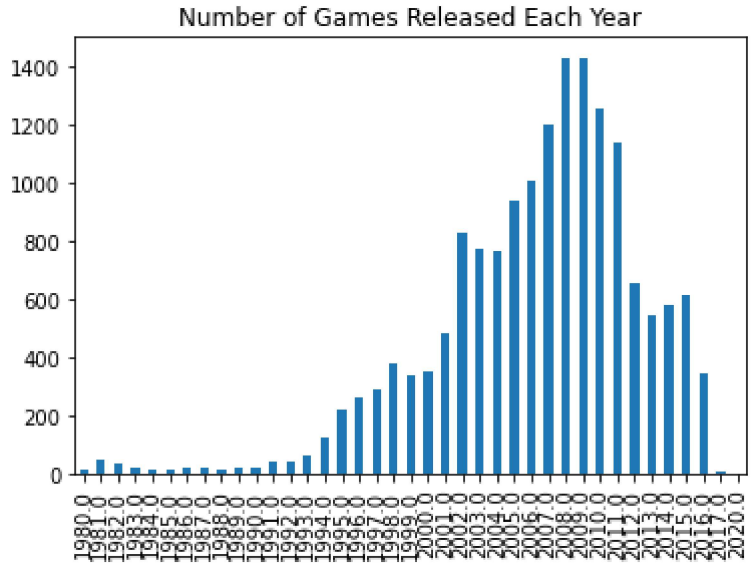
In [31]:
```
avg_sales_per_platform = df.groupby('Platform')['Global_Sales'].mean().sort_va
avg_sales_per_platform.plot(kind='bar', title='Average Sales per Game by Platf
plt.show()
```



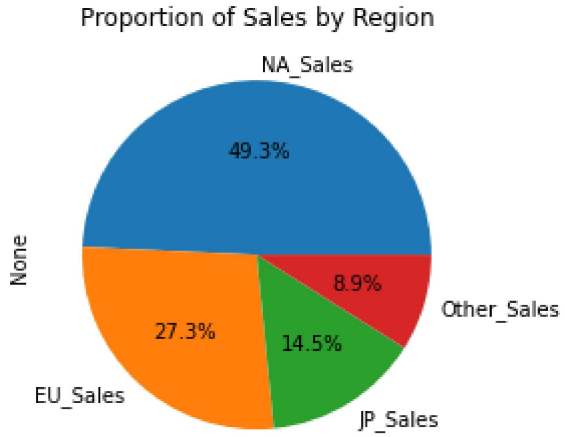
In [32]:
```
top_genres = df.groupby('Genre')['Global_Sales'].sum().sort_values(ascending=F
top_genres.plot(kind='bar', title='Top Genres with Highest Global Sales')
plt.show()
```



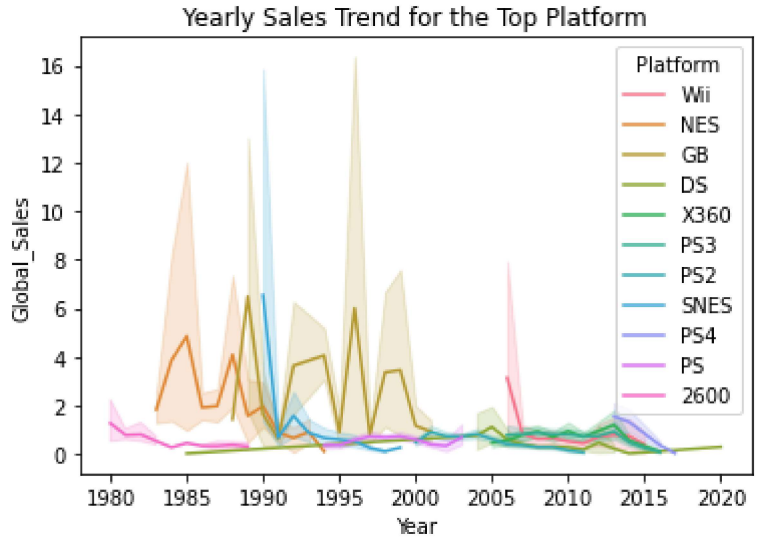
In [34]:
```
games_per_year = df['Year'].value_counts().sort_index()
games_per_year.plot(kind='bar', title='Number of Games Released Each Year')
plt.show()
```



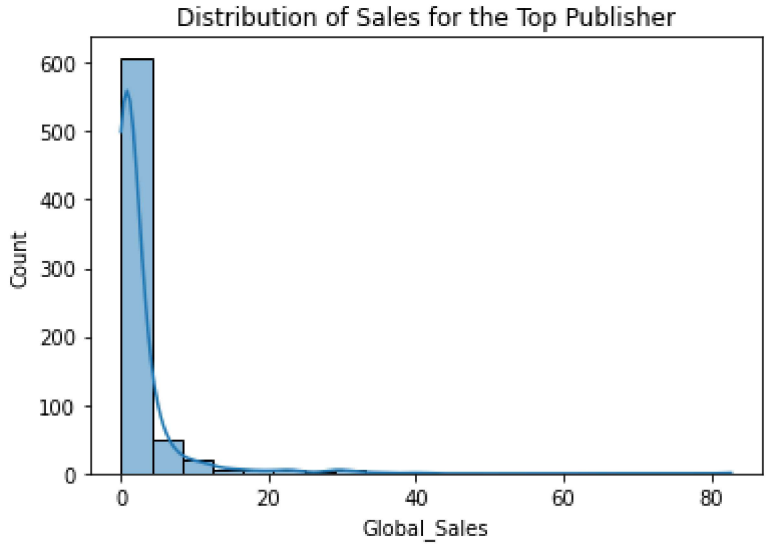
In [35]:
```
region_sales_sum = df[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']].sum
region_sales_sum.plot(kind='pie', autopct='%1.1f%%', title='Proportion of Sale
plt.show()
```



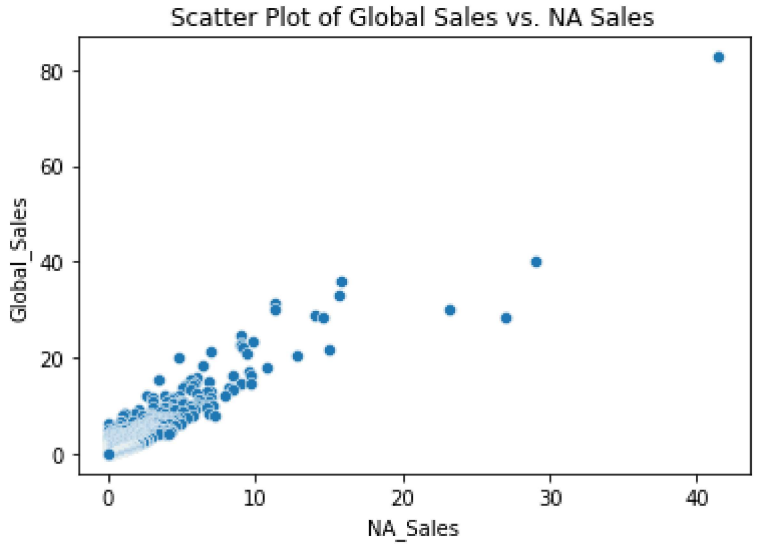
In [36]:
```
top_platform = df.groupby(['Year', 'Platform'])['Global_Sales'].sum().unstack(
top_platform_sales = df[df['Platform'].isin(top_platform.unique())]
sns.lineplot(x='Year', y='Global_Sales', hue='Platform', data=top_platform_sal
plt.title('Yearly Sales Trend for the Top Platform')
plt.show()
```



In [37]:
```
# 14. Distribution of sales for the top publisher
top_publisher = df.groupby(['Publisher'])['Global_Sales'].sum().idxmax()
top_publisher_sales = df[df['Publisher'] == top_publisher]
sns.histplot(x='Global_Sales', data=top_publisher_sales, bins=20, kde=True)
plt.title('Distribution of Sales for the Top Publisher')
plt.show()
```



In [38]:
```
# 15. Scatter plot of Global Sales vs. NA Sales
sns.scatterplot(x='NA_Sales', y='Global_Sales', data=df)
plt.title('Scatter Plot of Global Sales vs. NA Sales')
plt.show()
```



In [ ]: