

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from plotly import graph_objs as go
```

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

```
In [3]: df
```

```
Out[3]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22
...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00

16598 rows × 11 columns



```
In [4]: df.columns
```

```
Out[4]: Index(['Rank', 'Name', 'Platform', 'Year', 'Genre', 'Publisher', 'NA_Sales',
              'EU_Sales', 'JP_Sales', 'Other_Sales', 'Global_Sales'],
              dtype='object')
```

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

Out[5]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Globa
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82

In [6]: `df.sort_values(by=['Year'])['Year'].unique()`

Out[6]: array([1980., 1981., 1982., 1983., 1984., 1985., 1986., 1987., 1988.,
1989., 1990., 1991., 1992., 1993., 1994., 1995., 1996., 1997.,
1998., 1999., 2000., 2001., 2002., 2003., 2004., 2005., 2006.,
2007., 2008., 2009., 2010., 2011., 2012., 2013., 2014., 2015.,
2016., 2017., 2020., nan])

In [7]: `df[df.Year.isna()]`

Out[7]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
179	180	Madden NFL 2004	PS2	NaN	Sports	Electronic Arts	4.26	0.26	0.01
377	378	FIFA Soccer 2004	PS2	NaN	Sports	Electronic Arts	0.59	2.36	0.04
431	432	LEGO Batman: The Videogame	Wii	NaN	Action	Warner Bros. Interactive Entertainment	1.86	1.02	0.00
470	471	wwe Smackdown vs. Raw 2006	PS2	NaN	Fighting	NaN	1.57	1.02	0.00
607	608	Space Invaders	2600	NaN	Shooter	Atari	2.36	0.14	0.00
...
16307	16310	Freaky Flyers	GC	NaN	Racing	Unknown	0.01	0.00	0.00
16327	16330	Inversion	PC	NaN	Shooter	Namco Bandai Games	0.01	0.00	0.00
16366	16369	Hakuouki: Shinsengumi Kitan	PS3	NaN	Adventure	Unknown	0.01	0.00	0.00
16427	16430	Virtua Quest	GC	NaN	Role-Playing	Unknown	0.01	0.00	0.00
16493	16496	The Smurfs	3DS	NaN	Action	Unknown	0.00	0.01	0.00

271 rows × 11 columns



In [10]: df

Out[10]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22
...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00

16598 rows × 11 columns



In [11]: df.select_dtypes(include='object').columns.tolist()

Out[11]: ['Name', 'Platform', 'Genre', 'Publisher']

```

In [12]: # changing Categoricals to number
def catToNumeric(dfA):
    for x in dfA.select_dtypes(include='object').columns.tolist():
        ncol = 'id_'+x
        dfA[x] = pd.Categorical(dfA[x])
        dfA[ncol] = dfA[x].cat.codes
    return dfA

```

```
In [14]: dfNew = catToNumeric(df)
```

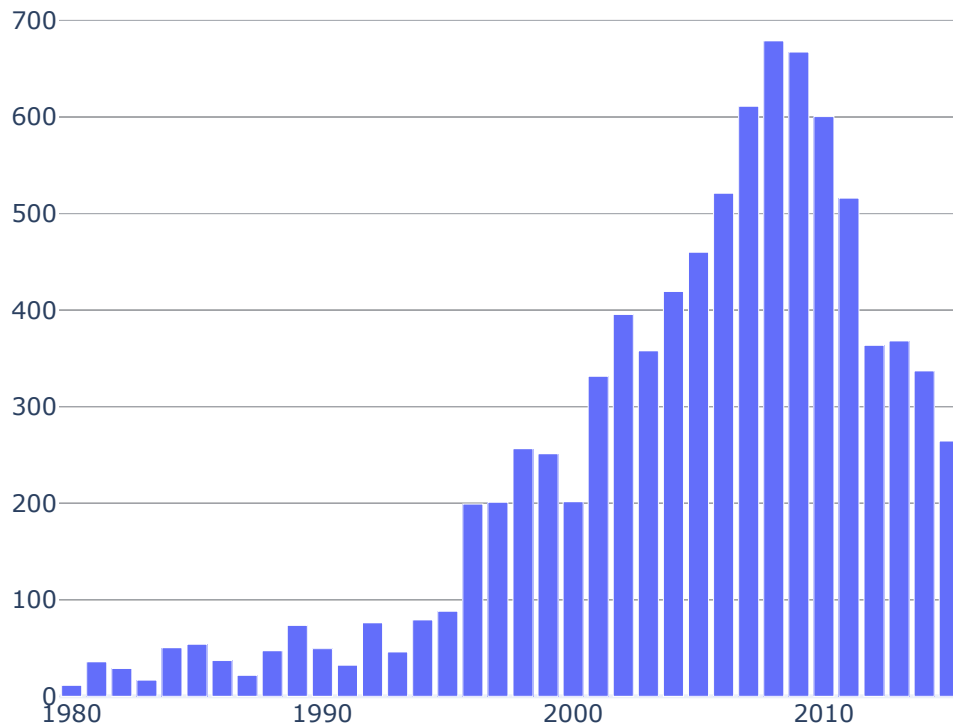
```
In [15]: dfNew.select_dtypes(include=np.number).columns.tolist()
```

```
Out[15]: ['Rank',  
          'Year',  
          'NA_Sales',  
          'EU_Sales',  
          'JP_Sales',  
          'Other_Sales',  
          'Global_Sales',  
          'id_Name',  
          'id_Platform',  
          'id_Genre',  
          'id_Publisher']
```

```
In [16]: salesYear = dfNew[((dfNew.Year>0) & (dfNew.Year < 2016))][['Global_Sales', 'Year']]
```

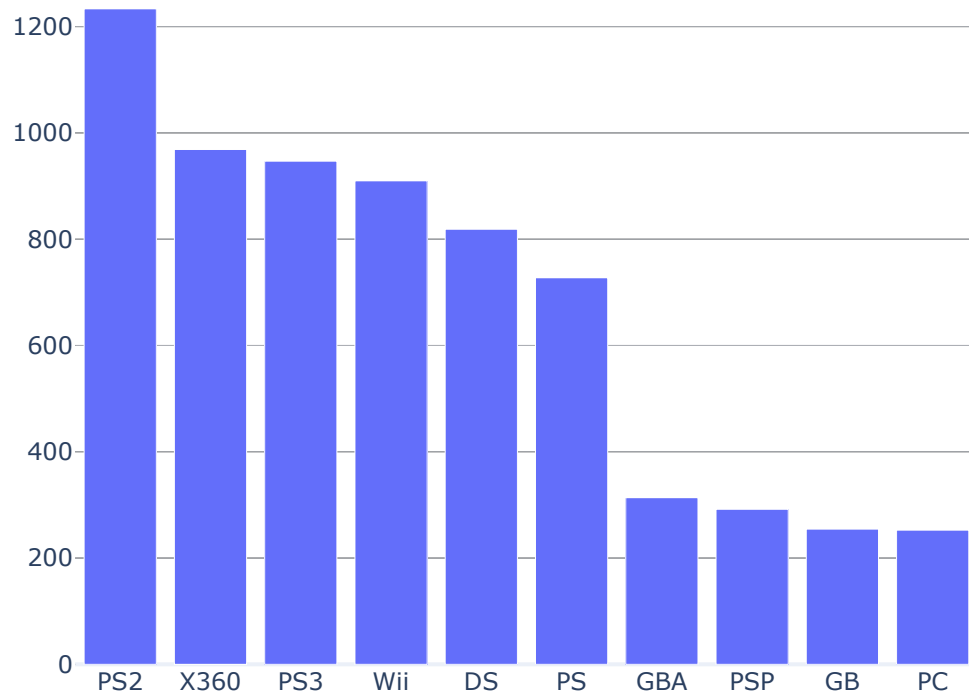
```
In [30]: fig = go.Figure([go.Bar(x=salesYear.index.tolist(), y=salesYear.values.reshape(-1))])  
fig.update_layout(template='plotly_white', title="Sales History (Million Dollars)")  
fig.show()
```

Sales History (Million Dollars)



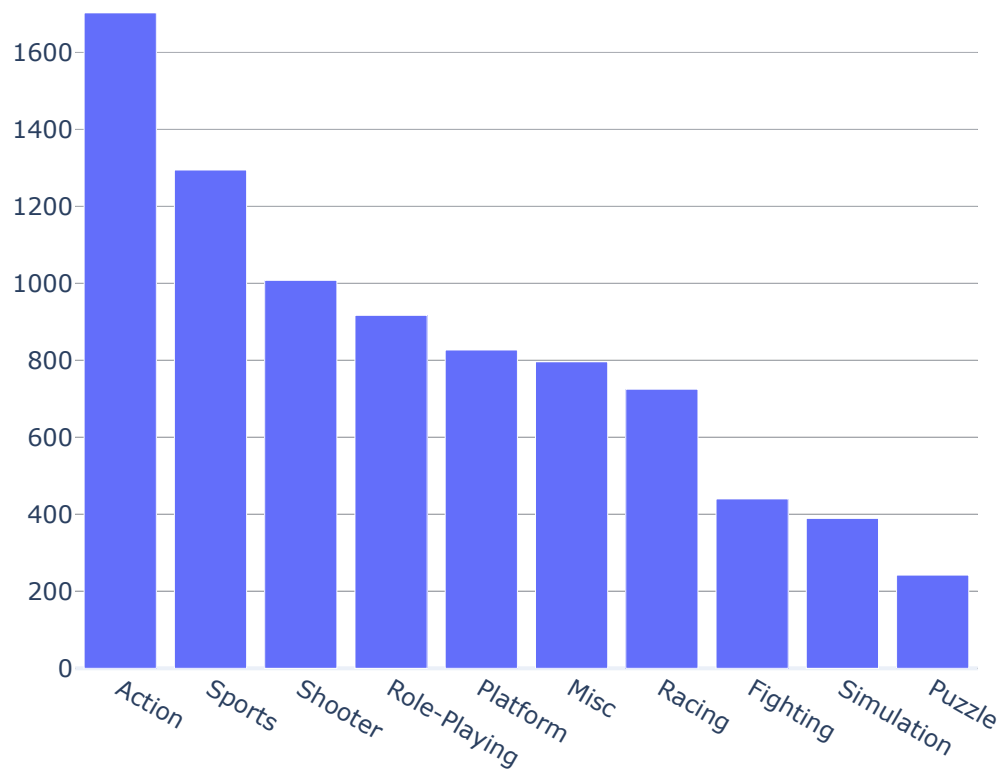
```
In [32]: >=1980')[['Platform', 'Global_Sales']].groupby(['Platform']).sum().sort_values(b
index.tolist(), y=salesPlatform.values.reshape(-1).tolist()))
', title="Games - Top 10 Platforms (Sales in Million Dollars)"]
```

Games - Top 10 Platforms (Sales in Million Dollars)



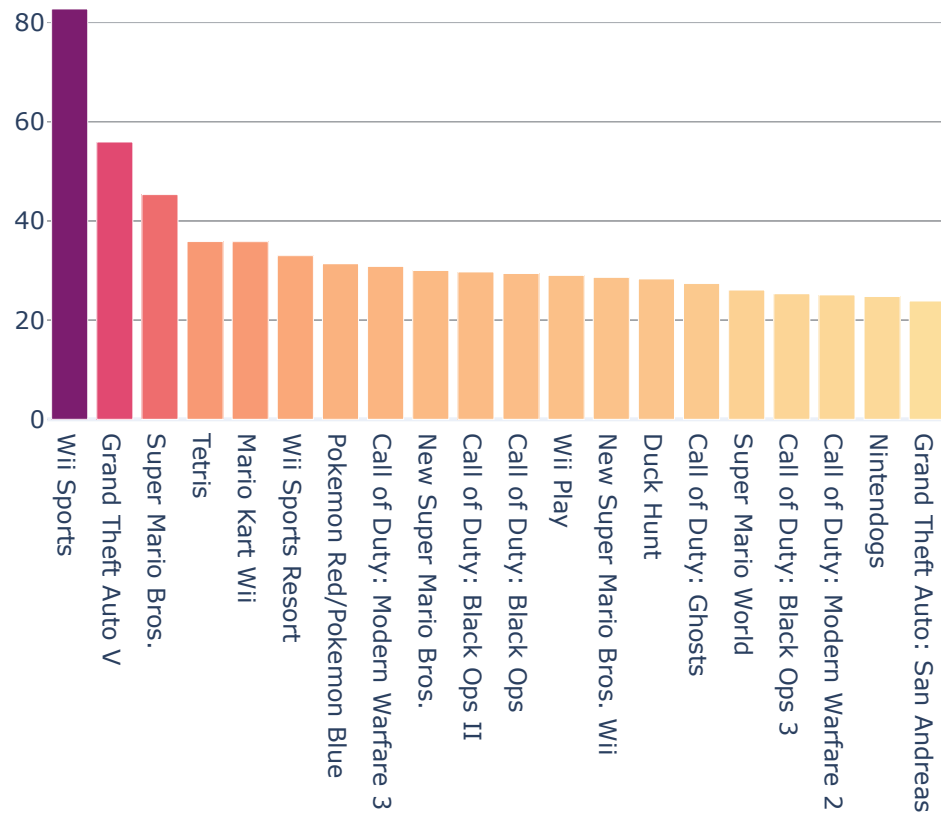

```
In [33]: > Year >=1980')[['Genre', 'Global_Sales']].groupby(['Genre']).sum().sort_values(b
Genre.index.tolist(), y=salesGenre.values.reshape(-1).tolist())
otly_white', title="Games - Top 10 Genres (Sales in Million Dollars)")
```

Games - Top 10 Genres (Sales in Million Dollars)



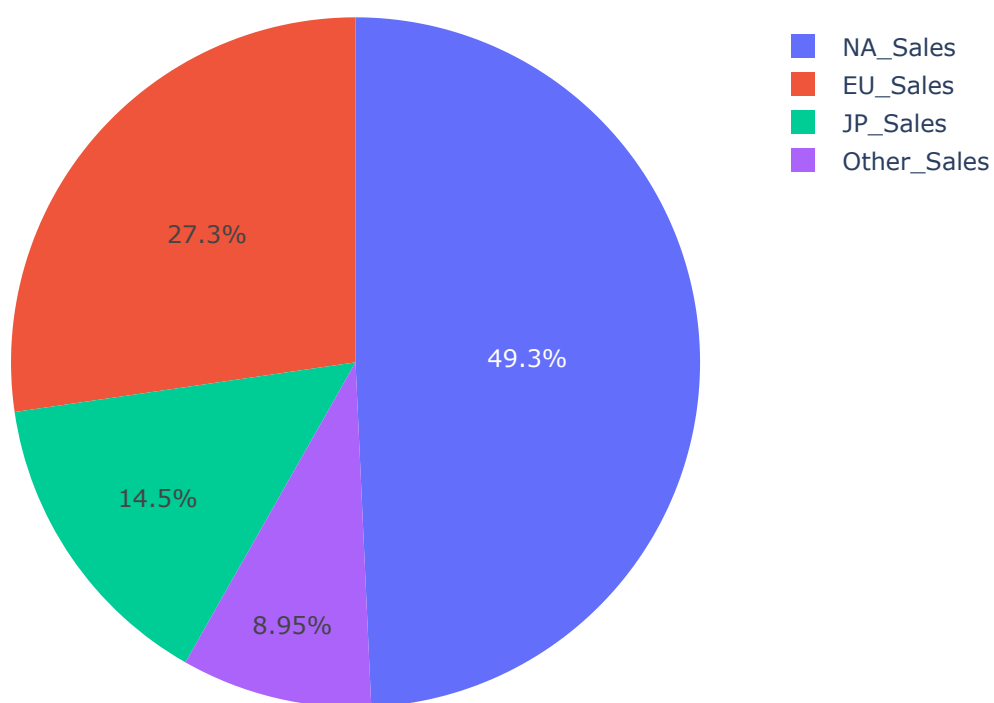
```
In [34]: = dfNew.query('2016 > Year >=1980')[['Name', 'Global_Sales']].groupby(['Name']).s
es.index.tolist()
es.values.reshape(-1).tolist()
figure([go.Bar(x=x, y=y, marker=dict(color=y, colorscale='sunsetdark'))])
layout(template='plotly_white', title="Games - Top 20 (Sales in Million Dollars)')
```

Games - Top 20 (Sales in Million Dollars)



```
In [37]: salesW = dfNew[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']].sum()
fig = go.Figure(data=[go.Pie(labels=salesW.index, values=salesW.values)])
fig.update_layout(template='plotly_white', title="World Sales (Million Dollars)")
fig.show()
```

World Sales (Million Dollars)



```
In [22]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16598 entries, 0 to 16597
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Rank                  16598 non-null  int64
1   Name                  16598 non-null  category
2   Platform              16598 non-null  category
3   Year                  16598 non-null  float64
4   Genre                 16598 non-null  category
5   Publisher             16598 non-null  category
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
11  id_Name               16598 non-null  int16
12  id_Platform           16598 non-null  int8
13  id_Genre              16598 non-null  int8
14  id_Publisher          16598 non-null  int16
dtypes: category(4), float64(6), int16(2), int64(1), int8(2)
memory usage: 1.5 MB
```

```
In [23]: df1 = pd.read_csv('vgsales.csv')
```

In [24]: df1.dropna()

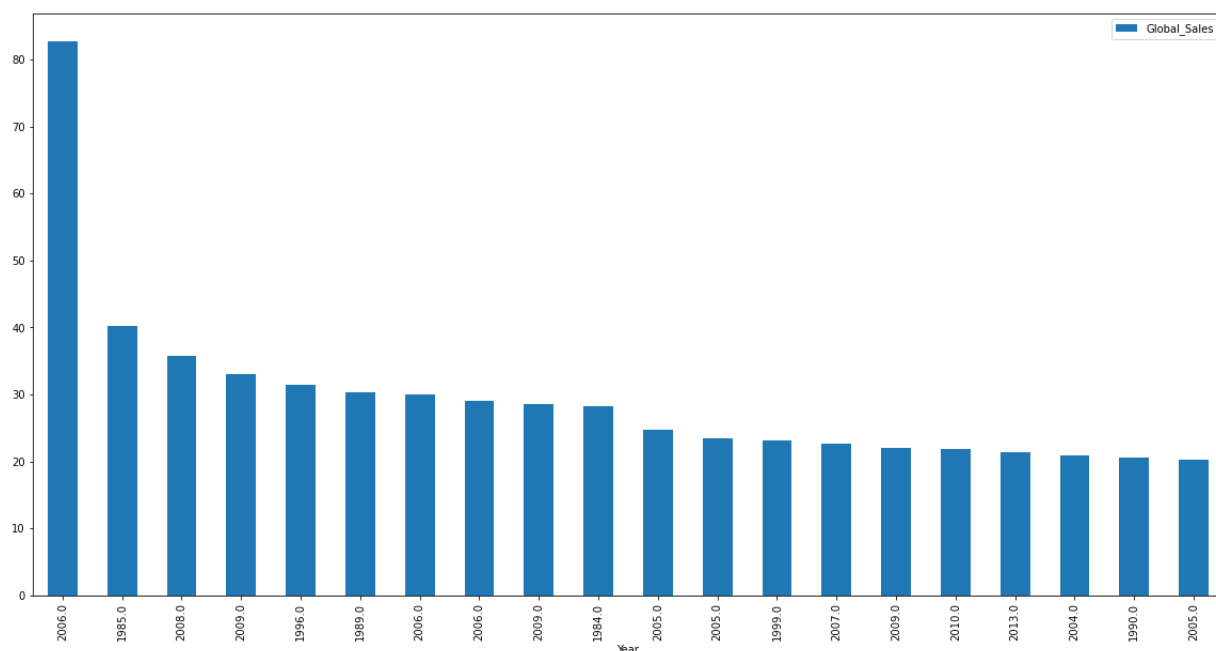
Out[24]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79
3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28
4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22
...
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00

16291 rows × 11 columns

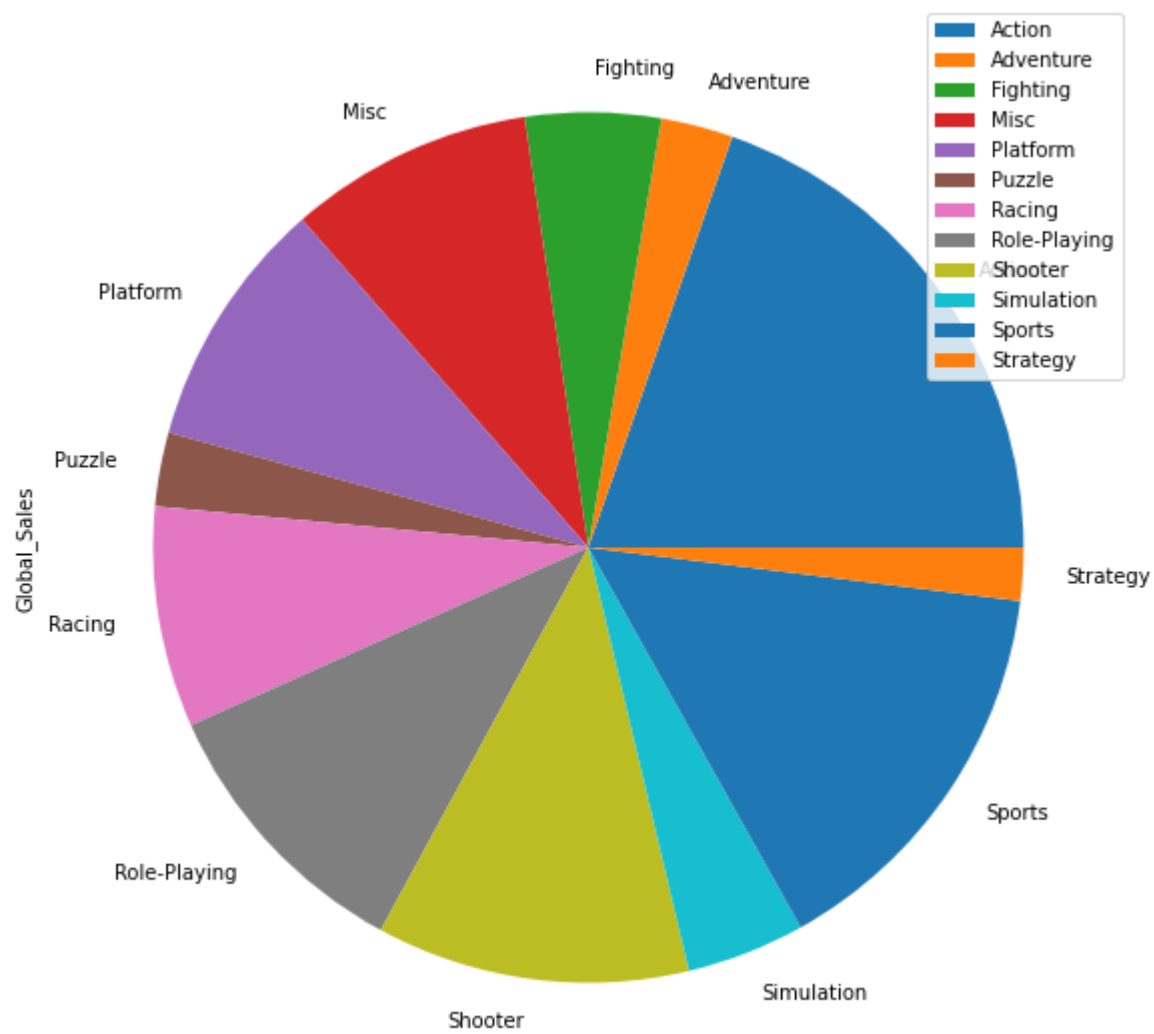


```
In [25]: df1.head(20).plot(kind='bar', x = 'Year', y = 'Global_Sales', figsize = (20,10))  
plt.show()
```

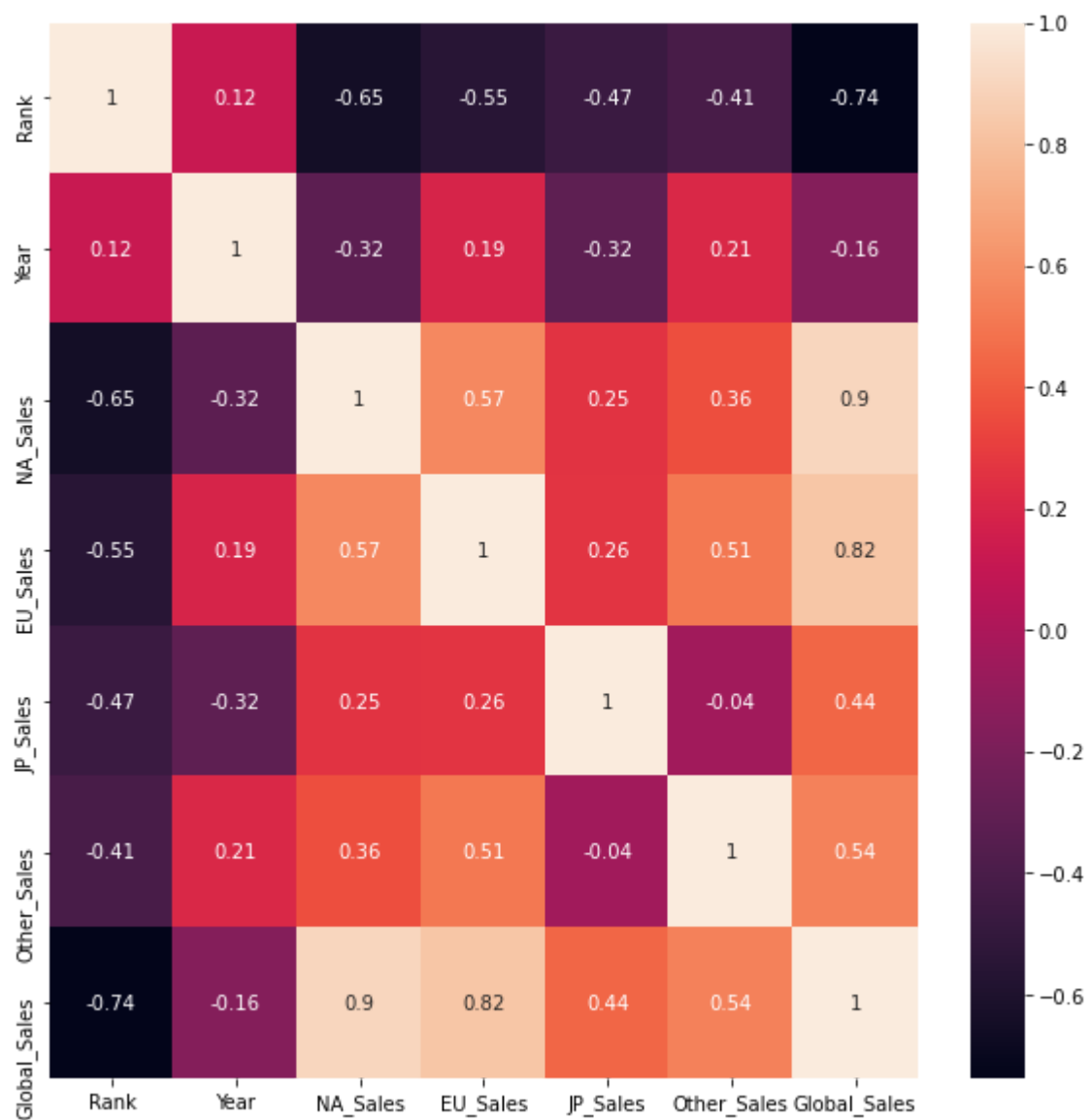


```
In [26]: df1 = df1.head(100)
```

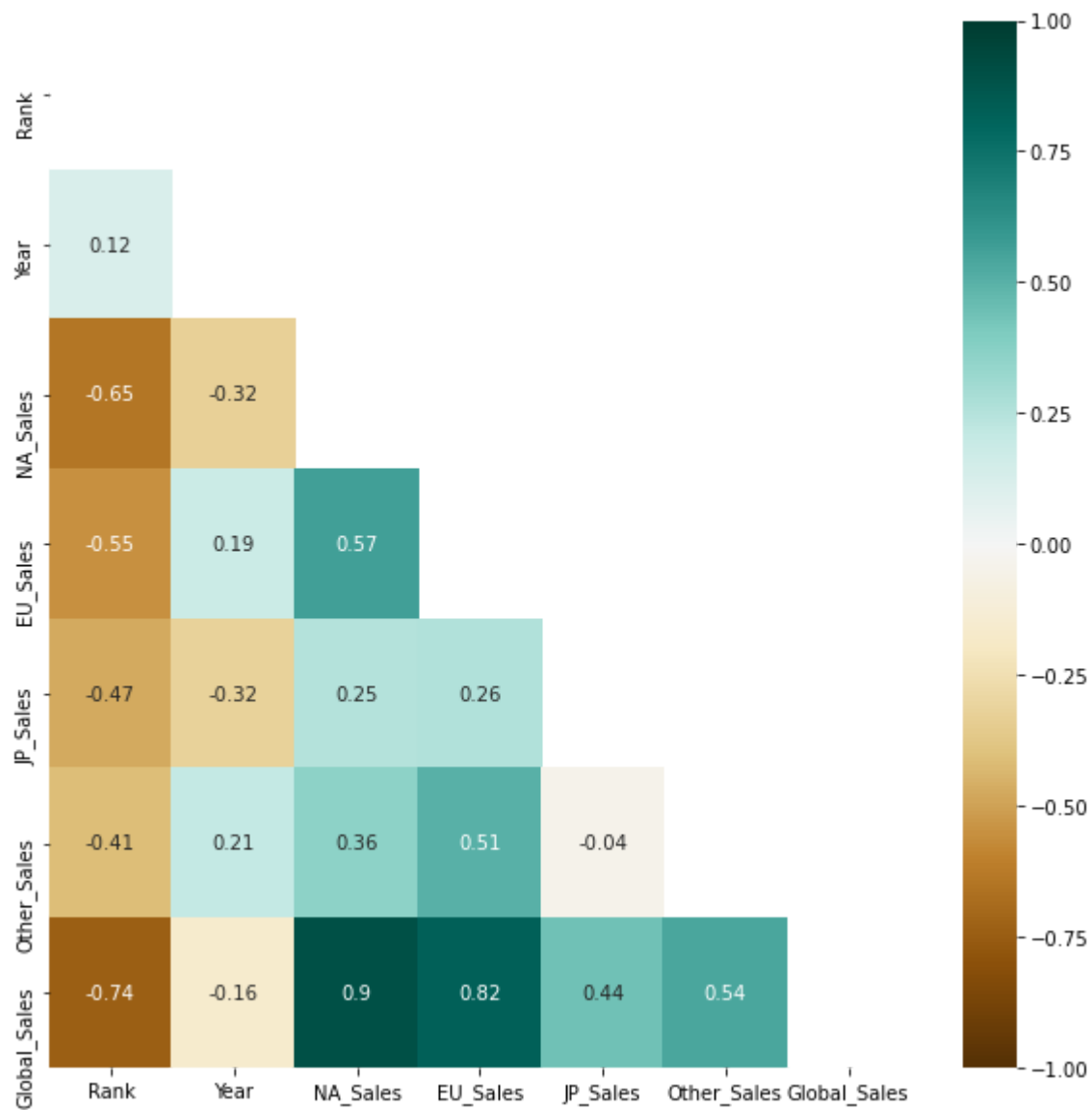
```
In [27]: df.groupby(['Genre']).sum().plot(kind = 'pie', y = 'Global_Sales', figsize = (10, 10), plt.show())
```



```
In [28]: plt.figure(figsize = (10,10))  
sns.heatmap(df1.corr(), annot=True)  
plt.show()
```




```
In [29]: plt.figure(figsize = (10,10))  
mask = np.triu(np.ones_like(df1.corr(), dtype=np.bool))  
heatmap = sns.heatmap(df1.corr(), mask=mask, vmin=-1, vmax=1, annot=True, cmap='E
```



```
In [ ]:
```

