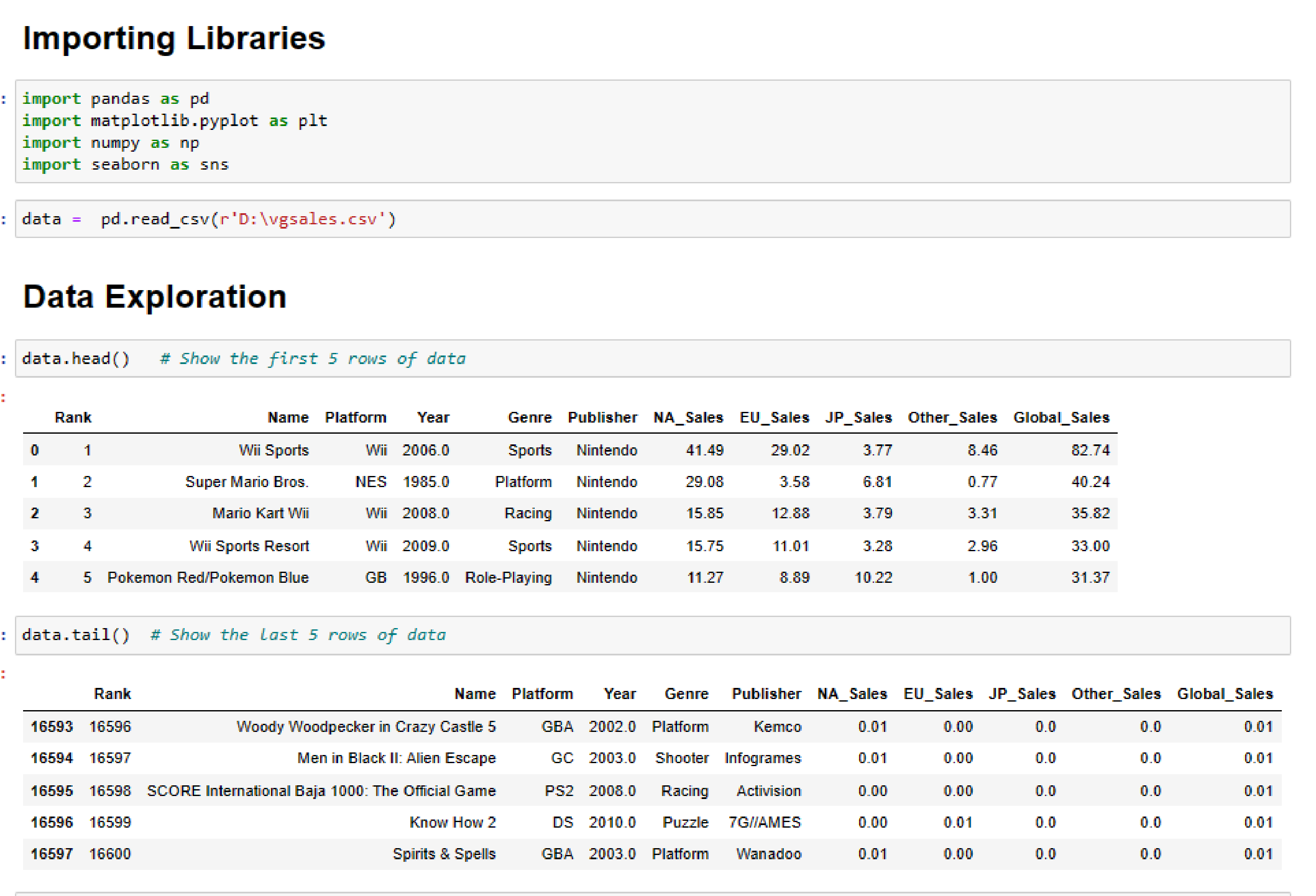


Importing Libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
data =  pd.read_csv(r'D:\vgsales.csv')
```

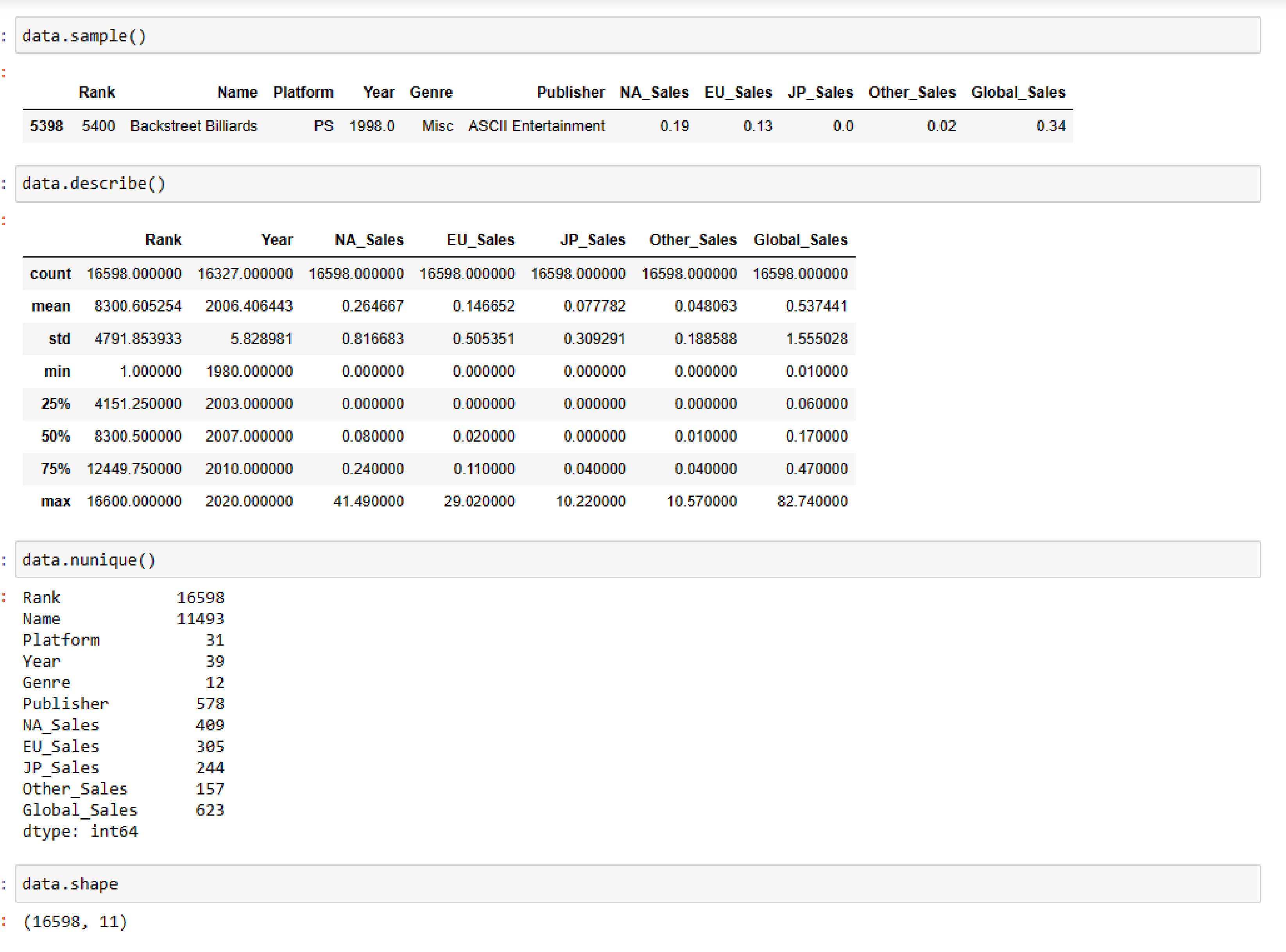
Data Exploration

```
data.head()   # Show the first 5 rows of data
```

	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

```
data.tail()  # Show the last 5 rows of data
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.0	0.0	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.0	0.0	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.0	0.0	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.0	0.0	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.0	0.0	0.01



```
data.sample()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
5398	5400	Backstreet Billiards	PS	1998.0	Misc	ASCII Entertainment	0.19	0.13	0.0	0.02	0.34

```
data.describe()
```

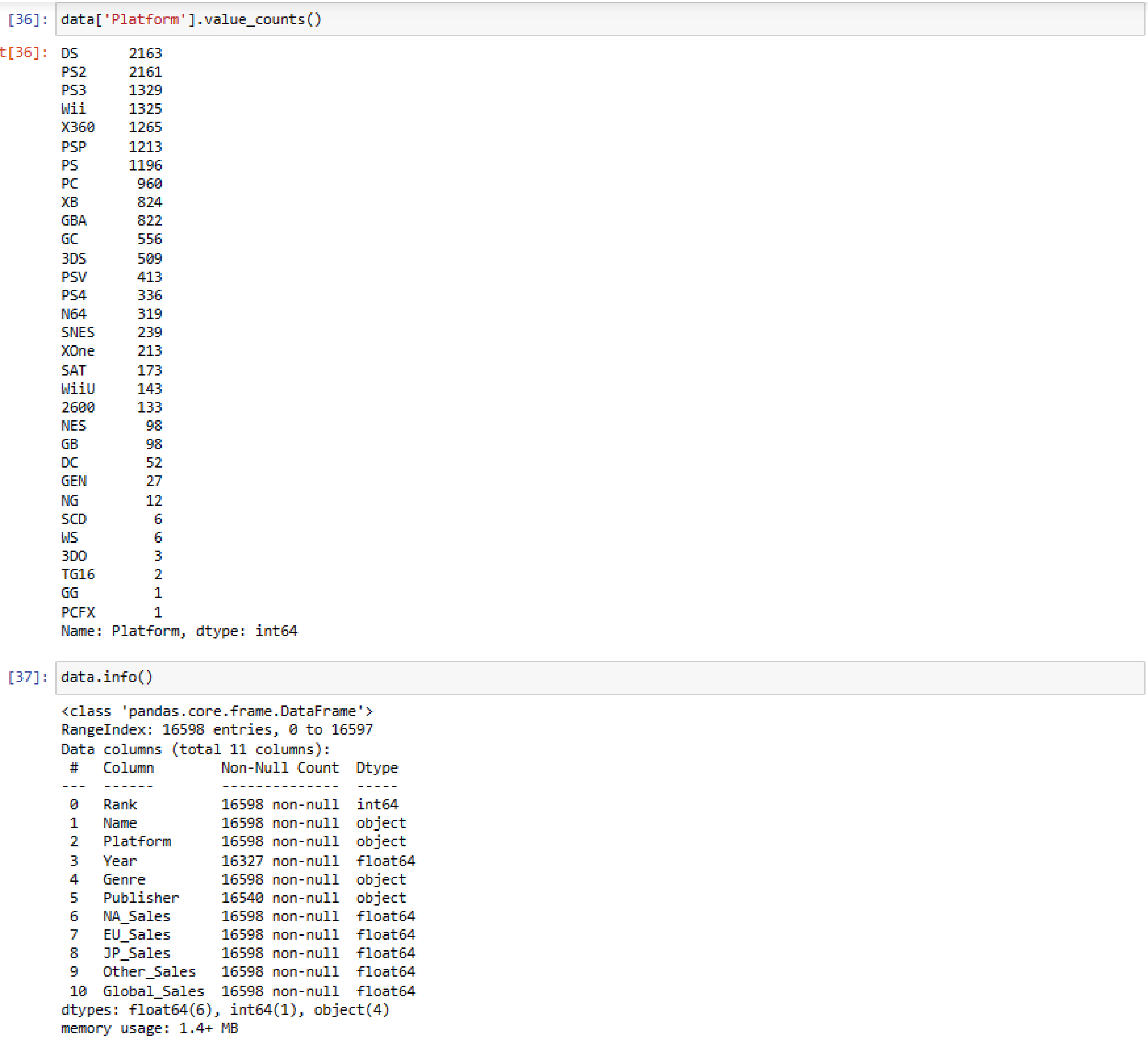
	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0.537441
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.555028
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.470000
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.740000

```
data.nunique()
```

```
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
```

```
data.shape
```

```
(16598, 11)
```

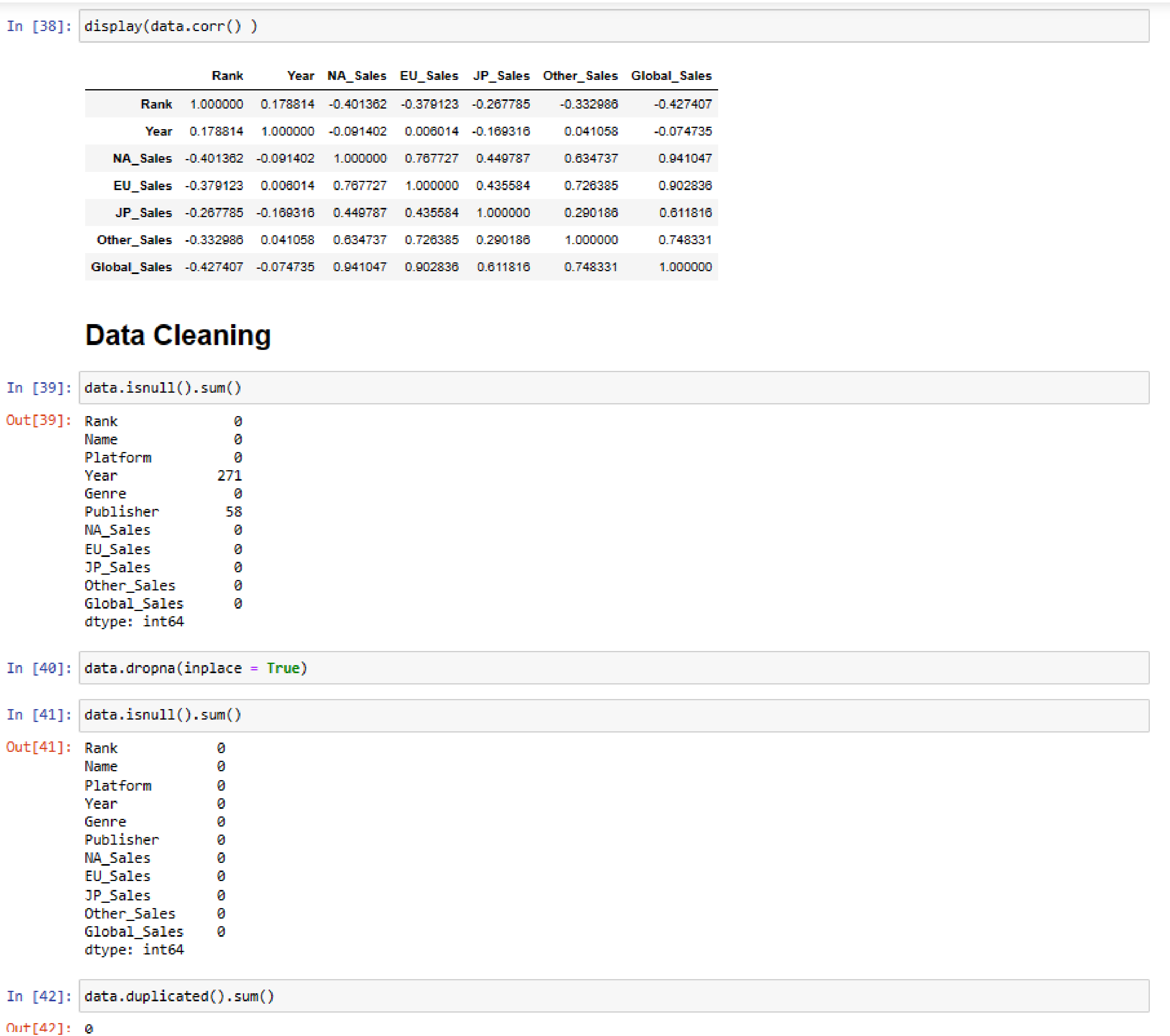


[36]:
```
data['Platform'].value_counts()
```

t[36]:
```
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
NES       98
GB        98
DC        52
GEN       27
NG        12
SCD        6
WS         6
3DO        3
TG16       2
GG         1
PCFX       1
Name: Platform, dtype: int64
```

[37]:
```
data.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 [38]: display(data.corr() )
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.178814	-0.401362	-0.379123	-0.267785	-0.332986	-0.427407
Year	0.178814	1.000000	-0.091402	0.006014	-0.169316	0.041058	-0.074735
NA_Sales	-0.401362	-0.091402	1.000000	0.767727	0.449787	0.634737	0.941047
EU_Sales	-0.379123	0.006014	0.767727	1.000000	0.435584	0.726385	0.902836
JP_Sales	-0.267785	-0.169316	0.449787	0.435584	1.000000	0.290186	0.611816
Other_Sales	-0.332986	0.041058	0.634737	0.726385	0.290186	1.000000	0.748331
Global_Sales	-0.427407	-0.074735	0.941047	0.902836	0.611816	0.748331	1.000000

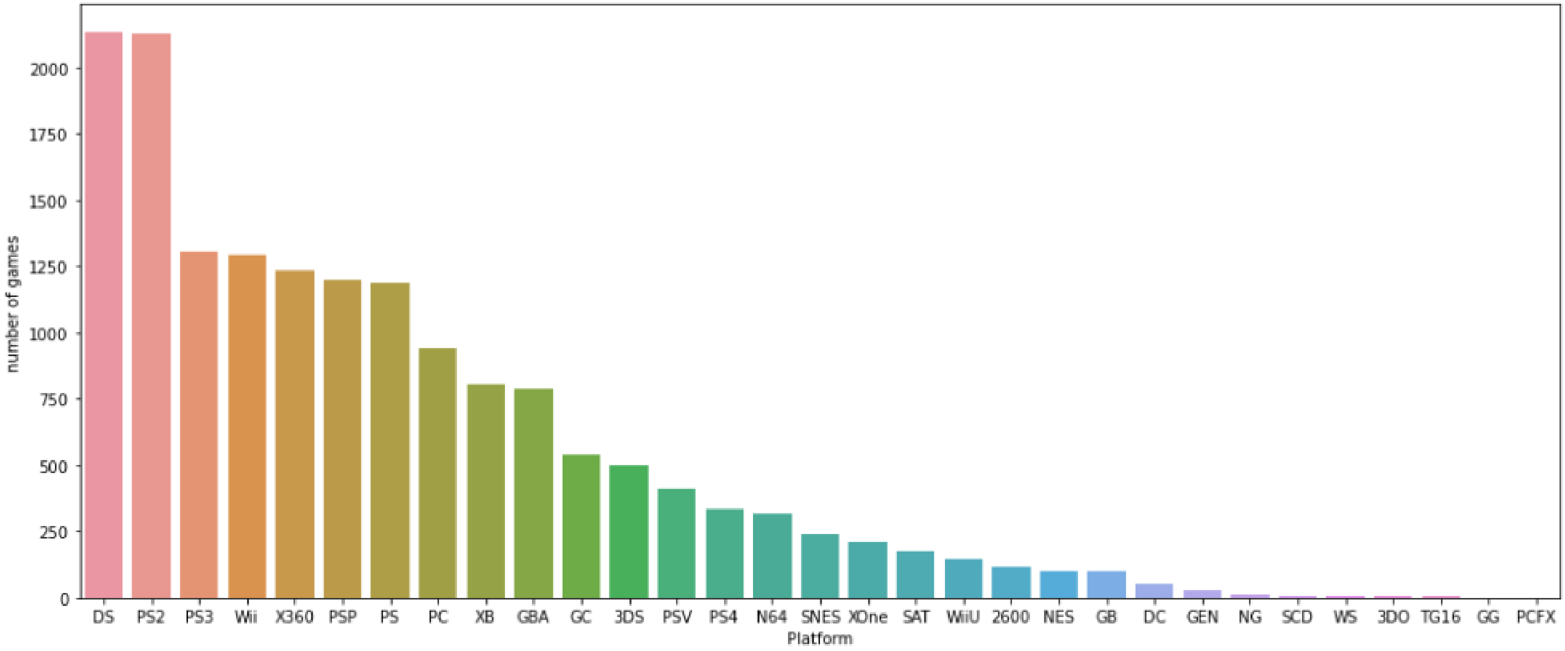
Data Cleaning

```
In [39]: data.isnull().sum()

Out[39]: Rank              0
         Name              0
         Platform          0
         Year            271
         Genre             0
         Publisher        58
         NA_Sales          0
         EU_Sales          0
         JP_Sales          0
         Other_Sales       0
         Global_Sales      0
         dtype: int64

In [40]: data.dropna(inplace = True)

In [41]: data.isnull().sum()

Out[41]: Rank            0
         Name            0
         Platform        0
         Year            0
         Genre           0
         Publisher       0
         NA_Sales        0
         EU_Sales        0
         JP_Sales        0
         Other_Sales     0
         Global_Sales    0
         dtype: int64

In [42]: data.duplicated().sum()

Out[42]: 0
```

Data visualization

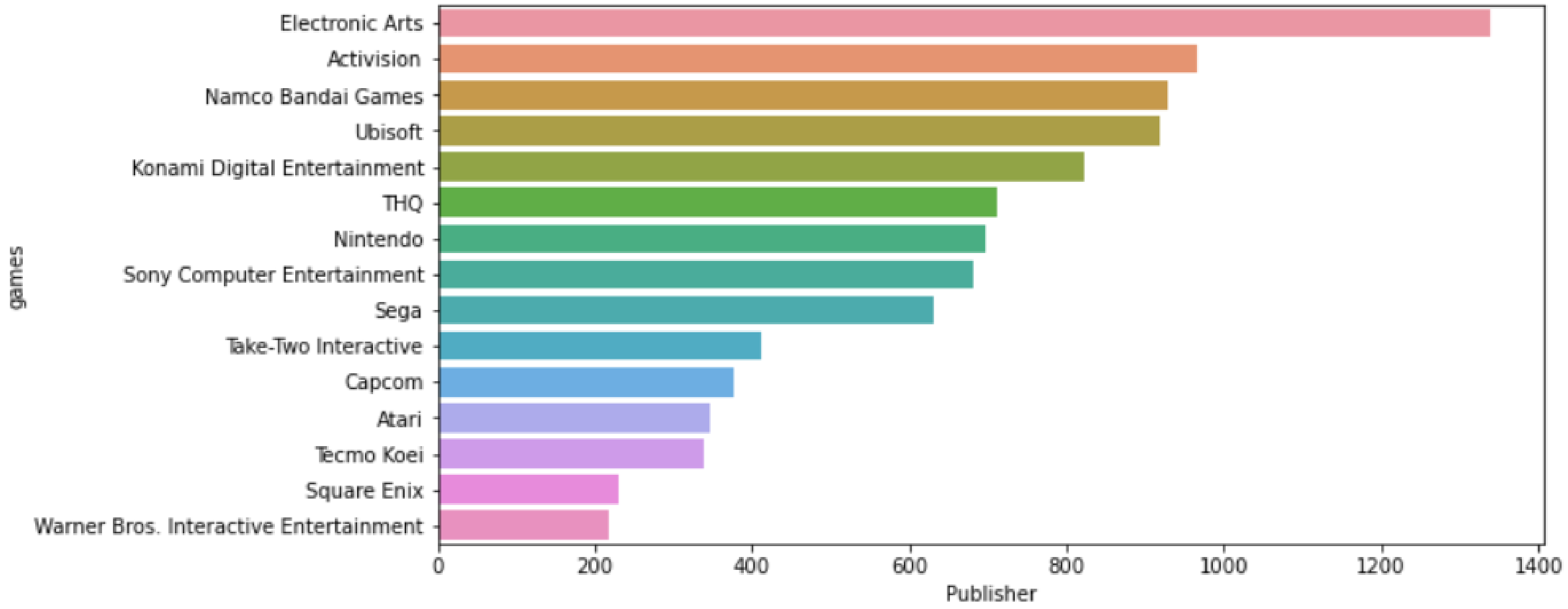
How many games are there on each platform ?

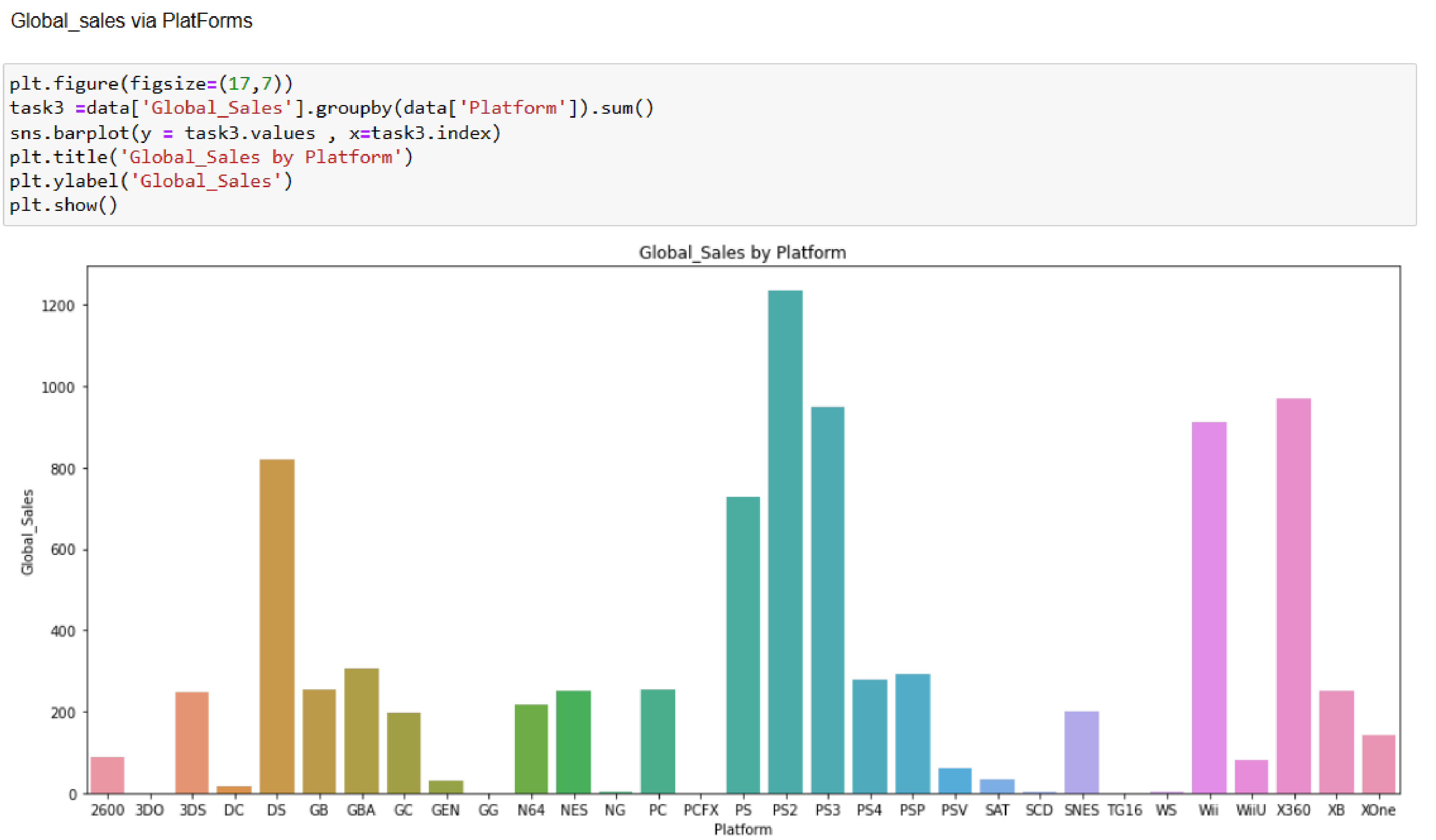
```
plt.figure(figsize=(17,7))
how = data['Platform'].value_counts()
sns.barplot(y=how.values, x=how.index)
plt.xlabel('Platform')
plt.ylabel('number of games')
plt.show()
```



How many games by publisher

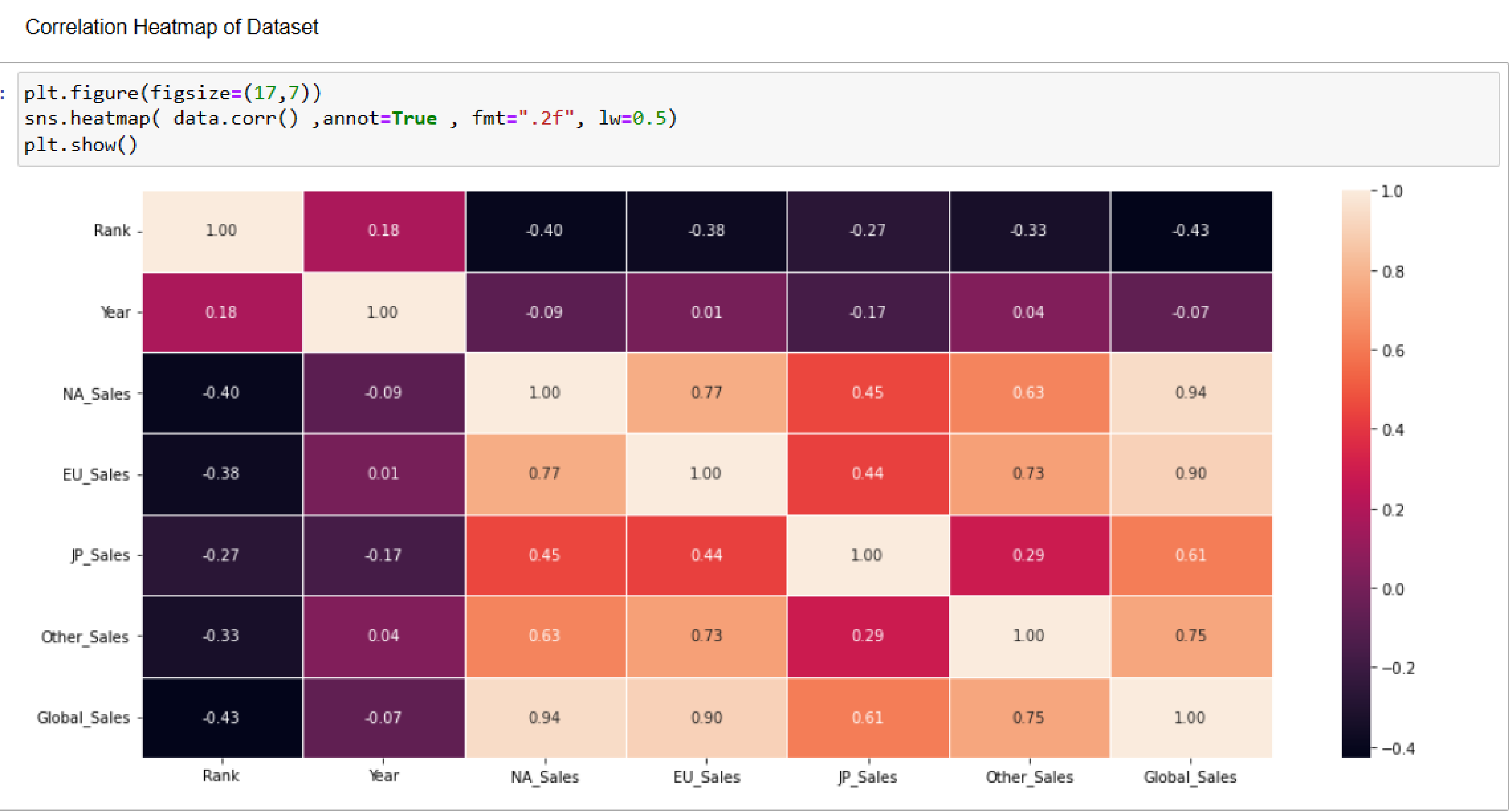
```
plt.figure(figsize=(10,5))
pub =data['Publisher'].value_counts().head(15)
sns.barplot(y = pub.index , x=pub.values)
plt.xlabel('Publisher')
plt.ylabel('games')
plt.show()
```





Global_sales via PlatForms

```
plt.figure(figsize=(17,7))
task3 =data['Global_Sales'].groupby(data['Platform']).sum()
sns.barplot(y = task3.values , x=task3.index)
plt.title('Global_Sales by Platform')
plt.ylabel('Global_Sales')
plt.show()
```



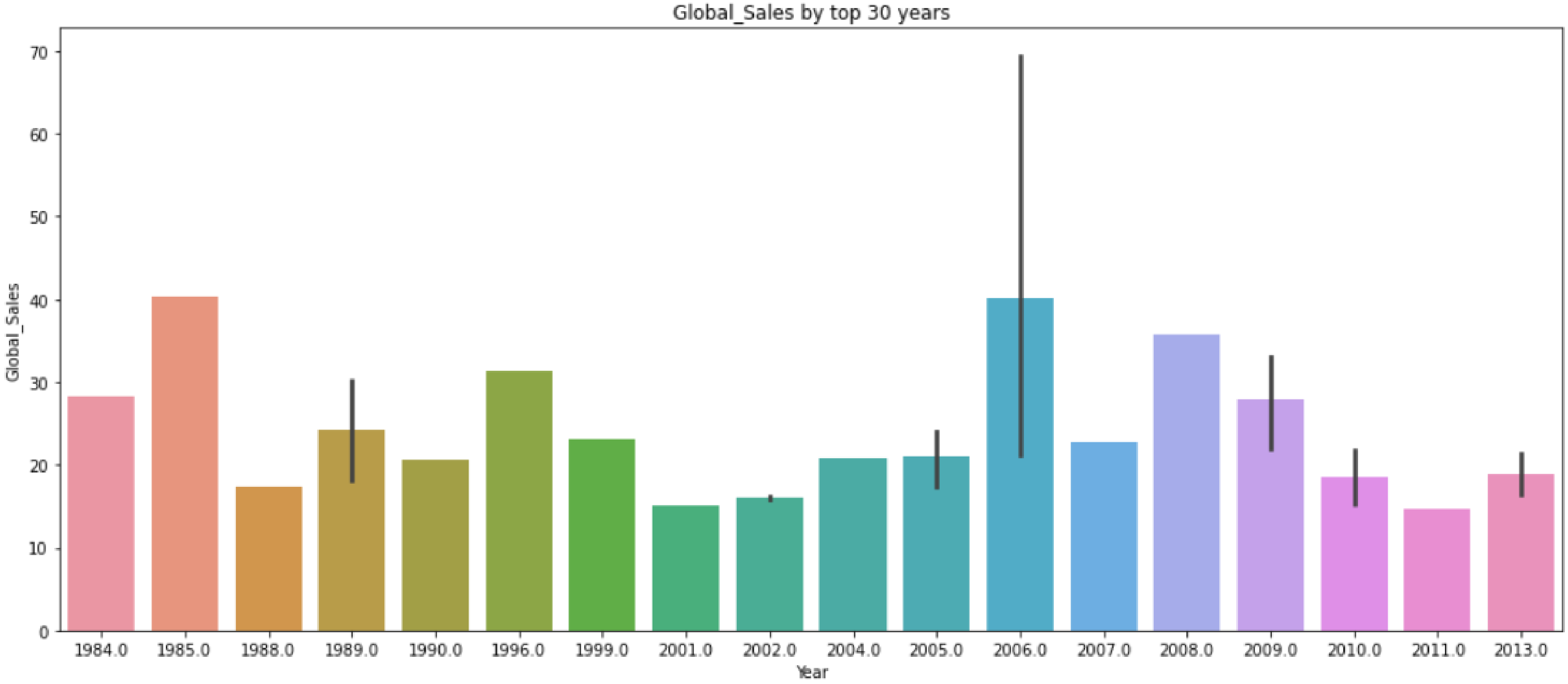
Correlation Heatmap of Dataset

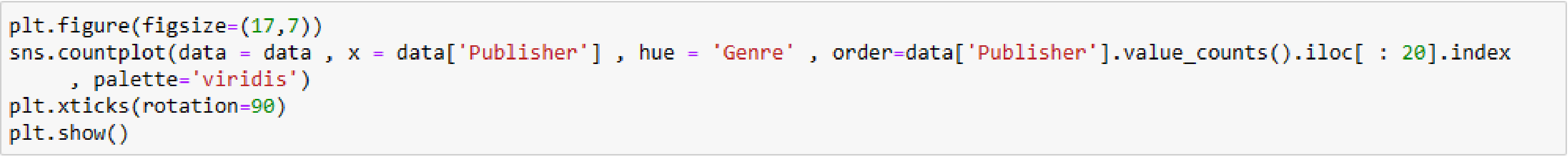
```
plt.figure(figsize=(17,7))
sns.heatmap( data.corr() ,annot=True , fmt=".2f", lw=0.5)
plt.show()
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.00	0.18	-0.40	-0.38	-0.27	-0.33	-0.43
Year	0.18	1.00	-0.09	0.01	-0.17	0.04	-0.07
NA_Sales	-0.40	-0.09	1.00	0.77	0.45	0.63	0.94
EU_Sales	-0.38	0.01	0.77	1.00	0.44	0.73	0.90
JP_Sales	-0.27	-0.17	0.45	0.44	1.00	0.29	0.61
Other_Sales	-0.33	0.04	0.63	0.73	0.29	1.00	0.75
Global_Sales	-0.43	-0.07	0.94	0.90	0.61	0.75	1.00

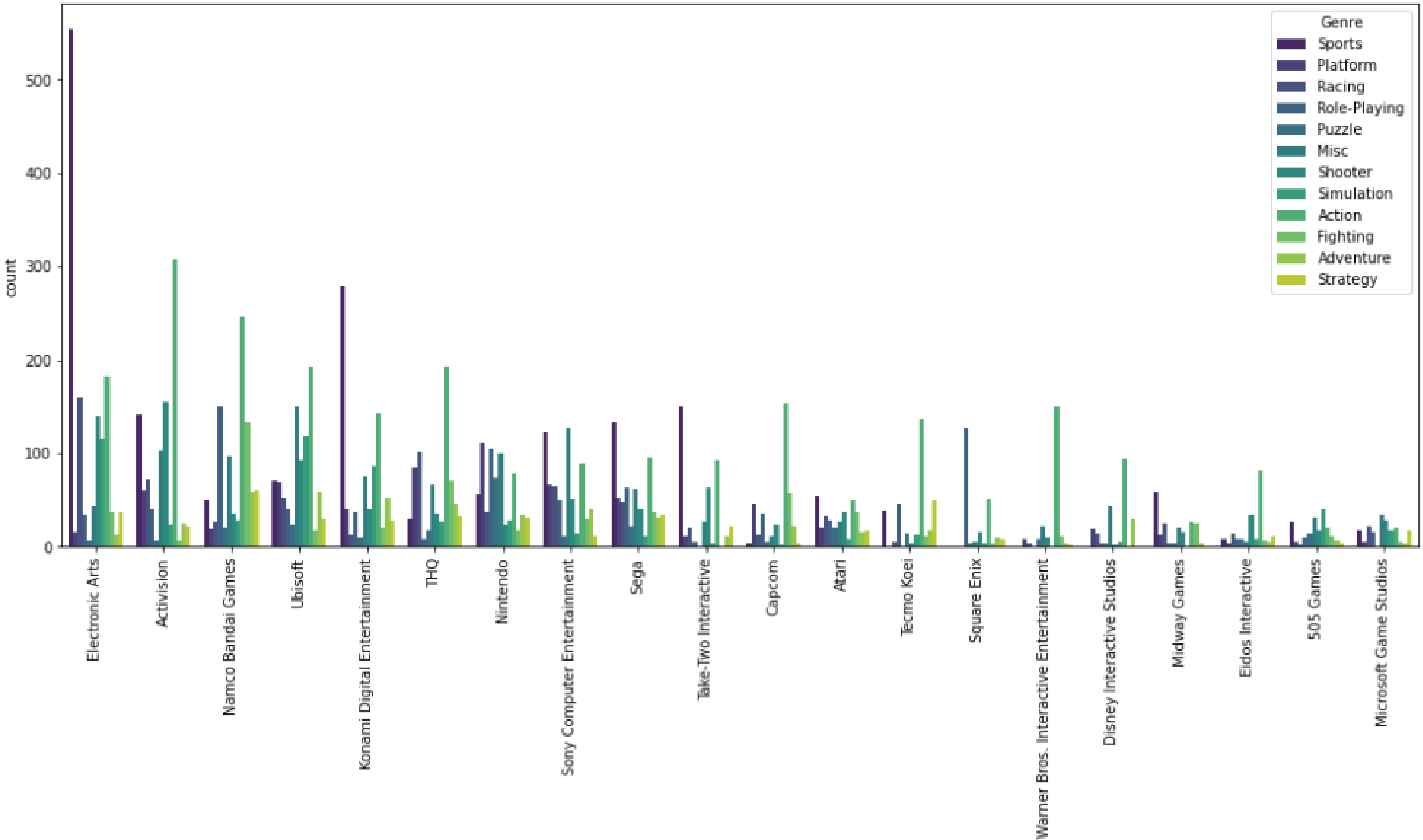
Global Sales by top 30 years

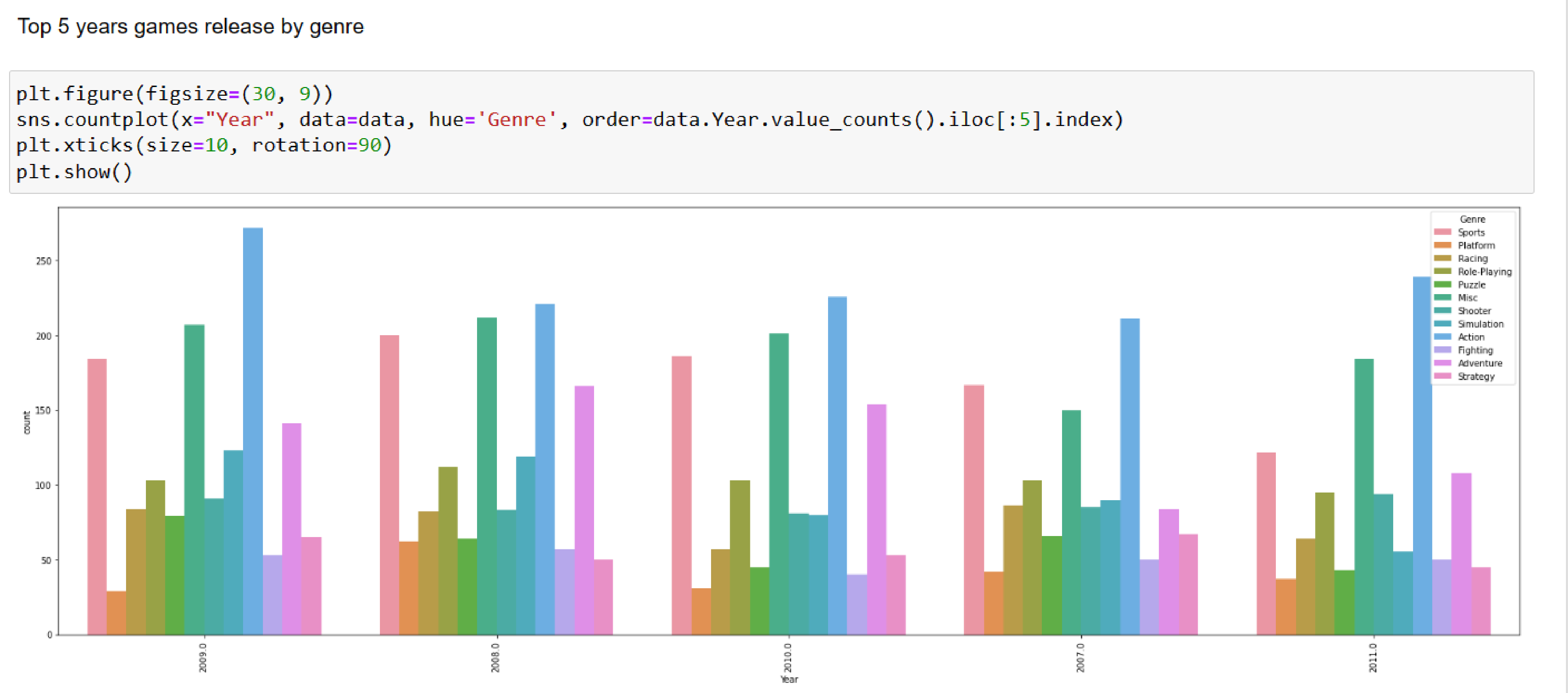
```
plt.figure(figsize=(17,7))
sns.barplot(y = 'Global_Sales' , x= (data['Year'].head(30)),data=data)
plt.title('Global_Sales by top 30 years')
plt.ylabel('Global_Sales')
plt.show()
```





```
plt.figure(figsize=(17,7))
sns.countplot(data = data , x = data['Publisher'] , hue = 'Genre' , order=data['Publisher'].value_counts().iloc[ : 20].index
     , palette='viridis')
plt.xticks(rotation=90)
plt.show()
```



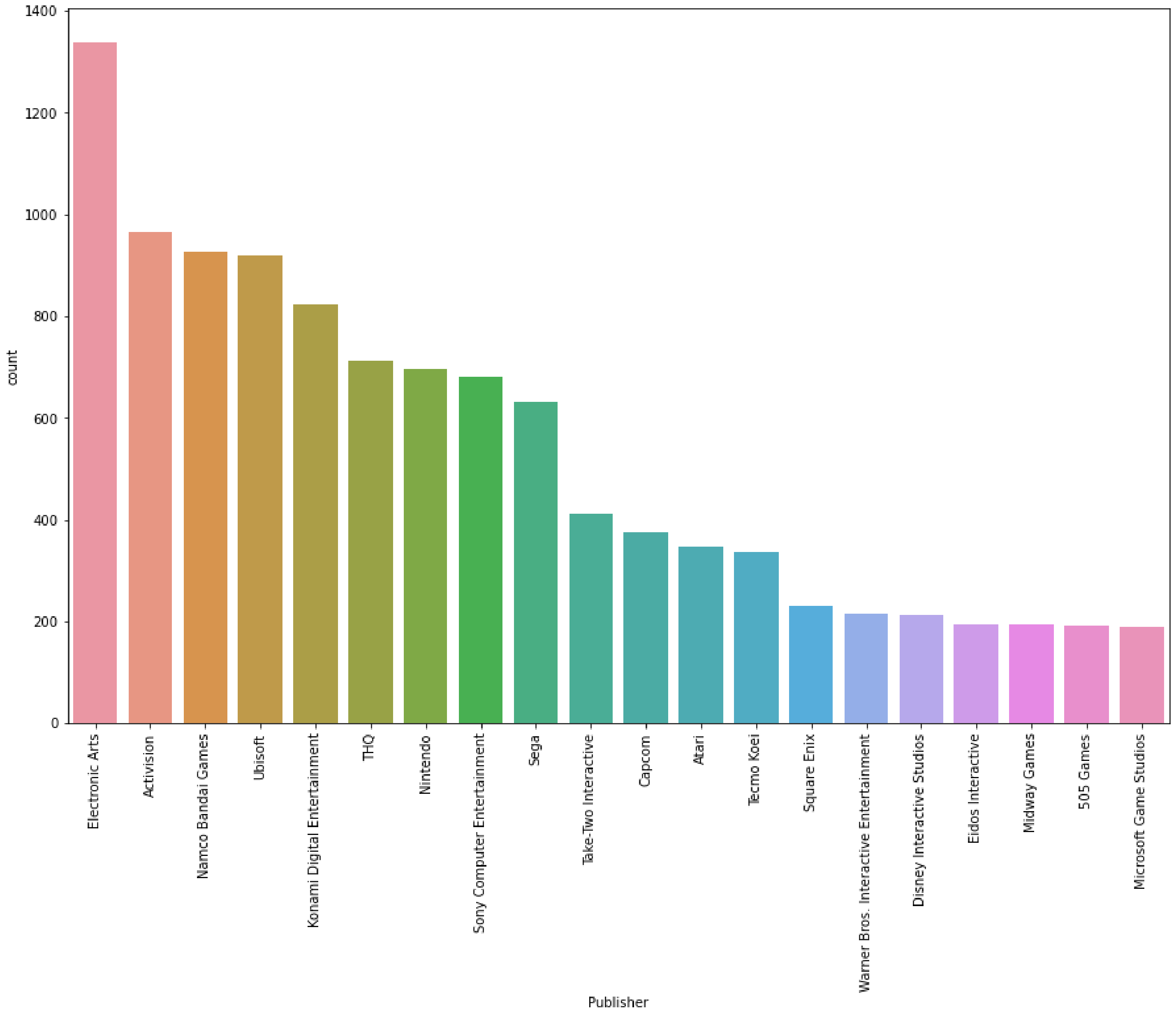


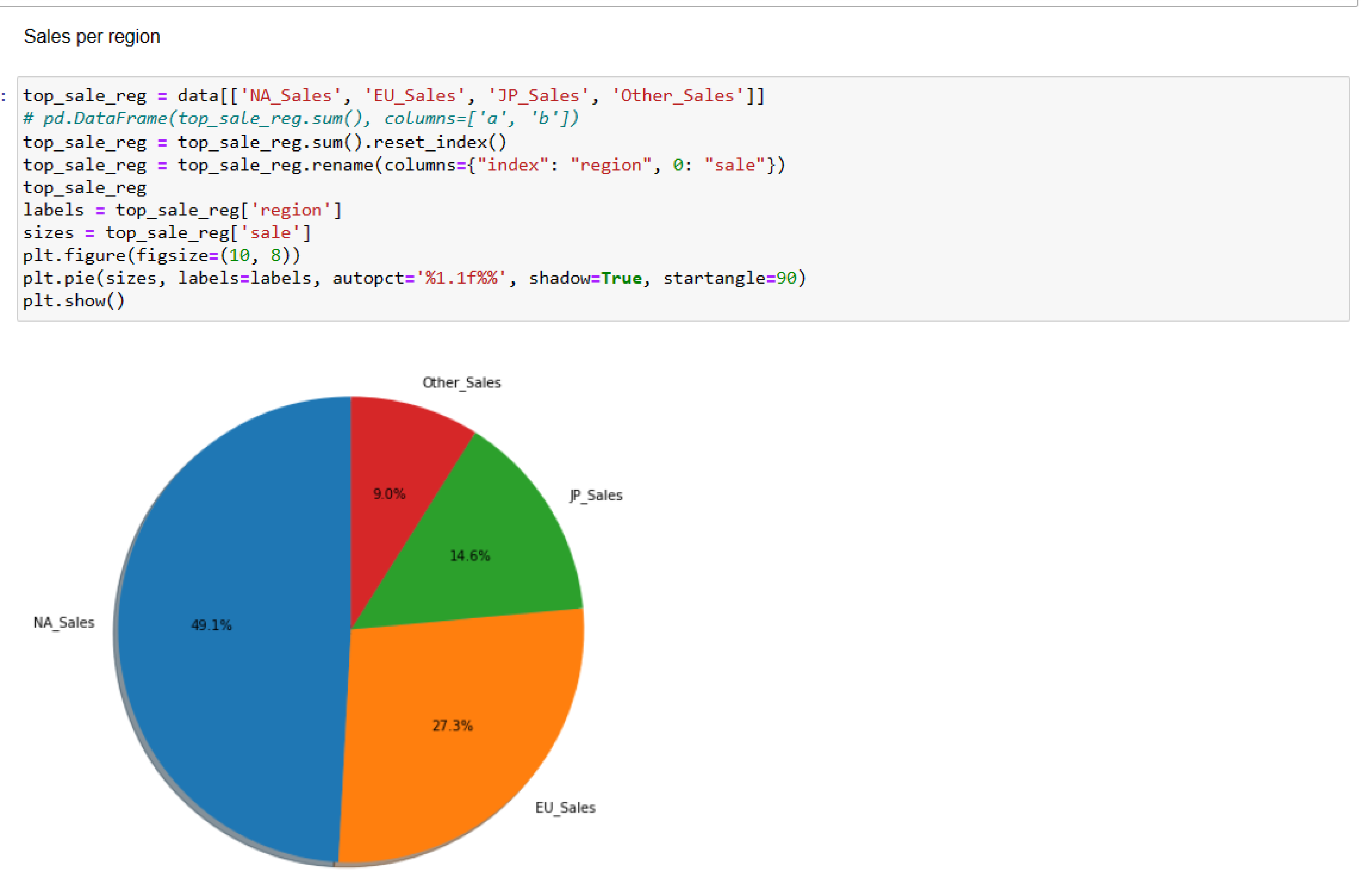
Top 5 years games release by genre

```
plt.figure(figsize=(30, 9))
sns.countplot(x="Year", data=data, hue='Genre', order=data.Year.value_counts().iloc[:5].index)
plt.xticks(size=10, rotation=90)
plt.show()
```

Top 20 Publisher

```
top_publisher = data.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=data, order = data.groupby(by=['Publisher'])
['Year'].count().sort_values(ascending=False).iloc[:20].index)
plt.xticks(rotation=90)
plt.show()
```





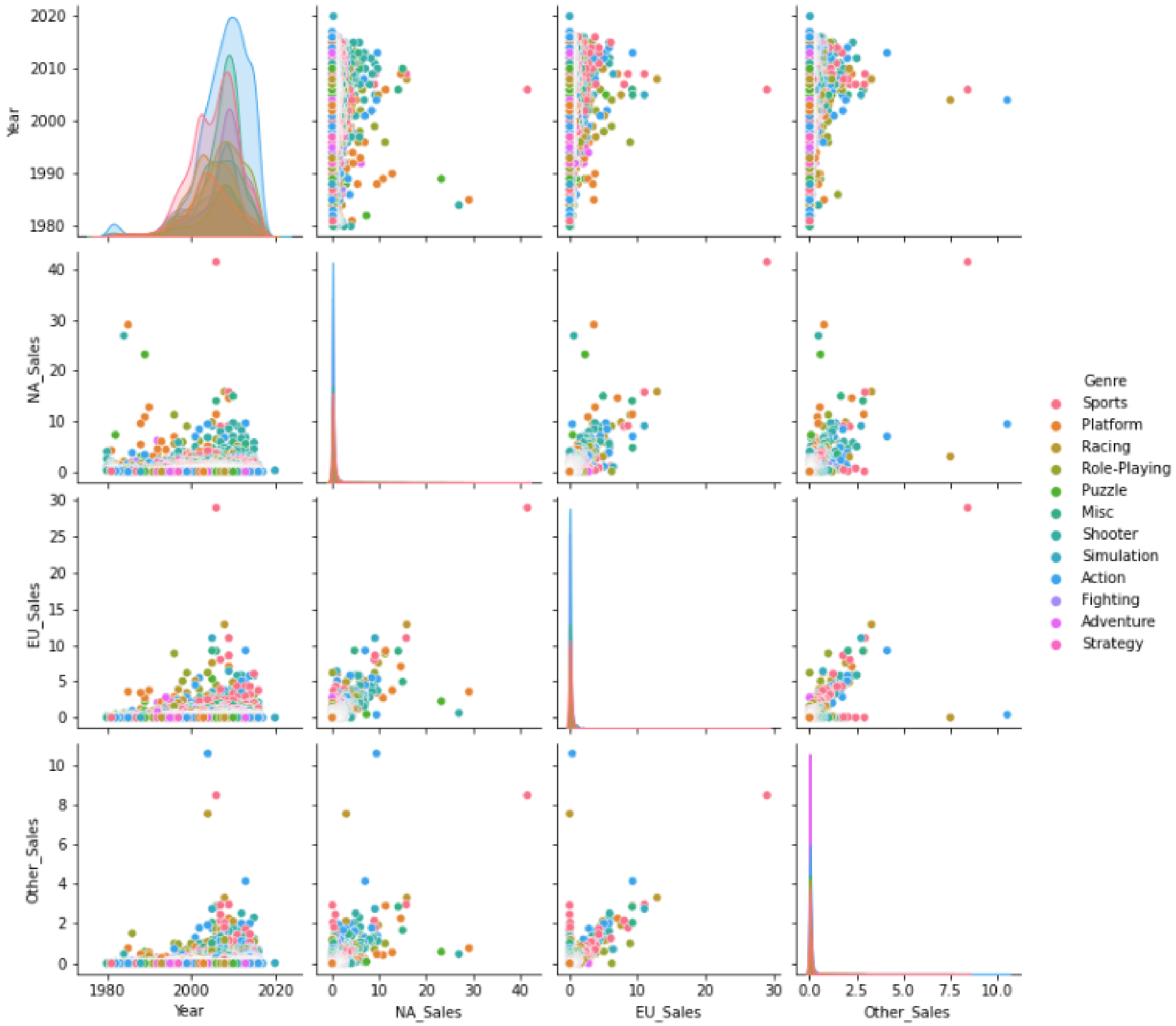
Sales per region

```
top_sale_reg = data[['NA_Sales', 'EU_Sales', 'JP_Sales', 'Other_Sales']]
# pd.DataFrame(top_sale_reg.sum(), columns=['a', 'b'])
top_sale_reg = top_sale_reg.sum().reset_index()
top_sale_reg = top_sale_reg.rename(columns={"index": "region", 0: "sale"})
top_sale_reg
labels = top_sale_reg['region']
sizes = top_sale_reg['sale']
plt.figure(figsize=(10, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
plt.show()
```

```
data_pair = data.loc[:,["Year","Platform", "Genre", "NA_Sales","EU_Sales", "Other_Sales"]]
data_pair.head()
```

	Year	Platform	Genre	NA_Sales	EU_Sales	Other_Sales
0	2006.0	Wii	Sports	41.49	29.02	8.46
1	1985.0	NES	Platform	29.08	3.58	0.77
2	2008.0	Wii	Racing	15.85	12.88	3.31
3	2009.0	Wii	Sports	15.75	11.01	2.96
4	1996.0	GB	Role-Playing	11.27	8.89	1.00

```
sns.pairplot(data_pair, hue='Genre')
plt.show()
```


Year
NA_Sales
EU_Sales
Other_Sales
Genre
Sports
Platform
Racing
Role-Playing
Puzzle
Misc
Shooter
Simulation
Action
Fighting
Adventure
Strategy