

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Read the data

```
In [16]: a=pd.read_excel("D:/Data Analytics/EDA/Video Games.xlsx")
```

View the data

```
In [7]: a
```

```
Out[7]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	
4	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	
...
91	92	Super Mario Galaxy 2	Wii	2010	Platform	Nintendo	3.66	2.42	0.98	
92	93	Star Wars Battlefront (2015)	PS4	2015	Shooter	Electronic Arts	2.93	3.29	0.22	
93	94	Call of Duty: Advanced Warfare	PS4	2014	Shooter	Activision	2.80	3.30	0.14	
94	95	The Legend of Zelda: Ocarina of Time	N64	1998	Action	Nintendo	4.10	1.89	1.45	
95	96	Crash Bandicoot 2: Cortex Strikes Back	PS	1997	Platform	Sony Computer Entertainment	3.78	2.17	1.31	

96 rows × 11 columns

To check the rows and columns

In [11]: `a.shape`

Out[11]: (96, 10)

To print top 5 records

In [12]: `a.head(5)`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Glo
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	8.46	
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	0.77	
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.31	
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	2.96	
4	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	1.00	

To print bottom 5 records

In [13]: `a.tail(5)`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Glo
91	92	Super Mario Galaxy 2	Wii	2010	Platform	Nintendo	3.66	2.42	0.64	
92	93	Star Wars Battlefront (2015)	PS4	2015	Shooter	Electronic Arts	2.93	3.29	1.23	
93	94	Call of Duty: Advanced Warfare	PS4	2014	Shooter	Activision	2.80	3.30	1.37	
94	95	The Legend of Zelda: Ocarina of Time	N64	1998	Action	Nintendo	4.10	1.89	0.16	
95	96	Crash Bandicoot 2: Cortex Strikes Back	PS	1997	Platform	Sony Computer Entertainment	3.78	2.17	0.31	

To print columns names

In [14]: `a.columns`

Out[14]: `Index(['Rank', 'Name', 'Platform', 'Year', 'Genre', 'Publisher', 'NA_Sales', 'EU_Sales', 'Other_Sales', 'Global_Sales'], dtype='object')`

To check the datatype

In [15]: `a.dtypes`

Out[15]:

Rank	int64
Name	object
Platform	object
Year	int64
Genre	object
Publisher	object
NA_Sales	float64
EU_Sales	float64
Other_Sales	float64
Global_Sales	float64
dtype:	object

To check null values

In [16]: `a.isnull()`

Out[16]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Global_Sales
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...
91	False	False	False	False	False	False	False	False	False	False
92	False	False	False	False	False	False	False	False	False	False
93	False	False	False	False	False	False	False	False	False	False
94	False	False	False	False	False	False	False	False	False	False
95	False	False	False	False	False	False	False	False	False	False

96 rows × 10 columns

Sum of null values

```
In [17]: a.isnull().sum()
```

```
Out[17]: Rank      0  
Name      0  
Platform  0  
Year      0  
Genre     0  
Publisher 0  
NA_Sales  0  
EU_Sales  0  
Other_Sales 0  
Global_Sales 0  
dtype: int64
```

To print the dimensions

```
In [18]: a.ndim
```

```
Out[18]: 2
```

To print the index

```
In [19]: a.index
```

```
Out[19]: RangeIndex(start=0, stop=96, step=1)
```

To check the size of data

```
In [20]: a.size
```

```
Out[20]: 960
```

To check the data is empty or not

```
In [21]: a.empty
```

```
Out[21]: False
```

To print the information

In [22]: a.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 96 entries, 0 to 95
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Rank              96 non-null      int64  
 1   Name              96 non-null      object  
 2   Platform          96 non-null      object  
 3   Year              96 non-null      int64  
 4   Genre             96 non-null      object  
 5   Publisher         96 non-null      object  
 6   NA_Sales          96 non-null      float64 
 7   EU_Sales          96 non-null      float64 
 8   Other_Sales       96 non-null      float64 
 9   Global_Sales      96 non-null      float64 
dtypes: float64(4), int64(2), object(4)
memory usage: 7.6+ KB
```

To print Statistical information

In [23]: a.describe()

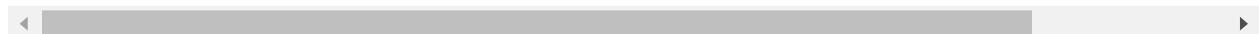
Out[23]:

	Rank	Year	NA_Sales	EU_Sales	Other_Sales	Global_Sales
count	96.000000	96.000000	96.000000	96.000000	96.000000	96.000000
mean	48.500000	2005.052083	7.157396	4.313542	1.351771	14.920417
std	27.856777	7.728784	5.969033	3.543298	1.581723	10.029910
min	1.000000	1982.000000	0.980000	0.010000	0.080000	7.580000
25%	24.750000	2001.000000	3.862500	2.410000	0.572500	9.275000
50%	48.500000	2007.000000	5.820000	3.430000	0.895000	11.590000
75%	72.250000	2011.000000	8.437500	5.060000	1.622500	16.207500
max	96.000000	2015.000000	41.490000	29.020000	10.570000	82.740000

In [24]: `a.describe(include="all")`

Out[24]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales
count	96.000000	96	96	96.000000	96	96	96.000000	96.000000	96.000000
unique	Nan	87	17	Nan	11	11	Nan	Nan	Nan
top	Nan	Grand Theft Auto V	Wii	Nan	Shooter	Nintendo	Nan	Nan	Nan
freq	Nan	3	15	Nan	20	50	Nan	Nan	Nan
mean	48.500000	Nan	Nan	2005.052083	Nan	Nan	7.157396	4.313542	1.35
std	27.856777	Nan	Nan	7.728784	Nan	Nan	5.969033	3.543298	1.58
min	1.000000	Nan	Nan	1982.000000	Nan	Nan	0.980000	0.010000	0.08
25%	24.750000	Nan	Nan	2001.000000	Nan	Nan	3.862500	2.410000	0.57
50%	48.500000	Nan	Nan	2007.000000	Nan	Nan	5.820000	3.430000	0.89
75%	72.250000	Nan	Nan	2011.000000	Nan	Nan	8.437500	5.060000	1.62
max	96.000000	Nan	Nan	2015.000000	Nan	Nan	41.490000	29.020000	10.57



check correlation()

In [38]: `a.corr()`

Out[38]:

	Rank	Year	NA_Sales	EU_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.127289	-0.657309	-0.537946	-0.399986	-0.739582
Year	0.127289	1.000000	-0.327936	0.192512	0.211917	-0.163147
NA_Sales	-0.657309	-0.327936	1.000000	0.564505	0.355117	0.901666
EU_Sales	-0.537946	0.192512	0.564505	1.000000	0.498915	0.819937
Other_Sales	-0.399986	0.211917	0.355117	0.498915	1.000000	0.534322
Global_Sales	-0.739582	-0.163147	0.901666	0.819937	0.534322	1.000000

To check duplicated values

```
In [25]: a.duplicated()
```

```
Out[25]: 0    False
1    False
2    False
3    False
4    False
...
91   False
92   False
93   False
94   False
95   False
Length: 96, dtype: bool
```

Sum of duplicated values

```
In [26]: a.duplicated().sum()
```

```
Out[26]: 0
```

No.of unique values in the dataset

```
In [27]: a.nunique()
```

```
Out[27]: Rank      96
Name      87
Platform  17
Year      27
Genre      11
Publisher 11
NA_Sales  91
EU_Sales  91
Other_Sales 79
Global_Sales 92
dtype: int64
```

Print Unique name in dataset

```
In [28]: a.Name.unique()
```

```
Out[28]: array(['Wii Sports', 'Super Mario Bros.', 'Mario Kart Wii',
 'Wii Sports Resort', 'Pokemon Red/Pokemon Blue', 'Tetris',
 'New Super Mario Bros.', 'Wii Play', 'New Super Mario Bros. Wii',
 'Duck Hunt', 'Nintendogs', 'Mario Kart DS',
 'Pokemon Gold/Pokemon Silver', 'Wii Fit', 'Wii Fit Plus',
 'Kinect Adventures!', 'Grand Theft Auto V',
 'Grand Theft Auto: San Andreas', 'Super Mario World', 'Brain Age',
 'Pokemon Diamond/Pokemon Pearl', 'Super Mario Land',
 'Super Mario Bros. 3', 'Grand Theft Auto: Vice City',
 'Pokemon Ruby/Pokemon Sapphire', 'Pokemon Black/Pokemon White',
 'Brain Age 2: More Training in Minutes a Day',
 'Gran Turismo 3: A-Spec', 'Call of Duty: Modern Warfare 3',
 'Pokemon Yellow: Special Pikachu Edition',
 'Call of Duty: Black Ops', 'Pokemon X/Pokemon Y',
 'Call of Duty: Black Ops 3', 'Call of Duty: Black Ops II',
 'Call of Duty: Modern Warfare 2', 'Grand Theft Auto III',
 'Super Smash Bros. Brawl', 'Animal Crossing: Wild World',
 'Mario Kart 7', 'Halo 3', 'Pokemon HeartGold/Pokemon SoulSilver',
 'Super Mario 64', 'Gran Turismo 4', 'Super Mario Galaxy',
 'Pokemon Omega Ruby/Pokemon Alpha Sapphire',
 'Super Mario Land 2: 6 Golden Coins', 'Grand Theft Auto IV',
 'Gran Turismo', 'Super Mario 3D Land', 'Gran Turismo 5',
 'Super Mario All-Stars', 'Pokemon FireRed/Pokemon LeafGreen',
 'Just Dance 3', 'Call of Duty: Ghosts', 'Halo: Reach',
 'Mario Kart 64', 'New Super Mario Bros. 2', 'Halo 4',
 'Final Fantasy VII', 'Just Dance 2', 'Gran Turismo 2',
 'Call of Duty 4: Modern Warfare', 'Donkey Kong Country',
 'Minecraft', 'Animal Crossing: New Leaf', 'Mario Party DS',
 'The Elder Scrolls V: Skyrim', 'Super Mario Kart', 'FIFA 16',
 'Wii Party', 'Halo 2', 'Mario Party 8',
 'Pokemon Black 2/Pokemon White 2', 'FIFA Soccer 13', 'The Sims 3',
 'GoldenEye 007', 'Mario & Sonic at the Olympic Games',
 'Final Fantasy X', 'Final Fantasy VIII',
 'Pokemon Platinum Version', 'Pac-Man',
 'Grand Theft Auto: Liberty City Stories', 'Super Mario Galaxy 2',
 'Star Wars Battlefront (2015)', 'Call of Duty: Advanced Warfare',
 'The Legend of Zelda: Ocarina of Time',
 'Crash Bandicoot 2: Cortex Strikes Back'], dtype=object)
```

Count the names

In [29]: `a.Name.value_counts()`

Out[29]:

Grand Theft Auto V	3
Call of Duty: Modern Warfare 2	2
Call of Duty: Black Ops	2
Call of Duty: Modern Warfare 3	2
Call of Duty: Ghosts	2
..	
Super Mario Kart	1
Just Dance 2	1
New Super Mario Bros. Wii	1
GoldenEye 007	1
Gran Turismo 3: A-Spec	1
Name: Name, Length: 87, dtype: int64	

To print random data

In [39]: `a.sample(5)`

Out[39]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	G
85	86	Mario & Sonic at the Olympic Games	Wii	2007	Sports	Sega	2.58	3.90		0.91
54	55	Gran Turismo 5	PS3	2010	Racing	Sony Computer Entertainment	2.96	4.88		2.12
93	94	Call of Duty: Advanced Warfare	PS4	2014	Shooter	Activision	2.80	3.30		1.37
50	51	Super Mario Land 2: 6 Golden Coins	GB	1992	Adventure	Nintendo	6.16	2.04		0.29
77	78	FIFA 16	PS4	2015	Sports	Electronic Arts	1.11	6.06		1.26

To show the maximum value of NA_Sales

In [31]: `a.sort_values('NA_Sales', ascending=False).head(5)`

Out[31]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	8.46	82.8
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	0.77	40.3
9	10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.47	28.1
5	6	Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	0.58	30.1
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.31	35.1

To show the maximum value of EU_Sales

In [32]: `a.sort_values("EU_Sales", ascending=False).head()`

Out[32]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Global_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	8.46	82.8
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.31	35.1
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	2.96	29.7
10	11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00	2.75	23.8
16	17	Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.01	9.27	4.14	20.4

To show the minimum value of NA_Sales

In [33]: `a.sort_values("NA_Sales", ascending=True).head()`

Out[33]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Glob
83	84	The Sims 3	PC	2009	Simulation	Electronic Arts	0.98	6.42		0.71
82	83	FIFA Soccer 13	PS3	2012	Action	Electronic Arts	1.06	5.05		2.01
77	78	FIFA 16	PS4	2015	Sports	Electronic Arts	1.11	6.06		1.26
78	79	Wii Party	Wii	2010	Misc	Nintendo	1.79	3.53		0.68
73	74	Animal Crossing: New Leaf	3DS	2012	Simulation	Nintendo	2.01	2.32		0.41

To show the minimum value of EU_Sales

In [34]: `a.sort_values("EU_Sales", ascending=True).tail()`

Out[34]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Glob
16	17	Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.01	9.27		4.14
10	11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00		2.75
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01		2.96
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88		3.31
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02		8.46

Top Genres

In [35]: `a.sort_values("Genre").head()`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales
		Pokemon HeartGold/Pokemon SoulSilver	DS	2009	Action	Nintendo	4.40	2.77	0.77
45	46	Grand Theft Auto: Vice City	PS2	2002	Action	Take-Two Interactive	8.41	5.49	1.78
24	25	The Legend of Zelda: Ocarina of Time	N64	1998	Action	Nintendo	4.10	1.89	0.16
94	95	Grand Theft Auto: Liberty City Stories	PSP	2005	Action	Take-Two Interactive	2.90	2.83	1.75
90	91	Grand Theft Auto V	X360	2013	Action	Take-Two Interactive	9.63	5.31	1.38
23	24								

To view the record of Grand Theft Auto V

In [36]: `a_Grand_Theft_Auto_V=a[a["Name"]=="Grand Theft Auto V"]
a_Grand_Theft_Auto_V`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	Other_Sales	Global_Sale
		Grand Theft Auto V	PS3	2013	Action	Take-Two Interactive	7.01	9.27	4.14	21.4
16	17	Grand Theft Auto V	X360	2013	Action	Take-Two Interactive	9.63	5.31	1.38	16.3
23	24	Grand Theft Auto V	PS4	2014	Action	Take-Two Interactive	3.80	5.81	2.02	11.9

view Last data of NA_Sales

In [37]: `a["NA_Sales"].iloc[-1]`

Out[37]: 3.78

Top Genres in North America

```
In [49]: top_US = a[['Name', 'Platform', 'Year', 'Genre', 'Publisher', 'NA_Sales']].sort_values(['NA_Sales']).head()
```

```
Out[49]: 0    41.49
1    29.08
9    26.93
5    23.20
2    15.85
Name: NA_Sales, dtype: float64
```

```
In [53]: top_Genres=top_US.groupby('Genre')['NA_Sales'].sum().to_frame(name='NA').sort_values(['NA'])
display(top_Genres)
```

NA	
Genre	
Shooter	146.37
Platform	132.69
Sports	78.96
Role-Playing	76.91
Action	69.25
Misc	61.28
Racing	60.21
Puzzle	33.92
Simulation	14.61
Fighting	6.75
Adventure	6.16

Top Genres in Europe

```
In [50]: top_EU=a[['Name', 'Platform', 'Year', 'Genre', 'Publisher', 'EU_Sales']].sort_values(['EU_Sales']).head()
```

```
Out[50]: 0    29.02
2    12.88
3    11.01
10   11.00
16   9.27
Name: EU_Sales, dtype: float64
```

```
In [54]: top_EU_genres=top_EU.groupby('Genre')[ 'EU_Sales'].sum().to_frame(name='EU').sort_
display(top_EU_genres)
```

EU	
Genre	
Shooter	68.25
Sports	66.61
Role-Playing	54.64
Platform	53.64
Action	50.19
Racing	44.81
Misc	39.98
Simulation	23.26
Puzzle	8.07
Fighting	2.61
Adventure	2.04

Top Genres in Japan

```
In [51]: top_JP=a[['Name','Platform','Year','Genre','Publisher','JP_Sales']].sort_values(b
top_JP['JP_Sales']).head()
```

```
Out[51]: 4      10.22
12     7.20
1      6.81
6      6.50
20     6.04
Name: JP_Sales, dtype: float64
```

```
In [42]: top_JP_genres=top_JP.groupby('Genre')[ 'JP_Sales'].sum().to_frame(name='JP').sort_
display(top_JP_genres)
```

JP	
Genre	
Role-Playing	63.75
Platform	45.94
Racing	24.64
Sports	13.90
Misc	13.41
Simulation	11.62
Puzzle	9.54
Action	8.93
Shooter	4.36
Adventure	2.69
Fighting	2.66

Extracting Data for Games Published in The Year 2010

In [57]:

```
year2010=d[d['Year']==2010]
year2010
```

Out[57]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	C
15	16	Kinect Adventures!	X360	2010	Misc	Microsoft Game Studios	14.97	4.94	0.24	
26	27	Pokemon Black/Pokemon White	DS	2010	Role-Playing	Nintendo	5.57	3.28	5.65	
31	32	Call of Duty: Black Ops	X360	2010	Shooter	Activision	9.67	3.73	0.11	
40	41	Call of Duty: Black Ops	PS3	2010	Shooter	Activision	5.98	4.44	0.48	
54	55	Gran Turismo 5	PS3	2010	Racing	Sony Computer Entertainment	2.96	4.88	0.81	
62	63	Halo: Reach	X360	2010	Shooter	Microsoft Game Studios	7.03	1.98	0.08	
68	69	Just Dance 2	Wii	2010	Misc	Ubisoft	5.84	2.89	0.01	
78	79	Wii Party	Wii	2010	Misc	Nintendo	1.79	3.53	2.49	
91	92	Super Mario Galaxy 2	Wii	2010	Platform	Nintendo	3.66	2.42	0.98	

Extracting Data for The Publisher- Nintendo

```
In [58]: d_nintendo=d[d['Publisher']=='Nintendo']
d_nintendo
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28
4	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22
5	6	Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	4.22
6	7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	6.50
7	8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	2.93
8	9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	4.70
9	10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28
10	11	Nintendogs	DS	2005	Simulation	Nintendo	9.07	11.00	1.93
11	12	Mario Kart DS	DS	2005	Racing	Nintendo	9.81	7.57	4.13
12	13	Pokemon Gold/Pokemon Silver	GB	1999	Role-Playing	Nintendo	9.00	6.18	7.20
13	14	Wii Fit	Wii	2007	Sports	Nintendo	8.94	8.03	3.60
14	15	Wii Fit Plus	Wii	2009	Sports	Nintendo	9.09	8.59	2.53
18	19	Super Mario World	SNES	1990	Platform	Nintendo	12.78	3.75	3.54
19	20	Brain Age	DS	2005	Misc	Nintendo	4.75	9.26	4.16
20	21	Pokemon Diamond/Pokemon Pearl	DS	2006	Role-Playing	Nintendo	6.42	4.52	6.04
21	22	Super Mario Land	GB	1989	Platform	Nintendo	10.83	2.71	4.18
22	23	Super Mario Bros. 3	NES	1988	Platform	Nintendo	9.54	3.44	3.84
25	26	Pokemon Ruby/Pokemon Sapphire	GBA	2002	Role-Playing	Nintendo	6.06	3.90	5.38
26	27	Pokemon Black/Pokemon White	DS	2010	Role-Playing	Nintendo	5.57	3.28	5.65
27	28	Brain Age 2: More Training in Minutes a Day	DS	2005	Puzzle	Nintendo	3.44	5.36	5.32
30	31	Pokémon Yellow: Special Pikachu Edition	GB	1998	Role-Playing	Nintendo	5.89	5.04	3.12
32	33	Pokemon X/Pokemon Y	3DS	2013	Role-Playing	Nintendo	5.17	4.05	4.34

Rank		Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales
39	40	Super Smash Bros. Brawl	Wii	2008	Fighting	Nintendo	6.75	2.61	2.66
41	42	Animal Crossing: Wild World	DS	2005	Simulation	Nintendo	2.55	3.52	5.33
42	43	Mario Kart 7	3DS	2011	Racing	Nintendo	4.74	3.91	2.67
45	46	Pokemon HeartGold/Pokemon SoulSilver	DS	2009	Action	Nintendo	4.40	2.77	3.96
46	47	Super Mario 64	N64	1996	Platform	Nintendo	6.91	2.85	1.91
48	49	Super Mario Galaxy	Wii	2007	Platform	Nintendo	6.16	3.40	1.20
49	50	Pokemon Omega Ruby/Pokemon Alpha Sapphire	3DS	2014	Role-Playing	Nintendo	4.23	3.37	3.08
50	51	Super Mario Land 2: 6 Golden Coins	GB	1992	Adventure	Nintendo	6.16	2.04	2.69
53	54	Super Mario 3D Land	3DS	2011	Platform	Nintendo	4.89	2.99	2.13
57	58	Super Mario All-Stars	SNES	1993	Platform	Nintendo	5.99	2.15	2.12
58	59	Pokemon FireRed/Pokemon LeafGreen	GBA	2004	Role-Playing	Nintendo	4.34	2.65	3.15
59	60	Super Mario 64	DS	2004	Platform	Nintendo	5.08	3.11	1.25
63	64	Mario Kart 64	N64	1996	Racing	Nintendo	5.55	1.94	2.23
64	65	New Super Mario Bros. 2	3DS	2012	Platform	Nintendo	3.66	3.07	2.47
71	72	Donkey Kong Country	SNES	1994	Platform	Nintendo	4.36	1.71	3.00
73	74	Animal Crossing: New Leaf	3DS	2012	Simulation	Nintendo	2.01	2.32	4.36
74	75	Mario Party DS	DS	2007	Misc	Nintendo	4.46	1.88	1.98
76	77	Super Mario Kart	SNES	1992	Racing	Nintendo	3.54	1.24	3.81
78	79	Wii Party	Wii	2010	Misc	Nintendo	1.79	3.53	2.49
80	81	Mario Party 8	Wii	2007	Misc	Nintendo	3.81	2.30	1.58
81	82	Pokemon Black 2/Pokemon White 2	DS	2012	Role-Playing	Nintendo	2.91	1.86	3.14
84	85	GoldenEye 007	N64	1997	Shooter	Nintendo	5.80	2.01	0.13
88	89	Pokémon Platinum Version	DS	2008	Role-Playing	Nintendo	2.82	1.78	2.69
91	92	Super Mario Galaxy 2	Wii	2010	Platform	Nintendo	3.66	2.42	0.98
94	95	The Legend of Zelda: Ocarina of Time	N64	1998	Action	Nintendo	4.10	1.89	1.45

Nintendo Global Sales Mean from 1984 to 2014

```
In [59]: nintendo_sales=d_nintendo.Global_Sales.sum()
nintendo_sales
```

Out[59]: 905.3700000000001

Nintendo's Sales in 2010

```
In [60]: d_nintendo2010=d_nintendo[d_nintendo['Year']==2010]
d_nintendo2010
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
	26	Pokemon Black/Pokemon White	DS	2010	Role-Playing	Nintendo	5.57	3.28	5.65	
	78	Wii Party	Wii	2010	Misc	Nintendo	1.79	3.53	2.49	
	91	Super Mario Galaxy 2	Wii	2010	Platform	Nintendo	3.66	2.42	0.98	

Nintendo's Global Sales in 2010

```
In [61]: nintendo_sales2010=d_nintendo2010.Global_Sales.sum()
nintendo_sales2010
```

Out[61]: 31.500000000000004

Nintendo Details for 2014

```
In [63]: d_nintendo2014=d_nintendo[d_nintendo['Year']==2014]
d_nintendo2014
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other
	49	Pokemon Omega Ruby/Pokemon Alpha Sapphire	3DS	2014	Role-Playing	Nintendo	4.23	3.37	3.08	

Nintendo's Global Sales in 2014

```
In [64]: nintendo_sales2014=d_nintendo2014.Global_Sales.sum()
nintendo_sales2014
```

```
Out[64]: 11.33
```

Nintendo's Global Sales in 2014 Compared to its 20 years Sales

```
In [65]: nintendo_percent2014=(nintendo_sales2014/nintendo_sales)*100
nintendo_percent2014
```

```
Out[65]: 1.2514220705346983
```

Extracting Data for The Publisher- Sony Computer Entertainment

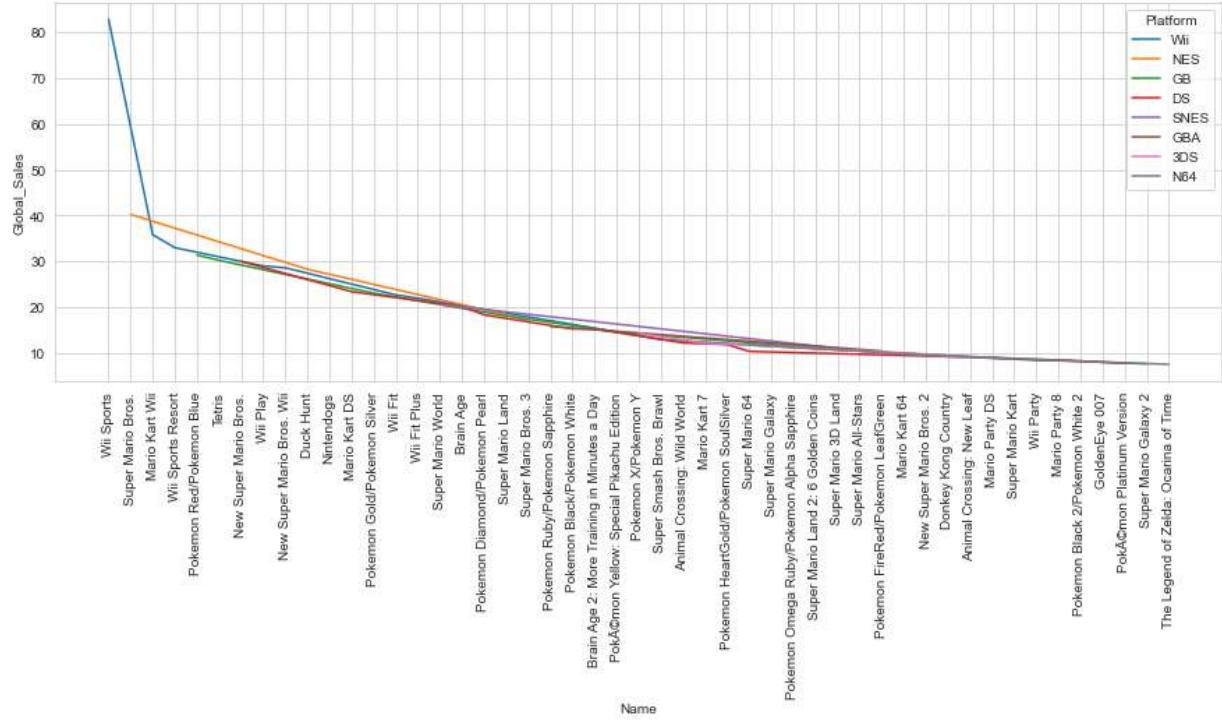
```
In [66]: d_sony=d[d['Publisher']=='Sony Computer Entertainment']
d_sony
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_
28	29	Gran Turismo 3: A-Spec	PS2	2001	Racing	Sony Computer Entertainment	6.85	5.09	1.87	
47	48	Gran Turismo 4	PS2	2004	Racing	Sony Computer Entertainment	3.01	0.01	1.10	
52	53	Gran Turismo	PS	1997	Racing	Sony Computer Entertainment	4.02	3.87	2.54	
54	55	Gran Turismo 5	PS3	2010	Racing	Sony Computer Entertainment	2.96	4.88	0.81	
66	67	Final Fantasy VII	PS	1997	Role-Playing	Sony Computer Entertainment	3.01	2.47	3.28	
69	70	Gran Turismo 2	PS	1999	Racing	Sony Computer Entertainment	3.88	3.42	1.69	
86	87	Final Fantasy X	PS2	2001	Role-Playing	Sony Computer Entertainment	2.91	2.07	2.73	
95	96	Crash Bandicoot 2: Cortex Strikes Back	PS	1997	Platform	Sony Computer Entertainment	3.78	2.17	1.31	

Data Visualisation

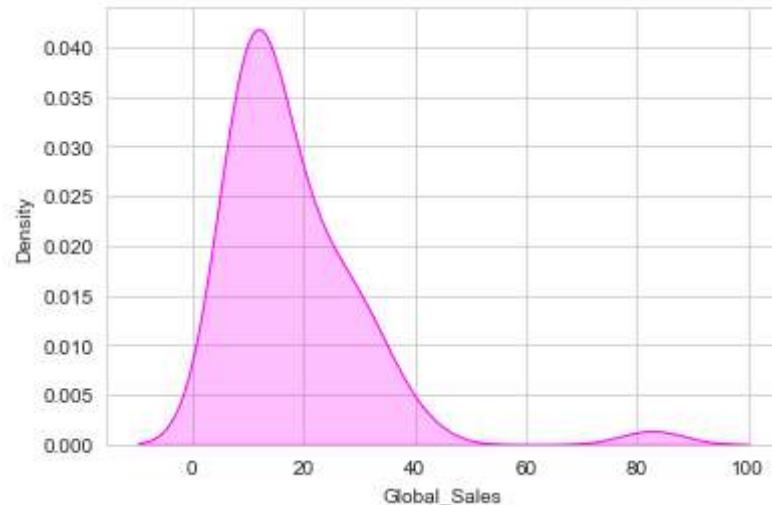
Line Plot for Game and Global Sales with Platform as Hue for Nintendo

```
In [68]: plt.figure(figsize=(15,5))
sns.lineplot(x='Name',y='Global_Sales',hue='Platform',data=d_nintendo)
plt.xticks(rotation=90)
plt.show()
```



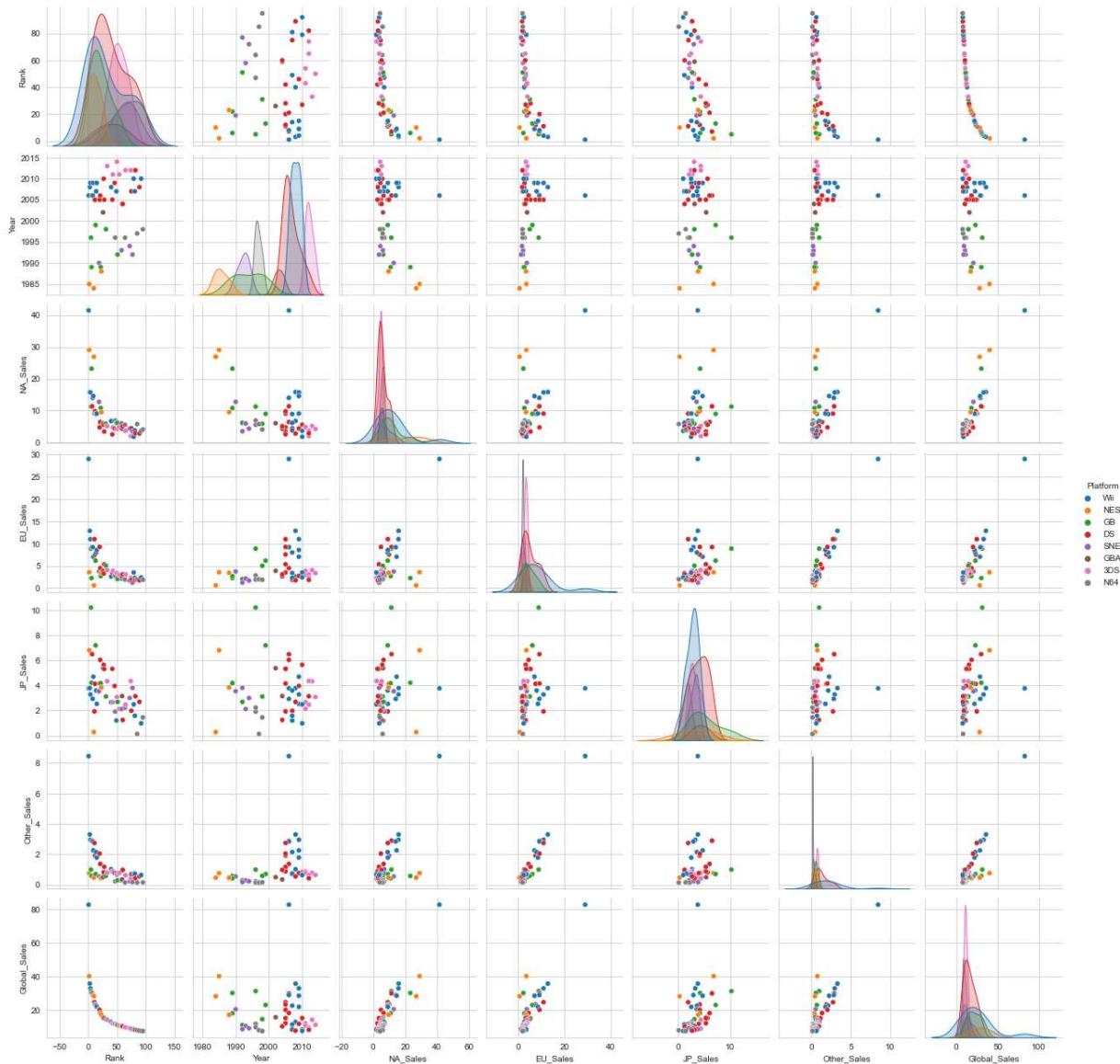
KDE Plot for Nintendo's Global Sale

```
In [69]: sns.kdeplot(d_nintendo['Global_Sales'],color='magenta',shade=True)  
plt.show()
```



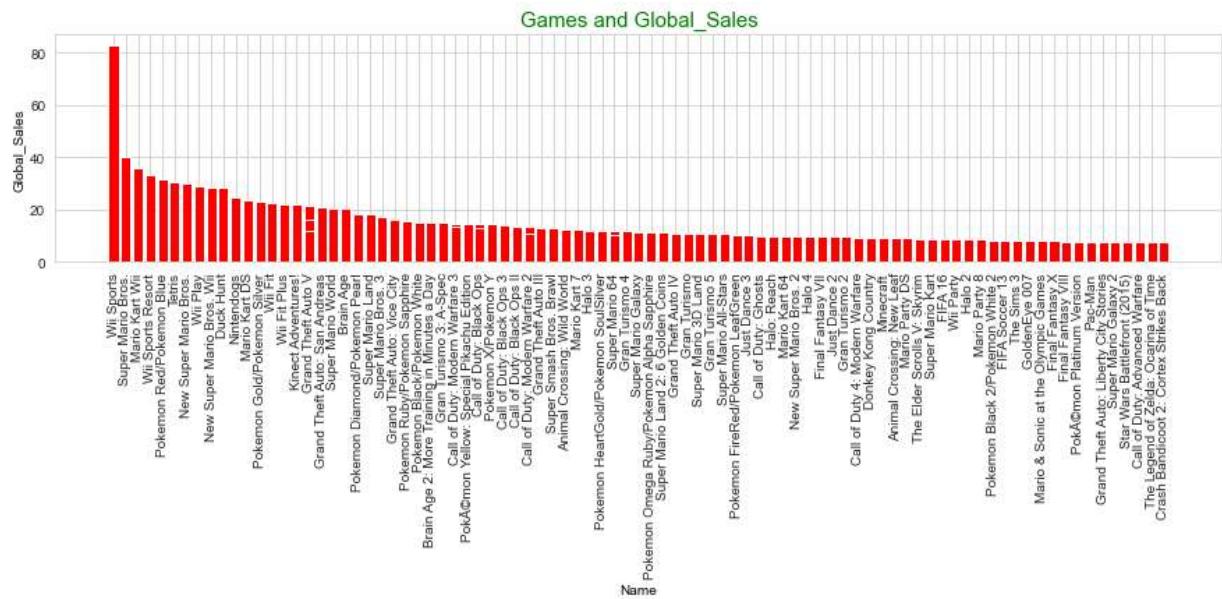
Pair Plot for Nintendo with Hue as Platform

```
In [70]: sns.pairplot(d_nintendo,diag_kind='kde',hue='Platform')
plt.show()
```



Games and Global_Sales

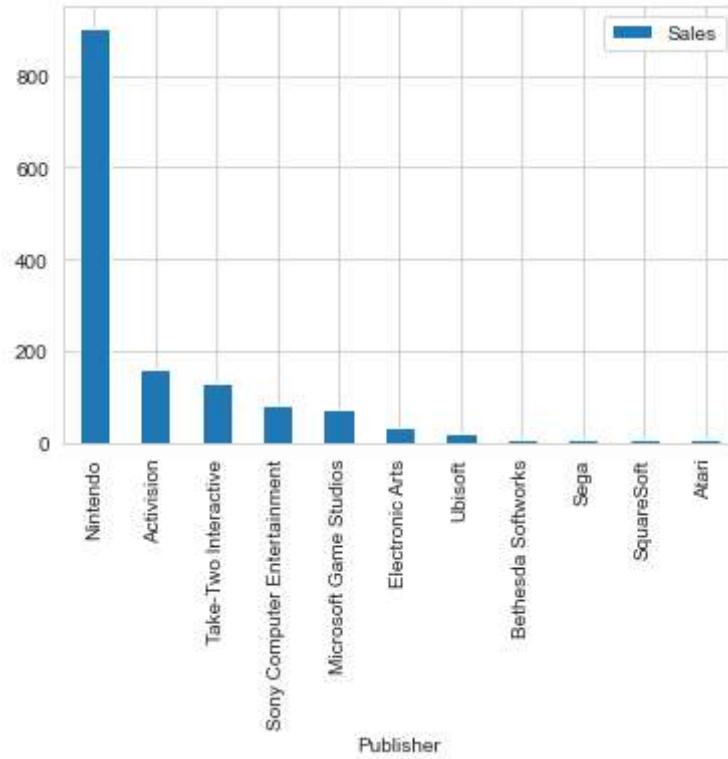
```
In [45]: plt.figure(figsize=(15,3))
plt.bar(a['Name'],a['Global_Sales'],color='r')
plt.title('Games and Global_Sales',fontsize=15,color='green')
plt.xlabel("Name",fontsize=10,color='black')
plt.xticks(rotation=90)
plt.ylabel("Global_Sales",fontsize=10,color='black')
plt.grid(True)
plt.show()
```



Plot for Publisher and Global Sales

```
In [37]: tp=a.groupby("Publisher")["Global_Sales"].sum().to_frame(name='Sales')  
tp.sort_values(by='Sales',ascending=False)[0:20].plot.bar()
```

```
Out[37]: <AxesSubplot:xlabel='Publisher'>
```



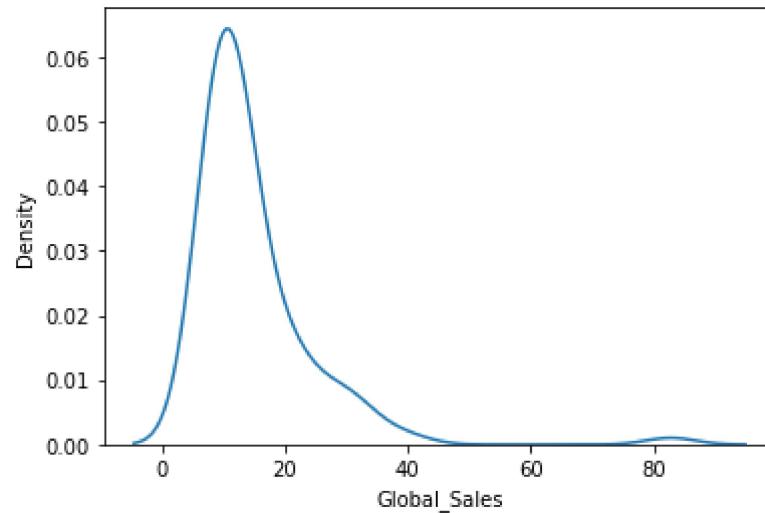
RELPOLT Between North American Sales and Global Sales for Each Games

```
In [46]: plt.figure(figsize=(16,18))
sns.relplot(x="NA_Sales",y="Global_Sales",hue='Name',data=a)
plt.show()
```

<Figure size 1152x1296 with 0 Axes>

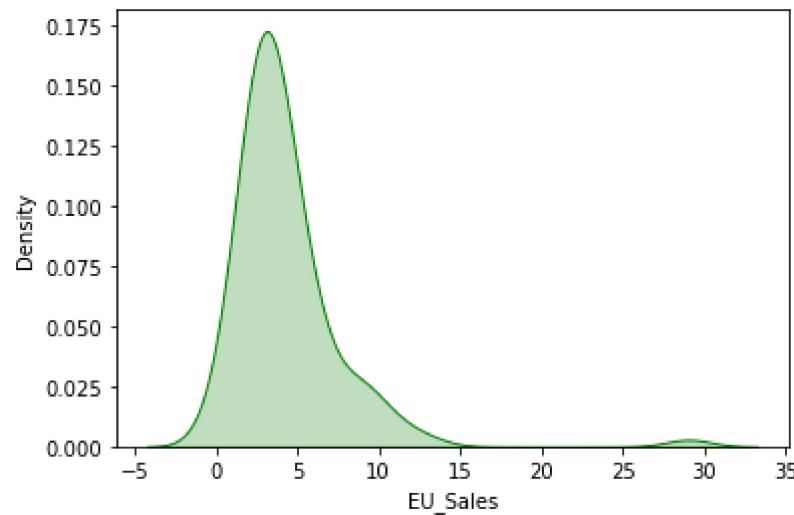
KDEPLOT for Global Sales

```
In [43]: sns.kdeplot(a['Global_Sales'])
plt.show()
```



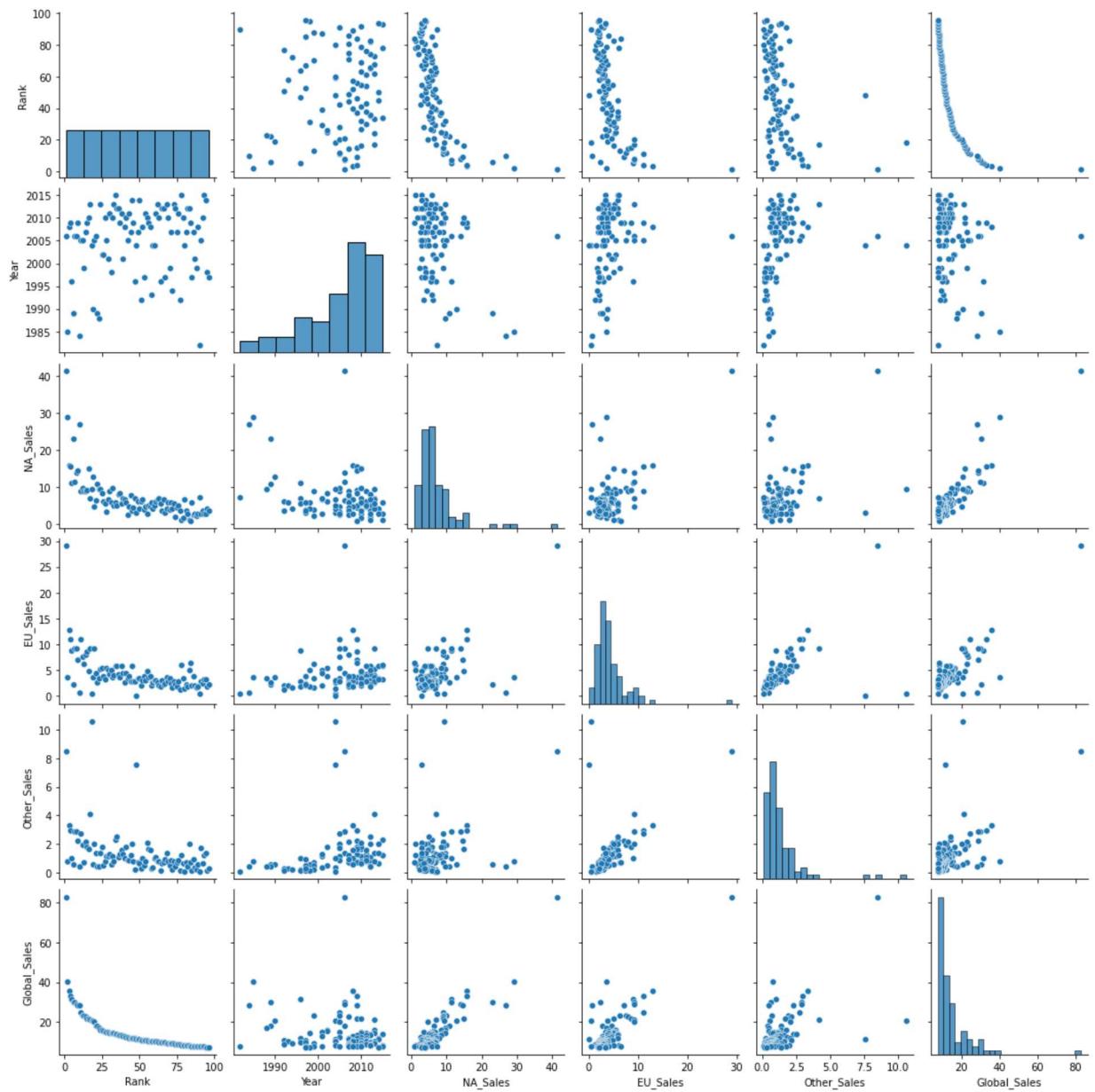
KDE Plot for Europe Sales

```
In [44]: sns.kdeplot(a['EU_Sales'],color='green',shade=True)  
plt.show()
```

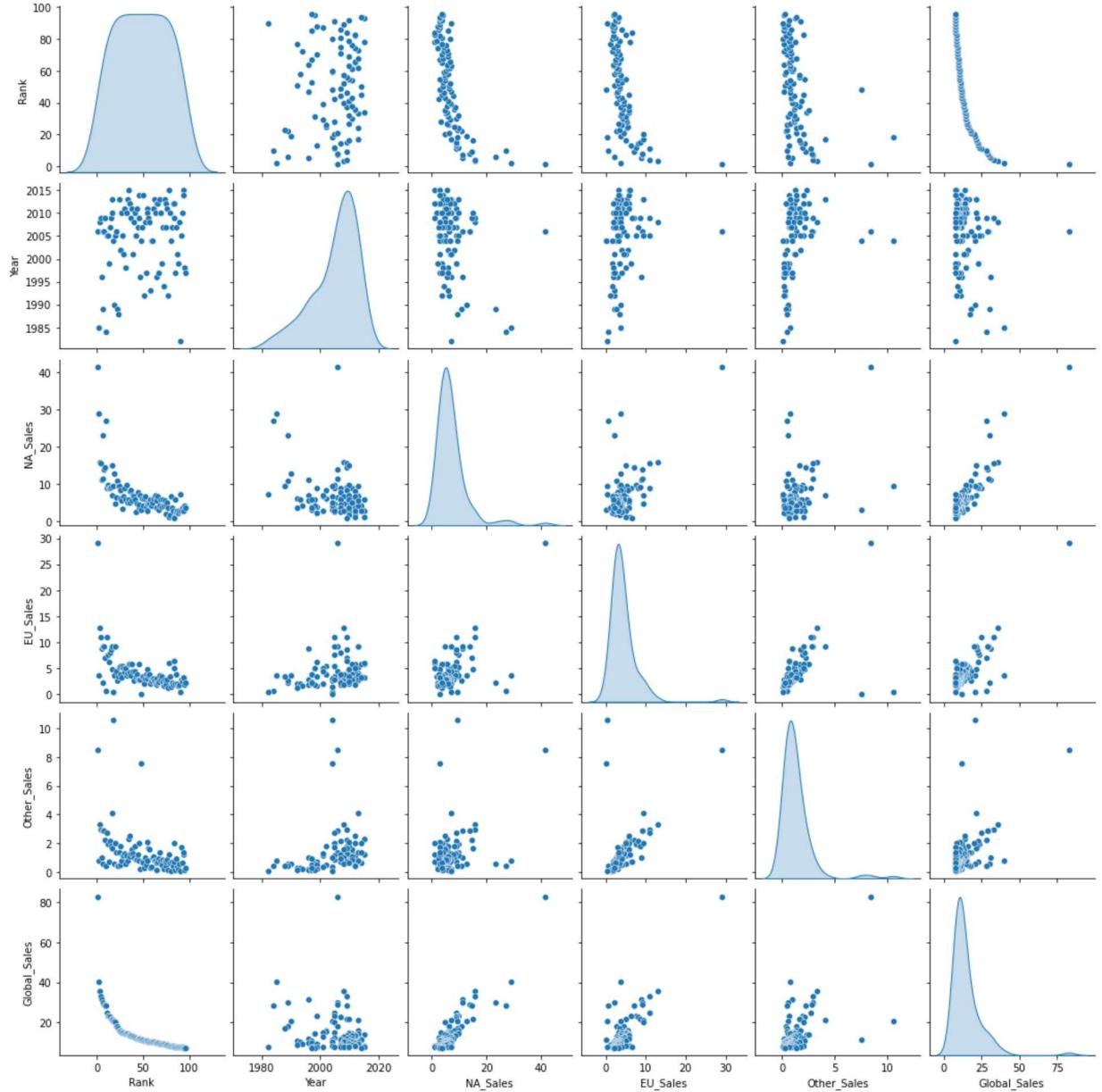


PAIRPLOT

```
In [45]: sns.pairplot(a)  
plt.show()
```

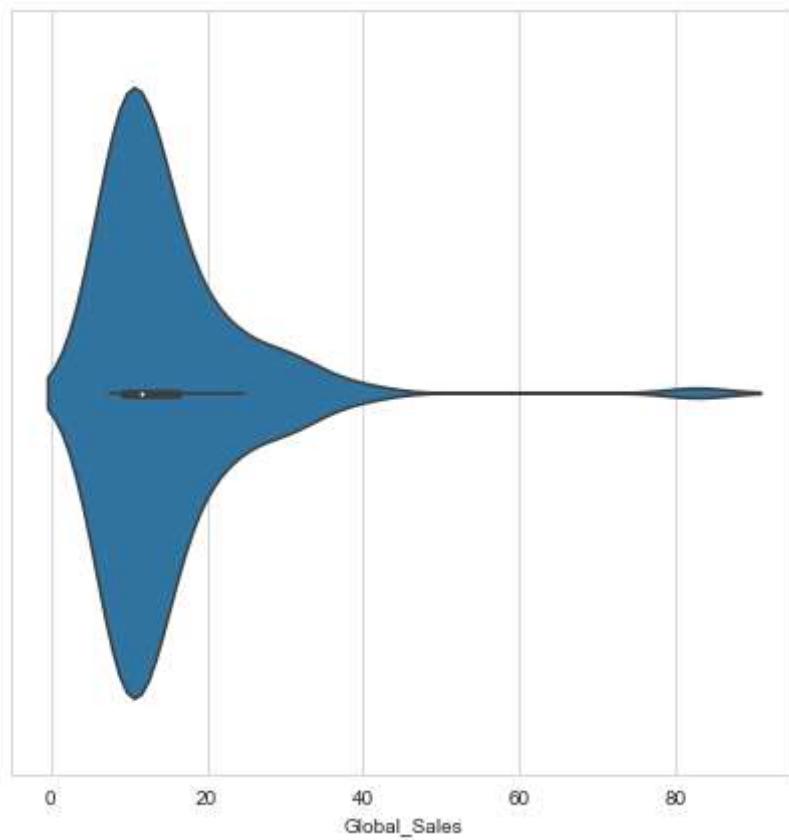


```
In [46]: sns.pairplot(a,diag_kind='kde')
plt.show()
```



VIOLIN PLOT of Global Sales

```
In [48]: plt.figure(figsize=(7,7))
sns.violinplot(a.Global_Sales)
plt.show()
```



Plot for 10 Games Sorted by Name and Global Sales

In [55]:

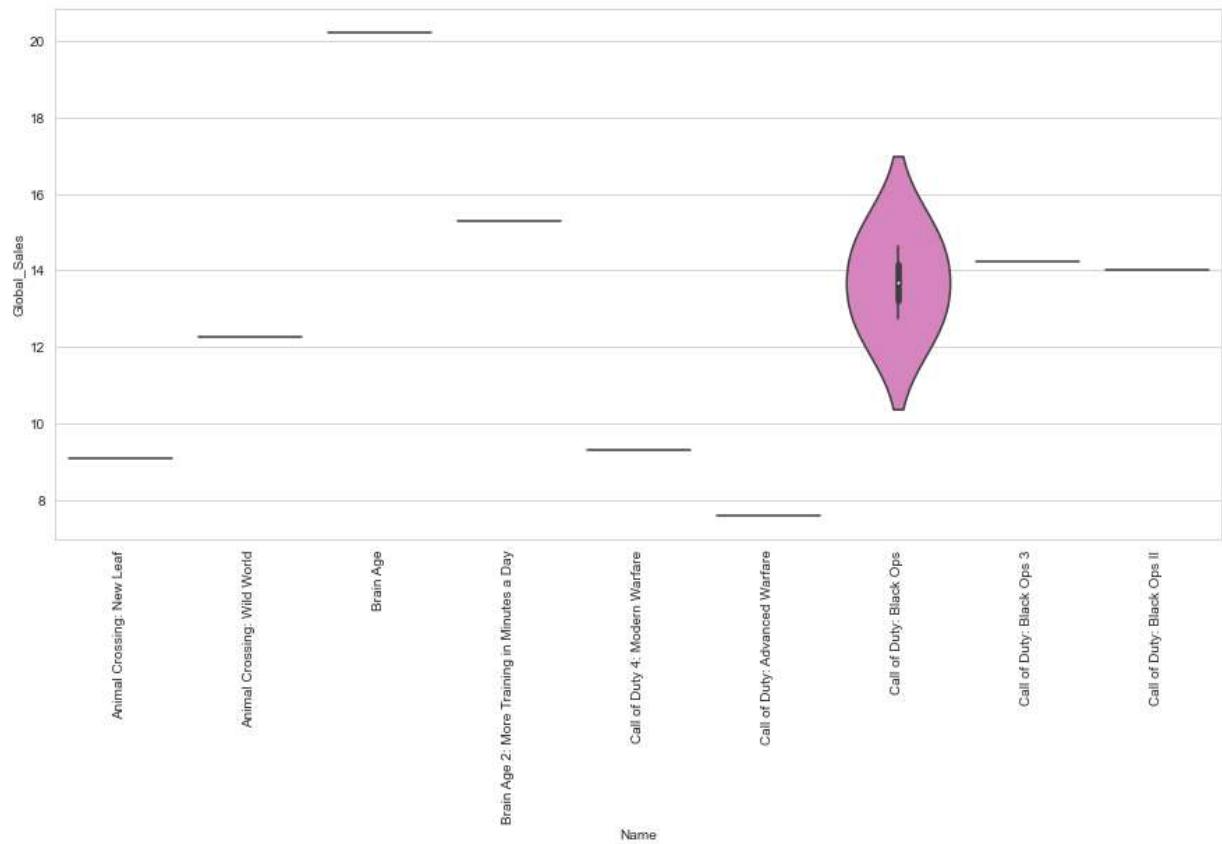
```
a2=a.sort_values('Name').head(10)
a2
```

Out[55]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
73	74	Animal Crossing: New Leaf	3DS	2012	Simulation	Nintendo	2.01	2.32	4.36	
41	42	Animal Crossing: Wild World	DS	2005	Simulation	Nintendo	2.55	3.52	5.33	
19	20	Brain Age	DS	2005	Misc	Nintendo	4.75	9.26	4.16	
27	28	Brain Age 2: More Training in Minutes a Day	DS	2005	Puzzle	Nintendo	3.44	5.36	5.32	
70	71	Call of Duty 4: Modern Warfare	X360	2007	Shooter	Activision	5.91	2.38	0.13	
93	94	Call of Duty: Advanced Warfare	PS4	2014	Shooter	Activision	2.80	3.30	0.14	
31	32	Call of Duty: Black Ops	X360	2010	Shooter	Activision	9.67	3.73	0.11	
40	41	Call of Duty: Black Ops	PS3	2010	Shooter	Activision	5.98	4.44	0.48	
33	34	Call of Duty: Black Ops 3	PS4	2015	Shooter	Activision	5.77	5.81	0.35	
34	35	Call of Duty: Black Ops II	PS3	2012	Shooter	Activision	4.99	5.88	0.65	

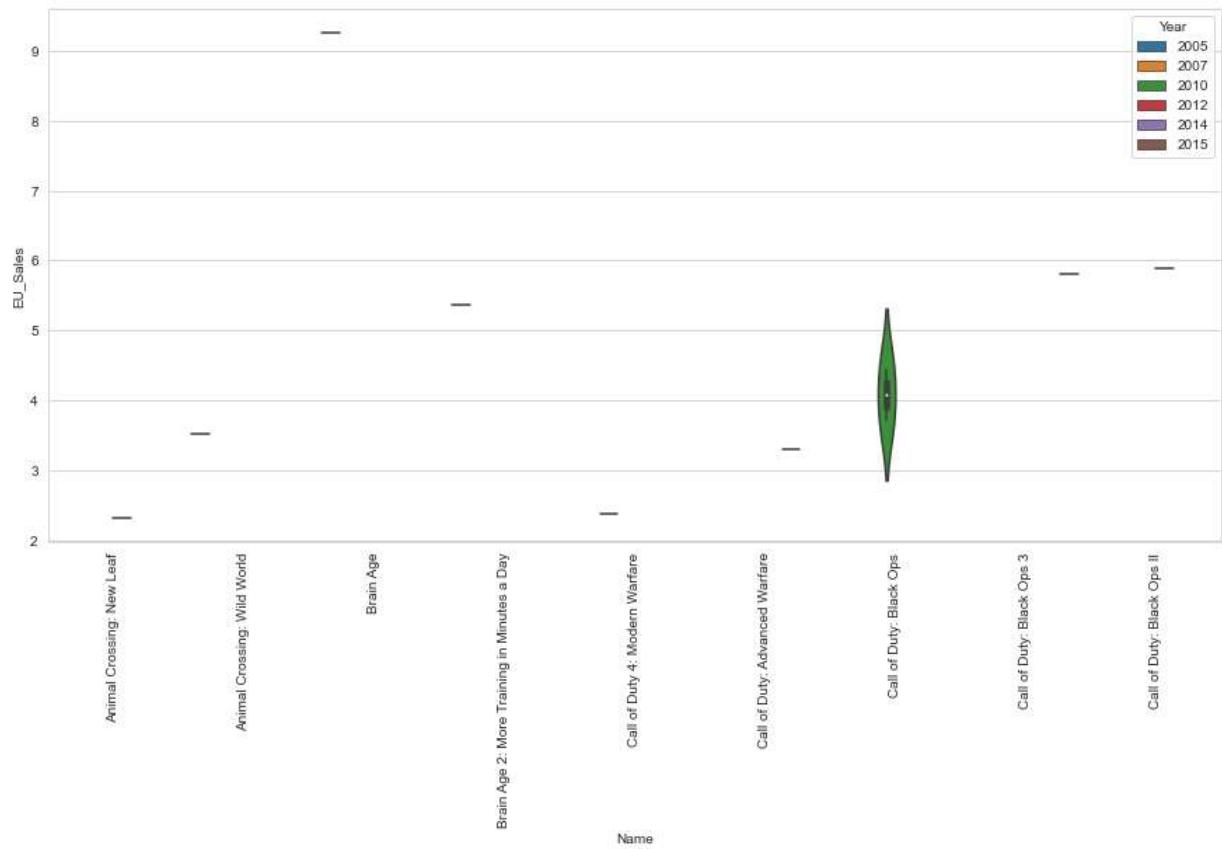


```
In [21]: plt.figure(figsize=(15,7))
sns.violinplot(x='Name',y='Global_Sales',data=a2)
plt.xticks(rotation=90)
plt.show()
```



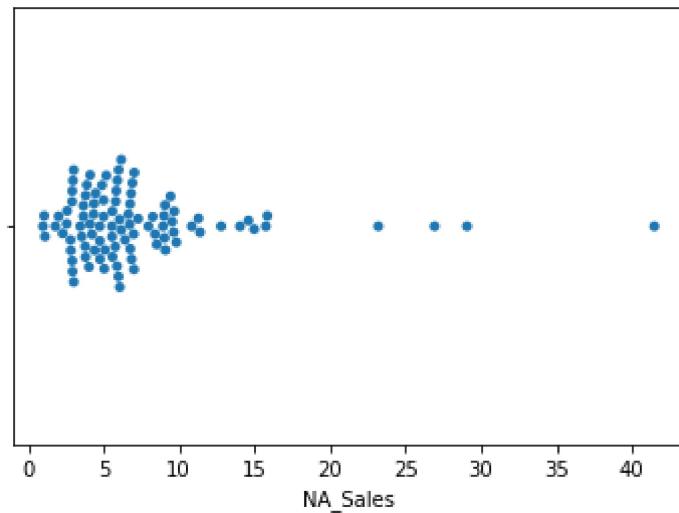
Plot for 10 Games Sorted by Name and Europe Sales

```
In [22]: plt.figure(figsize=(15,7))
sns.violinplot(x="Name",y='EU_Sales',hue='Year',data=a2)
plt.xticks(rotation=90)
plt.show()
```



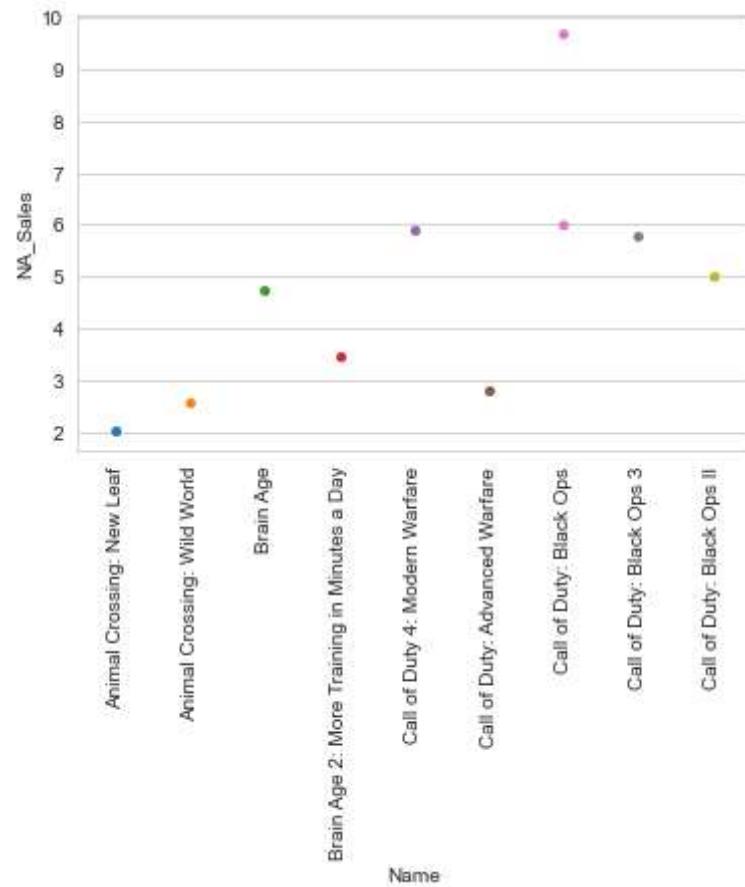
SWARM PLOT for North American Sales

```
In [10]: sns.swarmplot(a['NA_Sales'])
plt.show()
```



Swarm Plot for 10 Games Sorted by Name and North American Sales

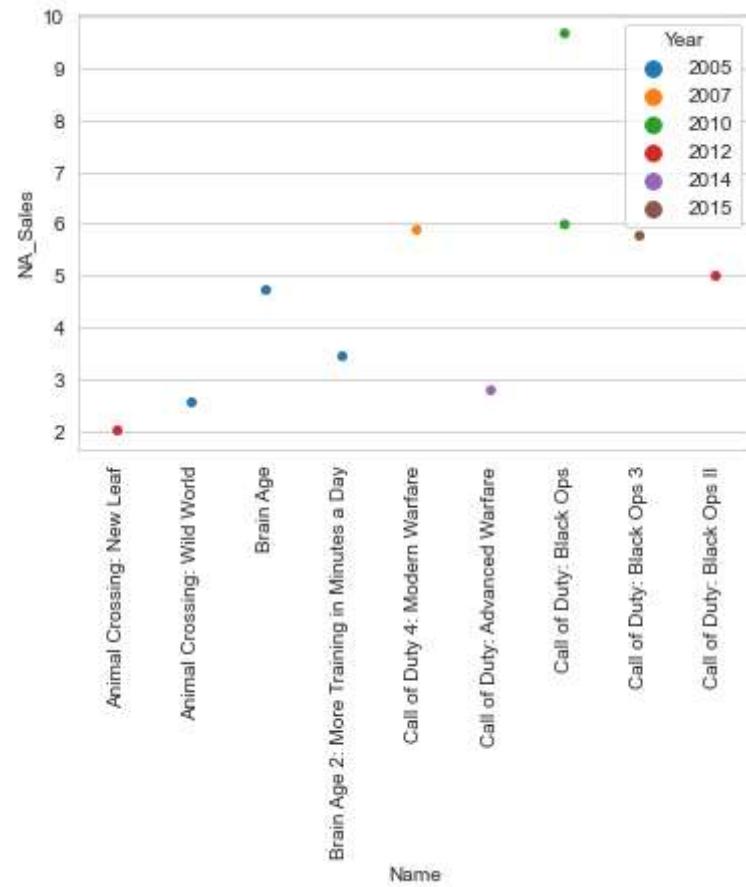
```
In [23]: sns.swarmplot(x='Name',y='NA_Sales',data=a2)
plt.xticks(rotation=90)
plt.show()
```



Swarm Plot for 10 Games Sorted by Name and North American Sales with

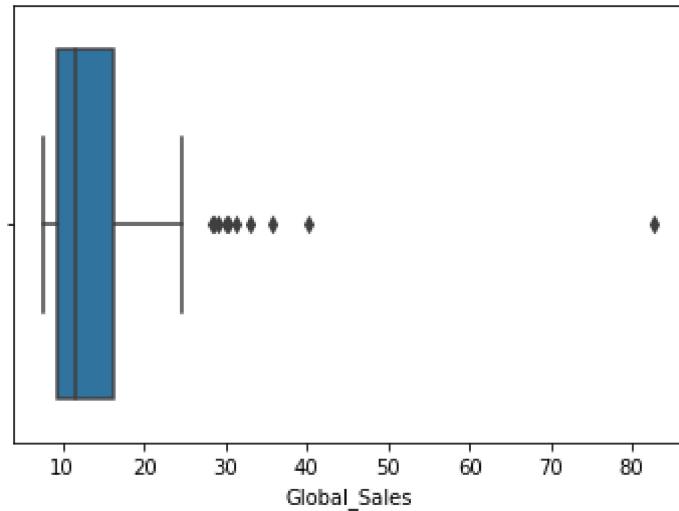
Year as Hue

```
In [24]: sns.swarmplot(x='Name',y='NA_Sales',hue='Year',data=a2)
plt.xticks(rotation=90)
plt.show()
```



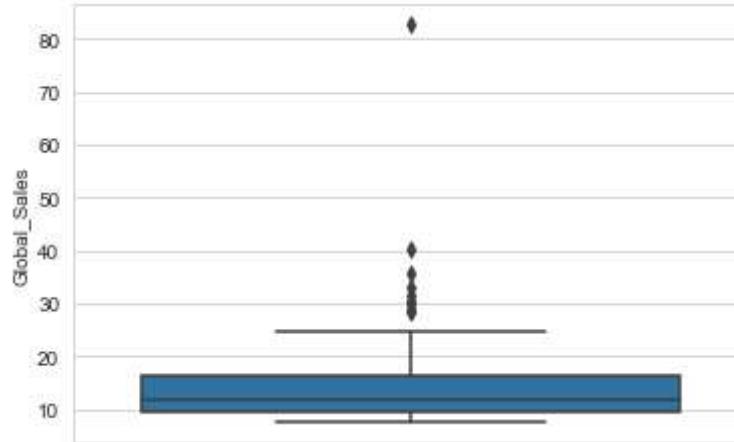
BOX PLOT for Global Sales

```
In [11]: sns.boxplot(a['Global_Sales'])
plt.show()
```



Inversion of the above box plot

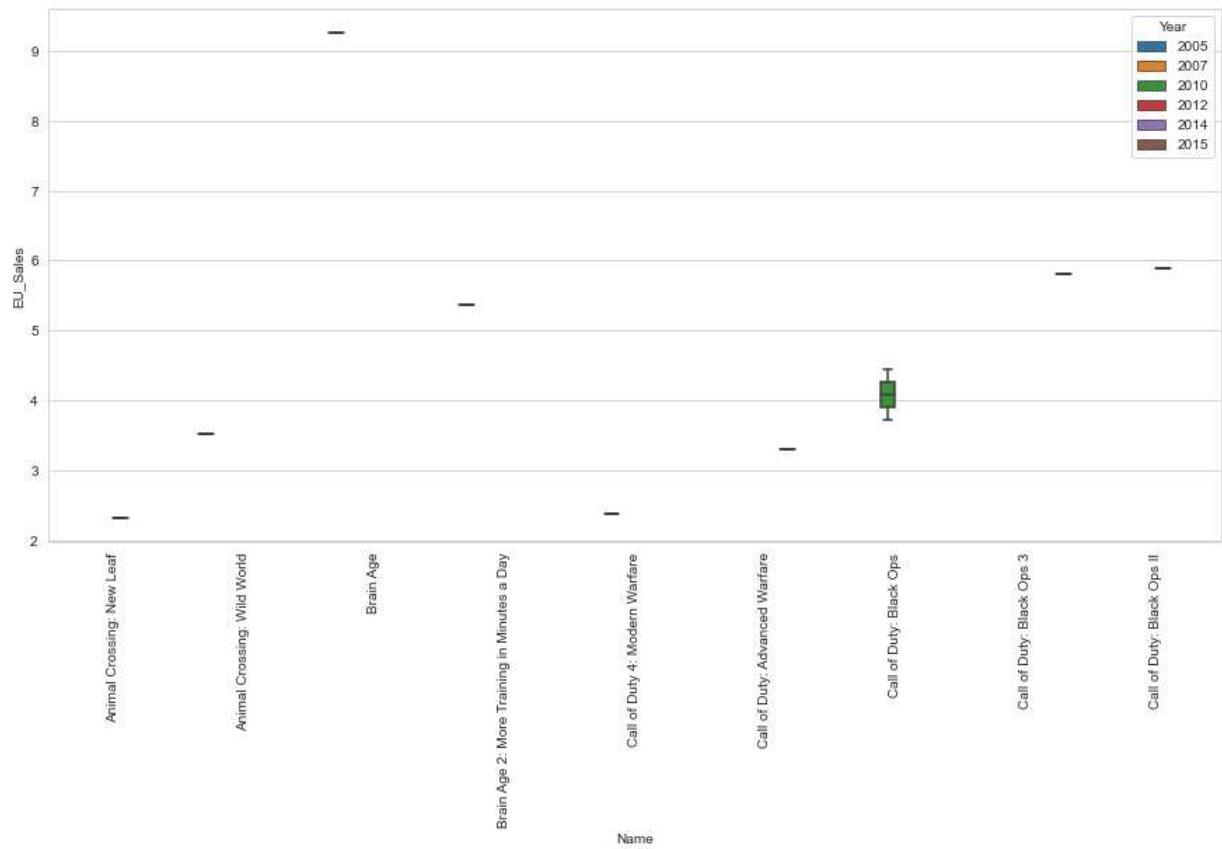
```
In [56]: sns.boxplot(y=a['Global_Sales'])
plt.show()
```



Box Plot Between Games and Europe Sales with

Year as Hue

```
In [26]: plt.figure(figsize=(15,7))
sns.boxplot(x='Name',y='EU_Sales',hue='Year',width=0.7,data=a2)
plt.xticks(rotation=90)
plt.show()
```

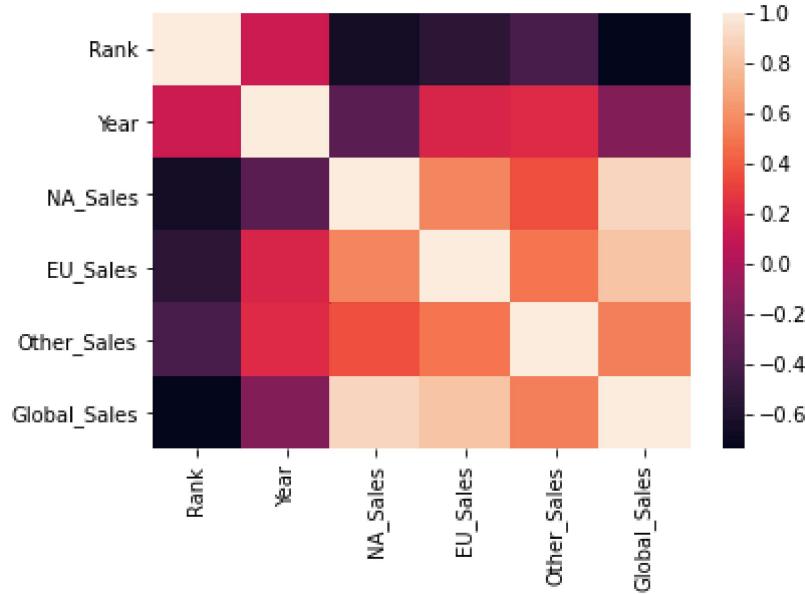


HEATMAP

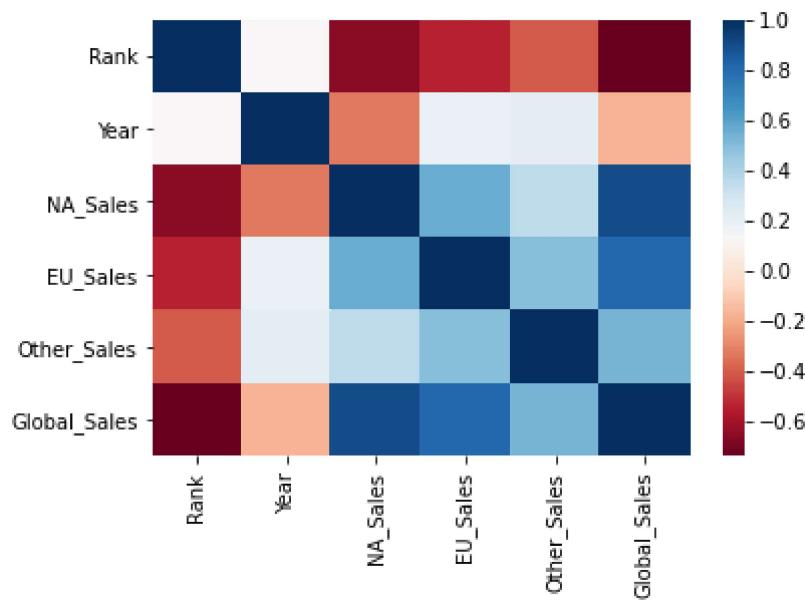
```
In [60]: a.corr()
```

	Rank	Year	NA_Sales	EU_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.127289	-0.657309	-0.537946	-0.399986	-0.739582
Year	0.127289	1.000000	-0.327936	0.192512	0.211917	-0.163147
NA_Sales	-0.657309	-0.327936	1.000000	0.564505	0.355117	0.901666
EU_Sales	-0.537946	0.192512	0.564505	1.000000	0.498915	0.819937
Other_Sales	-0.399986	0.211917	0.355117	0.498915	1.000000	0.534322
Global_Sales	-0.739582	-0.163147	0.901666	0.819937	0.534322	1.000000

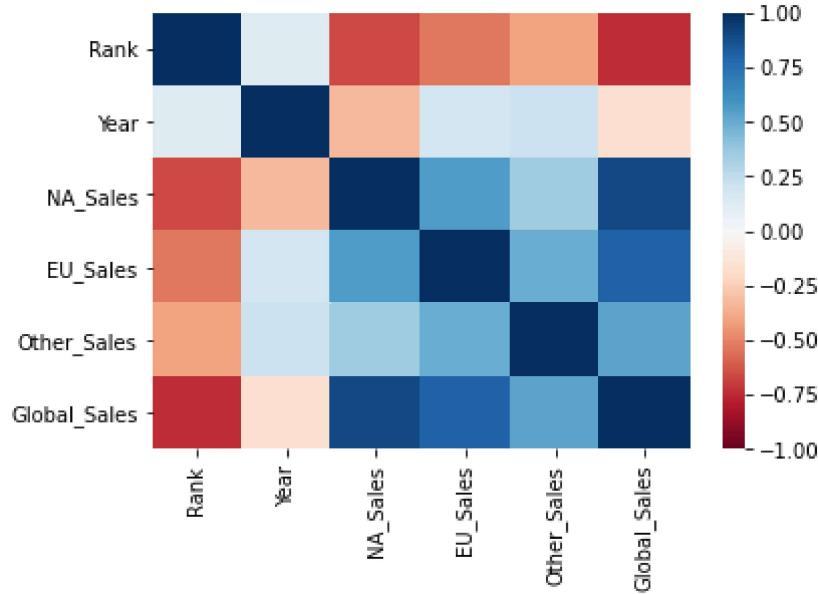
```
In [61]: sns.heatmap(a.corr())
plt.show()
```



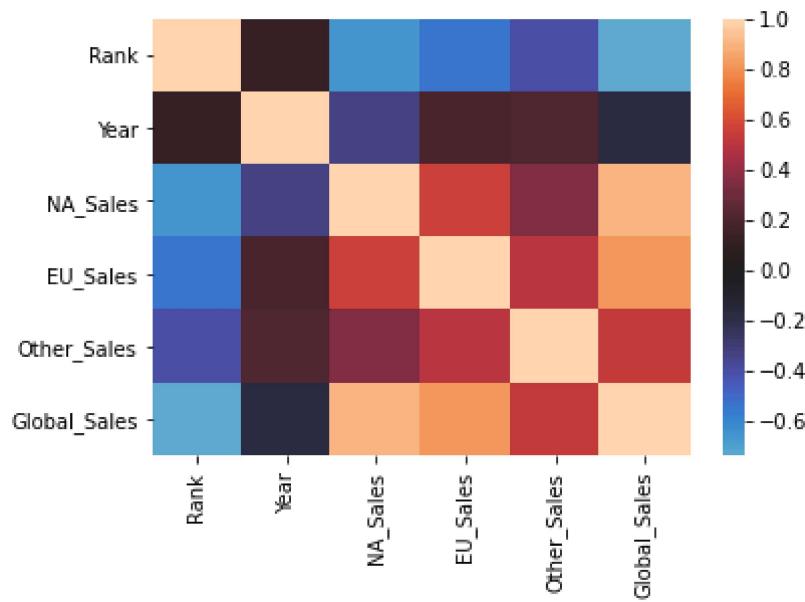
```
In [62]: sns.heatmap(a.corr(), cmap='RdBu')
plt.show()
```



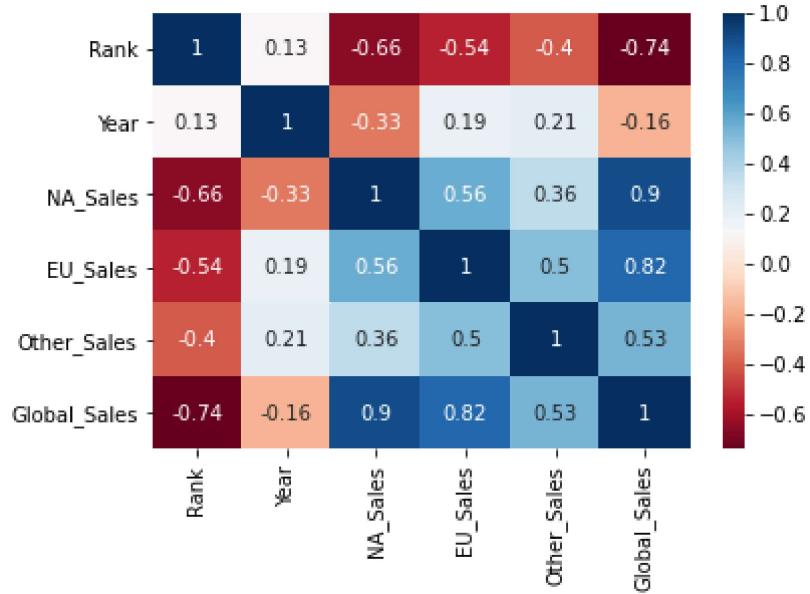
```
In [63]: sns.heatmap(a.corr(), cmap='RdBu', vmin=-1, vmax=1)  
plt.show()
```



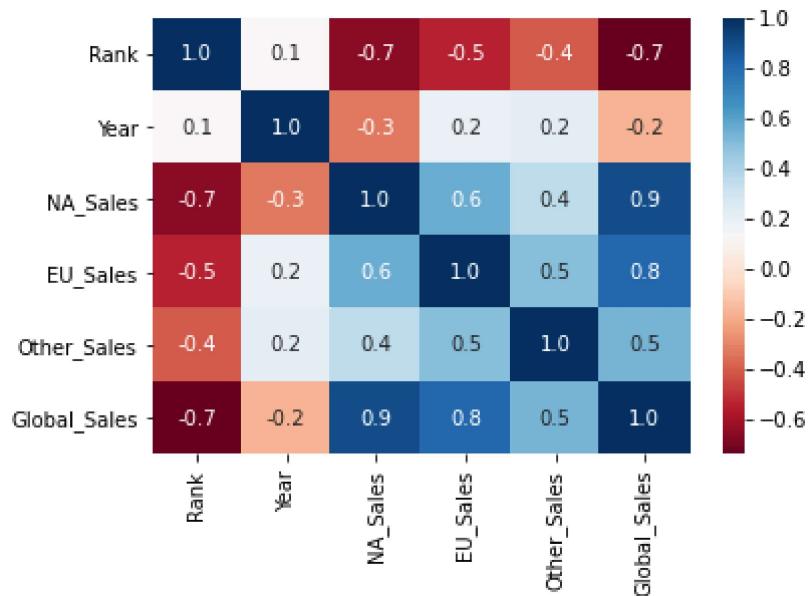
```
In [64]: sns.heatmap(a.corr(), center=0)  
plt.show()
```



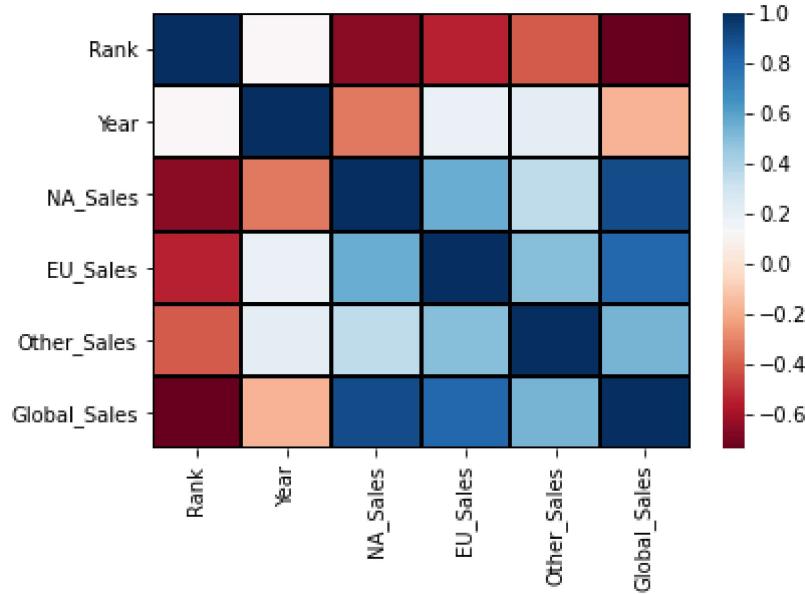
```
In [65]: sns.heatmap(a.corr(), cmap='RdBu', annot=True)
plt.show()
```



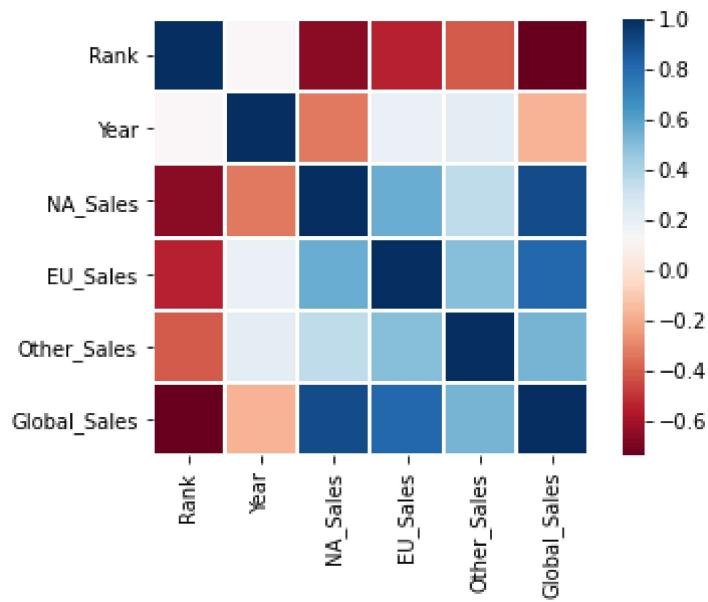
```
In [66]: sns.heatmap(a.corr(), cmap='RdBu', annot=True, fmt='.1f')
plt.show()
```



```
In [67]: sns.heatmap(a.corr(), cmap='RdBu', linewidth=1, linecolor='black')
plt.show()
```

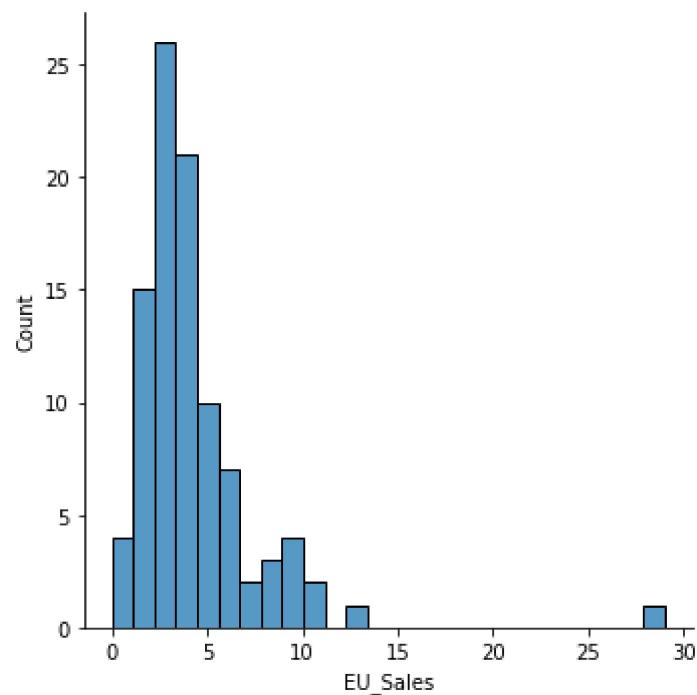


```
In [68]: sns.heatmap(a.corr(), cmap='RdBu', linewidth=1, square=True)
plt.show()
```



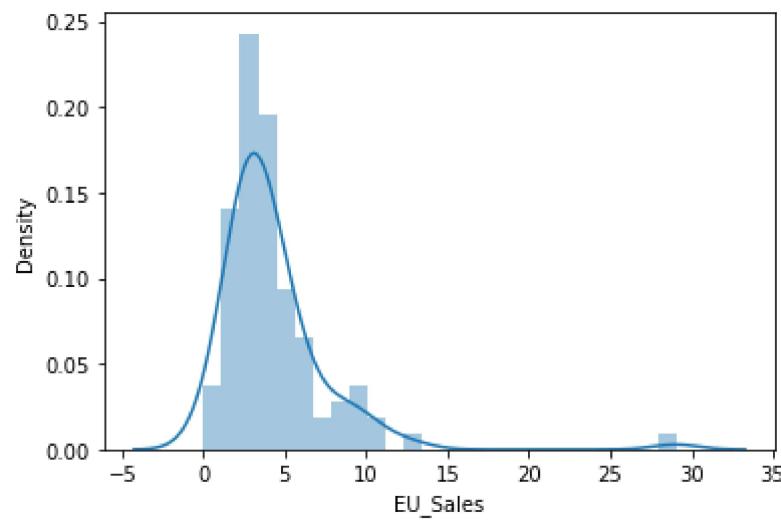
DISPLOT for Europe Sales

```
In [70]: sns.distplot(a.EU_Sales)  
plt.show()
```

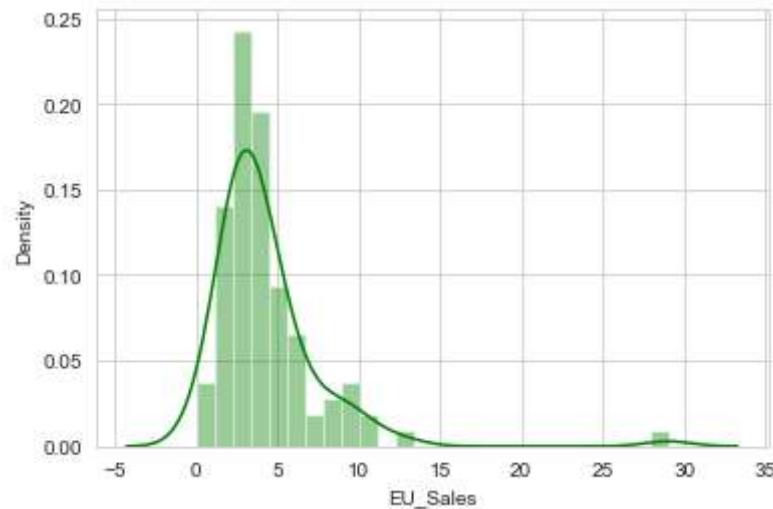


DISTPLOT for Europe Sales

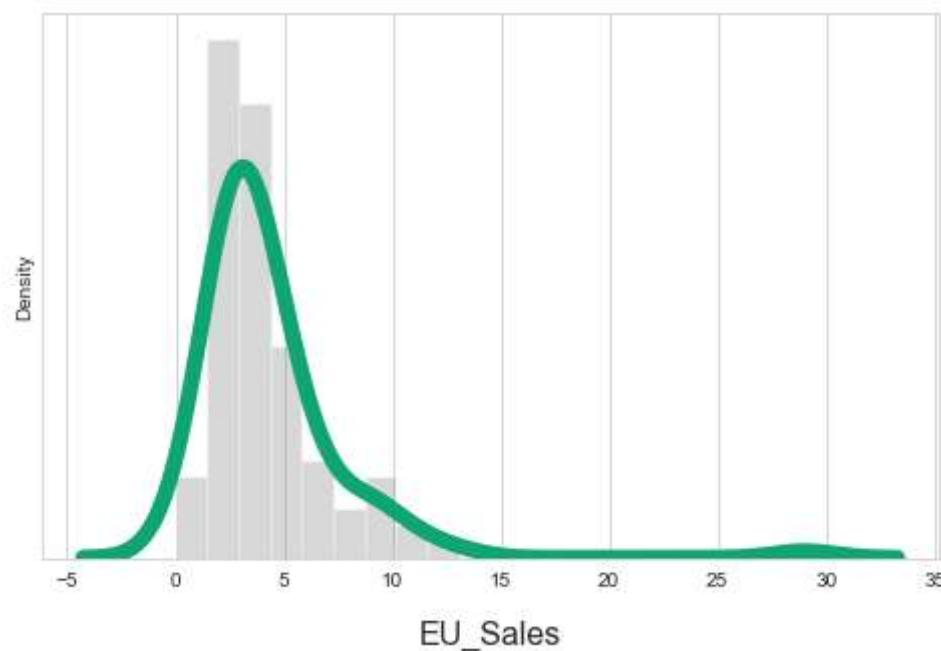
```
In [12]: sns.distplot(a.EU_Sales)  
plt.show()
```



```
In [13]: sns.set_style("whitegrid")
sns.distplot(a.EU_Sales,color='green')
plt.show()
```

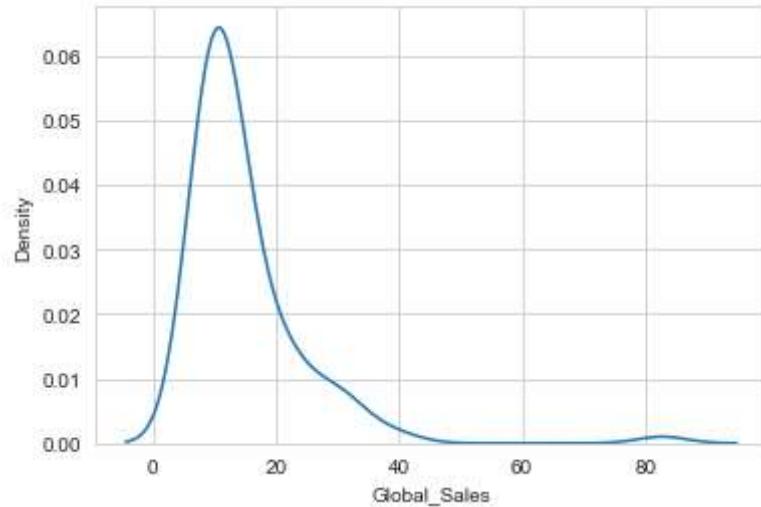


```
In [74]: plt.figure(figsize=(8,5))
sns.distplot(a.EU_Sales,
             bins=20,
             kde_kws={'lw':8,'color':'xkcd:bluish green'},
             hist_kws={'alpha':0.3,'color':'grey'})
plt.xlabel('EU_Sales', fontsize=16, labelpad=15)
plt.yticks([])
plt.show()
```

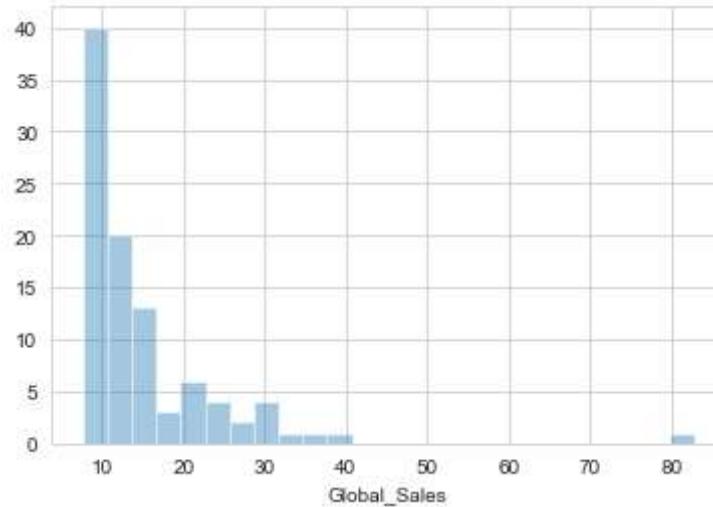


Distplot for Global Sales

```
In [14]: sns.distplot(a.Global_Sales, hist=False)  
plt.show()
```



```
In [15]: sns.distplot(a.Global_Sales, kde=False)  
plt.show()
```



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```