

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [32]: games_data=pd.read_csv('C:/Users/PREDATOR/Downloads/vgsales.csv')
```

```
In [4]: games_data.head()
```

Out[4]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_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	

```
In [5]: games_data.tail()
```

Out[5]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_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	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	

```
In [6]: games_data.shape
```

```
Out[6]: (16598, 11)
```

```
In [7]: games_data.describe()
```

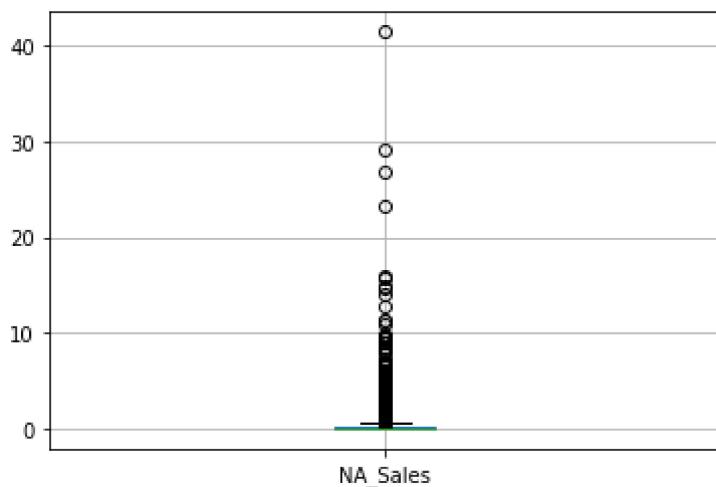
Out[7]:

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
--	-------------	-------------	-----------------	-----------------	-----------------	--------------------	---------------------

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_
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	0.51
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.51
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.01
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.06
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.11
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.41
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.74

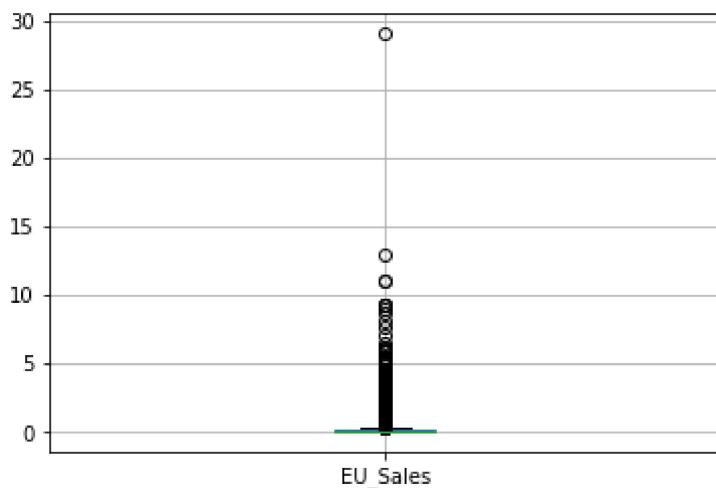
In [8]: `games_data.boxplot(column='NA_Sales')`

Out[8]: <AxesSubplot:>



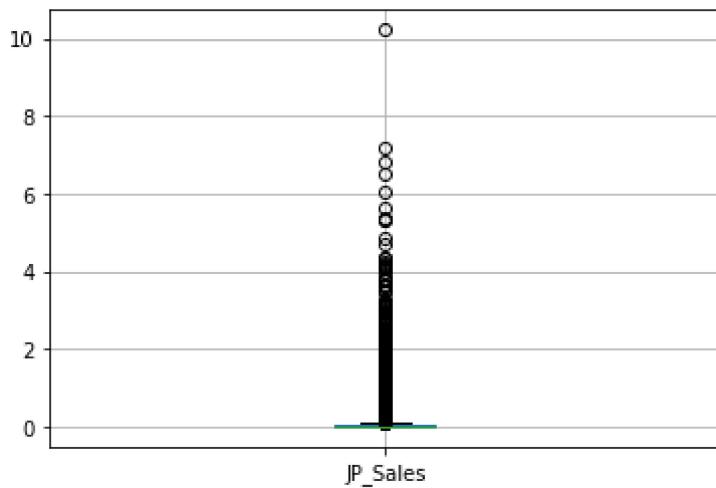
In [9]: `games_data.boxplot(column='EU_Sales')`

Out[9]: <AxesSubplot:>



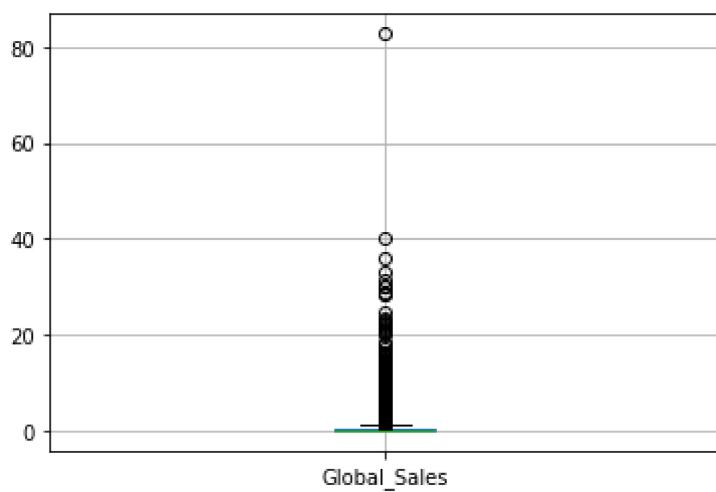
In [10]: `games_data.boxplot(column='JP_Sales')`

Out[10]: <AxesSubplot:>



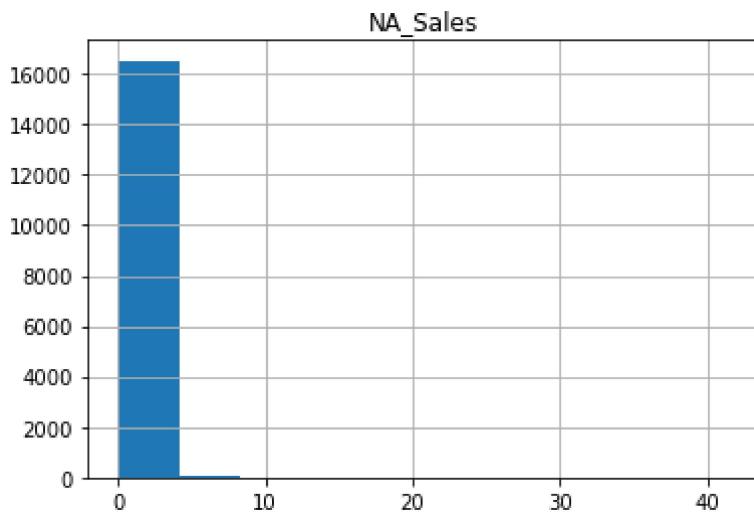
```
In [11]: games_data.boxplot(column='Global_Sales')
```

```
Out[11]: <AxesSubplot:>
```



```
In [12]: games_data.hist(column='NA_Sales')
```

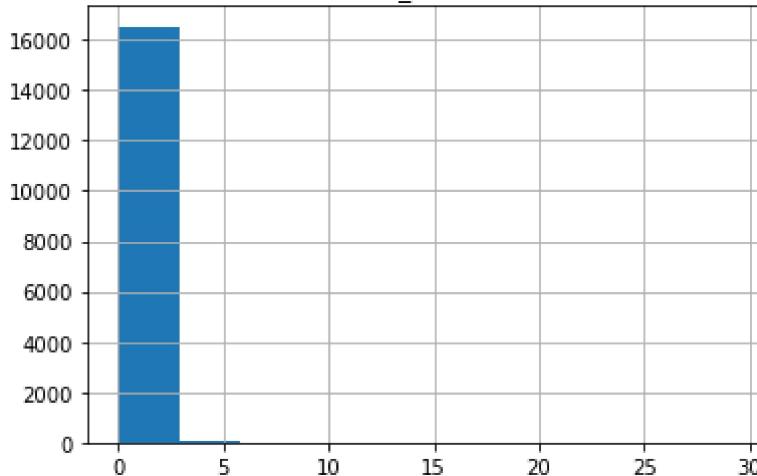
```
Out[12]: array([[<AxesSubplot:title={'center':'NA_Sales'}>]], dtype=object)
```



```
In [13]: games_data.hist(column='EU_Sales')
```

```
Out[13]: array([[<AxesSubplot:title={'center':'EU_Sales'}>]], dtype=object)
```

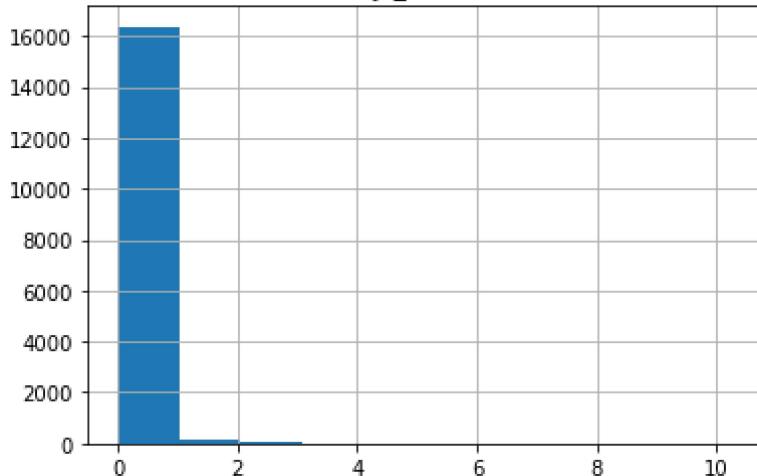
EU_Sales



```
In [14]: games_data.hist(column='JP_Sales')
```

```
Out[14]: array([[[<AxesSubplot:title={'center':'JP_Sales'}>]]], dtype=object)
```

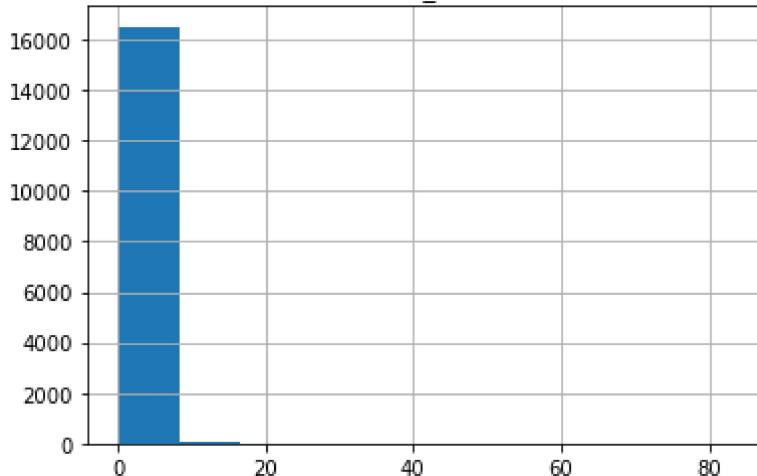
JP_Sales



```
In [15]: games_data.hist(column='Global_Sales')
```

```
Out[15]: array([[[<AxesSubplot:title={'center':'Global_Sales'}>]]], dtype=object)
```

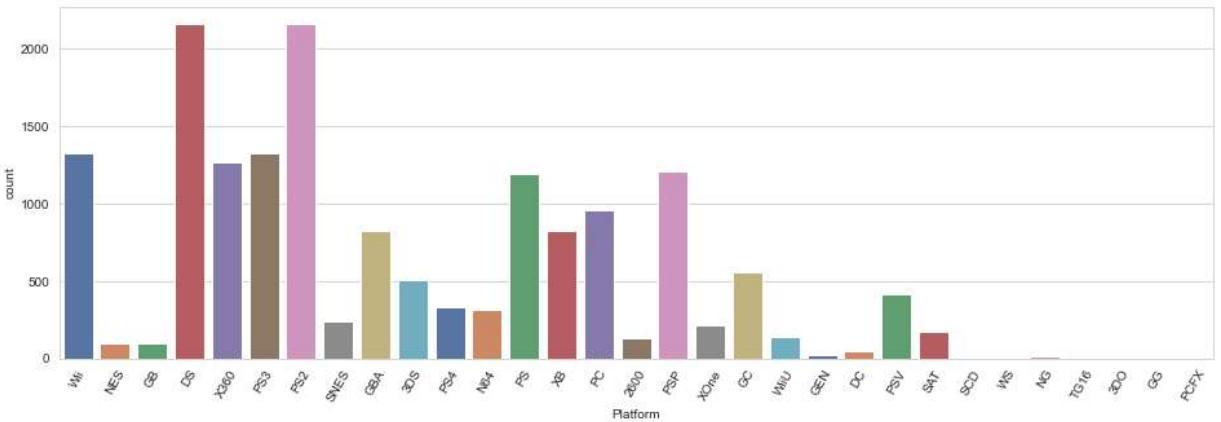
Global_Sales



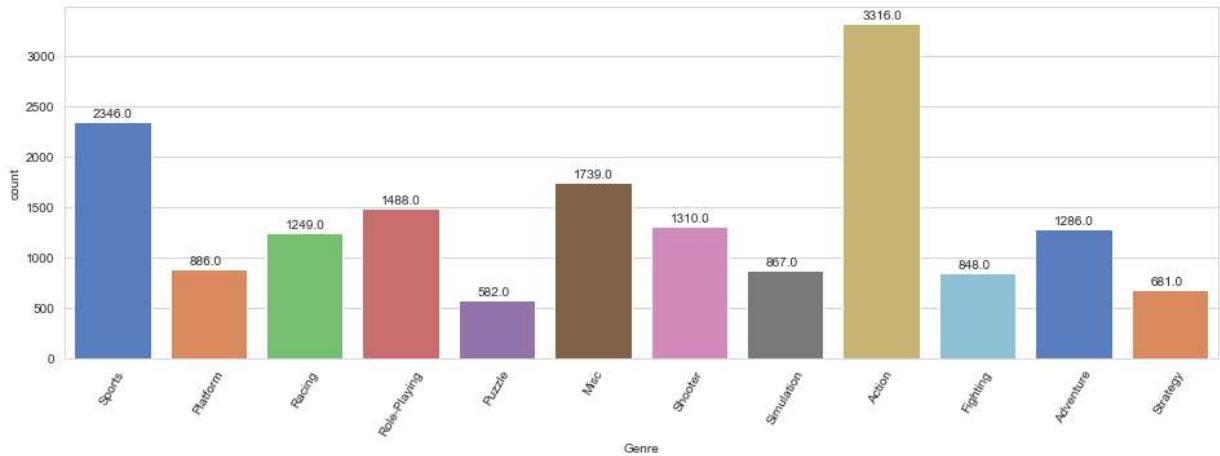
```
In [30]: plt.figure(figsize=(16,5))
sns.set_style('whitegrid')
sns.countplot(x='Platform', data=games_data, palette='deep')
plt.xticks(rotation=60)
```

```
Out[30]: (array([ 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, 'Wii'),
 Text(1, 0, 'NES'),
 Text(2, 0, 'GB'),
 Text(3, 0, 'DS'),
 Text(4, 0, 'X360'),
 Text(5, 0, 'PS3'),
 Text(6, 0, 'PS2'),
 Text(7, 0, 'SNES'),
 Text(8, 0, 'GBA'),
 Text(9, 0, '3DS'),
 Text(10, 0, 'PS4'),
 Text(11, 0, 'N64'),
 Text(12, 0, 'PS'),
 Text(13, 0, 'XB'),
 Text(14, 0, 'PC'),
 Text(15, 0, '2600'),
 Text(16, 0, 'PSP'),
 Text(17, 0, 'XOne'),
 Text(18, 0, 'GC'),
 Text(19, 0, 'WiiU'),
 Text(20, 0, 'GEN'),
 Text(21, 0, 'DC'),
 Text(22, 0, 'PSV'),
 Text(23, 0, 'SAT'),
 Text(24, 0, 'SCD'),
 Text(25, 0, 'WS'),
 Text(26, 0, 'NG'),
 Text(27, 0, 'TG16'),
 Text(28, 0, '3DO'),
 Text(29, 0, 'GG'),
 Text(30, 0, 'PCFX')])
```



```
In [60]: plt.figure(figsize=(16,5))
sns.set_style('whitegrid')
genre=sns.countplot(x='Genre',data=games_data,palette='muted')
plt.xticks(rotation=60)
for p in genre.patches:
    genre.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.2, p.get_height()+5))
```



In [61]: `games_data.isnull()`

Out[61]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_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
...
16593	False	False	False	False	False	False	False	False	False	False
16594	False	False	False	False	False	False	False	False	False	False
16595	False	False	False	False	False	False	False	False	False	False
16596	False	False	False	False	False	False	False	False	False	False
16597	False	False	False	False	False	False	False	False	False	False

16598 rows × 11 columns



In [62]: `games_data.isnull().sum()`

Out[62]:

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 [67]: `games_data.dropna(inplace=True)`

In [68]: `games_data.isnull().sum()`

Out[68]:

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
```

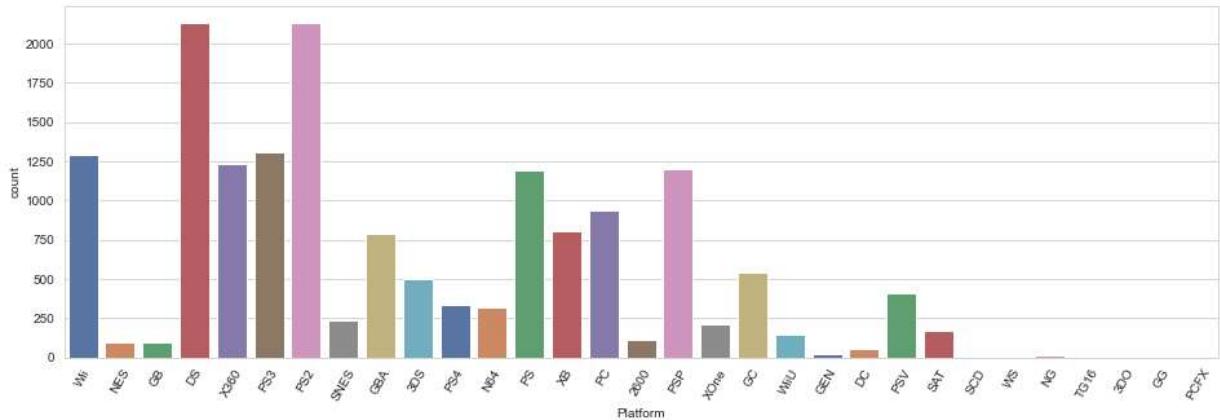
```
In [69]: games_data.shape
```

```
Out[69]: (16291, 11)
```

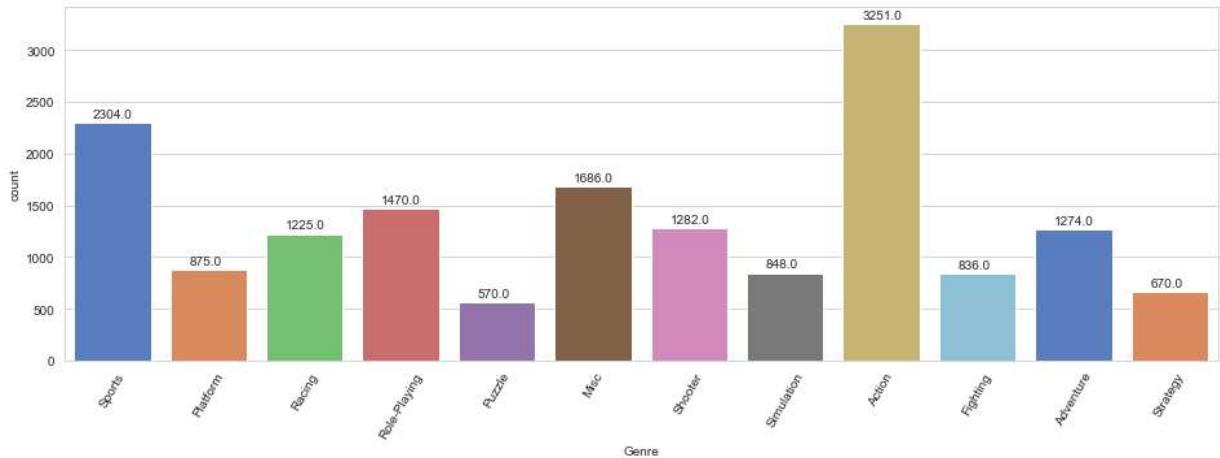
```
In [ ]: #AFTER DATA CLEANING
```

```
In [70]: plt.figure(figsize=(16,5))
sns.set_style('whitegrid')
sns.countplot(x='Platform', data=games_data, palette='deep')
plt.xticks(rotation=60)
```

```
Out[70]: (array([ 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, 'Wii'),  
 Text(1, 0, 'NES'),  
 Text(2, 0, 'GB'),  
 Text(3, 0, 'DS'),  
 Text(4, 0, 'X360'),  
 Text(5, 0, 'PS3'),  
 Text(6, 0, 'PS2'),  
 Text(7, 0, 'SNES'),  
 Text(8, 0, 'GBA'),  
 Text(9, 0, '3DS'),  
 Text(10, 0, 'PS4'),  
 Text(11, 0, 'N64'),  
 Text(12, 0, 'PS'),  
 Text(13, 0, 'XB'),  
 Text(14, 0, 'PC'),  
 Text(15, 0, '2600'),  
 Text(16, 0, 'PSP'),  
 Text(17, 0, 'XOne'),  
 Text(18, 0, 'GC'),  
 Text(19, 0, 'WiiU'),  
 Text(20, 0, 'GEN'),  
 Text(21, 0, 'DC'),  
 Text(22, 0, 'PSV'),  
 Text(23, 0, 'SAT'),  
 Text(24, 0, 'SCD'),  
 Text(25, 0, 'WS'),  
 Text(26, 0, 'NG'),  
 Text(27, 0, 'TG16'),  
 Text(28, 0, '3DO'),  
 Text(29, 0, 'GG'),  
 Text(30, 0, 'PCFX')])
```



```
In [71]: plt.figure(figsize=(16,5))
sns.set_style('whitegrid')
genre=sns.countplot(x='Genre',data=games_data,palette='muted')
plt.xticks(rotation=60)
for p in genre.patches:
    genre.annotate('{:.1f}'.format(p.get_height()), (p.get_x()+0.2, p.get_height()+5))
```



```
In [72]: games_data.head()
```

```
Out[72]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_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	



```
In [73]: games_data['Year']=games_data['Year'].apply(lambda x: int(x))
```

```
In [74]: games_data.head()
```

```
Out[74]:
```

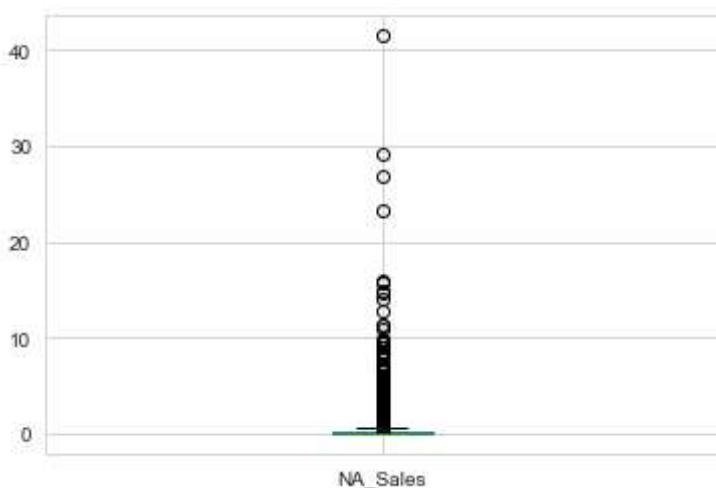
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
0	1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8

Rank		Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
1	2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0
2	3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3
3	4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2
4	5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1



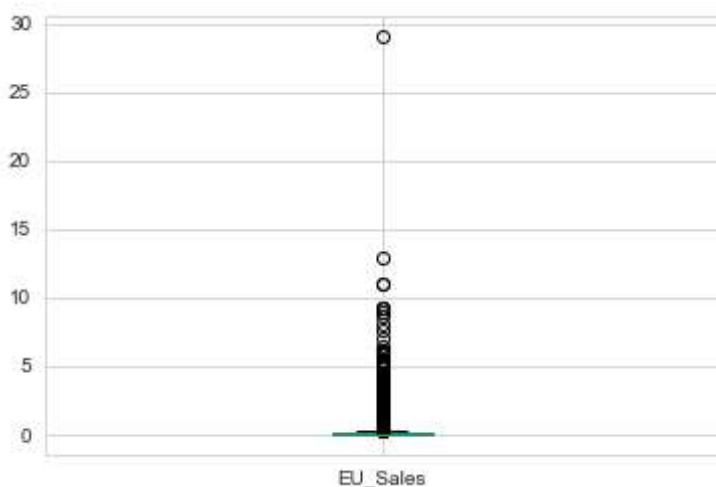
In [76]: `games_data.boxplot(column='NA_Sales')`

Out[76]: <AxesSubplot:>



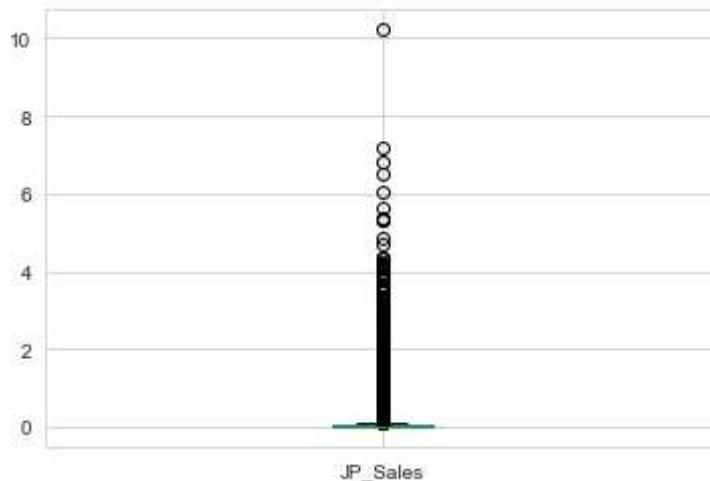
In [77]: `games_data.boxplot(column='EU_Sales')`

Out[77]: <AxesSubplot:>



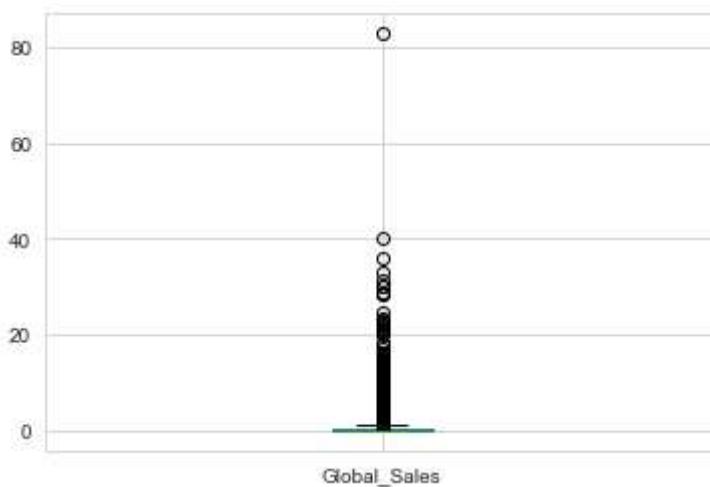
In [78]: `games_data.boxplot(column='JP_Sales')`

Out[78]: <AxesSubplot:>



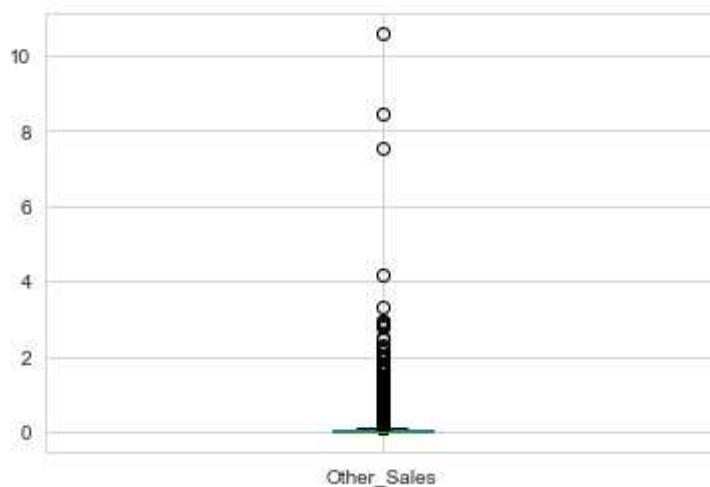
```
In [79]: games_data.boxplot(column='Global_Sales')
```

```
Out[79]: <AxesSubplot:>
```



```
In [80]: games_data.boxplot(column='Other_Sales')
```

```
Out[80]: <AxesSubplot:>
```



```
In [81]: grp=games_data.groupby('Genre')
x=grp['NA_Sales'].agg(np.mean)
y=grp['EU_Sales'].agg(np.mean)
z=grp['JP_Sales'].agg(np.mean)
```

```
In [82]: print(x)
print(y)
```

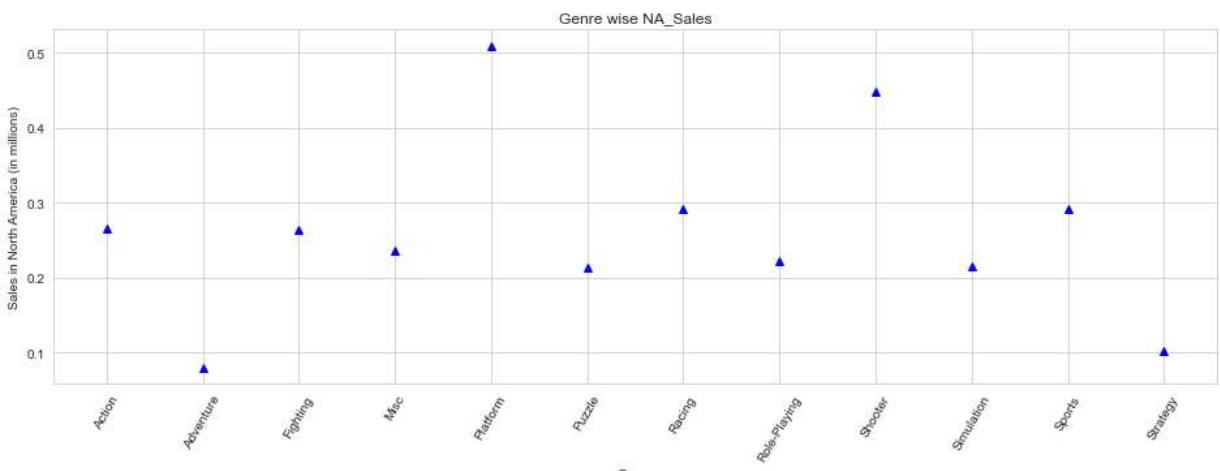
```
print(z)
```

```
Genre
Action      0.265078
Adventure   0.080008
Fighting    0.264043
Misc        0.235421
Platform    0.509703
Puzzle      0.214053
Racing      0.291371
Role-Playing 0.222109
Shooter     0.448643
Simulation  0.214363
Sports       0.290838
Strategy    0.101239
Name: NA_Sales, dtype: float64

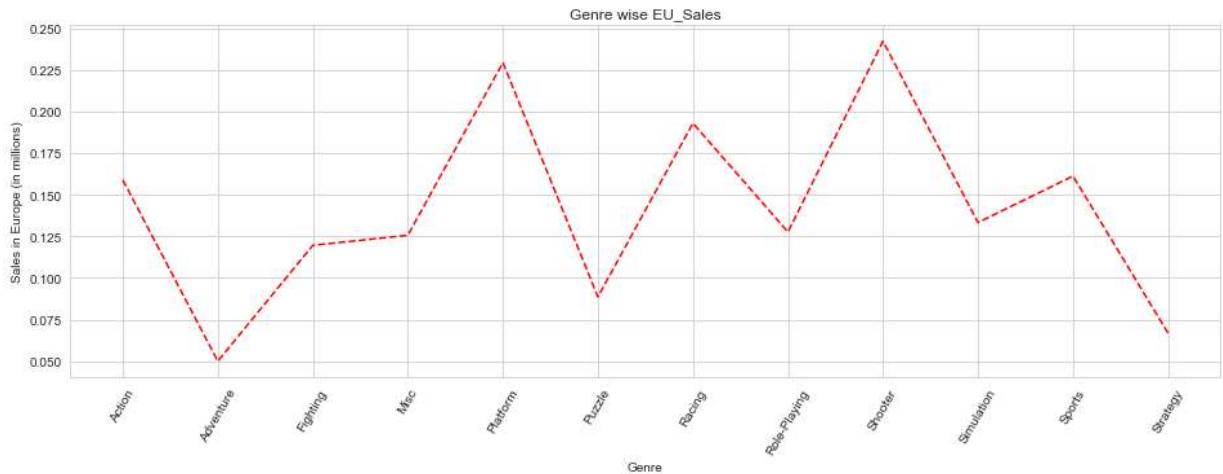
Genre
Action      0.158868
Adventure   0.050031
Fighting    0.119617
Misc        0.125605
Platform    0.229314
Puzzle      0.088632
Racing      0.192906
Role-Playing 0.127599
Shooter     0.242161
Simulation  0.133278
Sports       0.161172
Strategy    0.066925
Name: EU_Sales, dtype: float64

Genre
Action      0.048800
Adventure   0.040808
Fighting    0.104246
Misc        0.063268
Platform    0.149314
Puzzle      0.099439
Racing      0.046212
Role-Playing 0.238293
Shooter     0.029782
Simulation  0.074929
Sports       0.058490
Strategy    0.073284
Name: JP_Sales, dtype: float64
```

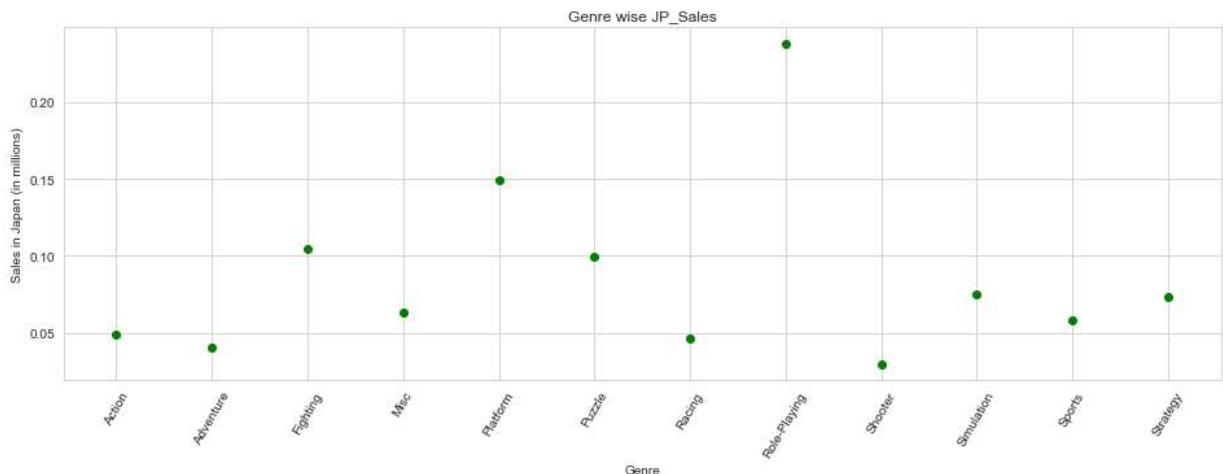
```
In [83]: plt.figure(figsize=(16,5))
plt.plot(x, 'g^', color='blue')
plt.xticks(rotation=60)
plt.title('Genre wise NA_Sales')
plt.xlabel('Genre')
plt.ylabel('Sales in North America (in millions)')
plt.show()
```



```
In [85]: plt.figure(figsize=(16,5))
plt.plot(y,'r--',color='red')
plt.xticks(rotation=60)
plt.title('Genre wise EU_Sales')
plt.xlabel('Genre')
plt.ylabel('Sales in Europe (in millions)')
plt.show()
```



```
In [86]: plt.figure(figsize=(16,5))
plt.plot(z,'ro',color='green')
plt.xticks(rotation=60)
plt.title('Genre wise JP_Sales')
plt.xlabel('Genre')
plt.ylabel('Sales in Japan (in millions)')
plt.show()
```

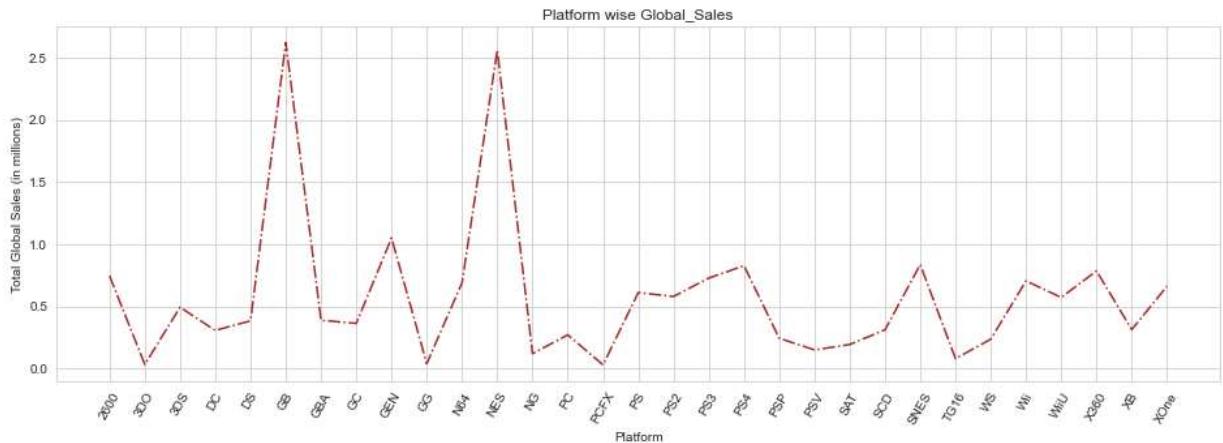


```
In [87]: grp=games_data.groupby('Platform')
a=grp['Global_Sales'].agg(np.mean)
b=grp['Other_Sales'].agg(np.mean)
print(a)
print(b)
```

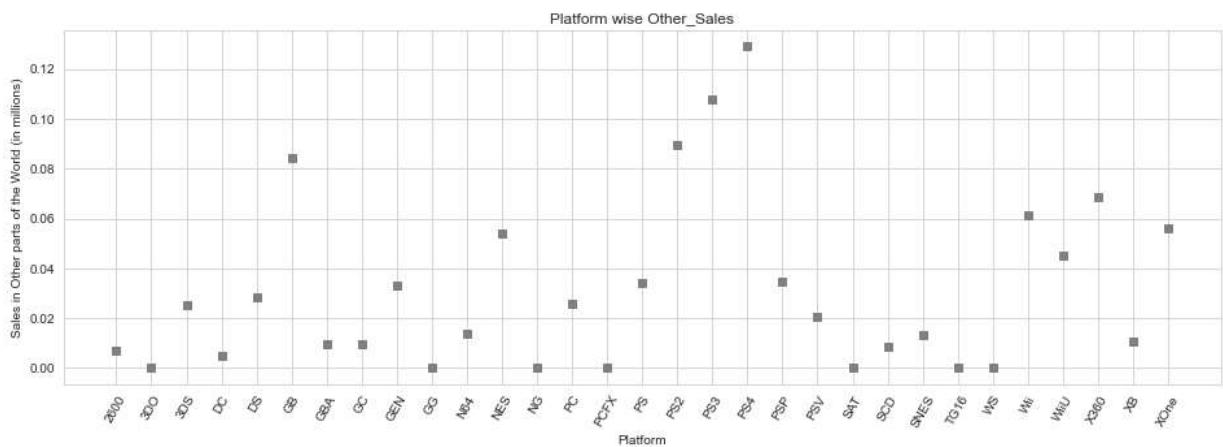
```
Platform
2600    0.746293
3DO     0.033333
3DS     0.493527
DC      0.307115
DS      0.384284
GB      2.622887
GBA     0.388830
GC      0.363727
GEN     1.050370
GG      0.040000
```

```
N64      0.690538
NES      2.561939
NG       0.120000
PC       0.271535
PCFX     0.030000
PS       0.611766
PS2      0.579906
PS3      0.728029
PS4      0.827679
PSP      0.243701
PSV      0.150244
SAT      0.194162
SCD      0.311667
SNES     0.837029
TG16     0.080000
WS       0.236667
Wii      0.705279
WiiU     0.572448
X360     0.785737
XB       0.313935
XOne     0.662254
Name: Global_Sales, dtype: float64
Platform
2600    0.007241
3DO      0.000000
3DS      0.025150
DC       0.005192
DS       0.028292
GB       0.084124
GBA      0.009555
GC       0.009465
GEN      0.032963
GG       0.000000
N64      0.013639
NES      0.054184
NG       0.000000
PC       0.025938
PCFX     0.000000
PS       0.034222
PS2      0.089549
PS3      0.107983
PS4      0.129048
PSP      0.034687
PSV      0.020512
SAT      0.000405
SCD      0.008333
SNES     0.013473
TG16     0.000000
WS       0.000000
Wii      0.061395
WiiU     0.045105
X360     0.068614
XB       0.010560
XOne     0.055962
Name: Other_Sales, dtype: float64
```

```
In [92]: plt.figure(figsize=(16,5))
plt.plot(a, '-.', color='Brown')
plt.xticks(rotation=60)
plt.title('Platform wise Global_Sales')
plt.xlabel('Platform')
plt.ylabel('Total Global Sales (in millions)')
plt.show()
```



```
In [94]: plt.figure(figsize=(16,5))
plt.plot(b,'bs',color='grey')
plt.xticks(rotation=60)
plt.title('Platform wise Other_Sales')
plt.xlabel('Platform')
plt.ylabel('Sales in Other parts of the World (in millions)')
plt.show()
```



```
In [41]: type(games_data)
```

```
Out[41]: pandas.core.frame.DataFrame
```

```
In [33]: games_data.head()
```

```
Out[33]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_S
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	

```
In [56]: nintendo=pd.DataFrame(games_data[games_data['Publisher']=='Nintendo'])
```

In [57]: `nintendo.head()`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_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	



In [58]: `nintendo.tail()`

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales
16269	16272	Slide Adventure: Mag Kid	DS	2007.0	Action	Nintendo	0.0	0.0	0.01	
16357	16360	Mario vs. Donkey Kong: Tipping Stars	WiiU	2015.0	Puzzle	Nintendo	0.0	0.0	0.01	
16456	16459	Art Academy: Home Studio	WiiU	2015.0	Misc	Nintendo	0.0	0.0	0.01	
16473	16476	Captain Rainbow	Wii	2008.0	Adventure	Nintendo	0.0	0.0	0.01	
16542	16545	Mario & Luigi: Paper Jam & Mario Kart 7 Double...	3DS	2015.0	Misc	Nintendo	0.0	0.0	0.01	



In [59]: `nintendo.describe()`

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	703.000000	696.000000	703.000000	703.000000	703.000000	703.000000	703.000000
mean	3860.830725	2003.83046	1.161977	0.595647	0.647824	0.135605	2.541337
std	4031.385208	7.77637	2.976782	1.736095	1.059476	0.462002	5.653126
min	1.000000	1983.00000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	719.000000	2000.00000	0.005000	0.000000	0.090000	0.000000	0.290000
50%	2335.000000	2005.00000	0.370000	0.120000	0.280000	0.030000	0.890000

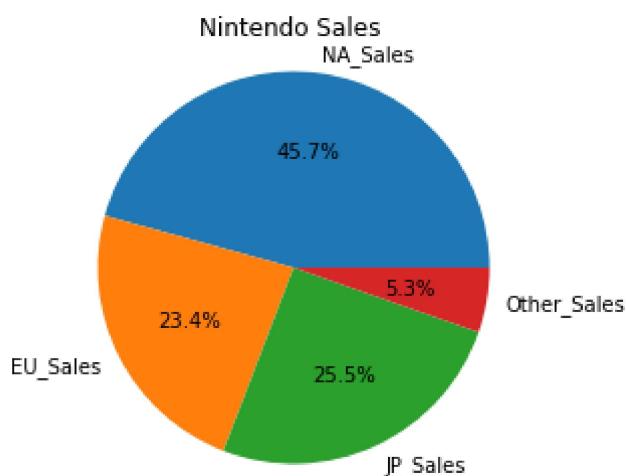
	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
75%	6009.500000	2009.00000	0.960000	0.480000	0.730000	0.090000	2.250000
max	16545.000000	2016.00000	41.490000	29.020000	10.220000	8.460000	82.740000

In [72]: `nin=nintendo.iloc[:,6:10].apply(np.mean)`

In [73]: `nin`

Out[73]: NA_Sales 1.161977
EU_Sales 0.595647
JP_Sales 0.647824
Other_Sales 0.135605
dtype: float64

In [74]: `nin_labels = 'NA_Sales','EU_Sales','JP_Sales','Other_Sales'
plt.pie(nin,labels=nin_labels,autopct='%1.1f%%')
plt.title('Nintendo Sales')
plt.axis('equal')
plt.show()`



In []: