Description

This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com.

Matrix column entries (attributes):

1. Rank - Ranking of overall sales
2. Name - The games name
3. Platform - Platform of the games release (i.e. PC,PS4, etc.)
4. Year - Year of the game's release
5. Genre - Genre of the game
6. Publisher - Publisher of the game
7. NA_Sales - Sales in North America (in millions)
8. EU_Sales - Sales in Europe (in millions)
9. JP_Sales - Sales in Japan (in millions)
10. Other_Sales - Sales in the rest of the world (in millions)
11. Global_Sales - Total worldwide sales.

Initial Data Exploration

- Find the popularity of each genre in different regions and globally
- Find global sales over the years
- Top 10 publishers
- Sales of games in different platforms
- Compare trends of different playstations and Pc over the years
- Frequency of different genres

Feature engineeing

- Handle duplicates
- Handle outliers
- Handle missing data
- Label encoding

Lib

```
import os #paths to file
import numpy as np # linear algebra
import pandas as pd # data processing
import warnings# warning filter

! pip install plotly --upgrade

#ploting libraries
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go


# ML libraries
from sklearn.preprocessing import LabelEncoder
```

```
Collecting plotly
  Downloading plotly-5.5.0-py2.py3-none-any.whl (26.5 MB)
Requirement already satisfied: six in c:\users\parth mehta\anaconda3\
lib\site-packages (from plotly) (1.15.0)
Collecting tenacity>=6.2.0
  Using cached tenacity-8.0.1-py3-none-any.whl (24 kB)
Installing collected packages: tenacity, plotly
Successfully installed plotly-5.5.0 tenacity-8.0.1
```

Preview Data

```
df=pd.read_csv("E:/projects/eda on video game sales data/vgsales.csv")
df.head()
```

```
   Rank                      Name Platform    Year         Genre
Publisher  \
0     1                Wii Sports      Wii  2006.0        Sports
Nintendo
1     2          Super Mario Bros.      NES  1985.0      Platform
Nintendo
2     3             Mario Kart Wii      Wii  2008.0        Racing
Nintendo
3     4          Wii Sports Resort      Wii  2009.0        Sports
Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  Role-Playing
Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
0     41.49     29.02      3.77         8.46         82.74
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
```

```
df.shape
```

```
(16598, 11)
```

```
df.dtypes
```

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

```
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
```

```
df.describe()
```

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

        Other_Sales  Global_Sales
count  16598.000000  16598.000000
mean       0.048063      0.537441
std        0.188588      1.555028
min        0.000000      0.010000
25%        0.000000      0.060000
50%        0.010000      0.170000
75%        0.040000      0.470000
max       10.570000     82.740000
```

```
df.isnull().sum()
```

```
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
```

Check duplication

```
df=df.drop_duplicates(keep='first')
```

Data Preprocessing

```
# null values are little so i will drop them
df.dropna(inplace=True)
df['Year']=df['Year'].astype(int)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16291 entries, 0 to 16597
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Rank          16291 non-null  int64
 1   Name          16291 non-null  object
 2   Platform      16291 non-null  object
 3   Year          16291 non-null  int32
 4   Genre         16291 non-null  object
 5   Publisher     16291 non-null  object
 6   NA_Sales      16291 non-null  float64
 7   EU_Sales      16291 non-null  float64
 8   JP_Sales      16291 non-null  float64
 9   Other_Sales   16291 non-null  float64
 10  Global_Sales  16291 non-null  float64
dtypes: float64(5), int32(1), int64(1), object(4)
memory usage: 1.4+ MB
```

```
df.head()
```

```
   Rank                     Name Platform  Year         Genre
Publisher  \
1     2         Super Mario Bros.      NES  1985      Platform
Nintendo
2     3            Mario Kart Wii      Wii  2008        Racing
Nintendo
3     4         Wii Sports Resort      Wii  2009        Sports
Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996  Role-Playing
Nintendo
5     6                    Tetris       GB  1989        Puzzle
Nintendo
```

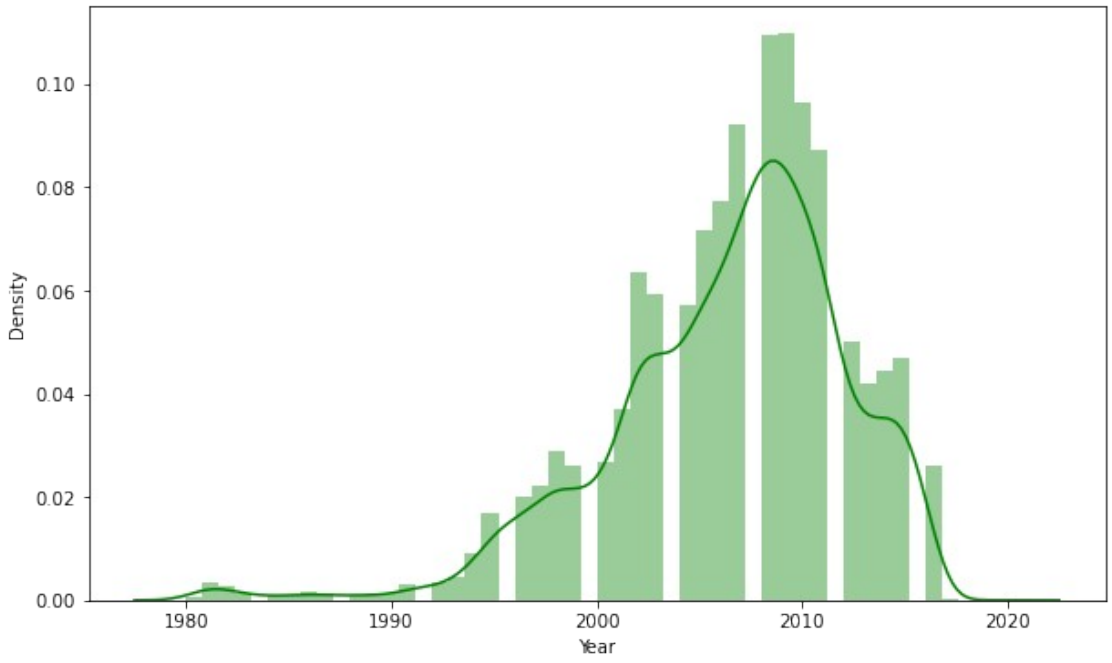
```
   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
5     23.20      2.26      4.22         0.58         30.26
```

EDA

```
ax=plt.figure(figsize=(10,6))
sns.distplot(df['Year'],color='green')
```

```
C:\Users\Parth Mehta\anaconda3\lib\site-packages\seaborn\
distributions.py:2557: FutureWarning: `distplot` is a deprecated
function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='Year', ylabel='Density'>
```



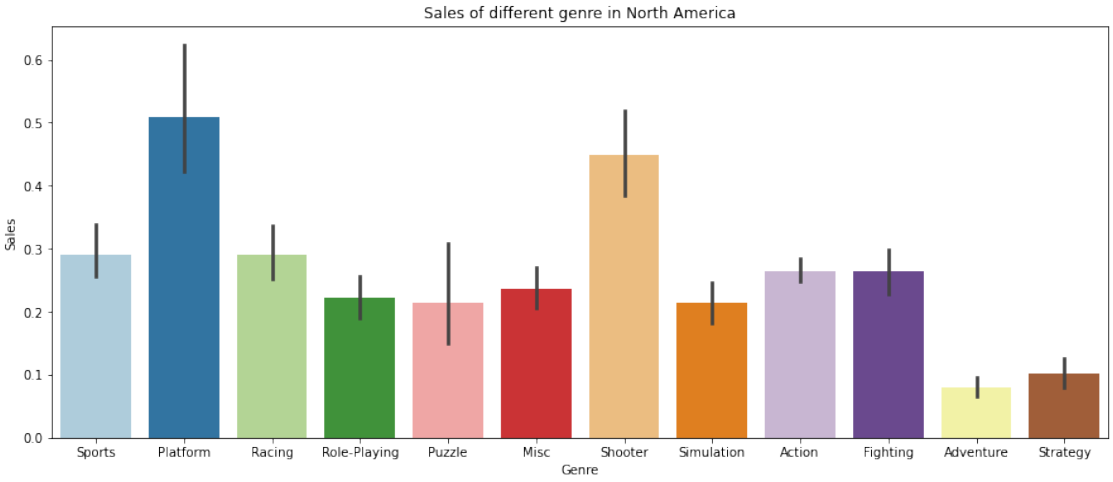
2009 has the highest numberof game releases

Sales of different genre in North America

```
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y='NA_Sales',data=df,palette='Paired')
```

```
plt.ylabel('Sales')
plt.title('Sales of different genre in North America')
```

```
Text(0.5, 1.0, 'Sales of different genre in North America')
```



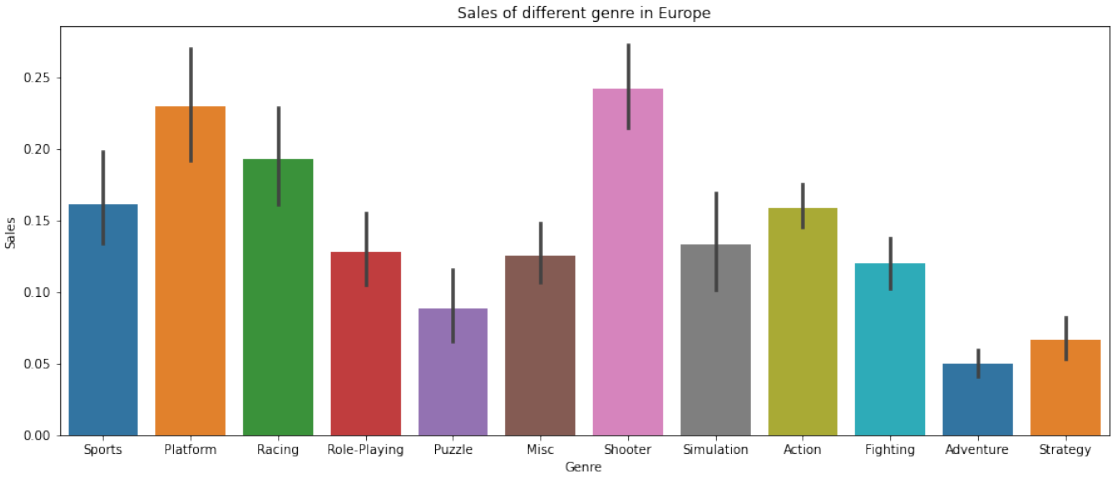
Platform and shooter games are the most played game genre in North America

---

Sales of different genre in Europe

```
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y='EU_Sales',data=df,palette='tab10')
plt.ylabel('Sales')
plt.title('Sales of different genre in Europe')
```

```
Text(0.5, 1.0, 'Sales of different genre in Europe')
```



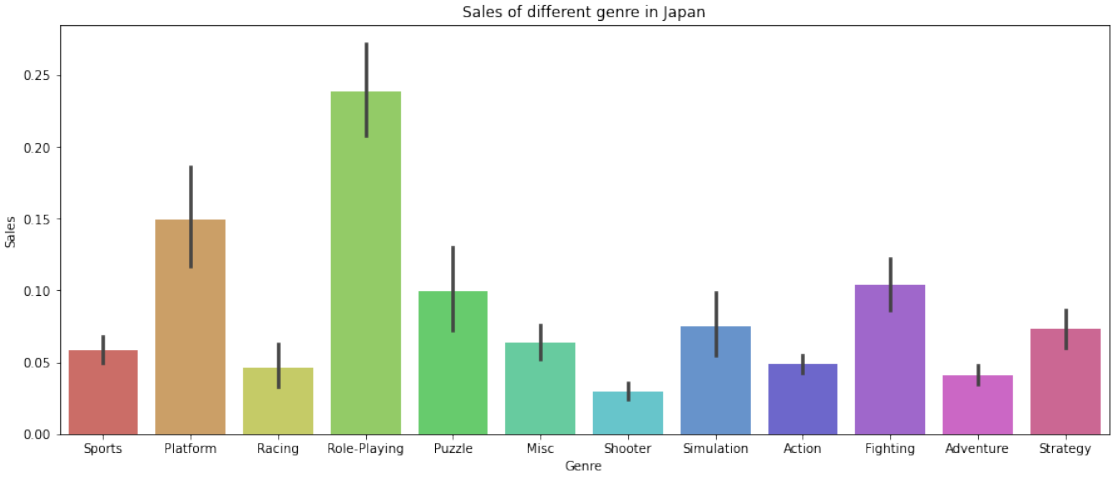
Platform and shooter games are the most played game genre in Europe

---

Sales of different genre in Japan

```
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y='JP_Sales',data=df,palette='hls')
plt.ylabel('Sales')
plt.title('Sales of different genre in Japan')
```

```
Text(0.5, 1.0, 'Sales of different genre in Japan')
```

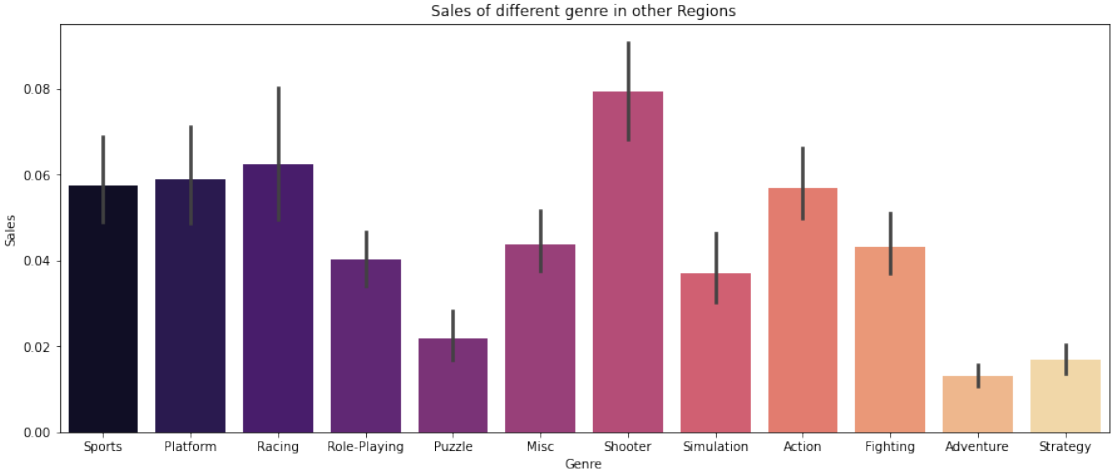


Role play games are highly played in Japan

Sales of different genre in other Regions

```
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y='Other_Sales',data=df,palette='magma')
plt.ylabel('Sales')
plt.title('Sales of different genre in other Regions')
```

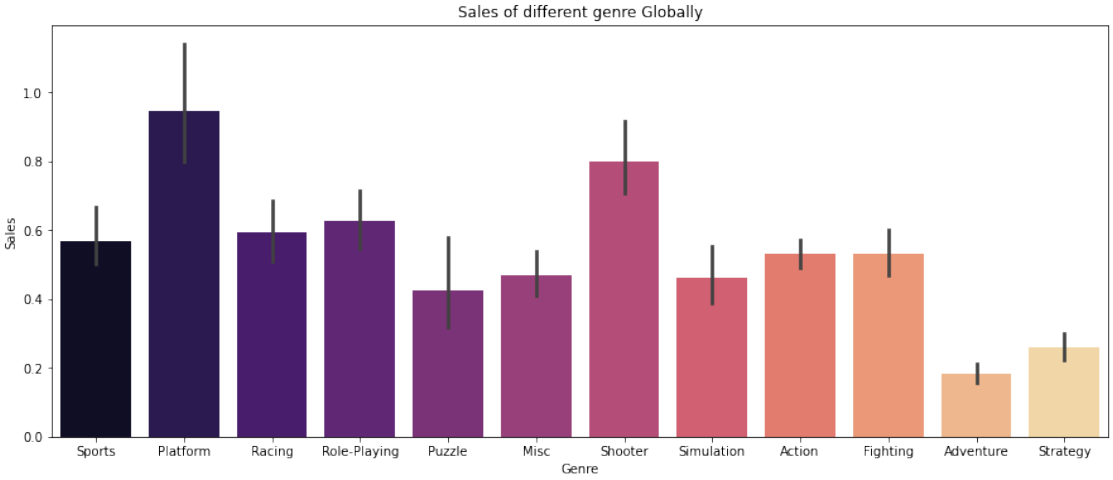
```
Text(0.5, 1.0, 'Sales of different genre in other Regions')
```



Sales of different genre Globally

```
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y='Global_Sales',data=df,palette='magma')
plt.ylabel('Sales')
plt.title('Sales of different genre Globally')
```

```
Text(0.5, 1.0, 'Sales of different genre Globally')
```



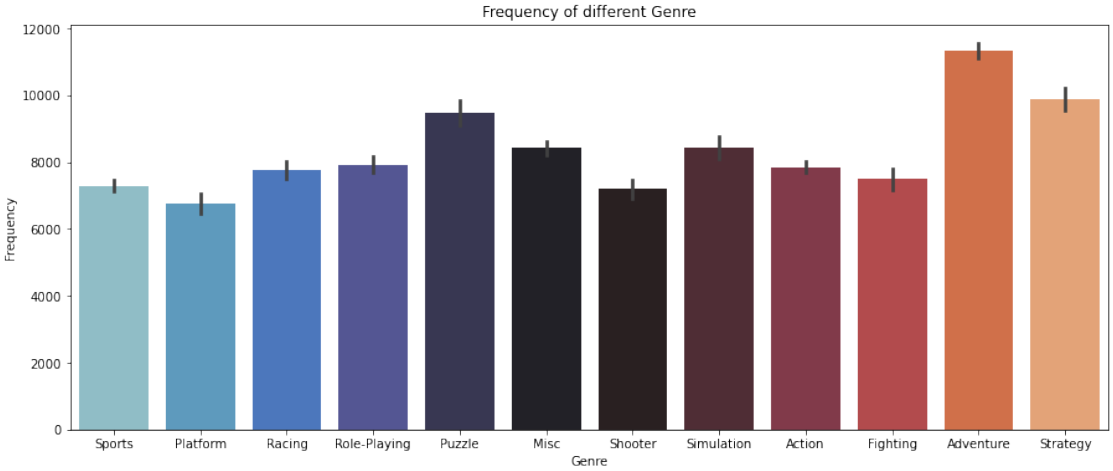
Platform and shooter games are the most played game genre Globally

---

Frequency of different Genre

```
a=np.arange(1,16292)
ax=plt.figure(figsize=(15,6))
sns.barplot(x='Genre',y=a,data=df,palette='icefire')
plt.ylabel('Frequency')
plt.title('Frequency of different Genre')
```

```
Text(0.5, 1.0, 'Frequency of different Genre')
```

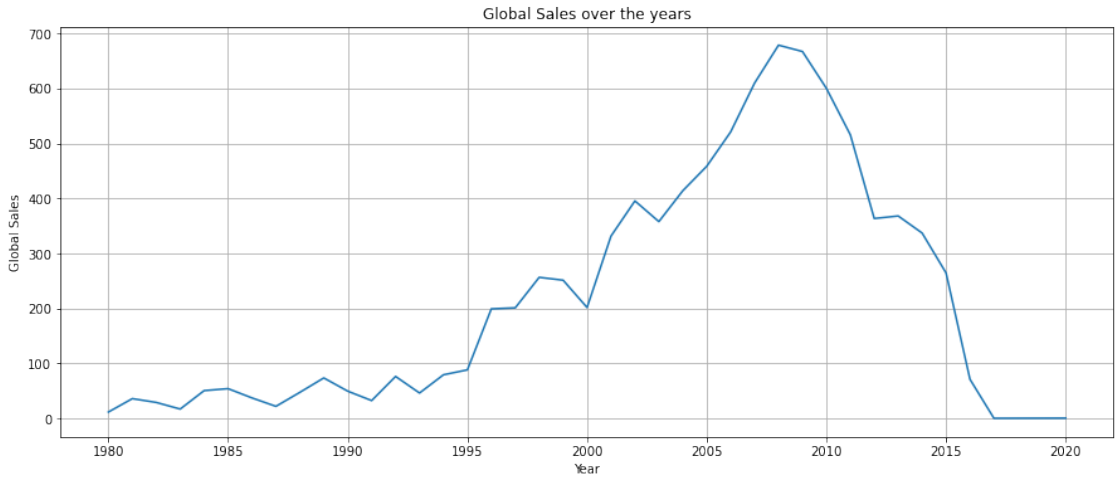


Adventure and strategy games are the highest in the dataset

Global Sales over the years

```
ax=plt.figure(figsize=(15,6))
df.groupby(['Year'])['Global_Sales'].sum().plot()
plt.grid()
plt.ylabel('Global Sales')
plt.title('Global Sales over the years')
```

```
Text(0.5, 1.0, 'Global Sales over the years')
```

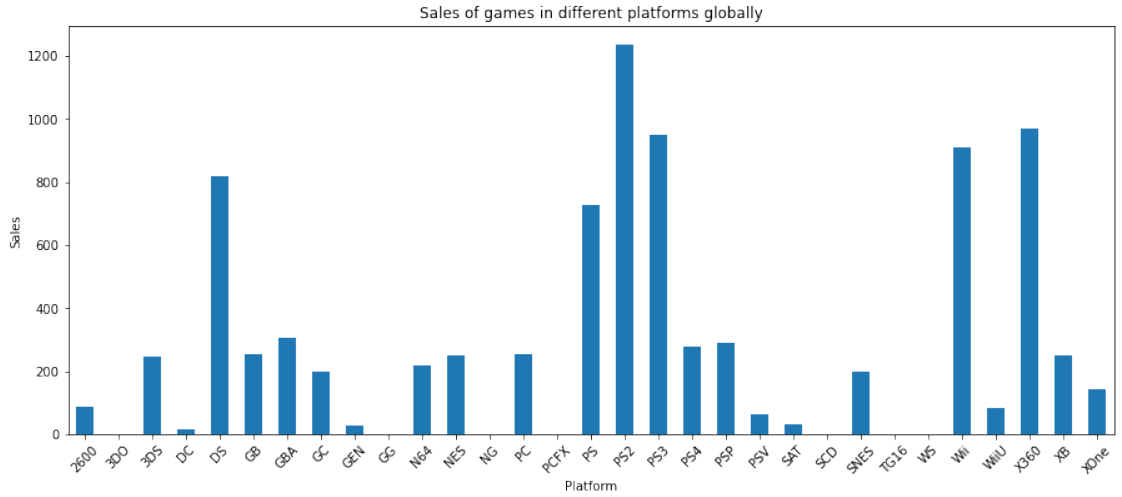


Highest sales have been recorded from 2007 to 2010

Sales of games in different platforms globally

```
ax=plt.figure(figsize=(15,6))
df.groupby(['Platform'])['Global_Sales'].sum().plot.bar()
plt.xticks(rotation=45)
plt.ylabel('Sales')
plt.title('Sales of games in different platforms globally')
```

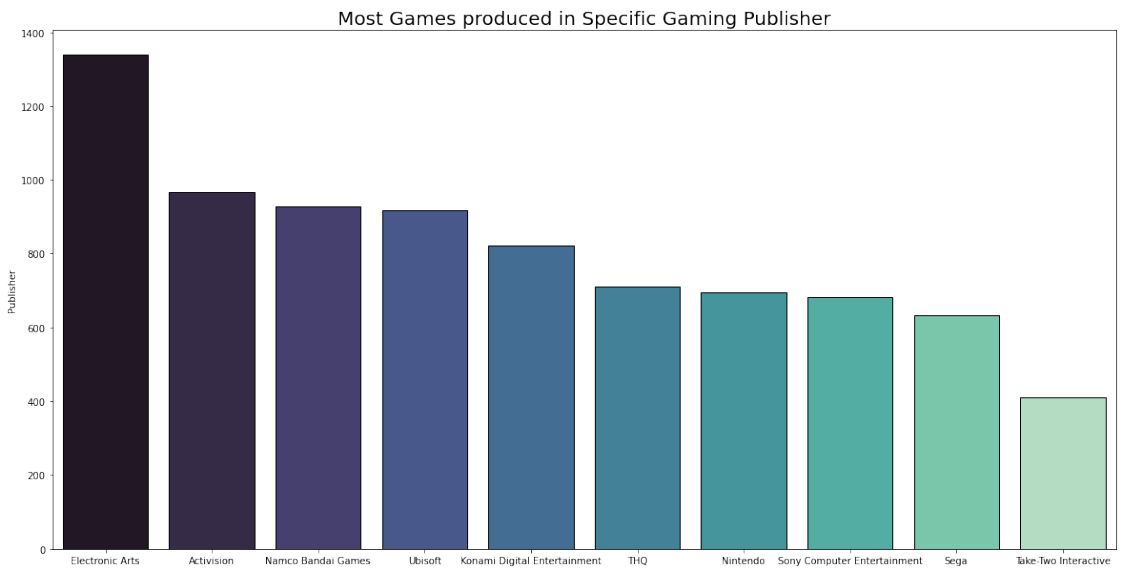
```
Text(0.5, 1.0, 'Sales of games in different platforms globally')
```



PS2 has recorded highest sales globally

```
top10=df["Publisher"].value_counts().index
plt.figure(figsize = (20,10))
sns.barplot(top10[:10], df["Publisher"].value_counts().iloc[:10]
            ,palette='mako'
            ,edgecolor='black'
            )
plt.title("Most Games produced in Specific Gaming
Publisher",fontsize=20)
plt.show()
```

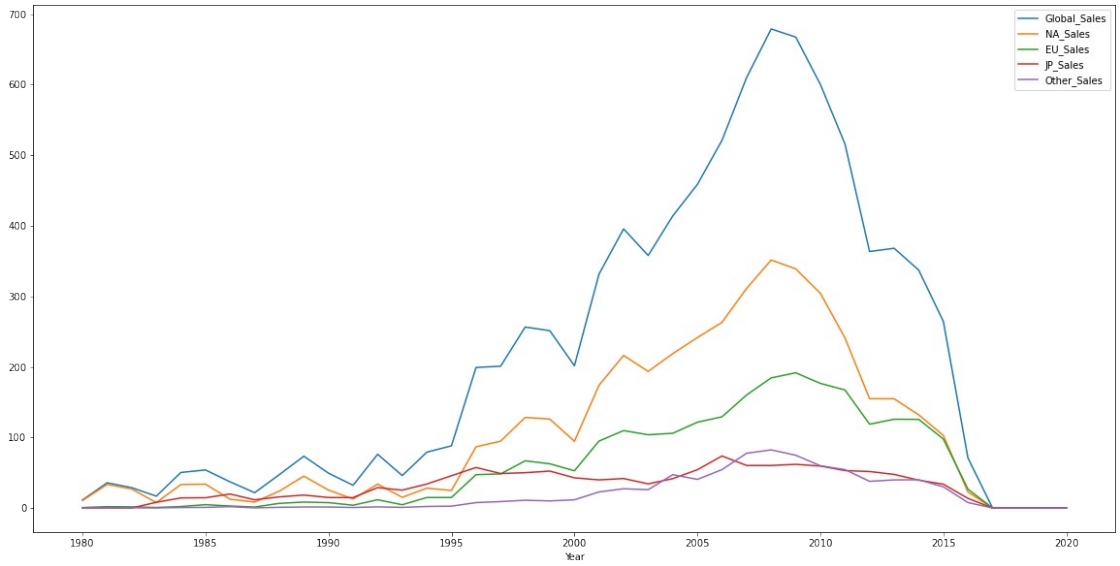
```
C:\Users\Parth Mehta\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning: Pass the following variables as
keyword args: x, y. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



Top 10 Publisher in order and most is Electronic Arts

```
GSales_Year = df.groupby('Year')
[['Global_Sales','NA_Sales','EU_Sales','JP_Sales','Other_Sales']].sum(
)
GSales_Year.plot(figsize = (20,10))
```

```
<AxesSubplot:xlabel='Year'>
```



1. most of sales between 2005 and 2010
2. lest sales in 1980 to 1990

Playstation vs PC Global Sales Comparison¶

```
PS = df[df['Platform'] == 'PS'].groupby('Year')
['Global_Sales'].sum().reset_index()
PS2 = df[df['Platform'] == 'PS2'].groupby('Year')
['Global_Sales'].sum().reset_index()
PS3 = df[df['Platform'] == 'PS3'].groupby('Year')
['Global_Sales'].sum().reset_index()
PS4 = df[df['Platform'] == 'PS4'].groupby('Year')
['Global_Sales'].sum().reset_index()
PC = df[df['Platform'] == 'PC'].groupby('Year')
['Global_Sales'].sum().reset_index()

fig = go.Figure()
fig.add_trace(go.Scatter(x=PS['Year'], y=PS['Global_Sales'],
                         name="PS Sales",
                         hovertext=PS['Global_Sales']))

fig.add_trace(go.Scatter(x=PS2['Year'], y=PS2['Global_Sales'],
                         name="PS2 Sales",
                         hovertext=PS2['Global_Sales']))

fig.add_trace(go.Scatter(x=PS3['Year'], y=PS3['Global_Sales'],
                         name="PS3 Sales",
                         hovertext=PS3['Global_Sales']))

fig.add_trace(go.Scatter(x=PS4['Year'], y=PS4['Global_Sales'],
                         name="PS4 Sales",
                         hovertext=PS4['Global_Sales']))

fig.add_trace(go.Scatter(x=PC['Year'], y=PC['Global_Sales'],
                         name="PC Sales",
                         hovertext=PC['Global_Sales']))

fig.update_layout(title_text='Playstation vs PC Global Sales
Comparison',
                  title_x=0.5, title_font=dict(size=22))
fig.update_layout(
    xaxis_title="Year",
    yaxis_title="Global Sales (M)")

fig.show()
```


```
{"data":[{"y":
[6.020000000000001,35.92000000000002,94.67999999999999,136.07999999999
99,169.58,144.5700000000001,96.27999999999993,35.520000000000024,6.689
999999999998,2.05],"x":
[1994,1995,1996,1997,1998,1999,2000,2001,2002,2003],"name":"PS
Sales","hovertext":
[6.020000000000001,35.92000000000002,94.67999999999999,136.07999999999
99,169.58,144.5700000000001,96.27999999999993,35.520000000000024,6.689
999999999998,2.05],"type":"scatter"},{"y":
```

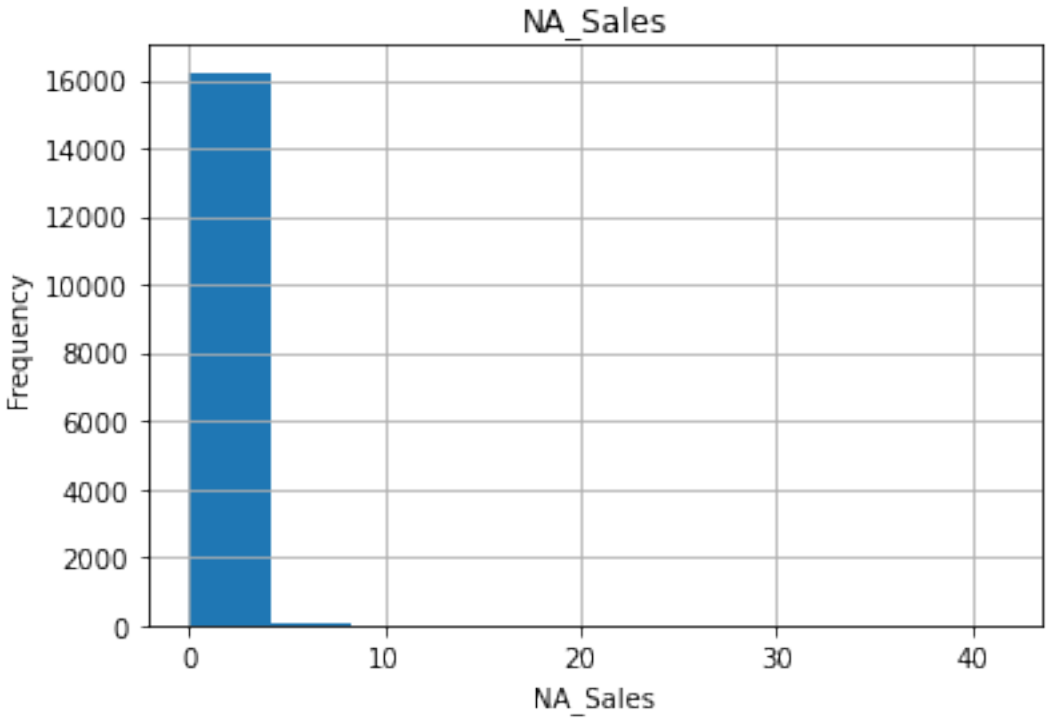
```
[39.11000000000001,166.43000000000006,205.40000000000006,184.289999999
99996,211.77999999999992,160.65000000000012,103.41999999999999,76,53.8
30000000000034,26.45,5.629999999999995,0.47],"x":
[2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011],"name":"
PS2 Sales","hovertext":
[39.11000000000001,166.43000000000006,205.40000000000006,184.289999999
99996,211.77999999999992,160.65000000000012,103.41999999999999,76,53.8
30000000000034,26.45,5.629999999999995,0.47],"type":"scatter"},{"y":
[21.070000000000004,73.81000000000006,119.69000000000001,132.339999999
99997,144.42000000000007,159.3700000000001,109.49000000000002,117.3899
9999999994,50.96000000000002,18.220000000000002,2.59],"x":
[2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016],"name":"PS3
Sales","hovertext":
[21.070000000000004,73.81000000000006,119.69000000000001,132.339999999
99997,144.42000000000007,159.3700000000001,109.49000000000002,117.3899
9999999994,50.96000000000002,18.220000000000002,2.59],"type":"scatter"
},{"y":
[24.760000000000005,98.76000000000003,115.29999999999997,39.2500000000
0002,3.0e-2],"x":[2013,2014,2015,2016,2017],"name":"PS4
Sales","hovertext":
[24.760000000000005,98.76000000000003,115.29999999999997,39.2500000000
0002,3.0e-2],"type":"scatter"},{"y":[3.0e-2,3.0e-
2,3.0199999999999996,12.85,4.2299999999999995,10.59,11.260000000000002
,3.2800000000000002,4.749999999999999,4.679999999999999,5.51,8.5999999
99999996,8.959999999999988,10.459999999999992,4.469999999999995,2.9699
99999999998,9.399999999999984,12.669999999999975,17.160000000000043,24
.300000000000004,35.06000000000003,23.53,12.829999999999995,13.3899999
99999993,8.069999999999997,2.5999999999999974],"x":
[1985,1988,1992,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004
,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016],"name":"
PC Sales","hovertext":[3.0e-2,3.0e-
2,3.0199999999999996,12.85,4.2299999999999995,10.59,11.260000000000002
,3.2800000000000002,4.749999999999999,4.679999999999999,5.51,8.5999999
99999996,8.959999999999988,10.459999999999992,4.469999999999995,2.9699
99999999998,9.399999999999984,12.669999999999975,17.160000000000043,24
.300000000000004,35.06000000000003,23.53,12.829999999999995,13.3899999
99999993,8.069999999999997,2.5999999999999974],"type":"scatter"}],"con
fig":{"plotlyServerURL":"https://plot.ly"},"layout":{"title":
{"x":0.5,"font":{"size":22},"text":"Playstation vs PC Global Sales
Comparison"},"xaxis":{"title":{"text":"Year"}},"template":{"data":
{"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"scattermapbox"
:[{"type":"scattermapbox","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
```

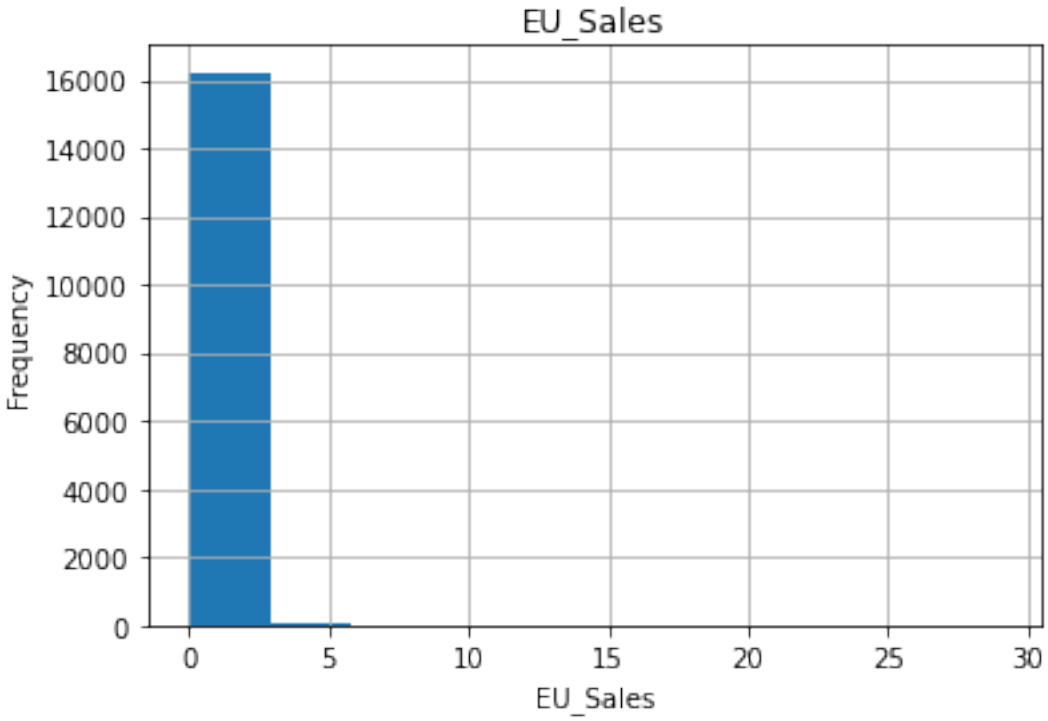
```
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"pie":
[{"automargin":true,"type":"pie"}],"carpet":[{"aaxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"baxis":
{"linecolor":"white","minorgridcolor":"white","endlinecolor":"#2a3f5f"
,"startlinecolor":"#2a3f5f","gridcolor":"white"},"type":"carpet"}],"ba
r":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"type":"bar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"barpolar":
[{"type":"barpolar","marker":{"line":
{"width":0.5,"color":"#E5ECF6"},"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"scatter3d":
[{"line":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d","marker":
{"colorbar":{"outlinewidth":0,"ticks":""}}}],"contour":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"histogram2d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"scatterpolar":
[{"type":"scatterpolar","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"histogram":
[{"type":"histogram","marker":{"pattern":
{"solidity":0.2,"fillmode":"overlay","size":10}}}],"histogram2dcontour
":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"parcoords":[{"line":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"scatterpolargl":
[{"type":"scatterpolargl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"scattercarpet":
```

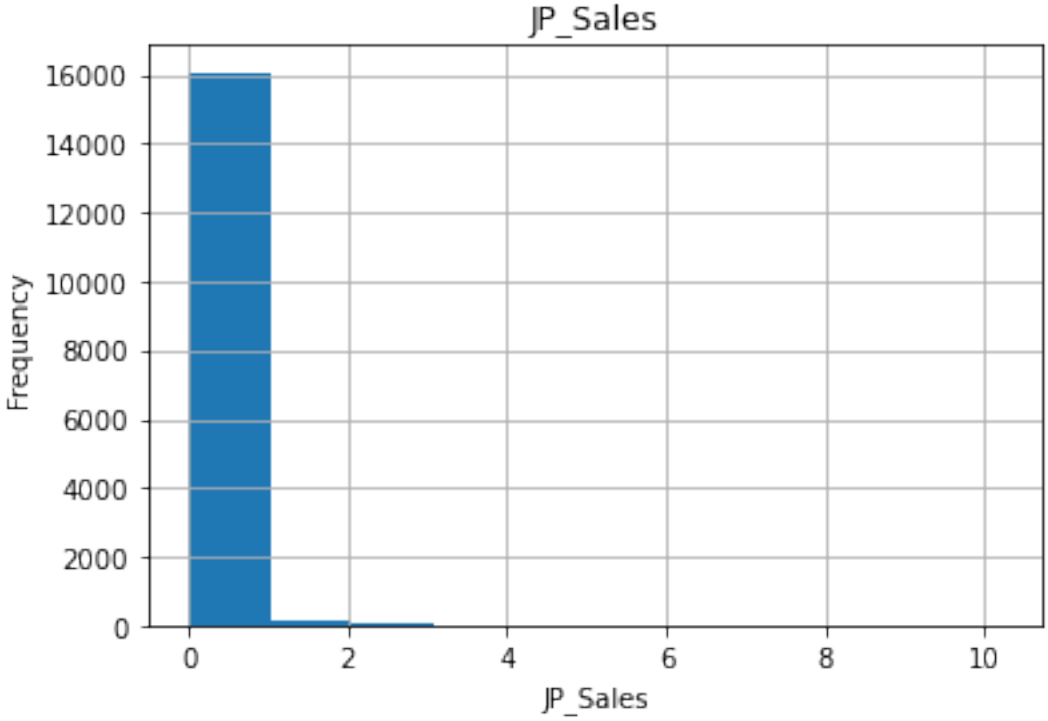
```
[{"type":"scattercarpet","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"choropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"scatterternary":
[{"type":"scatterternary","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"scatter":
[{"type":"scatter","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}],"scattergeo":
[{"type":"scattergeo","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}],"surface":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"scattergl":
[{"type":"scattergl","marker":{"colorbar":
{"outlinewidth":0,"ticks":""}}}]},"layout":{"ternary":{"aaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"baxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"caxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"autotypenumbers":"strict","shapedefaults":{"line":
{"color":"#2a3f5f"}},"annotationdefaults":
{"arrowwidth":1,"arrowcolor":"#2a3f5f","arrowhead":0},"coloraxis":
{"colorbar":{"outlinewidth":0,"ticks":""}},"title":{"x":5.0e-
2},"hoverlabel":{"align":"left"},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]]},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","scene":{"zaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"xaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
,"yaxis":
{"linecolor":"white","showbackground":true,"zerolinecolor":"white","gr
```

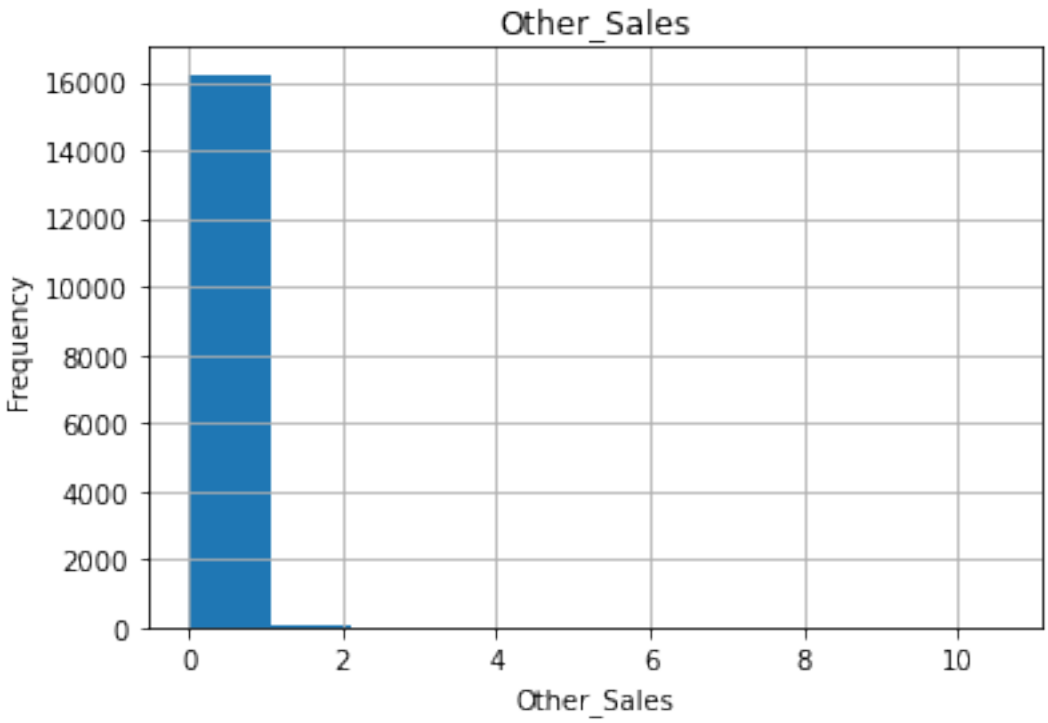
idwidth":2,"ticks":"","backgroundcolor":"#E5ECF6","gridcolor":"white"}
},"font":{"color":"#2a3f5f"},"xaxis":{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"polar":{"angularaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"radialaxis":
{"linecolor":"white","ticks":"","gridcolor":"white"},"bgcolor":"#E5ECF
6"},"plot_bgcolor":"#E5ECF6","geo":
{"subunitcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","sho
wland":true,"showlakes":true,"bgcolor":"white"},"yaxis":
{"linecolor":"white","title":
{"standoff":15},"zerolinewidth":2,"automargin":true,"zerolinecolor":"w
hite","ticks":"","gridcolor":"white"},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"]}},"yaxis":{"title":{"text":"Global
Sales (M)"}}}}

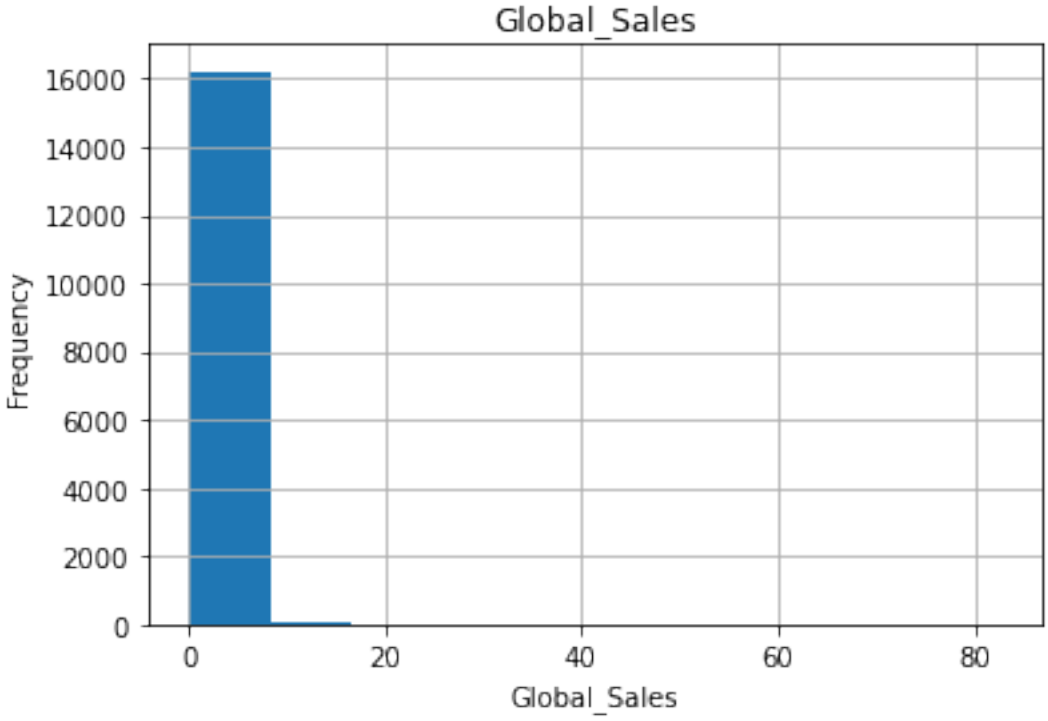
```
numerical_features=['NA_Sales','EU_Sales','JP_Sales','Other_Sales','Gl
obal_Sales']
for feature in numerical_features:
    df[feature].hist()
    plt.title(feature)
    plt.xlabel(feature)
    plt.ylabel("Frequency")
    plt.show()
```




EU_Sales
16000
14000
12000
10000
8000
6000
4000
2000
0
Frequency
0
5
10
15
20
25
30
EU_Sales


JP_Sales
16000
14000
12000
10000
8000
6000
4000
2000
0
Frequency
0
2
4
6
8
10
JP_Sales



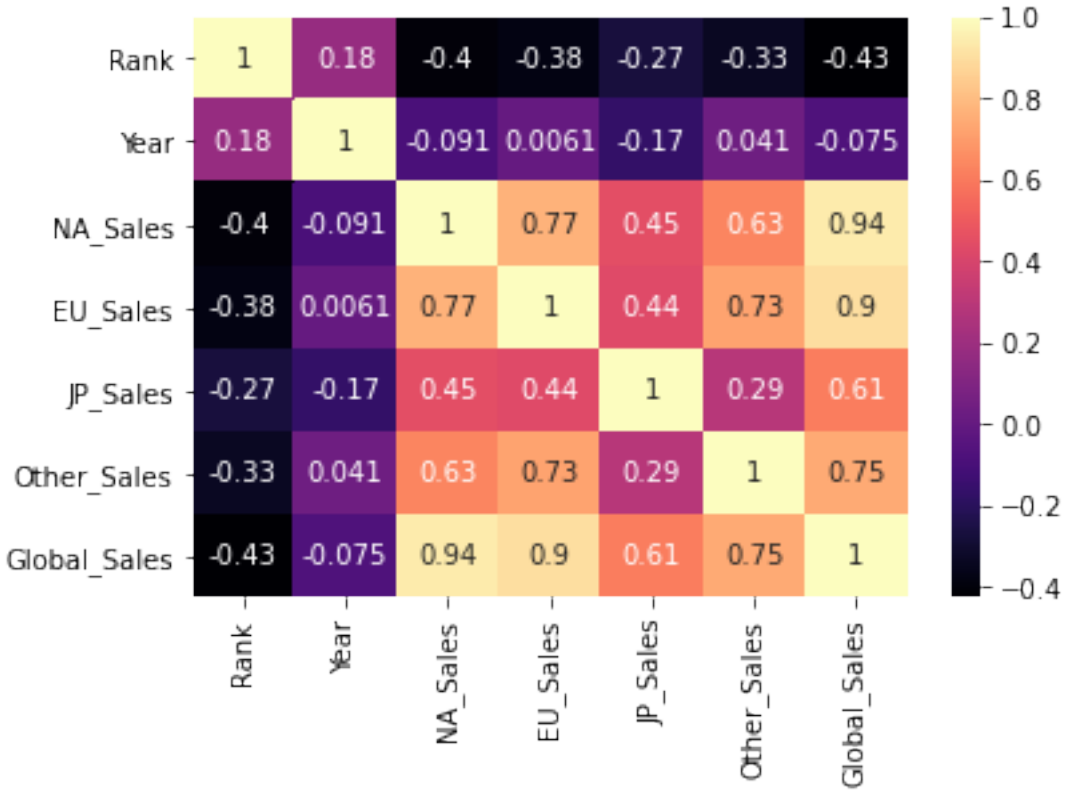


We can see that all of the data is right skewed

Dealing with Outliers

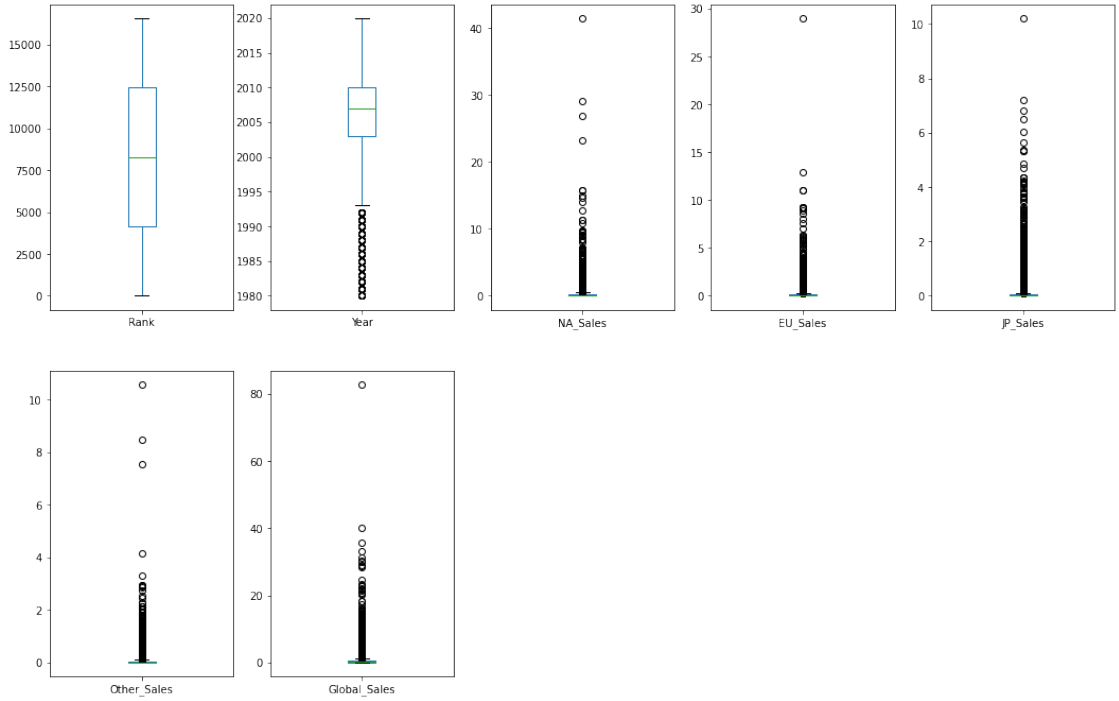
```
dfcopy=df.corr()

sns.heatmap(dfcopy ,annot = True, cmap="magma")
```

```
<AxesSubplot:>
```



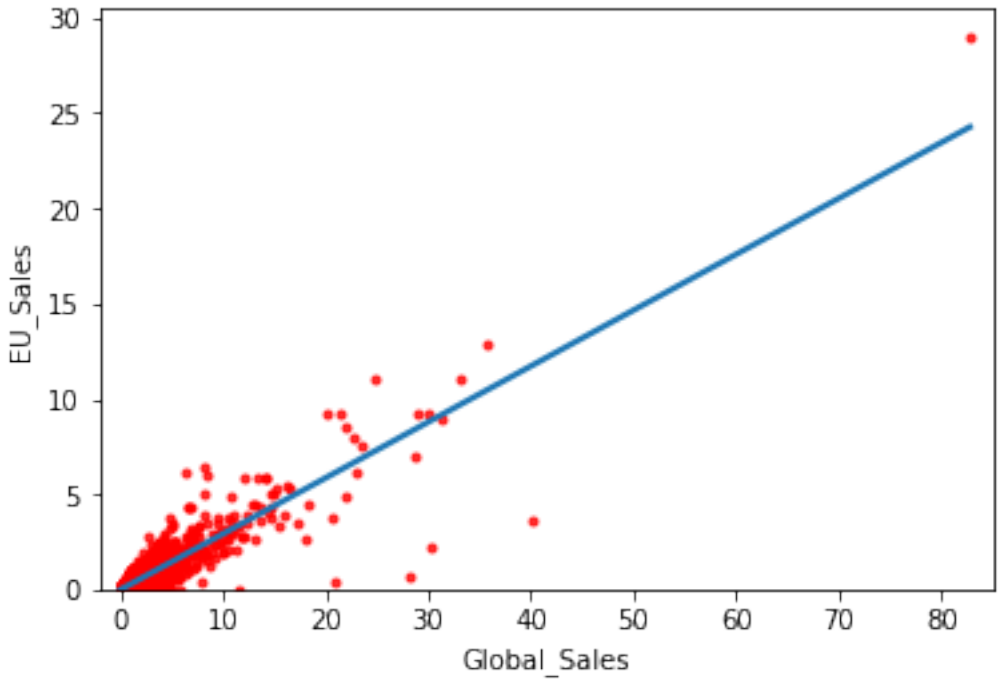

```
df.plot(kind = "box" , subplots = True , figsize = (18,18), layout =
(3,5))
```

```
Rank                AxesSubplot(0.125,0.657941;0.133621x0.222059)
Year              AxesSubplot(0.285345,0.657941;0.133621x0.222059)
NA_Sales           AxesSubplot(0.44569,0.657941;0.133621x0.222059)
EU_Sales          AxesSubplot(0.606034,0.657941;0.133621x0.222059)
JP_Sales          AxesSubplot(0.766379,0.657941;0.133621x0.222059)
Other_Sales          AxesSubplot(0.125,0.391471;0.133621x0.222059)
Global_Sales      AxesSubplot(0.285345,0.391471;0.133621x0.222059)
dtype: object
```



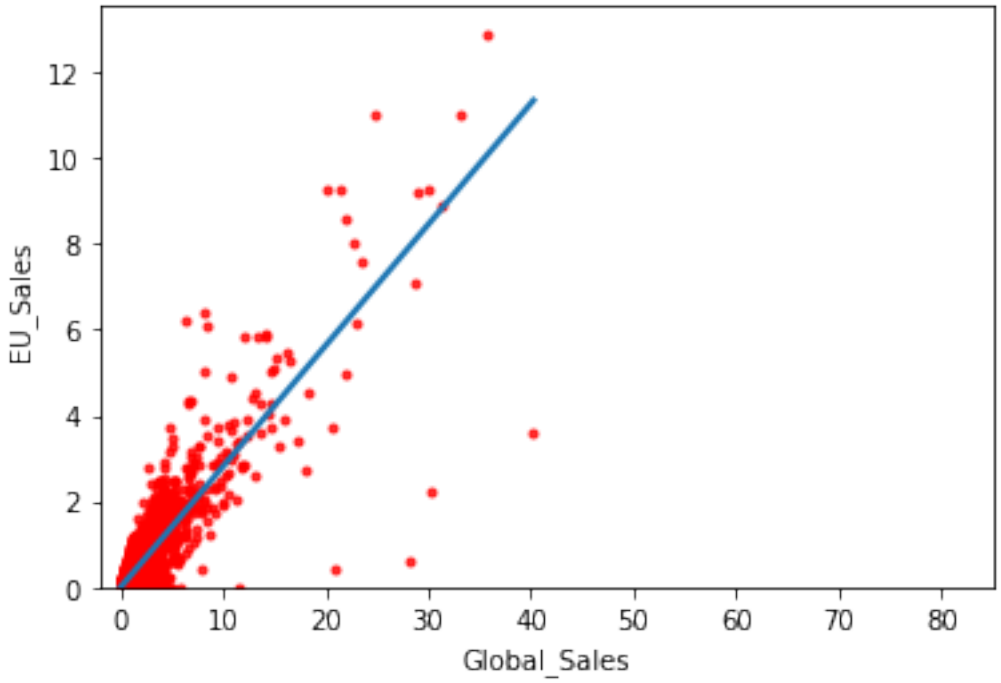

```
g = sns.regplot(df.Global_Sales,df.EU_Sales,ci=None,scatter_kws=
{"color":"r","s":9});
plt.xlim(-2,85)
plt.ylim(bottom=0)
```

```
C:\Users\Parth Mehta\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.


(0.0, 30.47140487188924)
```




```
df = df.drop([0],axis=0)

g = sns.regplot(df.Global_Sales,df.EU_Sales,ci=None,scatter_kws=
{"color":"r","s":9});
plt.xlim(-2,85)
plt.ylim(bottom=0)
```

```
C:\Users\Parth Mehta\anaconda3\lib\site-packages\seaborn\
_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12,
the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.


(0.0, 13.52411103334368)
```



Hypothesis Testing

1. Null Hypothesis - The average sales in North America and Europe Are same . Alternate Hypothesis - The average sales in North America and Europe Are same
2. Null Hypothesis - The Average sales in Japan and other regions are same . Alternate Hypothesis - The Average Sales in Japan and other regions are not same
3. Null Hypothesis - The average sales of activision and Electronic Arts is same . Alternate Hypothesis - The average sales of activision and Electronic Arts is same

We will test the first Hypothesis

```
import scipy.stats as stats
import math
np.random.seed(6)
sample_size=750
na_sample=np.random.choice(df['NA_Sales'],sample_size)
eu_sample=np.random.choice(df['EU_Sales'],sample_size)

from scipy.stats import ttest_1samp
ttest,p_value=ttest_1samp(na_sample,eu_sample.mean())

print(p_value)

0.00018810454545361085

if p_value < 0.05:
    print(" we are rejecting null hypothesis")
else:
    print("we are accepting null hypothesis")
```

```
 we are rejecting null hypothesis
```

Therefore,the average sales in North america and Europe are not same

Label Encoding and prepare X and y

```
df.head()
```

```
   Rank                      Name Platform  Year         Genre
Publisher  \
1     2         Super Mario Bros.      NES  1985      Platform
Nintendo
2     3            Mario Kart Wii      Wii  2008        Racing
Nintendo
3     4         Wii Sports Resort      Wii  2009        Sports
Nintendo
4     5  Pokemon Red/Pokemon Blue       GB  1996  Role-Playing
Nintendo
5     6                    Tetris       GB  1989        Puzzle
Nintendo

   NA_Sales  EU_Sales  JP_Sales  Other_Sales  Global_Sales
1     29.08      3.58      6.81         0.77         40.24
2     15.85     12.88      3.79         3.31         35.82
3     15.75     11.01      3.28         2.96         33.00
4     11.27      8.89     10.22         1.00         31.37
5     23.20      2.26      4.22         0.58         30.26
```

```
from sklearn.preprocessing import LabelEncoder

dff = df.copy()

le = LabelEncoder()

feature = ["Platform", "Genre"]


for col in feature:
    dff[col] = le.fit_transform(df[col])


dff.head()
```

```
   Rank                      Name  Platform  Year  Genre Publisher
NA_Sales  \
1     2         Super Mario Bros.        11  1985      4  Nintendo
29.08
```

```
2     3            Mario Kart Wii        26  2008      6  Nintendo
15.85
3     4         Wii Sports Resort        26  2009     10  Nintendo
15.75
4     5  Pokemon Red/Pokemon Blue         5  1996      7  Nintendo
11.27
5     6                    Tetris         5  1989      5  Nintendo
23.20

   EU_Sales  JP_Sales  Other_Sales  Global_Sales
1      3.58      6.81         0.77         40.24
2     12.88      3.79         3.31         35.82
3     11.01      3.28         2.96         33.00
4      8.89     10.22         1.00         31.37
5      2.26      4.22         0.58         30.26
```

```
X = dff[['Platform', 'Genre', 'NA_Sales', 'EU_Sales', 'JP_Sales',
'Other_Sales']].values

y = dff['Global_Sales'].values
```

Suggestions

- Create features that capture where a feature value lies relative to the members of a category it belongs to. In particular, calculate deviance of a row's feature value from the mean value of the category that row belongs to. This helps to capture information about a feature relative to the category's distribution.
- Create Paiplots and check if any polynomial features need to be added or not.
- Try performing Log tranformation on skewed variables.
- Handle rare categorical variables by putting them in other category or removing them.

Quality

This a fairly decent data and the most important part of proccessing has been done , so it can be fit to the model . If we had the sales data over more regions like Asia , Australia then that would've helped too.