

```
In [49]: greetings = "Assalam-o-Alaikum!"
print(greetings)
```

Assalam-o-Alaikum!

## Import Libraries

```
In [50]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## Import Dataset

```
In [51]: df = pd.read_csv("vgsales.csv")
df.head(10)
```

```
Out[51]:
```

	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
5	6	Tetris	GB	1989.0	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26
6	7	New Super Mario Bros.	DS	2006.0	Platform	Nintendo	11.38	9.23	6.50	2.90	30.01
7	8	Wii Play	Wii	2006.0	Misc	Nintendo	14.03	9.20	2.93	2.85	29.02
8	9	New Super Mario Bros. Wii	Wii	2009.0	Platform	Nintendo	14.59	7.06	4.70	2.26	28.62
9	10	Duck Hunt	NES	1984.0	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31

```
In [52]: 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
```

```
In [53]: df.isnull().sum()
```

```
Out[53]: 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 [54]: df.describe()
```

Out[54]:

	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 Analysis Section

1. Which platform has the highest global sales?

In [55]:

```
Highest_Sales = df.groupby("Platform")["Global_Sales"].agg("sum").to_frame().reset_index().sort_values("Global_Sales", ascending=False)
Highest_Sales = Highest_Sales.iloc[0]
print("Platform", "'"+Highest_Sales["Platform"]+"'", "has Highest Global Sales = $" + str(format(Highest_Sales["Global_Sales"], ".2f")))

Platform 'PS2' has Highest Global Sales = $1,255.64
```

2. What is the most popular genre in terms of global sales?

In [56]:

```
popular = df.groupby("Genre")["Global_Sales"].agg("sum").to_frame().reset_index().sort_values("Global_Sales", ascending=False)
popular = popular.iloc[0]
print("Genre", "'"+popular["Genre"]+"'", "have most popular in terms of global Sales = $" + str(format(popular["Global_Sales"], ".2f")))

Genre 'Action' have most popular in terms of global Sales = $1,751.18
```

In [57]:

```
df.head(1)
```

Out[57]:

	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

3. Which publisher has the highest total sales in North America?

In [58]:

```
Publisher_sale = df.groupby("Publisher")["NA_Sales"].agg("sum").to_frame().reset_index().sort_values("NA_Sales", ascending=False)
Publisher_sale = Publisher_sale.iloc[0]
print("Publisher", "'"+Publisher_sale["Publisher"]+"'", "has Highest Total sales in North America = $" + str(format(Publisher_sale["NA_Sales"], ".2f")))

Publisher 'Nintendo' has Highest Total sales in North America = $816.87
```

4. What is the average global sales for racing games?

In [59]:

```
Racing_games = df[df["Genre"] == "Racing"]
Racing_games
```

Out[59]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
11	12	Mario Kart DS	DS	2005.0	Racing	Nintendo	9.81	7.57	4.13	1.92	23.42
28	29	Gran Turismo 3: A-Spec	PS2	2001.0	Racing	Sony Computer Entertainment	6.85	5.09	1.87	1.16	14.98
42	43	Mario Kart 7	3DS	2011.0	Racing	Nintendo	4.74	3.91	2.67	0.89	12.21
47	48	Gran Turismo 4	PS2	2004.0	Racing	Sony Computer Entertainment	3.01	0.01	1.10	7.53	11.66
...	...	...	...	...	...	...	...	...	...	...	...
16530	16533	Sébastien Loeb Rally Evo	XOne	2016.0	Racing	Milestone S.r.l	0.00	0.01	0.00	0.00	0.01
16543	16546	Driving Simulator 2011	PC	2011.0	Racing	NaN	0.00	0.01	0.00	0.00	0.01
16573	16576	Mini Desktop Racing	Wii	2007.0	Racing	Popcorn Arcade	0.01	0.00	0.00	0.00	0.01
16574	16577	Yattaman Wii: BikkuriDokkiri Machine de Mou Ra...	Wii	2008.0	Racing	Takara Tomy	0.00	0.00	0.01	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01

1249 rows × 11 columns

In [60]:

```
AGSR = Racing_games["Global_Sales"].agg("mean").round(2)
print("The average global sales for racing games = $" + str(AGSR))
```

The average global sales for racing games = \$0.59

## 5. How many games were released in each year?

```
In [61]: Games_in_Year = df.groupby("Year")["Name"].count().to_frame().reset_index()
Games_in_Year["Year"] = Games_in_Year["Year"].astype(int)
Games_in_Year
```

```
Out[61]:
```

	Year	Name
0	1980	9
1	1981	46
2	1982	36
3	1983	17
4	1984	14
5	1985	14
6	1986	21
7	1987	16
8	1988	15
9	1989	17
10	1990	16
11	1991	41
12	1992	43
13	1993	60
14	1994	121
15	1995	219
16	1996	263
17	1997	289
18	1998	379
19	1999	338
20	2000	349
21	2001	482
22	2002	829
23	2003	775
24	2004	763
25	2005	941
26	2006	1008
27	2007	1202
28	2008	1428
29	2009	1431
30	2010	1259
31	2011	1139
32	2012	657
33	2013	546
34	2014	582
35	2015	614
36	2016	344
37	2017	3
38	2020	1

## 6. What is the total sales for all Nintendo games?

```
In [62]: df["Total Sales"] = df["NA_Sales"] + df["EU_Sales"] + df["JP_Sales"] + df["Other_Sales"] + df["Global_Sales"]
df.head(1)
```

```
Out[62]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	165.48

```
In [63]: df["Total Sales"] = df["NA_Sales"] + df["EU_Sales"] + df["JP_Sales"] + df["Other_Sales"] + df["Global_Sales"]
NIN_Games_Sales = df[df["Publisher"] == "Nintendo"]
NIN_Games_Sales = NIN_Games_Sales["Total Sales"].sum()
```

```
print("The total sales for all Nintendo games = $" + str(NIN_Games_Sales))
```

The total sales for all Nintendo games = \$3572.92

## 7. Which game has the highest sales in Japan?

```
In [64]: Game_sale = df.groupby("Name")["JP_Sales"].agg("sum").to_frame().reset_index().sort_values("JP_Sales", ascending=False)
Game_sale = Game_sale.iloc[0]
print("Game Category", "'"+Game_sale["Name"]+"'", "has Highest sales in japan = $" + str(format(Game_sale["JP_Sales"], ".2f")))
```

Game Category 'Pokemon Red/Pokemon Blue' has Highest sales in japan = \$10.22

## 8. What is the total sales for the top 5 games?

```
In [65]: df.head(1)
```

```
Out[65]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	165.48

```
In [66]: TSTG = df.groupby("Name")["Total Sales"].sum().to_frame().reset_index().sort_values("Total Sales", ascending=False)
TSTG
```

```
Out[66]:
```

	Name	Total Sales
11007	Wii Sports	165.48
3712	Grand Theft Auto V	111.84
9327	Super Mario Bros.	90.62
9715	Tetris	71.68
5573	Mario Kart Wii	71.65

## 9. How many games were published by Take-Two Interactive?

```
In [67]: PTTI = df[df["Publisher"] == "Take-Two Interactive"]
PTTIC = PTTI["Name"].count()
print("The games were published by Take-Two Interactive = " + str(PTTIC))
```

The games were published by Take-Two Interactive = 413

## 10. What is the total sales for role-playing games?

```
In [68]: RPG = df[df["Genre"] == "Role-Playing"]
TSRPG = RPG["Total Sales"].agg("sum").round(2)
print("the total sales for role-playing games = $" + str((TSRPG)))
```

the total sales for role-playing games = \$1854.63

```
In [69]: df.head(2)
```

```
Out[69]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	165.48
1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	80.48

# Data Visualization Section

## 1. Create a bar chart to compare the sales of the top 10 games in North America.

```
In [70]: top_games = df.groupby("Name")["NA_Sales"].agg("sum").to_frame().reset_index().sort_values("NA_Sales", ascending=False)
top_games
```

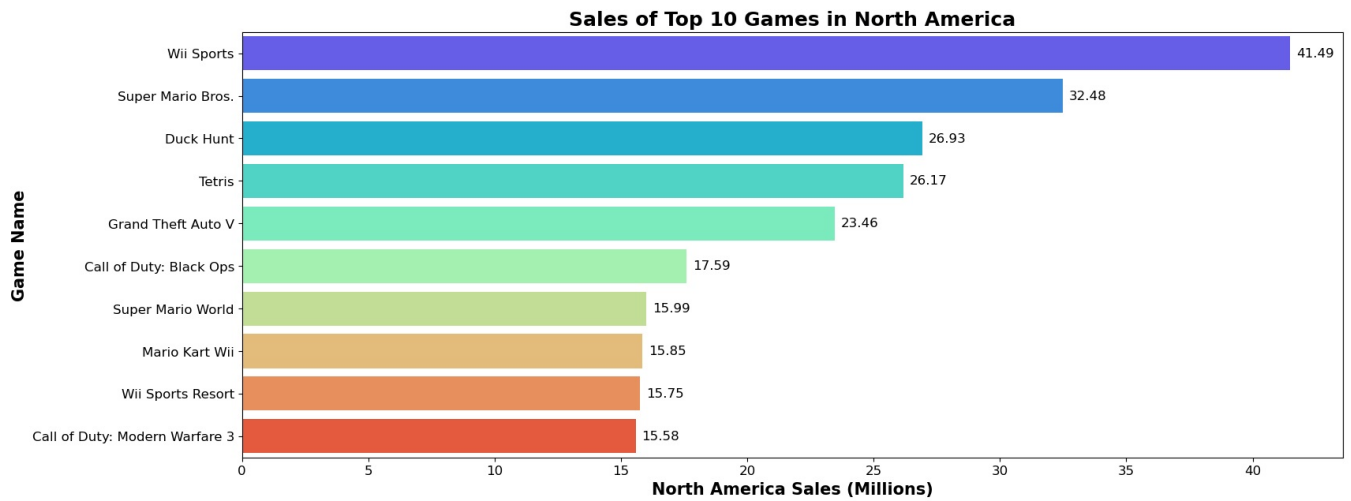
```
Out[70]:
```

	Name	NA_Sales
11007	Wii Sports	41.49
9327	Super Mario Bros.	32.48
2590	Duck Hunt	26.93
9715	Tetris	26.17
3712	Grand Theft Auto V	23.46
1222	Call of Duty: Black Ops	17.59
9341	Super Mario World	15.99
5573	Mario Kart Wii	15.85
11009	Wii Sports Resort	15.75
1228	Call of Duty: Modern Warfare 3	15.58

```
In [71]: plt.figure(figsize=(18, 7))
```

```
plt.figure(figsize=(10, 7))
graph = sns.barplot(y="Name", x="NA_Sales", data=top_games, palette="rainbow")
for p in graph.patches:
    width = p.get_width()
    plt.text(width + 0.25, p.get_y() + p.get_height() / 2, '{:.2f}'.format(width), ha='left', va='center', fontweight='bold')

plt.title("Sales of Top 10 Games in North America", weight="bold", size=18)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.xlabel("North America Sales (Millions)", weight="bold", size=15)
plt.ylabel("Game Name", weight="bold", size=15)
plt.show()
```

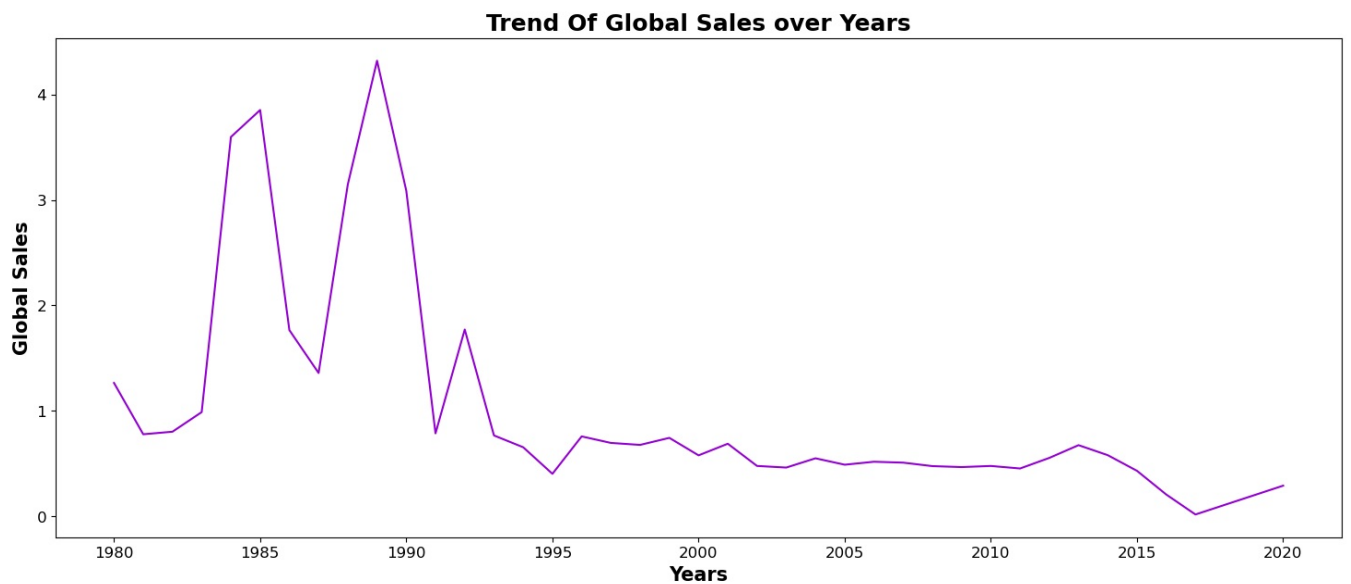


2. Plot a line graph showing the trend of global sales over the years.

```
In [72]: Dataframe = df[["Year", "Global_Sales"]]
Dataframe = Dataframe.dropna()
Dataframe.isnull().sum()
```

```
Out[72]: Year      0
Global_Sales  0
dtype: int64
```

```
In [73]: plt.figure(figsize=(18, 7))
sns.lineplot(x="Year", y="Global_Sales", data=Dataframe, color="darkviolet", errorbar=None)
plt.title("Trend Of Global Sales over Years", weight="bold", size=18)
plt.xticks(size=12)
plt.yticks(size=12)
plt.xlabel("Years", weight="bold", size=15)
plt.ylabel("Global Sales", weight="bold", size=15)
plt.show()
```



```
In [74]: df.head(1)
```

```
Out[74]:
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales
0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	165.48

3. Create a bar chart to visualize the distribution of sales across different genres.

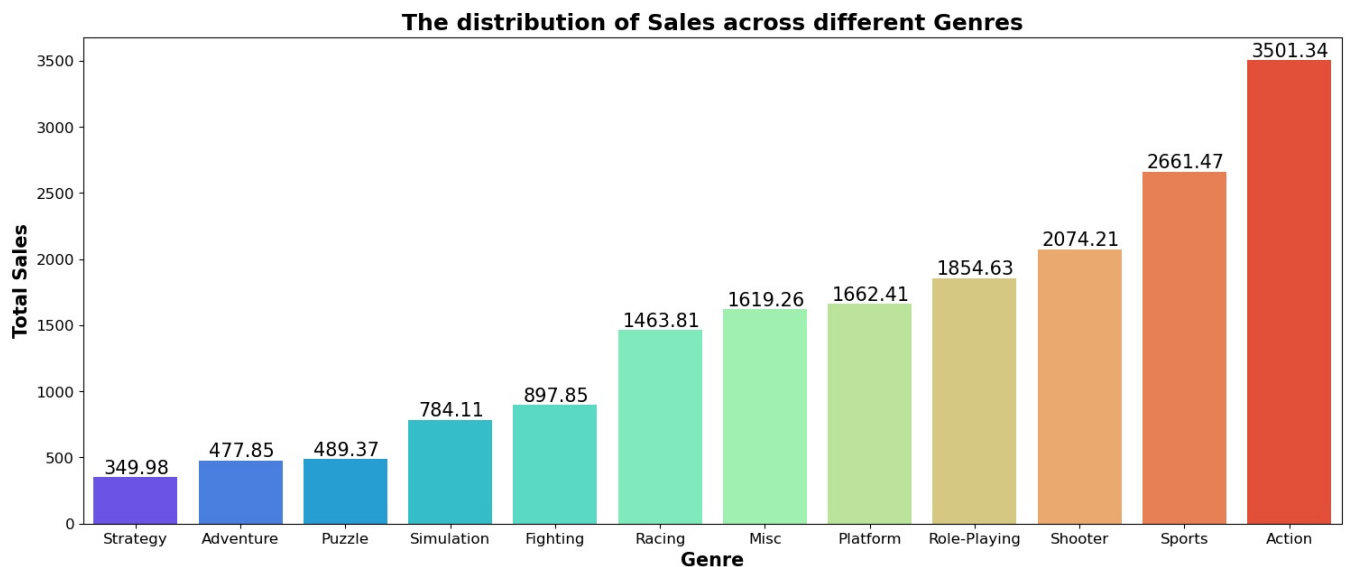
```
In [75]: visualization = df.groupby("Genre")["Total Sales"].agg("sum").to_frame().reset_index()
visualization
```

Out[75]:

	Genre	Total Sales
0	Action	3501.34
1	Adventure	477.85
2	Fighting	897.85
3	Misc	1619.26
4	Platform	1662.41
5	Puzzle	489.37
6	Racing	1463.81
7	Role-Playing	1854.63
8	Shooter	2074.21
9	Simulation	784.11
10	Sports	2661.47
11	Strategy	349.98

In [76]:

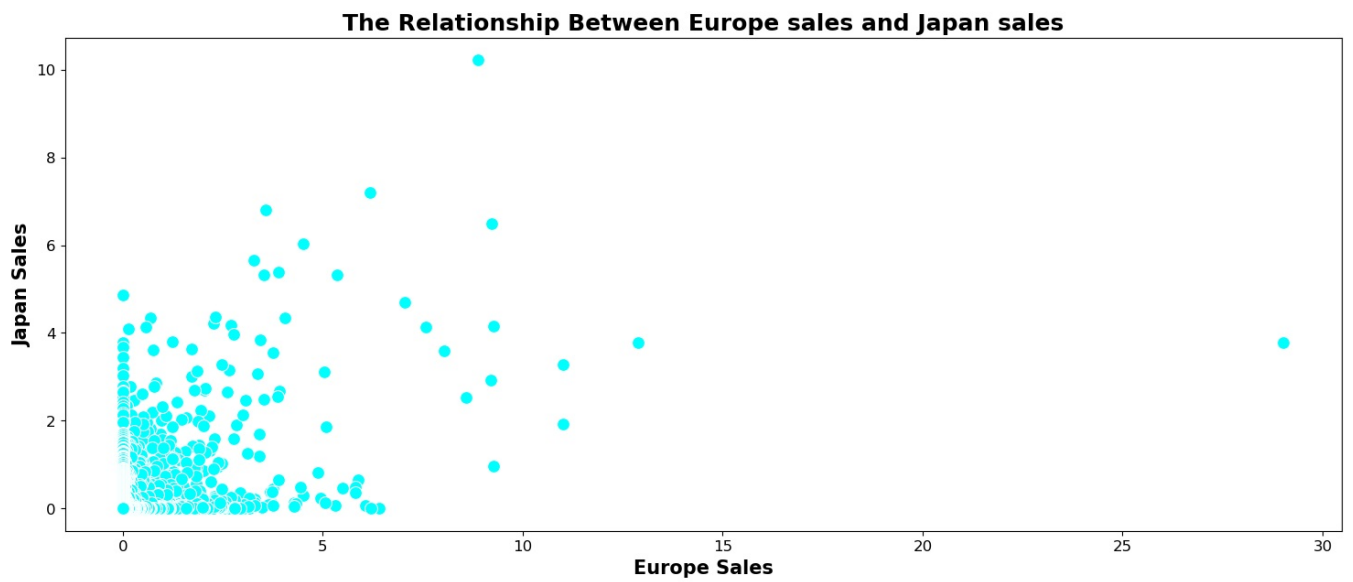
```
plt.figure(figsize =(18, 7))
graph = sns.barplot(x = "Genre", y = "Total Sales", data = visualization.sort_values("Total Sales"), palette =
for p in graph.patches:
    graph.annotate('{:.02f}'.format(p.get_height()),
                    (p.get_x()+0.41, p.get_height()),
                    ha='center', va='bottom',color= 'black', size = 15)
plt.title("The distribution of Sales across different Genres", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Genre", weight = "bold", size = 15)
plt.ylabel("Total Sales", weight = "bold", size = 15)
plt.show()
```



4. Generate a scatter plot to explore the relationship between EU sales and JP sales.

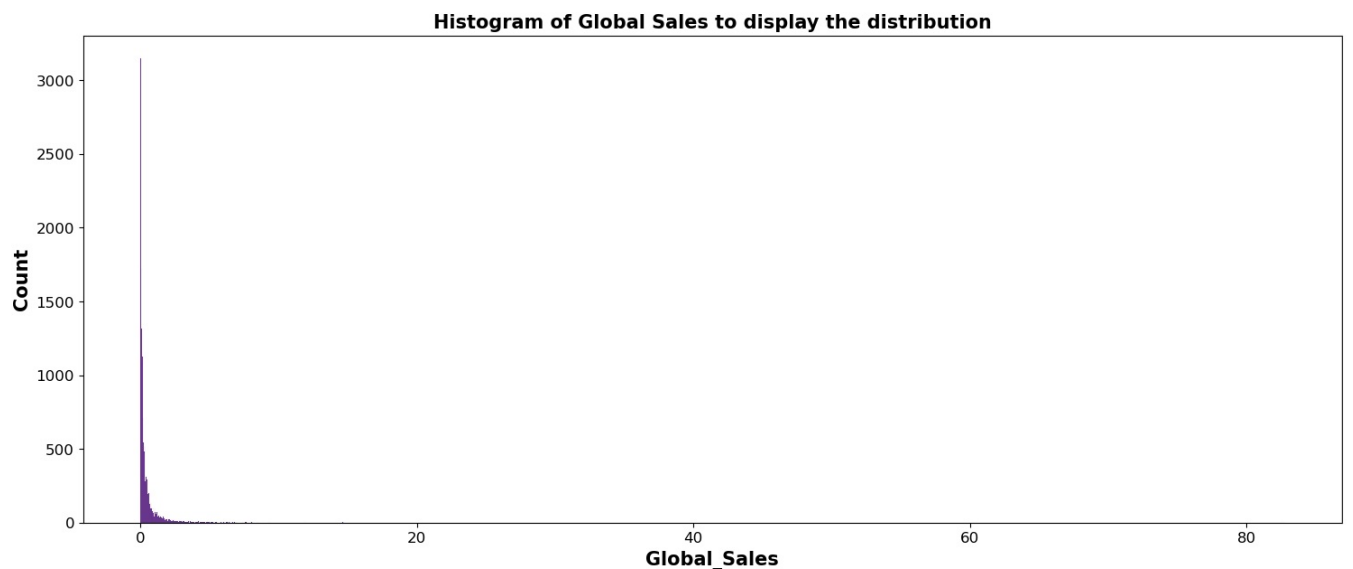
In [77]:

```
plt.figure(figsize =(18, 7))
sns.scatterplot(x = "EU_Sales", y = "JP_Sales", data = df, color = "aqua", s = 100)
plt.title("The Relationship Between Europe sales and Japan sales", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Europe Sales", weight = "bold", size = 15)
plt.ylabel("Japan Sales", weight = "bold", size = 15)
plt.show()
```



5. Plot a histogram to display the distribution of global sales.

```
In [78]: plt.figure(figsize= (18, 7))
sns.histplot(x ="Global_Sales", data = df, color = "indigo")
plt.title("Histogram of Global Sales to display the distribution", weight ="bold", size = 15 )
plt.xticks(size =12)
plt.yticks(size =12)
plt.xlabel("Global_Sales", weight ="bold", size =15)
plt.ylabel("Count", weight ="bold", size =15)
plt.show()
```



6. Create a stacked bar chart to compare the sales of Nintendo games across different platforms.

```
In [86]: NGDP = df[df["Publisher"] == "Nintendo"]
NGDP
```

Out[86]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales	
	0	1	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	165.48
	1	2	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	80.48
	2	3	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	71.65
	3	4	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00	66.00
	4	5	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	62.75
	...	...	...	...	...	...	...	...	...	...	...	...	...
16269	16272	Slide Adventure: Mag Kid	DS	2007.0	Action	Nintendo	0.00	0.00	0.01	0.00	0.01	0.01	0.02
16357	16360	Mario vs. Donkey Kong: Tipping Stars	WiiU	2015.0	Puzzle	Nintendo	0.00	0.00	0.01	0.00	0.01	0.01	0.02
16456	16459	Art Academy: Home Studio	WiiU	2015.0	Misc	Nintendo	0.00	0.00	0.01	0.00	0.01	0.01	0.02
16473	16476	Captain Rainbow	Wii	2008.0	Adventure	Nintendo	0.00	0.00	0.01	0.00	0.01	0.01	0.02
16542	16545	Mario & Luigi: Paper Jam & Mario Kart 7 Double...	3DS	2015.0	Misc	Nintendo	0.00	0.00	0.01	0.00	0.01	0.01	0.02

703 rows × 12 columns

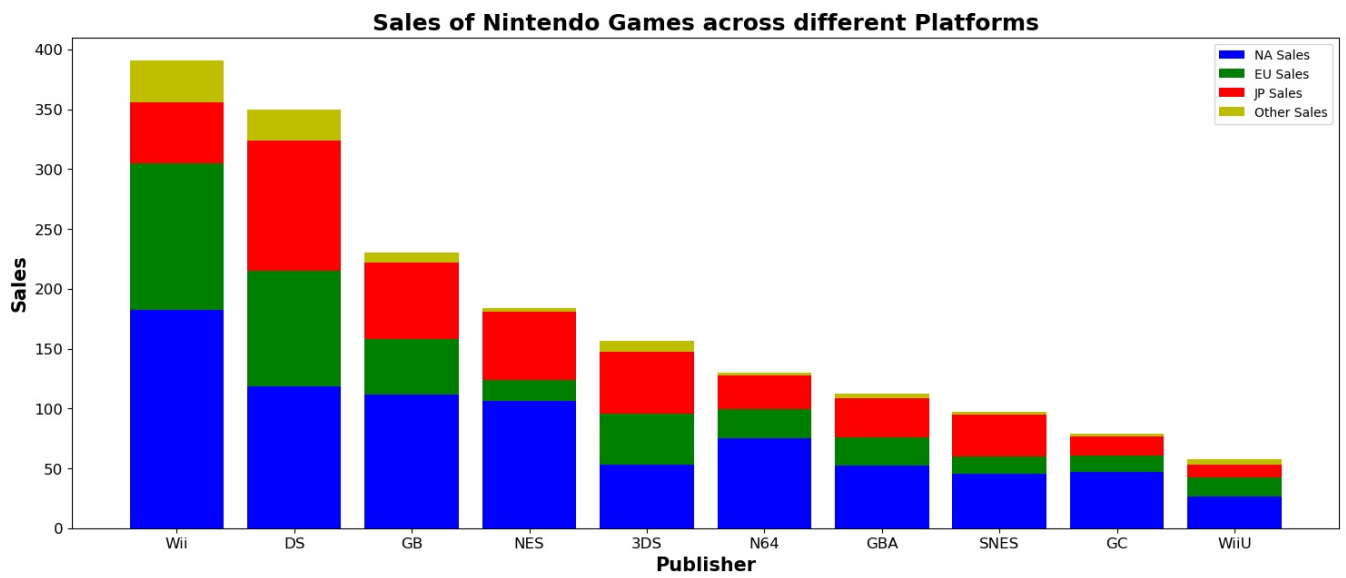
```
In [89]: NGDP = NGDP.groupby("Platform")[["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales", "Global_Sales", "Total Sales"]
NGDP
```

Out[89]:

	Platform	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Total Sales
8	Wii	182.02	123.01	51.04	34.40	390.46	780.93
1	DS	118.39	96.89	108.50	25.78	349.75	699.31
2	GB	111.66	46.59	63.83	8.00	230.09	460.17
6	NES	106.29	17.67	56.71	3.27	183.97	367.91
0	3DS	52.73	42.61	52.23	8.86	156.45	312.88
5	N64	75.30	24.14	27.92	2.56	129.95	259.87
3	GBA	52.65	22.94	32.94	3.50	112.00	224.03
7	SNES	45.14	15.06	34.44	2.18	96.84	193.66
4	GC	46.60	13.78	16.57	2.23	79.15	158.33
9	WiiU	26.09	16.05	11.24	4.55	57.90	115.83

```
In [90]: plt.figure(figsize =(18, 7))
plt.bar(NGDP['Platform'], NGDP['NA_Sales'], label='NA Sales', color='b')
plt.bar(NGDP['Platform'], NGDP['EU_Sales'], bottom=NGDP['NA_Sales'], label='EU Sales', color='g')
plt.bar(NGDP['Platform'], NGDP['JP_Sales'], bottom=NGDP['NA_Sales'] + NGDP['EU_Sales'], label='JP Sales', color='r')
plt.bar(NGDP['Platform'], NGDP['Other_Sales'], bottom=NGDP['NA_Sales'] + NGDP['EU_Sales'] + NGDP['JP_Sales'], label='Other Sales', color='m')
plt.title("Sales of Nintendo Games across different Platforms", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Publisher", weight = "bold", size = 15)
plt.ylabel("Sales", weight = "bold", size = 15)
plt.legend()
plt.show()
```





7. Generate a box plot to compare the sales of different genres.

```
In [95]: genre_df = df[["Genre", "NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales", "Global_Sales"]]
genre_df = genre_df.melt("Genre", var_name = "Area", value_name = "Sales")
genre_df
```

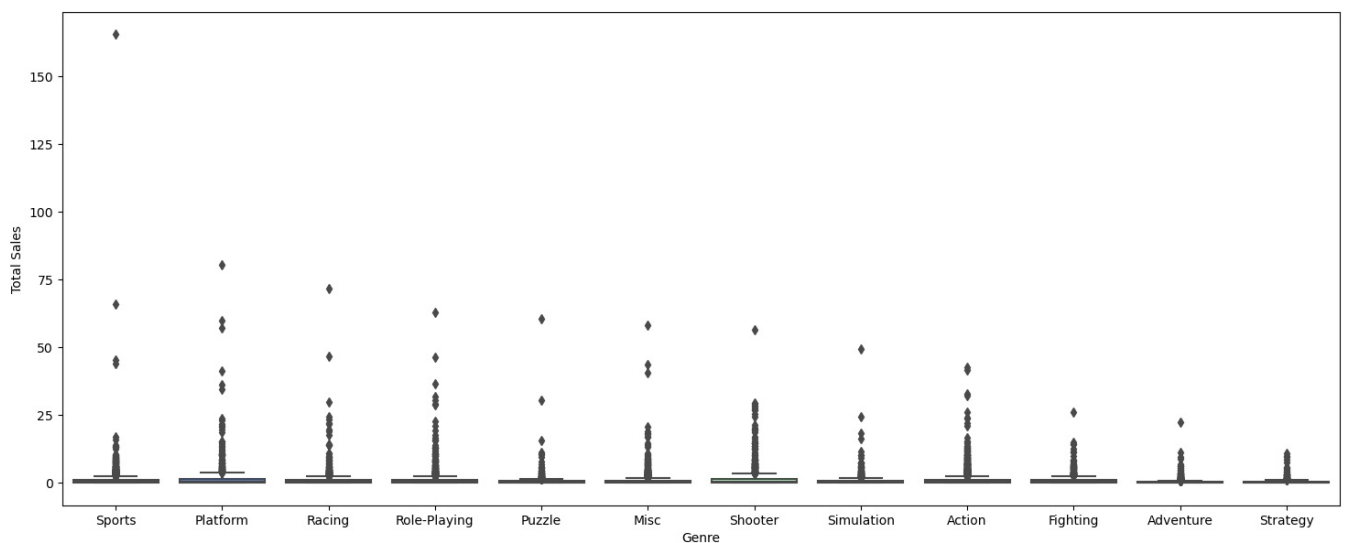
```
Out[95]:
```

	Genre	Area	Sales
0	Sports	NA_Sales	41.49
1	Platform	NA_Sales	29.08
2	Racing	NA_Sales	15.85
3	Sports	NA_Sales	15.75
4	Role-Playing	NA_Sales	11.27
...	...	...	...
82985	Platform	Global_Sales	0.01
82986	Shooter	Global_Sales	0.01
82987	Racing	Global_Sales	0.01
82988	Puzzle	Global_Sales	0.01
82989	Platform	Global_Sales	0.01

82990 rows × 3 columns

```
In [100]: plt.figure(figsize=(18, 7))
sns.boxplot(x = "Genre", y = "Total Sales", data = df, palette = "rainbow")
```

```
Out[100]: <Axes: xlabel='Genre', ylabel='Total Sales'>
```



8. Create a stacked area chart to visualize the cumulative sales over the years.

```
In [107]: null_year = df.dropna(subset=["Year"])
```

```
In [113]: commulative_area = null_year.groupby("Year")["Total Sales"].agg("sum").to_frame().reset_index()
```

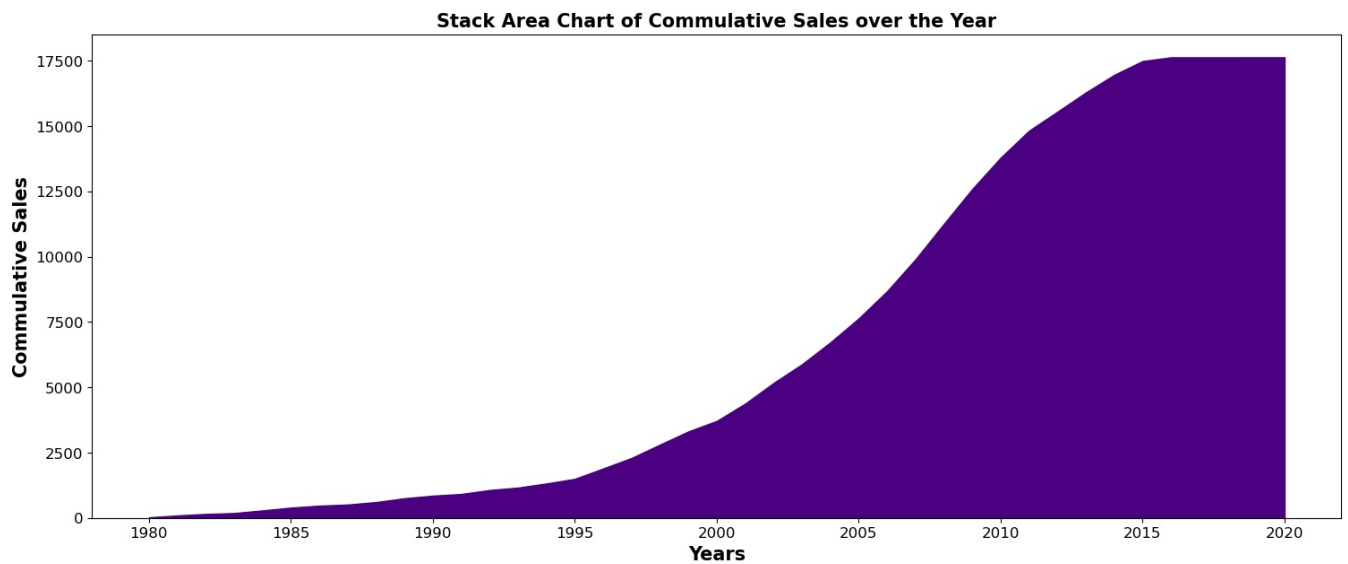
```
commulative_area["Commulative Sales"] = commulative_area["Total Sales"].cumsum()
commulative_area
```

Out[113]:

	Year	Total Sales	Commulative Sales
0	1980.0	22.76	22.76
1	1981.0	71.45	94.21
2	1982.0	57.74	151.95
3	1983.0	33.59	185.54
4	1984.0	100.71	286.25
5	1985.0	107.89	394.14
6	1986.0	74.15	468.29
7	1987.0	43.44	511.73
8	1988.0	94.43	606.16
9	1989.0	146.90	753.06
10	1990.0	98.76	851.82
11	1991.0	64.46	916.28
12	1992.0	152.30	1068.58
13	1993.0	91.97	1160.55
14	1994.0	158.39	1318.94
15	1995.0	176.22	1495.16
16	1996.0	398.30	1893.46
17	1997.0	402.05	2295.51
18	1998.0	512.80	2808.31
19	1999.0	502.39	3310.70
20	2000.0	403.19	3713.89
21	2001.0	662.96	4376.85
22	2002.0	790.49	5167.34
23	2003.0	715.46	5882.80
24	2004.0	838.16	6720.96
25	2005.0	919.35	7640.31
26	2006.0	1041.56	8681.87
27	2007.0	1221.57	9903.44
28	2008.0	1357.39	11260.83
29	2009.0	1334.40	12595.23
30	2010.0	1200.81	13796.04
31	2011.0	1031.92	14827.96
32	2012.0	726.84	15554.80
33	2013.0	736.09	16290.89
34	2014.0	674.15	16965.04
35	2015.0	528.70	17493.74
36	2016.0	141.80	17635.54
37	2017.0	0.10	17635.64
38	2020.0	0.58	17636.22

In [120]:

```
plt.figure(figsize=(18, 7))
plt.stackplot(commulative_area["Year"], commulative_area["Commulative Sales"], data = commulative_area, color =
plt.title("Stack Area Chart of Commulative Sales over the Year", weight = "bold", size = 15 )
plt.xticks(size =12)
plt.yticks(size =12)
plt.xlabel("Years", weight = "bold", size =15)
plt.ylabel("Commulative Sales", weight = "bold", size =15)
plt.show()
```



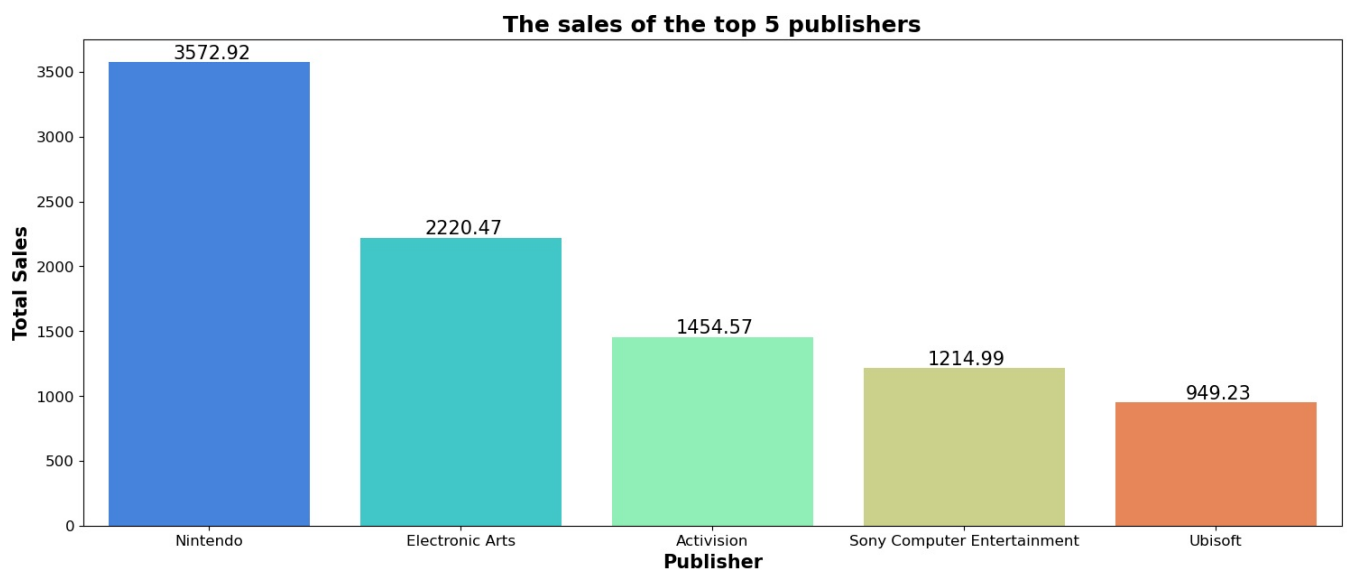
9. Plot a bar chart to compare the sales of the top 5 publishers.

```
In [127]: top_publishers = df.groupby("Publisher")["Total Sales"].agg("sum").to_frame().reset_index().sort_values("Total Sales").head(5)
```

```
Out[127]:
```

	Publisher	Total Sales
359	Nintendo	3572.92
138	Electronic Arts	2220.47
21	Activision	1454.57
456	Sony Computer Entertainment	1214.99
525	Ubisoft	949.23

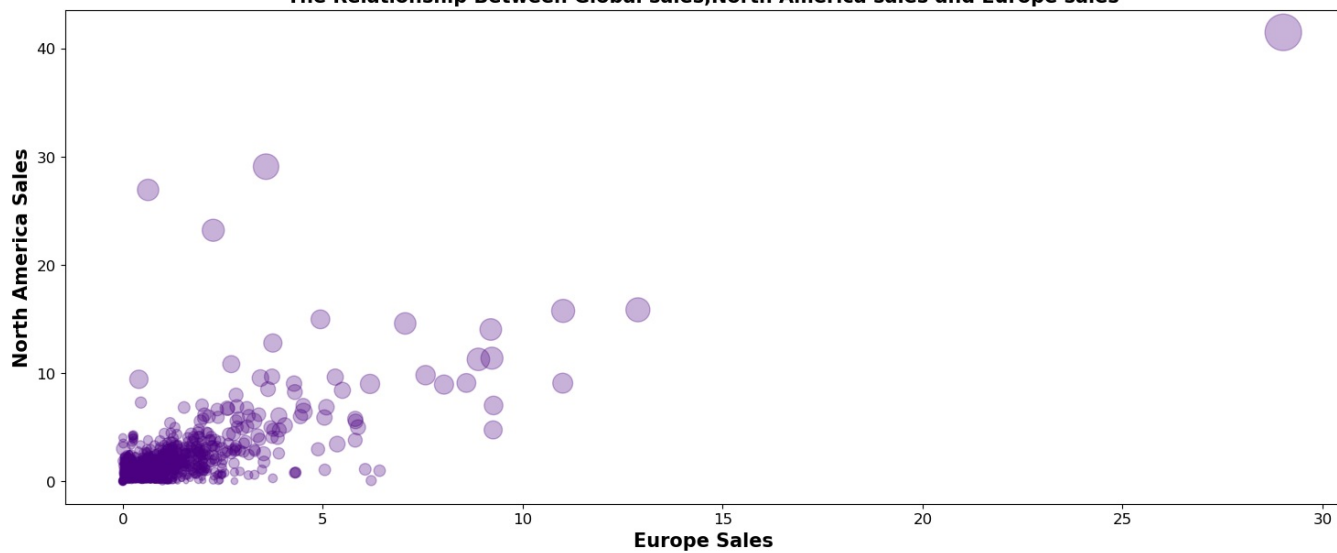
```
In [135]: plt.figure(figsize =(18, 7))
graph = sns.barplot(x = "Publisher", y = "Total Sales", data = top_publishers.head(5), palette = "rainbow")
for p in graph.patches:
    graph.annotate('{:.02f}'.format(p.get_height()),
                  (p.get_x()+0.41, p.get_height()),
                  ha='center', va='bottom',color= 'black', size = 15)
plt.title("The sales of the top 5 publishers", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Publisher", weight = "bold", size = 15)
plt.ylabel("Total Sales", weight = "bold", size = 15)
plt.show()
```



10. Generate a bubble chart to explore the relationship between global sales, NA sales, and EU sales.

```
In [140]: plt.figure(figsize =(18, 7))
plt.scatter(df["EU_Sales"], df["NA_Sales"], color = "indigo", s = df["Global_Sales"]*10, alpha = 0.3 )
plt.title("The Relationship Between Global sales, North America sales and Europe sales", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Europe Sales", weight = "bold", size = 15)
plt.ylabel("North America Sales", weight = "bold", size = 15)
plt.show()
```

The Relationship Between Global sales, North America sales and Europe sales



Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js