

## INRODUCTION TO MATPLOTLIB AND SEABORN

```
!pip install matplotlib
!pip install seaborn
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.45.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.23.5)
Requirement already satisfied: pandas>=0.25 in /usr/local/lib/python3.10/dist-packages (from seaborn) (1.5.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/local/lib/python3.10/dist-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (4.45.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (1.4.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib!=3.6.1,>=3.1) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.25->seaborn) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

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

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/original/final_vg1_-_final_vg_%281%29.csv?1670840
```

```
--2023-12-07 02:44:14-- https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/original/final_vg1_-_final_vg_%281%29.csv?1670840
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.238.59.211, 18.238.59.58, 18.238.59.71, ..
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.238.59.211|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2041483 (1.9M) [text/plain]
Saving to: 'final_vg1_-_final_vg_(1).csv?1670840'

final_vg1_-_final_v 100%[=====] 1.95M --.-KB/s in 0.06s

2023-12-07 02:44:14 (34.7 MB/s) - 'final_vg1_-_final_vg_(1).csv?1670840' saved [2041483/2041483]
```

```
data= pd.read_csv('final_vg1_-_final_vg_(1).csv?1670840')
data
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_S
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.00
1	9137	jShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.40
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.70
3	8359	.hack//G.U. Vol.1//Rebirth	PS2	2006.0	Role- Playing	Namco Bandai Games	2.031986	1.30
4	7109	.hack//G.U. Vol.2//Reminisce	PS2	2006.0	Role- Playing	Namco Bandai Games	2.792725	2.50
...	...	...	...	...	...	...	...	...
16647	7925	Zumba Fitness Rush	X360	2012.0	Sports	505 Games	4.409308	3.10
16648	6279	Zumba Fitness:	Wii	2013.0	Misc	Majesco	3.033887	2.70

```
data.to_csv('final_vg.csv',sep=",")
```

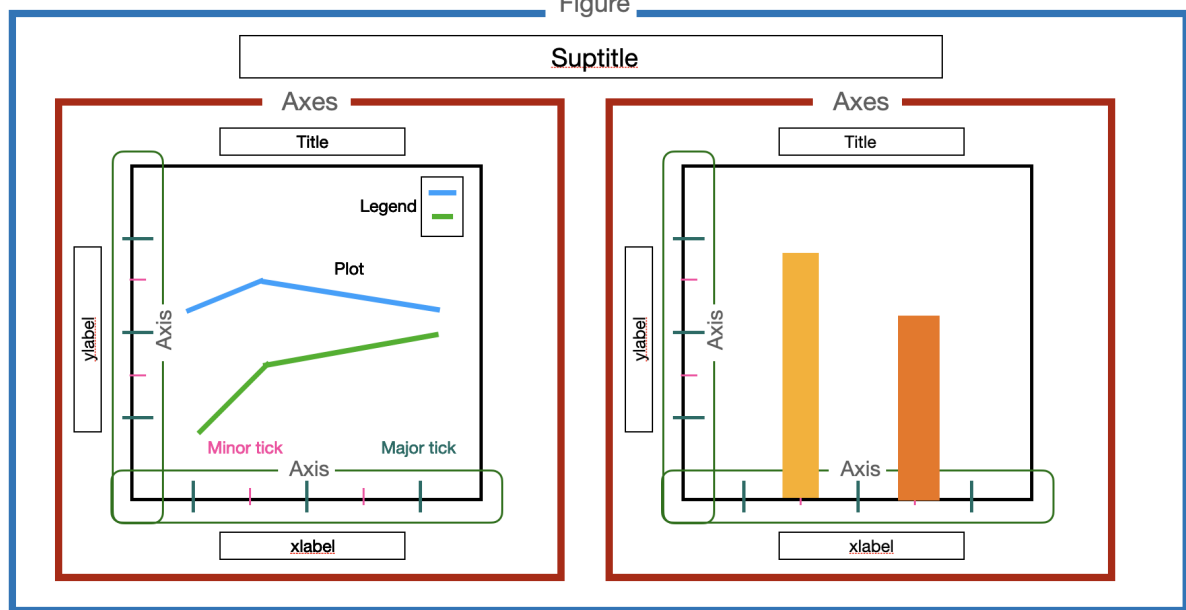
```
!gdown 15I3g3TBZvN6-WxLWmWf1i_h8oeT6gA7G
```

Downloading...  
From: [https://drive.google.com/uc?id=15I3g3TBZvN6-WxLWMwFi1\\_h8oeT6gA7G](https://drive.google.com/uc?id=15I3g3TBZvN6-WxLWMwFi1_h8oeT6gA7G)  
To: /content/final\_vg.csv  
100% 2.15M/2.15M [00:00<00:00, 57.9MB/s]

```
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.033887	3.439352	1.991671	12.802935
1	9137	iShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.493442	3.033887	0.394830	7.034163
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339	1.493442	0.408693	4.982552
3	14280	.hack: Significance of Two Memories	PS3	2012.0	Action	Namco	1.145709	1.762339	1.493442	0.408693	4.982552

Figure

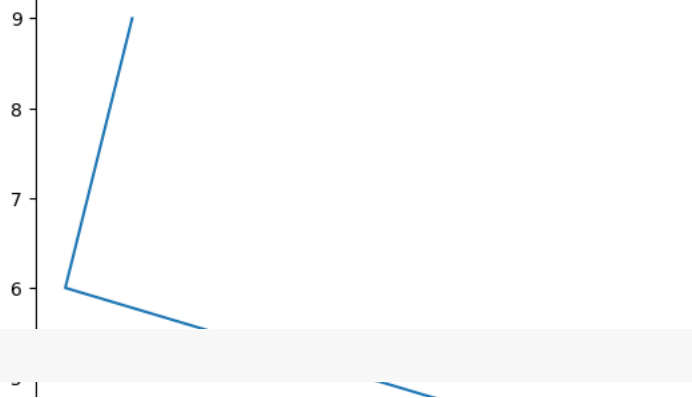


```
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.033887
1	9137	iShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.493442
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339
3	14280	.hack: Significance of Two Memories	PS3	2012.0	Action	Namco	1.145709	1.762339

```
x = [10,1,2]
y = [4,6,9]
plt.plot(x,y)
```

```
[<matplotlib.lines.Line2D at 0x7b920ee7f940>]
```



Find the top 5 genre of Video Games

Univariate Analysis - 1

- Distribution of the each category
- What proportion each category has on the total

```
data['Genre']
```

```
0      Shooter
1      Platform
2      Action
3      Role-Playing
4      Role-Playing
...
16647    Sports
16648    Misc
16649    Misc
16650    Role-Playing
16651    Action
Name: Genre, Length: 16652, dtype: object
```

```
cat_counts = data['Genre'].value_counts().sort_values(ascending=False)
cat_counts
```

```
Action      3316
Sports      2400
Misc        1739
Role-Playing 1488
Shooter      1310
Adventure    1286
Racing       1249
Platform      886
Simulation    867
Fighting      848
Strategy       681
Puzzle        582
Name: Genre, dtype: int64
```

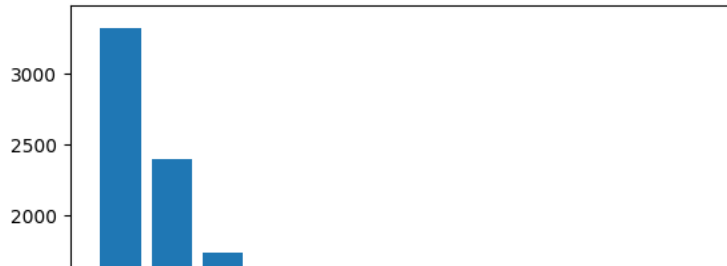
```
cat_counts.index
```

```
Index(['Action', 'Sports', 'Misc', 'Role-Playing', 'Shooter', 'Adventure',
      'Racing', 'Platform', 'Simulation', 'Fighting', 'Strategy', 'Puzzle'],
      dtype='object')
```

```
## Bar chart to visualise this
```

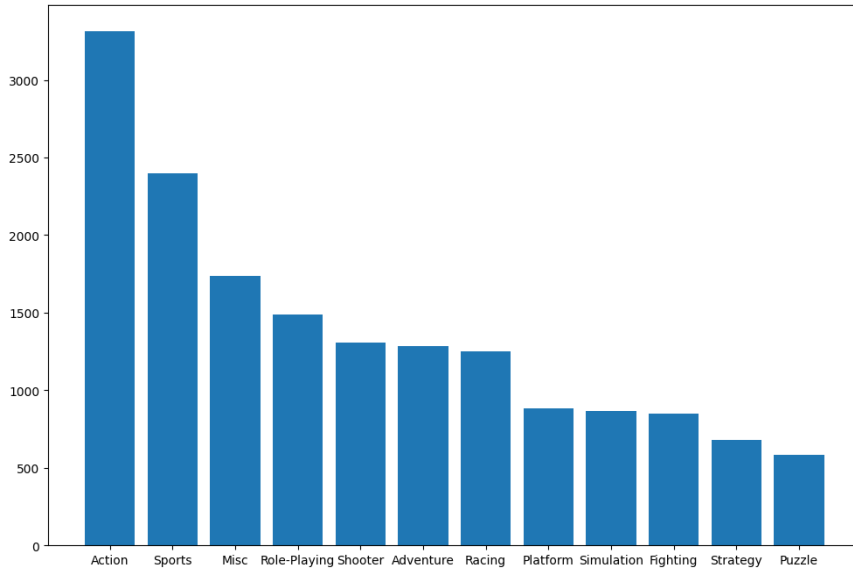
```
x_bar = cat_counts.index
y_bar = cat_counts
plt.bar(x_bar,y_bar)
```

<BarContainer object of 12 artists>



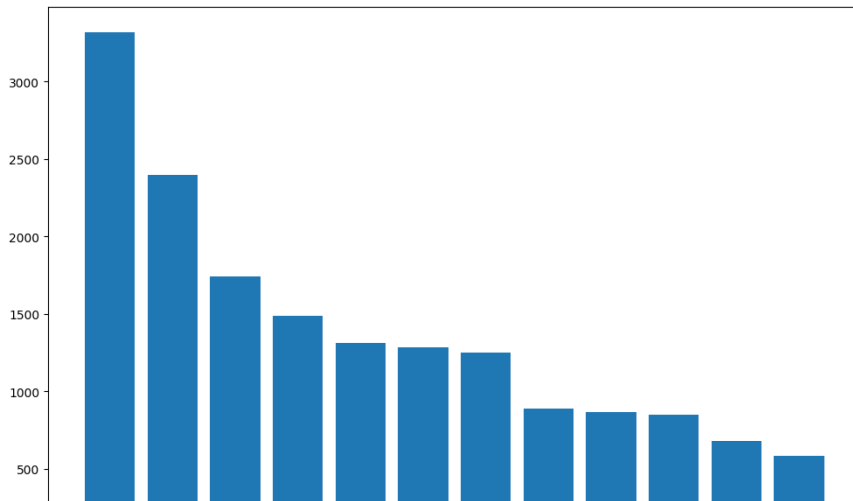
```
x_bar = cat_counts.index
y_bar = cat_counts
plt.figure(figsize=(12,8))
plt.bar(x_bar,y_bar)
```

<BarContainer object of 12 artists>



```
x_bar = cat_counts.index
y_bar = cat_counts
plt.figure(figsize=(12,8))
plt.bar(x_bar,y_bar)
plt.xticks(rotation=65,fontsize = 12)
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
 [Text(0, 0, 'Action'),
  Text(1, 0, 'Sports'),
  Text(2, 0, 'Misc'),
  Text(3, 0, 'Role-Playing'),
  Text(4, 0, 'Shooter'),
  Text(5, 0, 'Adventure'),
  Text(6, 0, 'Racing'),
  Text(7, 0, 'Platform'),
  Text(8, 0, 'Simulation'),
  Text(9, 0, 'Fighting'),
  Text(10, 0, 'Strategy'),
  Text(11, 0, 'Puzzle')])
```



```
x_bar = cat_counts.index
y_bar = cat_counts
plt.figure(figsize=(12,8))
plt.bar(x_bar,y_bar,width=0.5,color='green')
plt.xticks(rotation=65,fontsize = 12)
plt.title("Games per Genre", fontsize = 12)
plt.xlabel("Genre",fontsize=12)
plt.ylabel("Count",fontsize=12)
plt.show()
```

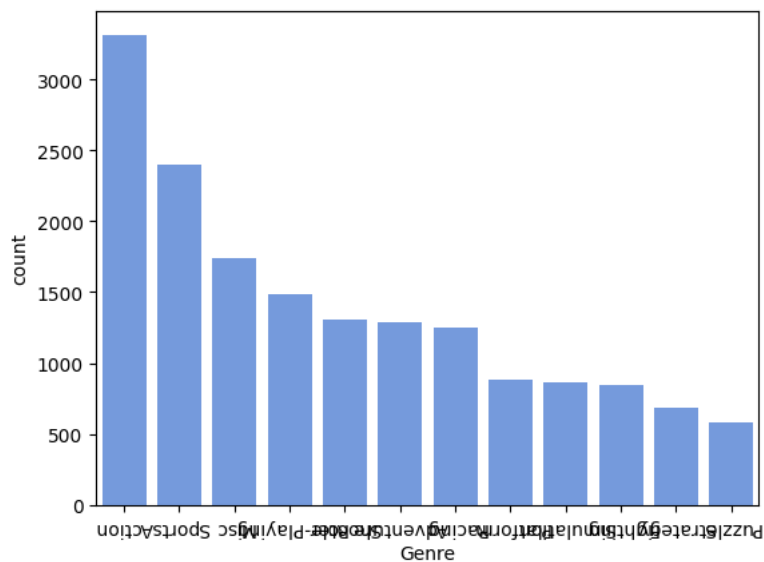
```

plt.bar?

plt.xlabel?

##Plot bargraph in Seaborn
sns.countplot(x='Genre',data=data,color='cornflowerblue',order=data['Genre'].value_counts().index,)
plt.xticks(rotation=180)
plt.show()

```



```

## Contribution to the total

Sales_data = data[["NA_Sales","EU_Sales","JP_Sales","Other_Sales"]]
Region_sales = Sales_data.sum(axis=0)

plt.pie(Region_sales,labels = ["NA_Sales","EU_Sales","JP_Sales","Other_Sales"])
addlabels(Region_sales)
plt.show()

```

```
NameError                                Traceback (most recent call last)
<ipython-input-26-436bbc0da49a> in <cell line: 8>()
6
```

```
Sales_data = data[["NA_Sales","EU_Sales","JP_Sales","Other_Sales"]]
```

```
plt.show()
```

```
plt.pie?
```

```
NameError: name 'addlabel' is not defined
```

```
Region_sales
```

SEARCH STACK OVERFLOW

```
x_bar = cat_counts.index
y_bar = cat_counts
plt.figure(figsize=(12,8))
plt.bar(y_bar,width=0.5,color='green')
plt.xticks(rotation=65,fontsize = 12)
plt.title("Games per Genre", fontsize = 12)
plt.xlabel("Genre",fontsize=12)
plt.ylabel("Count",fontsize=12)
plt.show()
```

```
data.head()
```

```
# Univariate Analysis – Numerical data
```

```
## How to identify the popularity of video games year by year
```

```
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.033887
1	9137	iShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.493442
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339
3	14280	Project X Zone 2	PS3	2012.0	Action	Namco	1.145709	1.762339

```
data['Year']
```

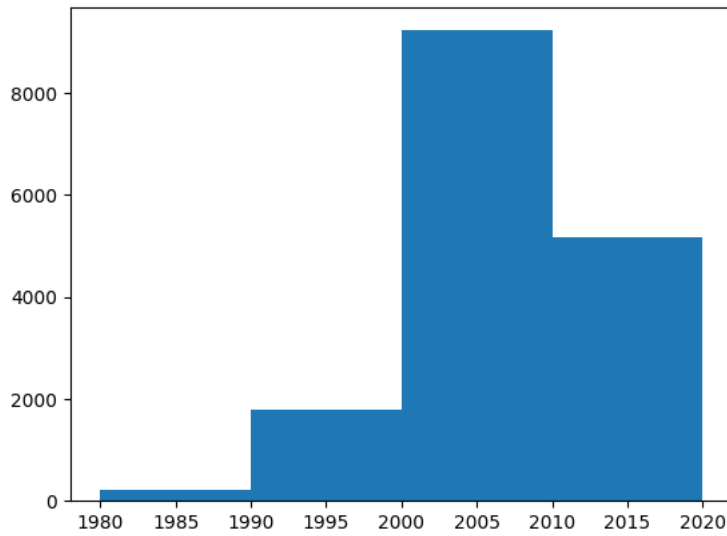
```
0      1985.0
1      2007.0
2      2012.0
3      2006.0
4      2006.0
...
16647   2012.0
16648   2013.0
16649   2013.0
16650   2008.0
16651   2013.0
Name: Year, Length: 16652, dtype: float64
```

```
plt.hist(data['Year'])
plt.show()
```

```

5000
## I want to analyse decade wise
plt.hist(data['Year'],bins=4)
plt.show()

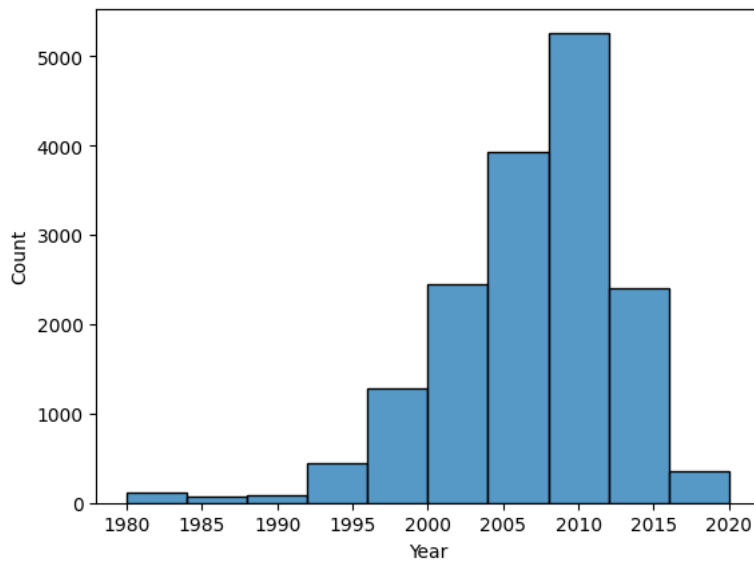
```



```
## Using Seaborn
```

```
sns.histplot(data['Year'],bins=10)
```

```
<Axes: xlabel='Year', ylabel='Count'>
```



```
sns.histplot(data['Year'])
```



```
<Axes: xlabel='Year', ylabel='Count'>
```



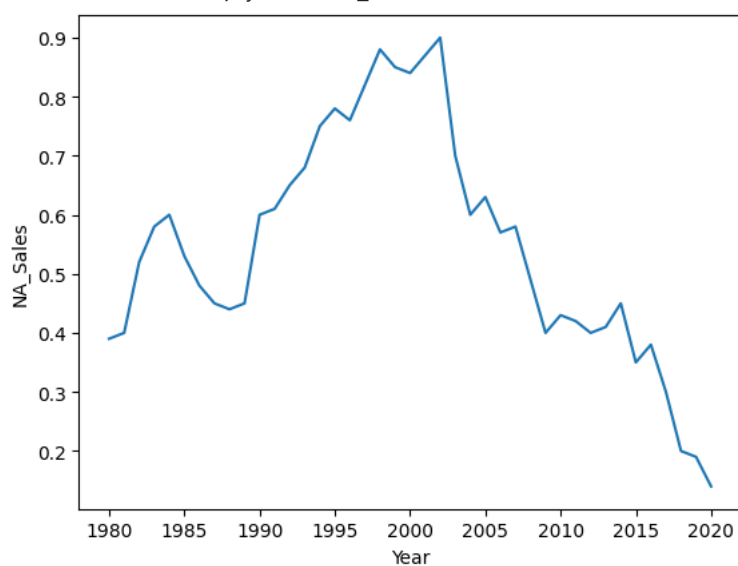
```
## Bivariate Analysis  
#2 types of data points
```

```
data.loc[data["Name"]=="Ice Hockey"]
```

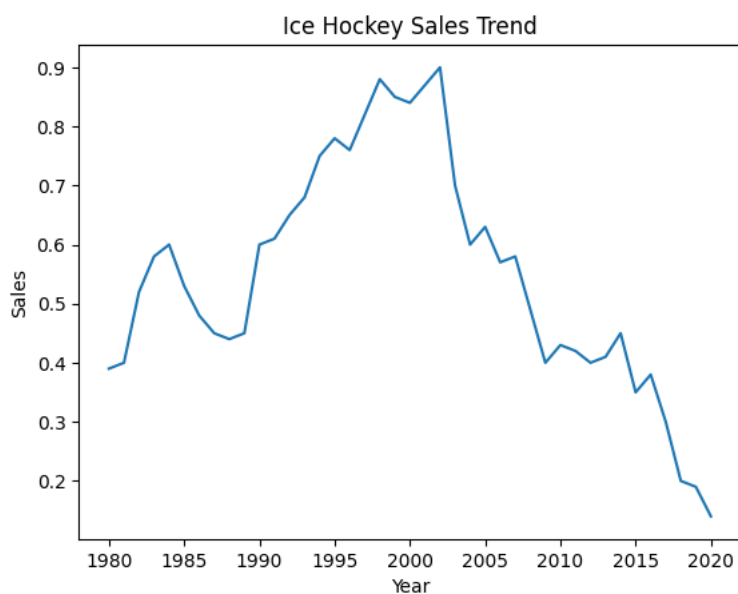
```
ih = data.loc[data["Name"]=="Ice Hockey"]
```

```
sns.lineplot(x='Year',y='NA_Sales',data=ih)
```

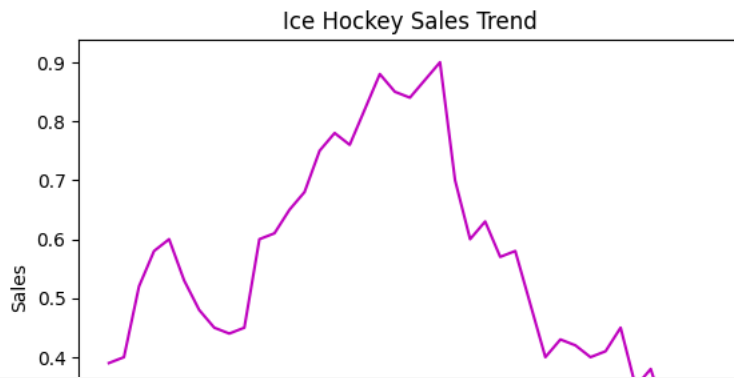
<Axes: xlabel='Year', ylabel='NA\_Sales'>



```
plt.title("Ice Hockey Sales Trend")
plt.xlabel("Year")
plt.ylabel("Sales")
ih = data.loc[data["Name"]=="Ice Hockey"]
sns.lineplot(x='Year',y='NA_Sales',data=ih)
plt.show()
```

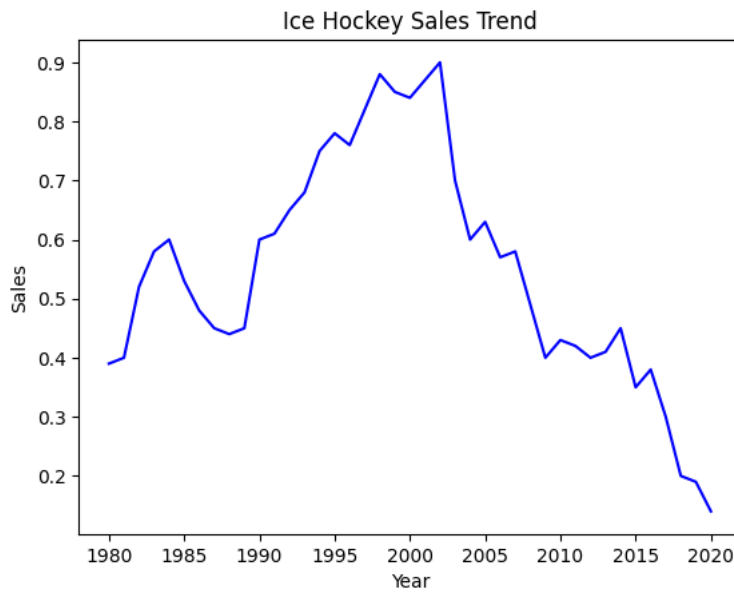


```
plt.title("Ice Hockey Sales Trend")
plt.xlabel("Year")
plt.ylabel("Sales")
ih = data.loc[data["Name"]=="Ice Hockey"]
sns.lineplot(x='Year',y='NA_Sales',data=ih,color="m")
plt.show()
```



	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_S
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.00
1	9137	iShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.40
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.70
3	8359	.hack//G.U. Vol.1//Rebirth	PS2	2006.0	Role- Playing	Namco Bandai Games	2.031986	1.30
4	7109	.hack//G.U. Vol.2//Reminisce	PS2	2006.0	Role- Playing	Namco Bandai Games	2.792725	2.50
...	...	...	...	...	...	...	...	...
16647	7925	Zumba Fitness Rush	X360	2012.0	Sports	505 Games	4.409308	3.10
16648	6279	Zumba Fitness:	Wii	2013.0	Misc	Majesco	3.033887	2.70

```
plt.title("Ice Hockey Sales Trend")
plt.xlabel("Year")
plt.ylabel("Sales")
ih = data.loc[data["Name"]=="Ice Hockey"]
sns.lineplot(x='Year',y='NA_Sales',data=ih,color="b")
plt.show()
```

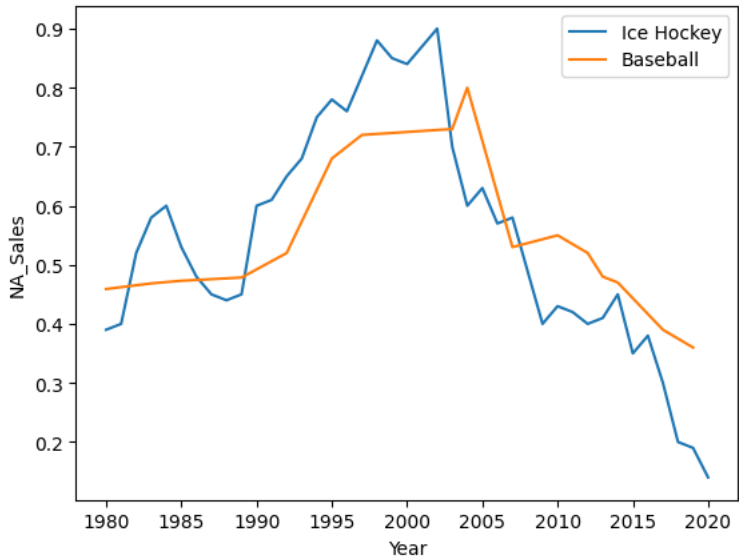


```
baseball = data.loc[data["Name"]=="Baseball"]
baseball
```

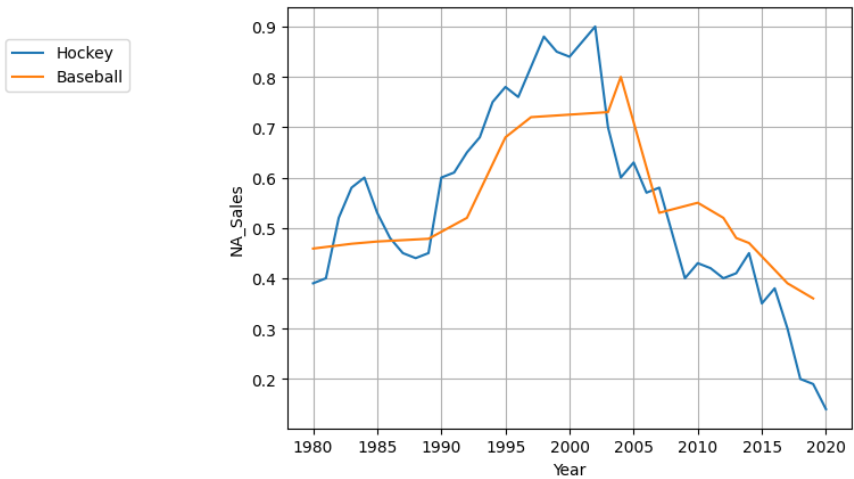
	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sa
941	324	Baseball	NES	1980.0	Sports	Nintendo	0.459000	2.320000	5.230
942	422	Baseball	NES	1983.0	Sports	Nintendo	0.468529	2.697415	5.854
943	231	Baseball	GB	1985.0	Sports	Nintendo	0.473000	3.074830	6.478
944	1144	Baseball	GB	1989.0	Sports	Nintendo	0.478448	3.452245	7.103
945	134	Baseball	GB	1992.0	Sports	Nintendo	0.520000	3.829660	7.727
946	134	Baseball	GB	1995.0	Sports	Nintendo	0.680000	3.829660	7.727
947	134	Baseball	PS	1997.0	Sports	Nintendo	0.720000	4.207075	8.352
948	2321	Baseball	Wii	2003.0	Sports	Nintendo	0.730000	4.584489	8.976
949	134	Baseball	GB	2004.0	Sports	Nintendo	0.800000	3.829660	7.727
950	1311	Baseball	Wii	2005.0	Sports	Nintendo	0.710000	4.961904	9.600
951	134	Baseball	X361	2007.0	Sports	Nintendo	0.530000	3.829660	7.727
952	1124	Baseball	X360	2010.0	Sports	Nintendo	0.550000	5.339319	10.225
953	2341	Baseball	X360	2012.0	Sports	Nintendo	0.520000	5.716734	10.849
954	134	Baseball	X362	2013.0	Sports	Nintendo	0.480000	3.829660	7.727

```
sns.lineplot(x='Year',y='NA_Sales',data=ih,label = "Ice Hockey")
sns.lineplot(x='Year',y='NA_Sales',data=baseball, label = "Baseball")
```

<Axes: xlabel='Year', ylabel='NA\_Sales'>



```
sns.lineplot(x='Year',y='NA_Sales',data=ih,label="Hockey")
sns.lineplot(x='Year',y='NA_Sales',data=baseball,label="Baseball")
plt.legend(loc=(-0.5,0.8))
plt.grid()
```



```
## Scatter Plot
plt.bar_label(bars, fmt='%d', label_type='edge', fontsize=10, color='blue') # Adjust formatting options as needed

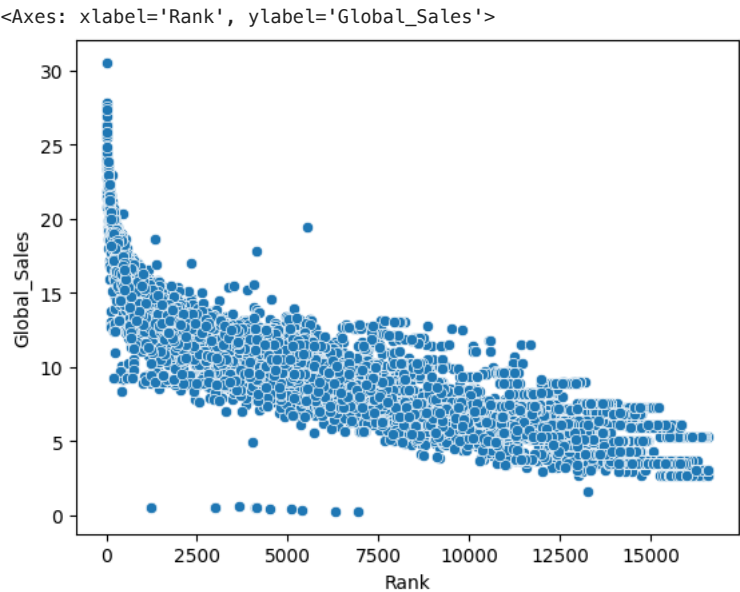
#Suppose you want to find the relation between 2 data
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-84-a23f89e528fd> in <cell line: 2>()
      1 ## Scatter Plot
----> 2 plt.bar_label(bars, fmt='%d', label_type='edge', fontsize=10,
      3 color='blue') # Adjust formatting options as needed
      4 #Suppose you want to find the relation between 2 data

NameError: name 'bars' is not defined
```

SEARCH STACK OVERFLOW

```
sns.scatterplot(data,x='Rank',y='Global_Sales')
```

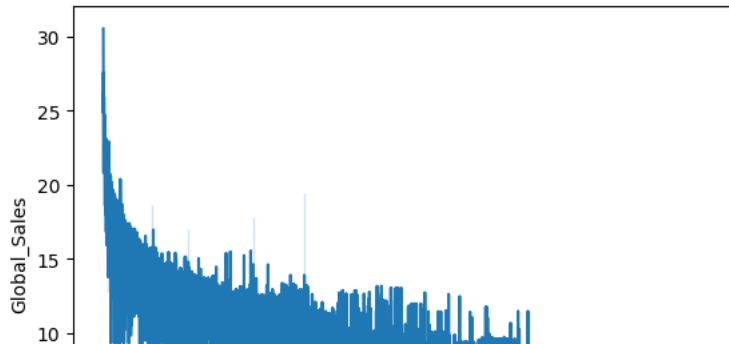


data

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_S
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.00
1	9137	jShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.40
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.70
3	8359	.hack//G.U. Vol.1//Rebirth	PS2	2006.0	Role- Playing	Namco Bandai Games	2.031986	1.30
4	7109	.hack//G.U. Vol.2//Reminisce	PS2	2006.0	Role- Playing	Namco Bandai Games	2.792725	2.50
...	...	...	...	...	...	...	...	...
16647	7925	Zumba Fitness Rush	X360	2012.0	Sports	505 Games	4.409308	3.10
16648	6279	Zumba Fitness:	Wii	2013.0	Misc	Majesco	3.033887	2.70

```
sns.lineplot(data,x='Rank',y='Global_Sales')
```

<Axes: xlabel='Rank', ylabel='Global\_Sales'>



```
## Categorical - Categorical data
```

```
data.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.033887
1	9137	iShin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.493442
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339
3	15000	15000	DS	2009.0	Platform	Namco	1.145709	1.762339

```
top3_Pub = data['Publisher'].value_counts().index[:3]
top3_Gen = data['Genre'].value_counts().index[:3]
top3_plat = data['Platform'].value_counts().index[:3]
```

```
top3_Pub
```

```
Index(['Electronic Arts', 'Activision', 'Namco Bandai Games'], dtype='object')
```

```
top3_Gen
```

```
Action    3316
Sports    2400
Misc      1739
Name: Genre, dtype: int64
```

```
top3_plat
```

```
DS      2163
PS2     2161
PS3     1330
Name: Platform, dtype: int64
```

```
top3_dataset = data.loc[(data["Publisher"].isin(top3_Pub)) & (data["Platform"].isin(top3_plat)) & (data["Genre"].isin(top3_Gen))]
```

```
top3_dataset.head()
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_S
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339	1.493442
13	2742	[Prototype 2]	PS3	2012.0	Action	Activision	3.978349	3.727034	0.843750
16	1604	[Prototype]	PS3	2009.0	Action	Activision	4.569217	4.108402	1.145709

```
# Dodged bar chart
```

```
sns.countplot(x = 'Publisher',data=data,hue='Platform')
plt.ylabel('Count')
```

```

KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-122-5b3e48b85e82> in <cell line: 1>()
----> 1 sns.countplot(x = 'Publisher',data=data,hue='Platform')
      2 plt.ylabel('Count')

```

⏮ 7 frames

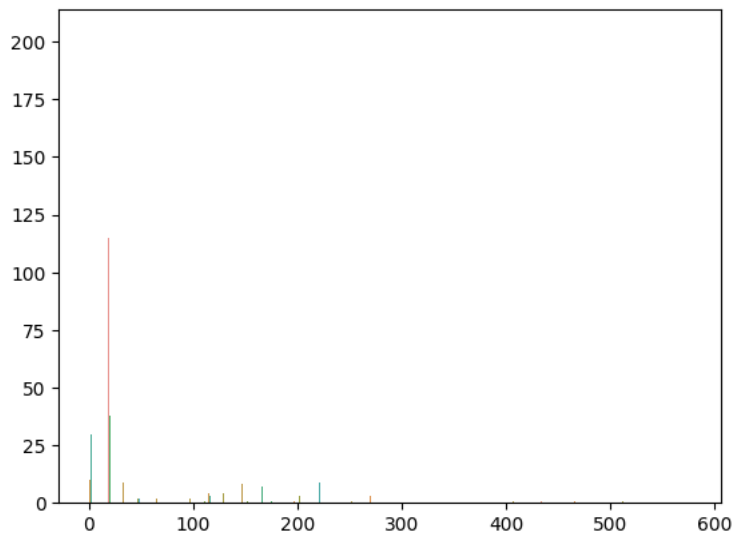
```

/usr/local/lib/python3.10/dist-packages/matplotlib/Bezier.py in
__call__(self, t)
    198     self._px = (self._cpoints.T * coeff).T
    199
--> 200     def __call__(self, t):
    201         """
    202         Evaluate the Bézier curve at point(s) *t* in [0, 1].

```

KeyboardInterrupt:

SEARCH STACK OVERFLOW



<ipython-input-133-55fe13e52a85>:1: FutureWarning: The default value of numer.

