Colab: https://colab.research.google.com/drive/1_8kAoExECvH09c_A5ApycfS9CJ3siLO2?usp=sharing

```
import numpy as np
import pandas as pd
```

```
# Matplotlib
# Seaborn
```

```
# 1. Exploratory - EDA, looking for patterns, analysing the data
# 2. Explanatory - Storytelling, Dashboarding
```

```
# Science - anatomy of plot, choosing the right plot
# Art - right scale, labels, axis ticks, remove clutter, highlight some information
```

```
import matplotlib.pyplot as plt
import seaborn as sns
# matplotlib+pandas
# why not plotly - is creates dynamic plots
# more code to write
# more difficult to grasp for a beginner
# not used a lot in Industry
# M+S, Tableau
```

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/origin
```

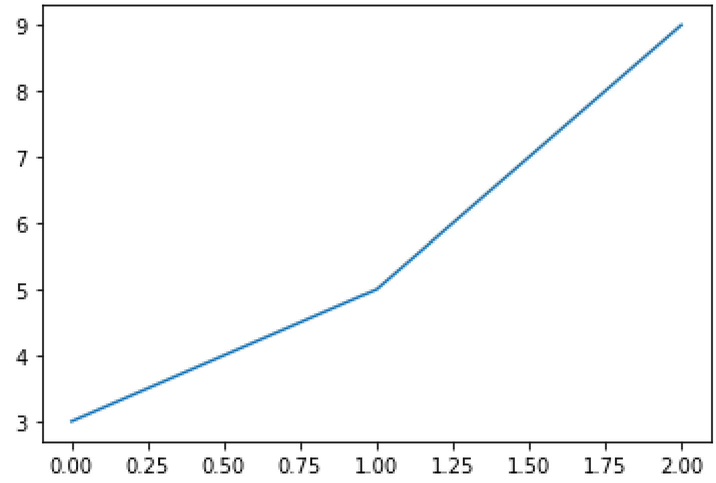
```
--2022-12-14 15:48:12--  https://d2beiqkhq929f0.cloudfront.net/public_assets/a
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99
HTTP request sent, awaiting response... 200 OK
Length: 2041483 (1.9M) [text/plain]
Saving to: ‘final_vg.csv’

final_vg.csv        100%[===================>]   1.95M  --.-KB/s    in 0.08s

2022-12-14 15:48:12 (23.6 MB/s) - ‘final_vg.csv’ saved [2041483/2041483]
```

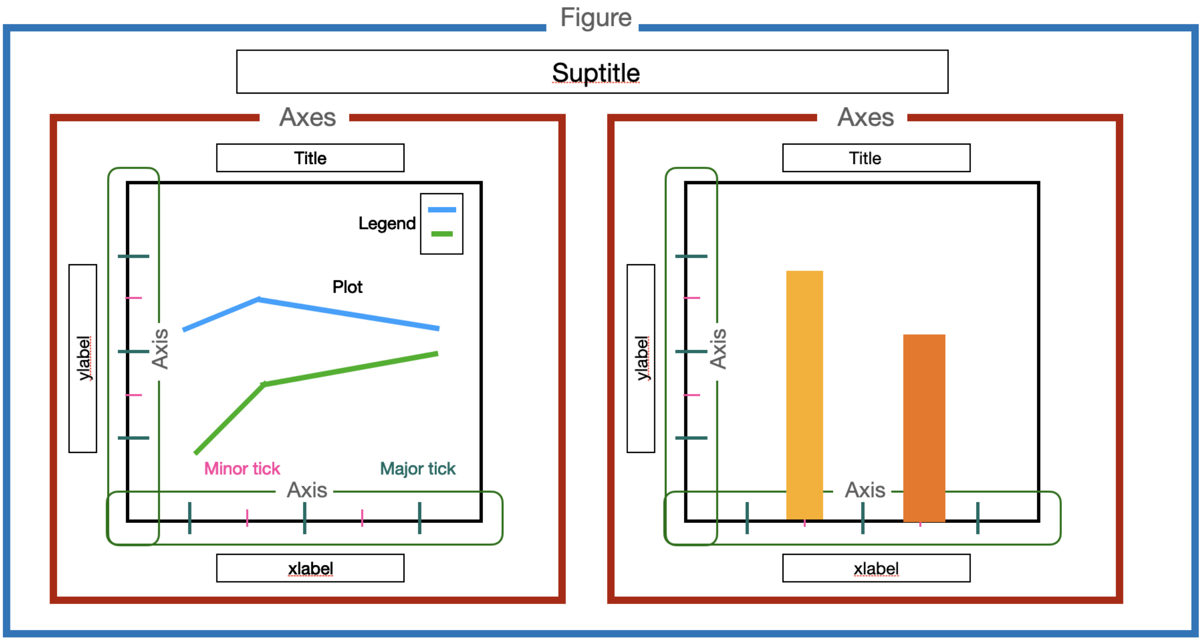
```
data = pd.read_csv('final_vg.csv')
data.head()
```

```
    Rank                         Name  Platform   Year      Genre         Publi
```

```
# (0, 3), (1, 5), (2, 9)
x_val = [0, 1, 2]
y_val = [3, 5, 9]
plt.plot(x_val, y_val)
```

```
[<matplotlib.lines.Line2D at 0x7f027e7c54f0>]
```





```
# Jupyter Cell - shown after the code
# Terminal - figure will be displayed as a sep window
# IDE - Seperate very small window will pop in the IDE itself
```

```
# Choosing is the right plot?
# Number of variables involved in answering a question
# Q1- How many variables of interest are involved?
# Q2 - Whether these variables are numerical or categorical?
# How many variables of interest are involved?
# 1 Variable - Univariate Data Visualisation
# 2 Variable - Bi-variate Data Visualisation
# 2+ Variables - Multivariate Data Visualisation
```

```
# Univariate
  # Numercical
  # Categorical
# Bivariate
  # Num, Num
  # Num, Cat
  # Cat, Cat
# Multivariate - n-dimensional, 3D
 # Num, Num, Cat
 # Cat, Cat, Num
 # Cat, Cat, Cat
 # Num, Num, Num
# Subplots
```

```
# Categorical - count of each categories, share/fraction component of each category
```

```
# How can you find the top-N Genres?
```

```
data["Genre"].value_counts()
# whenever you see a cat variable, start thinking about placing some bars
```

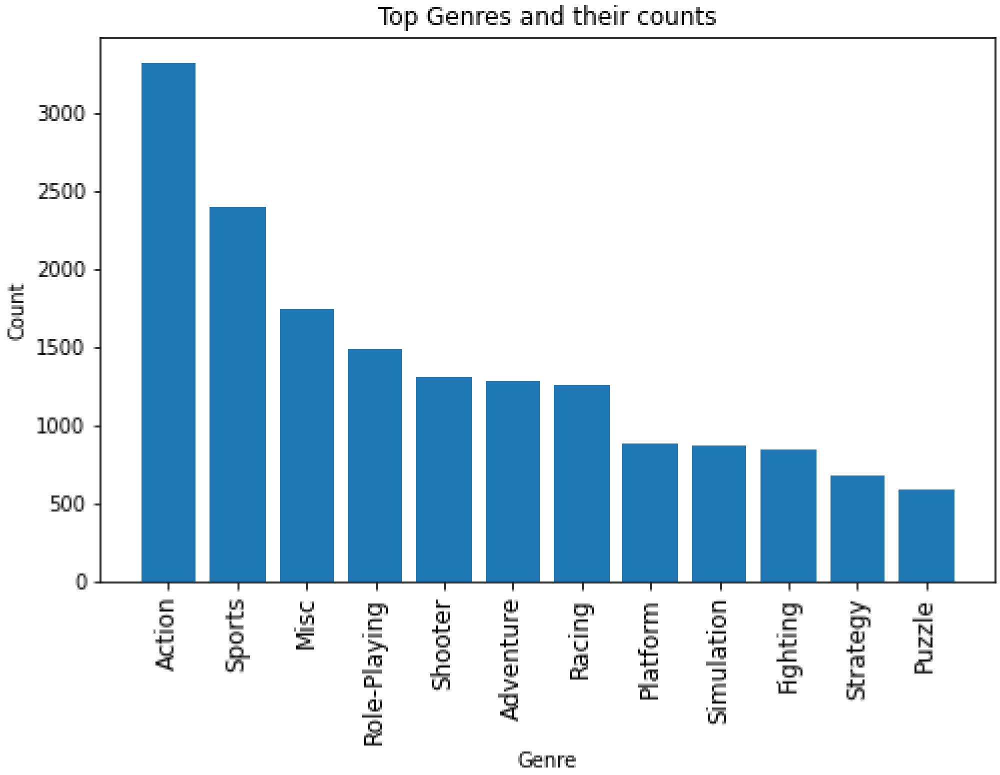
```
Action          3316
Sports          2400
Misc            1739
Role-Playing    1488
Shooter         1310
Adventure       1286
Racing          1249
Platform         886
Simulation       867
Fighting         848
Strategy         681
Puzzle           582
Name: Genre, dtype: int64
```

```
x_val = data["Genre"].value_counts().index
y_val = data["Genre"].value_counts().to_list()
plt.bar(x_val, y_val)
```

```
<BarContainer object of 12 artists>
```

```
plt.figure(figsize=(8,5))
plt.bar(x_val, y_val) # sns.____plot()
plt.xticks(rotation=90, fontsize=12)
plt.xlabel("Genre")
plt.ylabel("Count")
plt.title("Top Genres and their counts",  fontsize=12)
```

```
Text(0.5, 1.0, 'Top Genres and their counts')
```

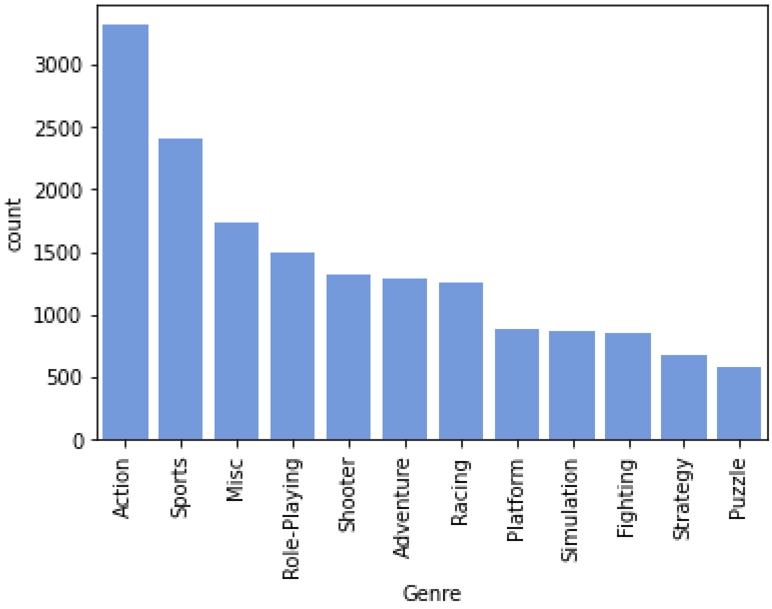


```
plt.figure(figsize=(8,5))
plt.bar(x_val, y_val, width=0.2, color="orange") # sns.____plot()
plt.xticks(rotation=90, fontsize=12)
```

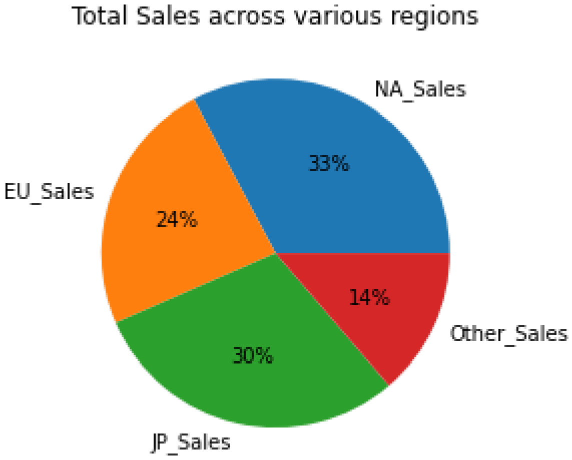
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11],
 <a list of 12 Text major ticklabel objects>)
```

3000
2500
2000

```
sns.countplot(data=data,
              x="Genre",
              order=data["Genre"].value_counts().index,
              color="cornflowerblue")
plt.xticks(rotation=90)
plt.show() # telling Python, that, hey now you should display all of the stuff
```



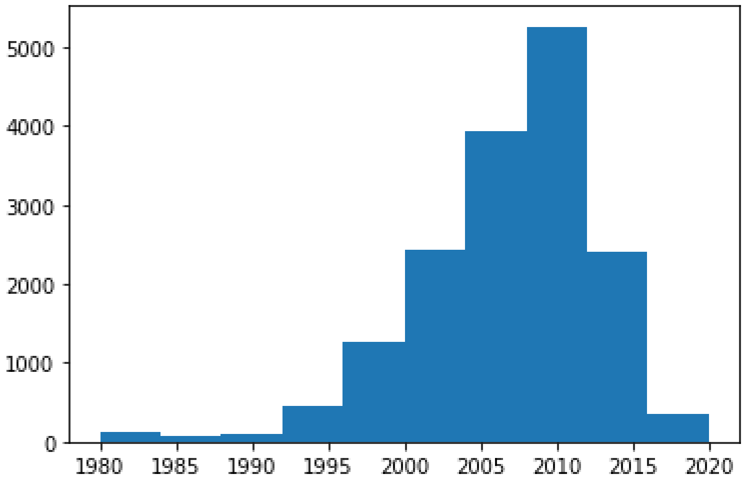
```
# Pie chart - it is not very well received by scientific, seaborn doesn't piechart

# piecharts in matplotlib, verbose - post-read
```



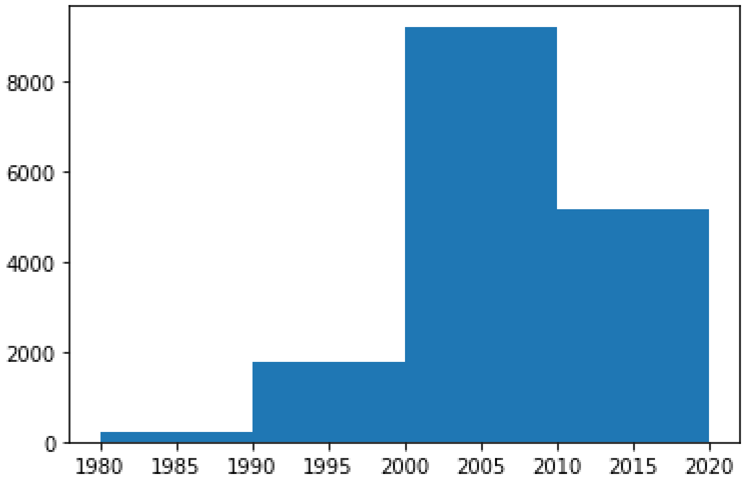
```
# Univariate - Numerical
```

```
# popularity of video games in general year-by-year? --> distribution of games publ
```

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

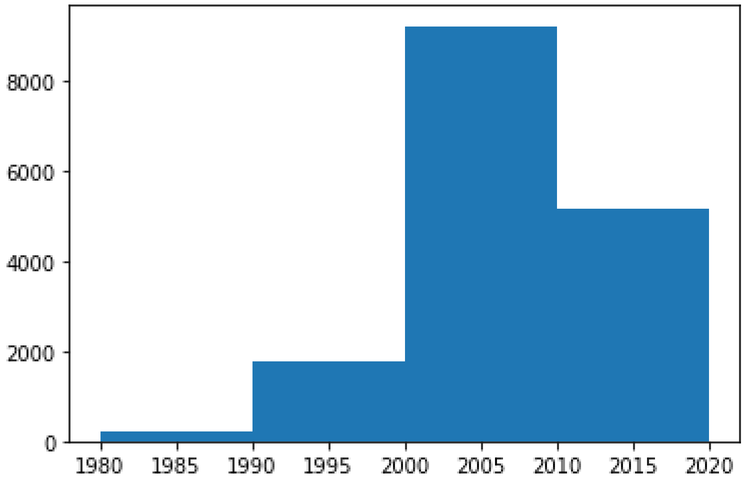


```
plt.hist(data["Year"], bins=4)
plt.show()
```



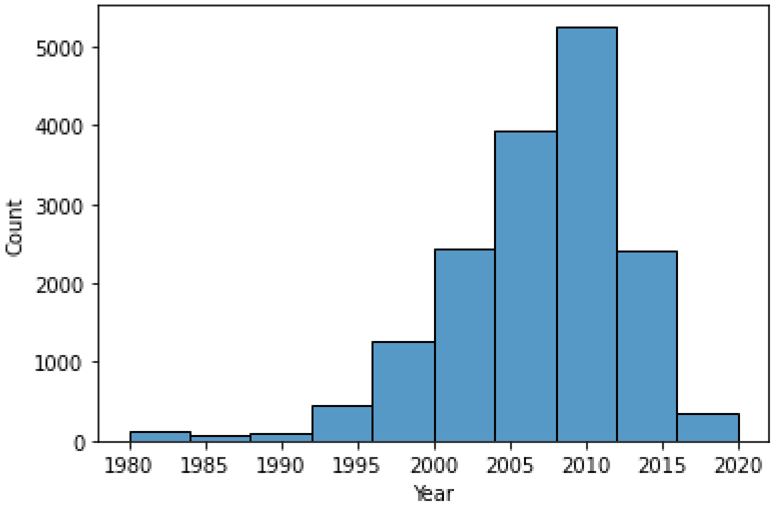
```
plt.hist(data["Year"], bins=20)
plt.show()
```

```
count, bins, _ = plt.hist(data["Year"], bins=4)
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f027ad30370>
```

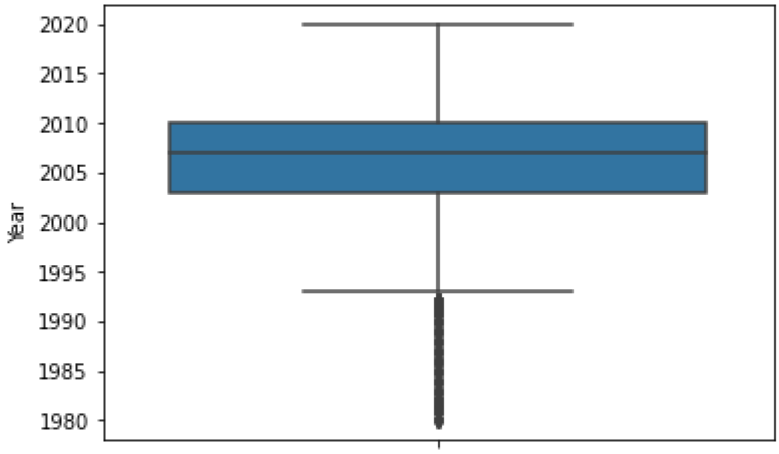


```
sns.kdeplot(data["Year"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f027aa91c40>
```

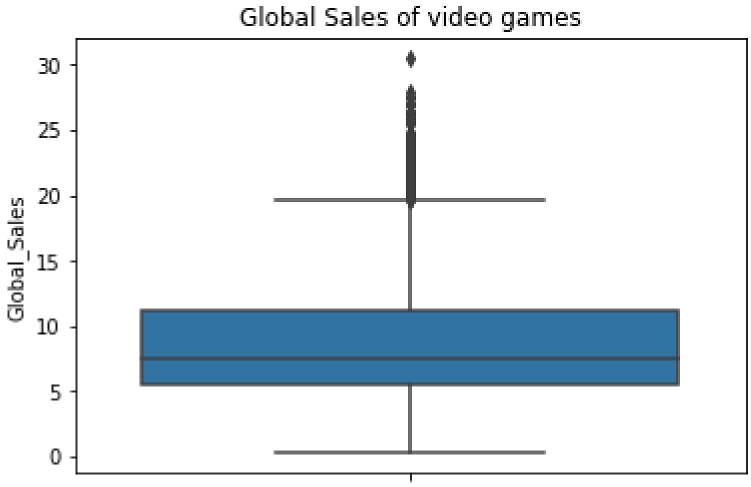
```
sns.boxplot(y=data["Year"])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f02780d3a30>
```



```
sns.boxplot(y=data["Global_Sales"])
plt.title('Global Sales of video games')
```

```
Text(0.5, 1.0, 'Global Sales of video games')
```



Categorical - Barplot, Pie Chart

Numerical - Histogram, KDE, Boxplot

Others - Violion Plot (KDE+Boxplot)

Bivariate - CC

```
# Sales trends over the years for some game (longest running game)?
```

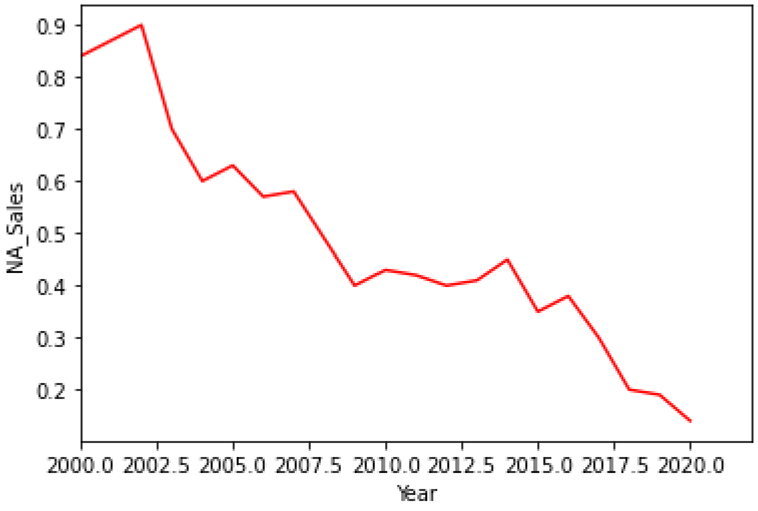
```
data['Name'].value_counts()
```

```
Ice Hockey                                        41
Baseball                                          17
```

```
Need for Speed: Most Wanted                         12
Ratatouille                                          9
FIFA 14                                              9
                                                    ..
Indy 500                                             1
Indy Racing 2000                                     1
Indycar Series 2005                                  1
inFAMOUS                                             1
Zyuden Sentai Kyoryuger: Game de Gaburincho!!        1
Name: Name, Length: 11493, dtype: int64
```

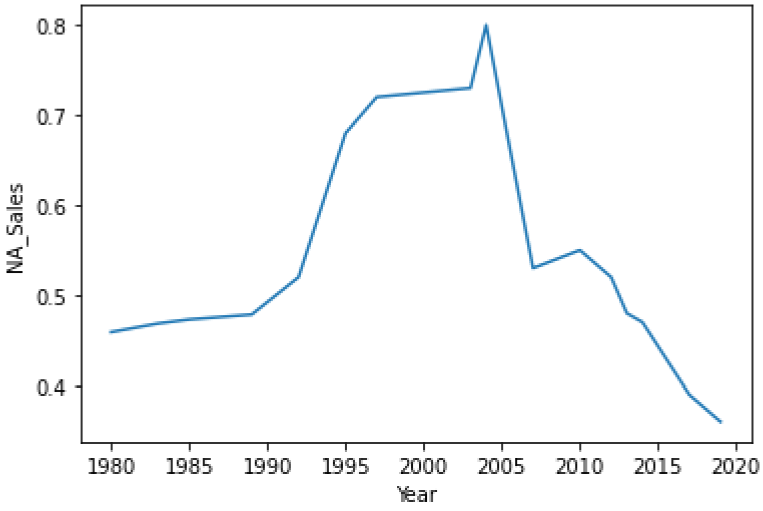
```
ih = data.loc[data['Name']=='Ice Hockey']
sns.lineplot(x="Year", y="NA_Sales", data=ih, color="red")
plt.xlim(left=2000) #ylim
```

```
(2000.0, 2022.0)
```

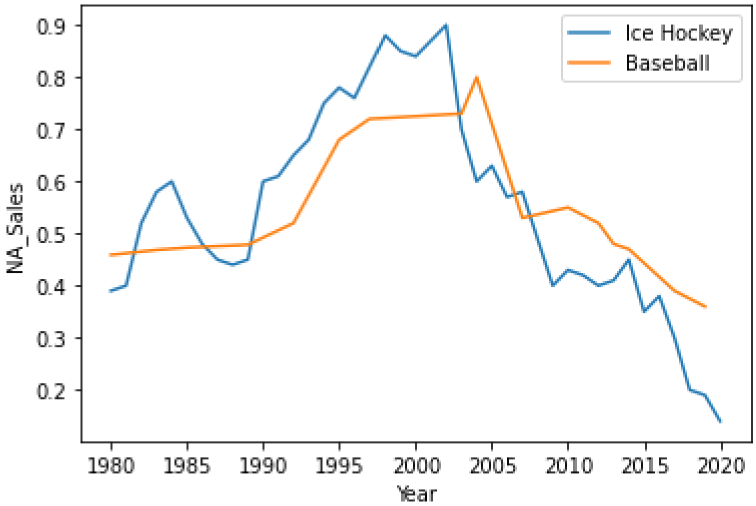


```
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x="Year", y="NA_Sales", data=baseball)
```

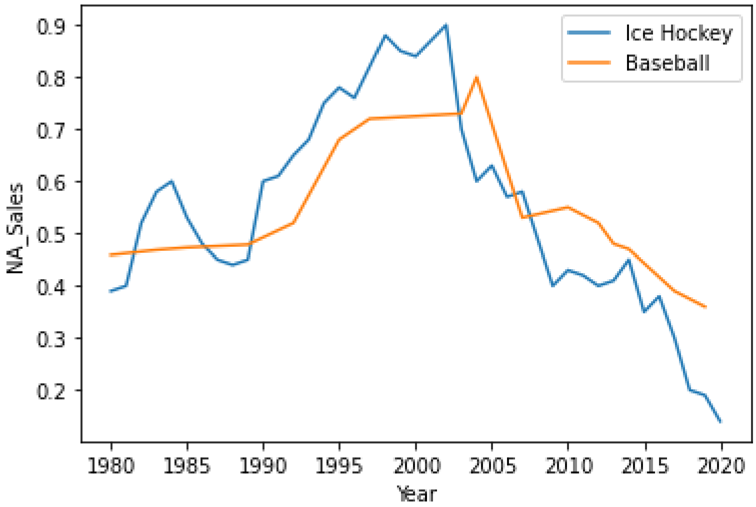
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0277e868e0>
```



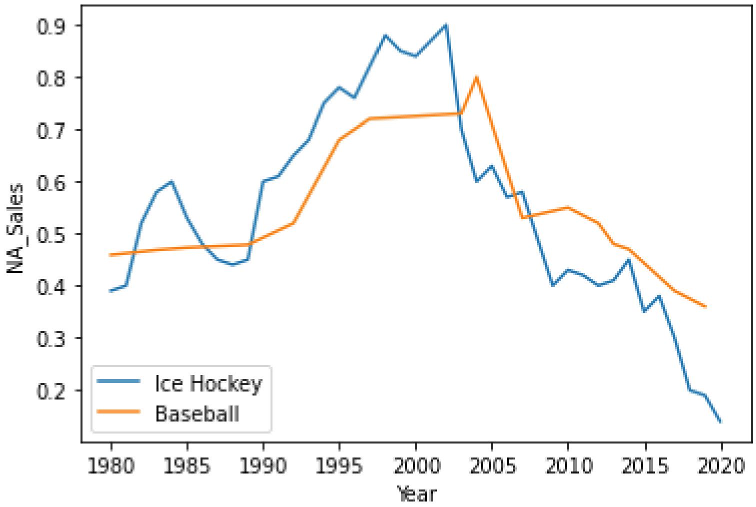
```
sns.lineplot(x="Year", y="NA_Sales", data=ih, label="Ice Hockey")
sns.lineplot(x="Year", y="NA_Sales", data=baseball, label="Baseball")
plt.show()
```



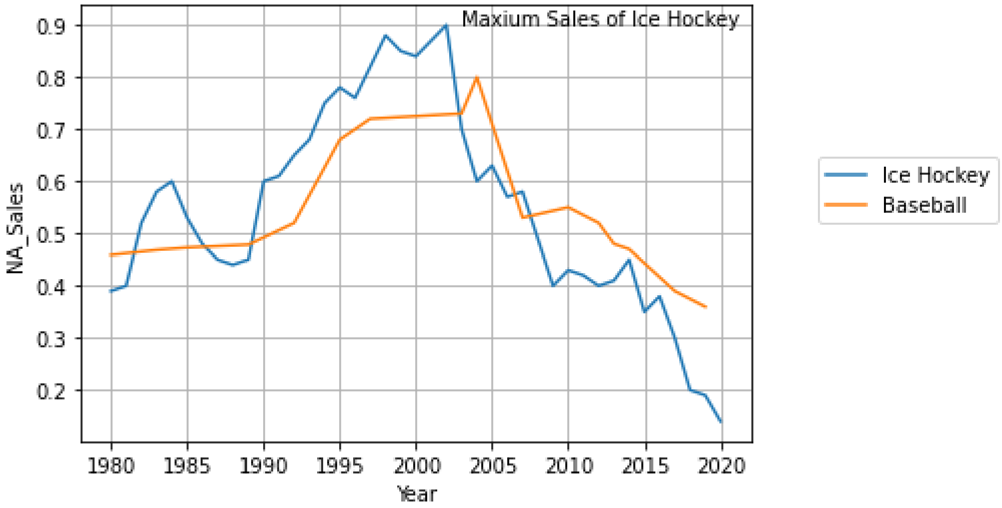
```
sns.lineplot(x='Year', y='NA_Sales', data=ih)
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
plt.legend(['Ice Hockey','Baseball'])
plt.show()
```



```
sns.lineplot(x='Year', y='NA_Sales', data=ih)
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
plt.legend(['Ice Hockey','Baseball'], loc="lower left")
plt.show()
```



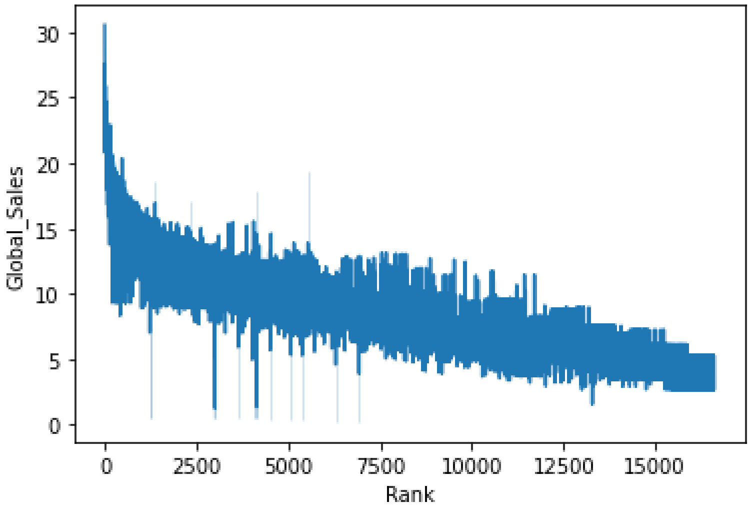
```
sns.lineplot(x='Year', y='NA_Sales', data=ih)
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
plt.legend(['Ice Hockey','Baseball'], loc=(1.1,0.5))
plt.text(2003, 0.9, "Maxium Sales of Ice Hockey")
plt.grid()
plt.show()
```



```
# Relation between rank and sales?
```

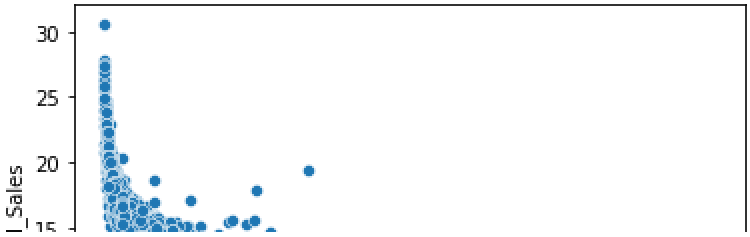
```
sns.lineplot(y="Global_Sales", x="Rank", data=data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0277855340>
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0277b557c0>
```



```
# Bivariate - Cat Cat
```

```
data.describe(include=object)
```

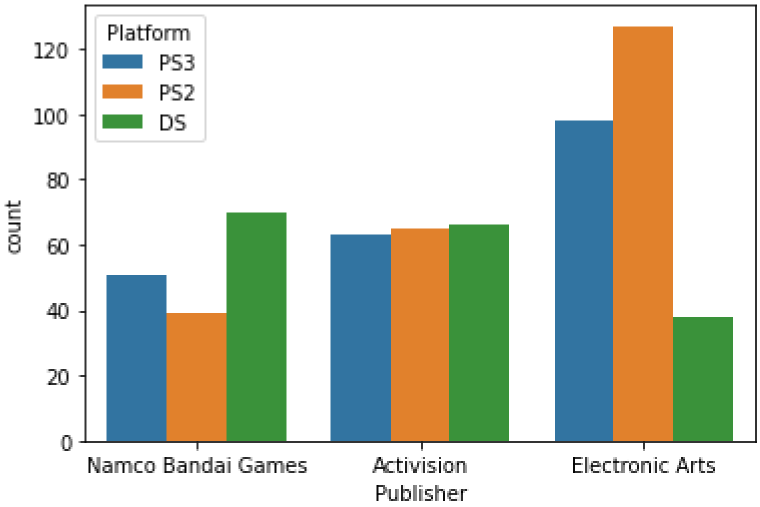
	Name	Platform	Genre	Publisher
count	16652	16652	16652	16594
unique	11493	33	12	578
top	Ice Hockey	DS	Action	Electronic Arts
freq	41	2163	3316	1351

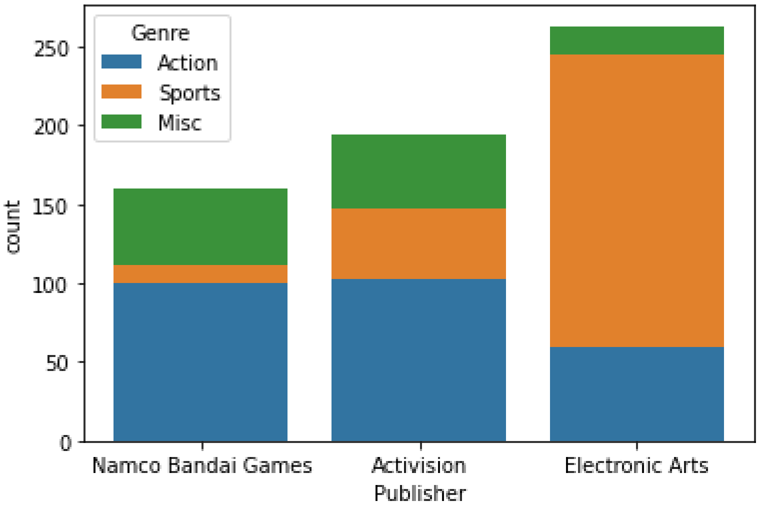
```
top3_pub = data['Publisher'].value_counts().index[:3]
top3_gen = data['Genre'].value_counts().index[:3]
top3_plat = data['Platform'].value_counts().index[:3]
top3_data = data.loc[(data["Publisher"].isin(top3_pub)) & (data["Platform"].isin(to
top3_data
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	J
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339	
13	2742	[Prototype 2]	PS3	2012.0	Action	Activision	3.978349	3.727034	
16	1604	[Prototype]	PS3	2009.0	Action	Activision	4.569217	4.108402	
19	1741	007: Quantum of Solace	PS3	2008.0	Action	Activision	4.156030	4.346074	
21	4501	007: Quantum of Solace	PS2	2008.0	Action	Activision	3.228043	2.738800	
...	...	...	...	...	...	...	...	...	
		Yes! Precure 5 Go Go				Namco			

```
sns.countplot(x="Publisher", data=top3_data, hue="Platform")
# Dodged Bar(Count) Plot
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f027796e070>
```

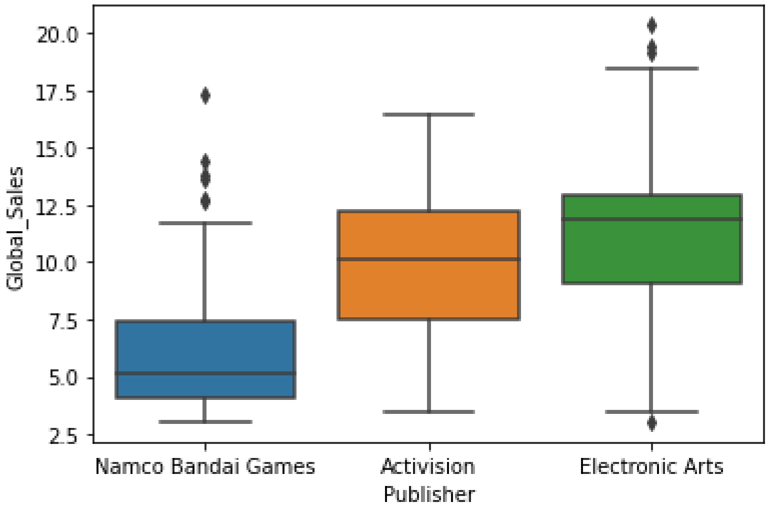




```
# Cat, Cont
```

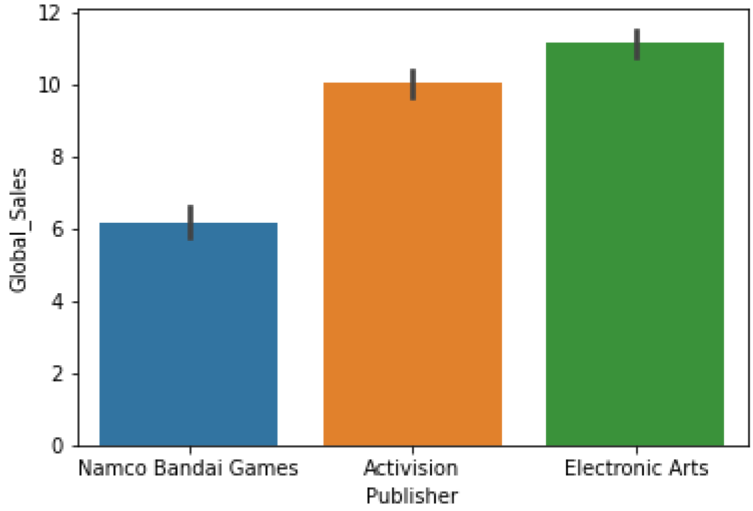
```
# distribution of sales for top-3 publishers
sns.boxplot(data = top3_data, x="Publisher", y="Global_Sales")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0277b4a2b0>
```



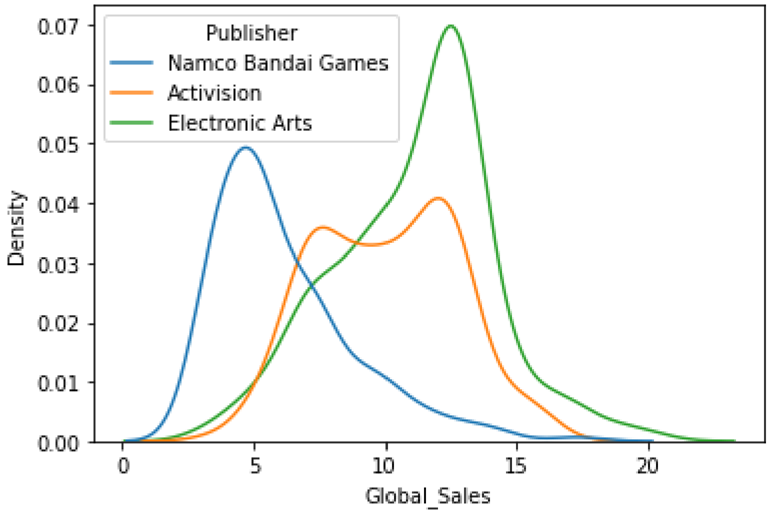
```
sns.barplot(data = top3_data, x="Publisher", y="Global_Sales", estimator=np.mean) #
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f027ad79ee0>
```



```
sns.kdeplot(x="Global_Sales", hue="Publisher", data=top3_data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f0277fabeb0>
```



1. NN - Scatterplot, Lineplot
2. CC - Dodged Countplot, Stacked Count plot
3. NC - Dodged Boxplots, Barplot, KDE plots

- Multivariate Plots
- Subplots - creating multiplots in same figure
- Meshgrid

Friday no class

Harshit --> Monday -> Web API and Scraping

Anant --> Wednesday --> Matplolib and Seaborn-3

✓ 0s completed at 23:01 ● ×