Link: https://colab.research.google.com/drive/1BOyZUnmNjbDYcILb4h5hxXNilCfWAWgy?usp=sharing

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

```
# Where all do you think Data Voz. is helpful or needed?
# Exploratory - see some difficult patterns, EDA
# Explanatory - I want to create a stort out of it and I want to stake holders
```

```
# Art and Science Data Visualisation
```

Loading...

```
# Science
# Understanding the anatomy of a plot
# How to choose which plot is right to answer my question?'

# Art
# Choosing thje right scale, axis, ticks and labels
# Identify and removing clutters
# Ways to highlight the information
```

```
# Intro to Matplotlib
```

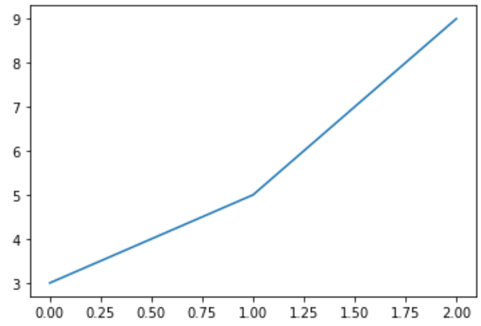
```
import matplotlib.pyplot as plt
import seaborn as sns
```

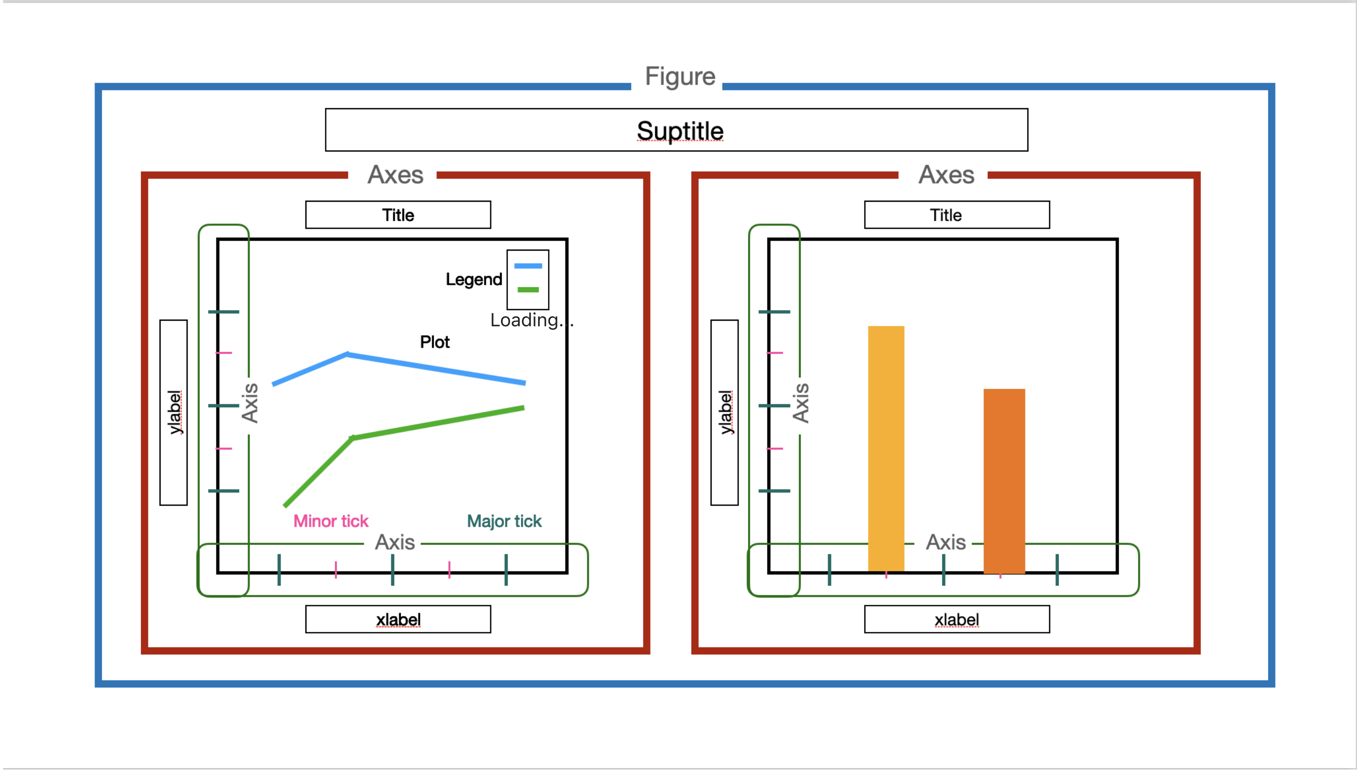
```
# (0,3) (1,5) (2,9)
```

```
x_val = [0,1,2]
y_val = [3,5,9]
```

```
plt.plot(x_val, y_val)
```

```
[<matplotlib.lines.Line2D at 0x7fdd26add400>]
```





```
# how to choose the plot


# Data
# Rows - Records, Samples, Volume, Data-points
# Columns - Features, Attributes, Characteristics


# Columns - Continous, Categorical
# Categorical
# Oridinal - inherant ordering
# Nominal - no ordering


# Thumb for deciding the right plot?

# Question
# 1. How many variables/features are involved in answering my question?
  # Univatiate
  # Bivariate
  # Multi-variate
# 2. What are data-types of different variables involved?
  # Numerical
  # Categorical


# Univariate Data Visualisation
  # N - histogram, boxplot, swarm plot, density plot, violin
  # C - pie-chart, donut, bar chart
# Bivariate Data Visualisation
  # N, N - scatter, ...
  # N, C -
  # C, C -
# Multi-variate Data Visualisatiob (3 variables)
  # N, N, N -
  # N, N, C -
  # C, C, N -
  # C, C, C -
```

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

```
--2023-02-09 15:41:57--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/original/final_vg1_-_fina
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.84.178.93, 99.84.178.132, 99.84.178.172, ..
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99.84.178.93|:443... connected.
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 0.02s
```

```
2023-02-09 15:41:57 (109 MB/s) - ‘final_vg.csv’ saved [2041483/2041483]
```

```
import pandas as pd
import numpy as np
data = pd.read_csv('final_vg.csv')
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	¡Shin Chan Flipa en colores!	DS	2007.0	Loading... 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	8359	.hack//G.U. Vol.1//Rebirth	PS2	2006.0	Role-Playing	Namco Bandai Games	2.031986	1.389856	3.228043	0.394830	7.226880
4	7109	.hack//G.U. Vol.2//Reminisce	PS2	2006.0	Role-Playing	Namco Bandai Games	2.792725	2.592054	1.440483	1.493442	8.363113

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16652 entries, 0 to 16651
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Rank          16652 non-null  int64  
 1   Name          16652 non-null  object 
 2   Platform      16652 non-null  object 
 3   Year          16381 non-null  float64
 4   Genre         16652 non-null  object 
 5   Publisher     16594 non-null  object 
 6   NA_Sales      16652 non-null  float64
 7   EU_Sales      16652 non-null  float64
 8   JP_Sales      16652 non-null  float64
 9   Other_Sales   16652 non-null  float64
 10  Global_Sales  16652 non-null  float64
dtypes: float64(6), int64(1), object(4)
memory usage: 1.4+ MB
```

```
data["Genre"].unique()
```

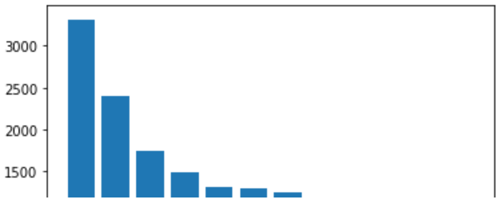
```
array(['Shooter', 'Platform', 'Action', 'Role-Playing', 'Racing', 'Misc',
       'Adventure', 'Sports', 'Puzzle', 'Simulation', 'Strategy',
       'Fighting'], dtype=object)
```

```
cat_counts = data["Genre"].value_counts()
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
```

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

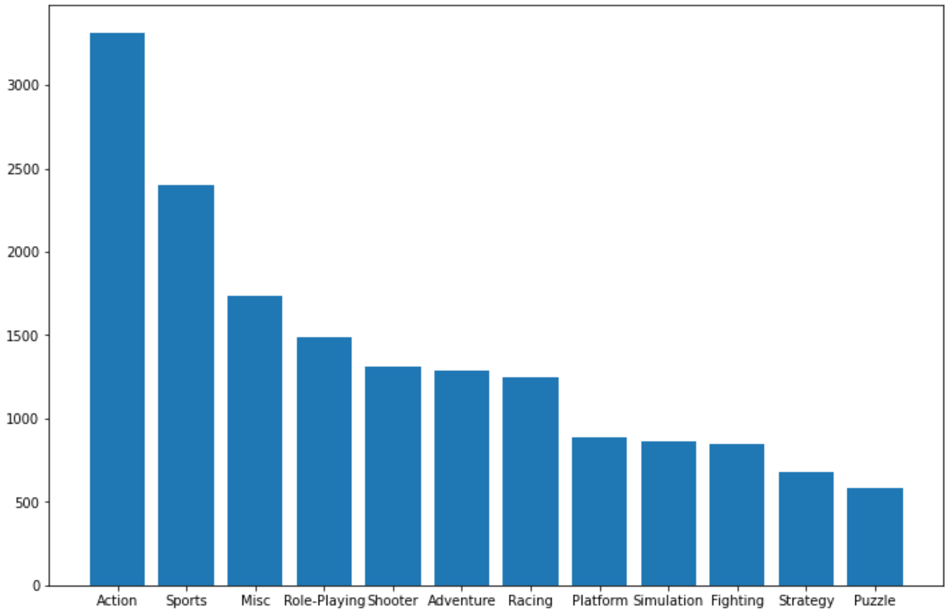
```
<BarContainer object of 12 artists>
```



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

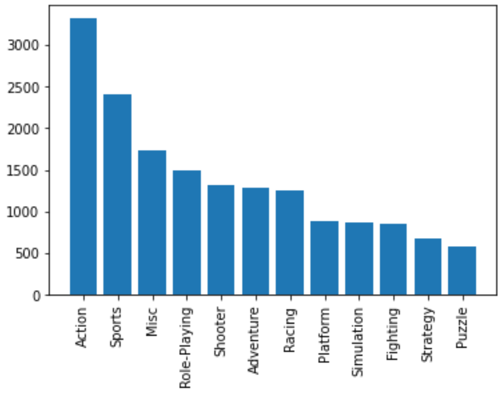
```
<BarContainer object of 12 artists>
```

Loading...



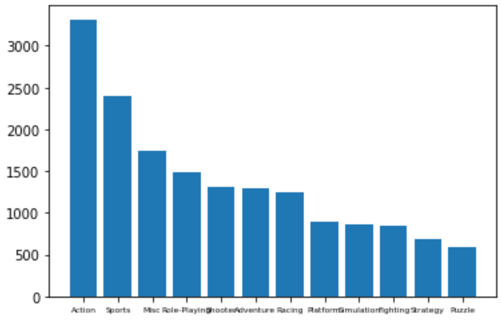
```
plt.bar(x_bar, y_bar)
plt.xticks(rotation=90)
```

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



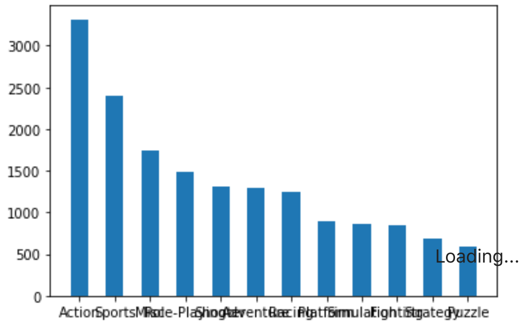
```
plt.bar(x_bar, y_bar)
plt.xticks(fontsize=6)
```

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

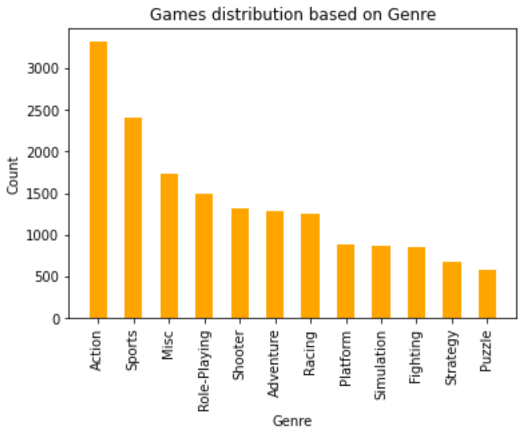


```
plt.bar(x_bar, y_bar, width=0.5)
```

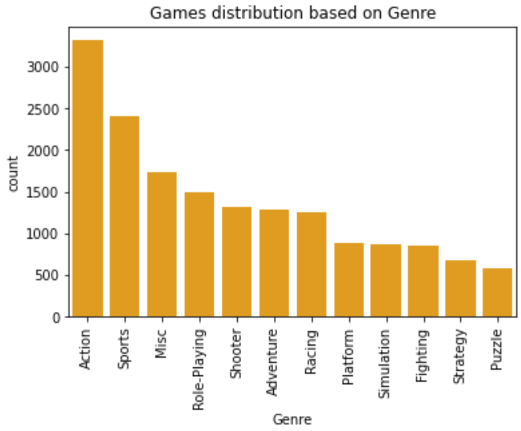
```
<BarContainer object of 12 artists>
```

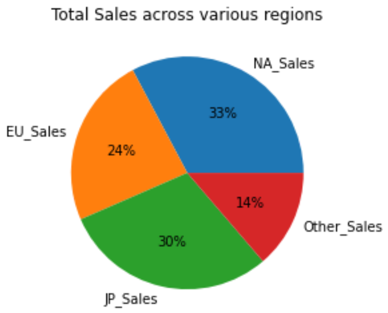


```
plt.bar(x_bar, y_bar, color="orange", width=0.5)
plt.xticks(rotation=90)
plt.title("Games distribution based on Genre",fontsize=12)
plt.xlabel("Genre", fontsize=10)
plt.ylabel("Count", fontsize=10)
plt.show()
```



```
sns.countplot(data = data, x = "Genre", color="orange",
              order = data["Genre"].value_counts().index)
plt.xticks(rotation=90)
plt.title("Games distribution based on Genre",fontsize=12)
plt.show()
```

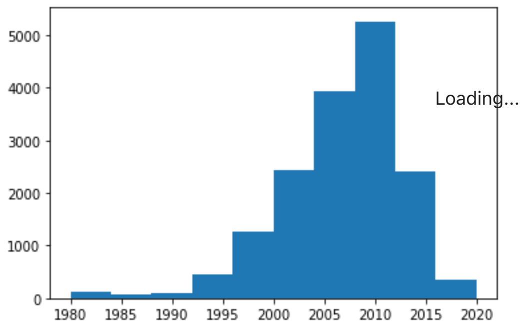




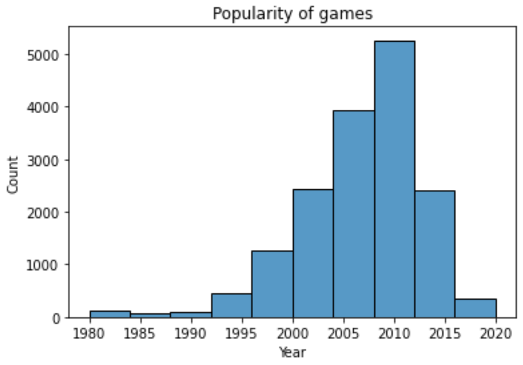
```
# Univariate, Numerical (Cont.)
```

```
# Popularity of video games (no. of games in market) year by year?
```

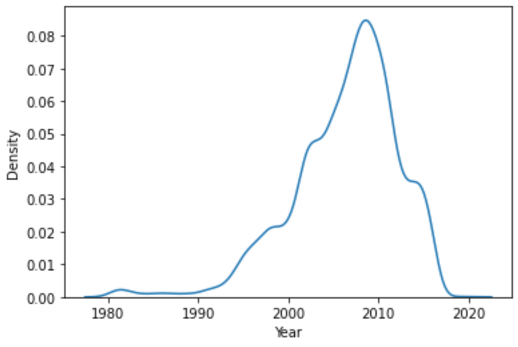
```
plt.hist(data["Year"],bins=10)
plt.show()
```



```
sns.histplot(data["Year"],bins=10)
plt.title("Popularity of games")
plt.show()
```



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



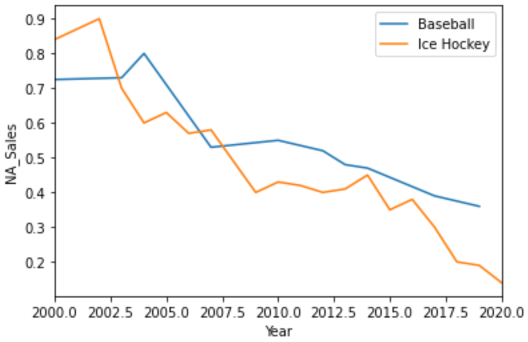
```
# Typical earnings from a video game
```

```
plt.figure(figsize=(5,5))
sns.boxplot(y = data["Global_Sales"])
plt.yticks(fontsize=12)
plt.ylabel('Global Sales', fontsize=12)
plt.title('Global Sales of video games', fontsize=12)
```

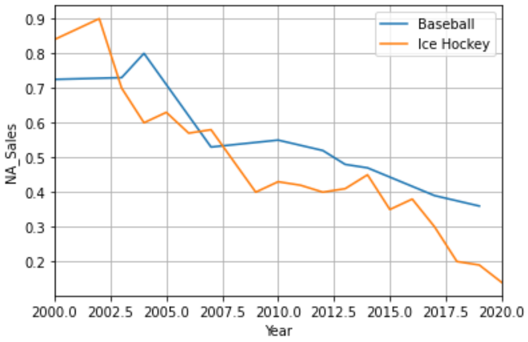
```
Text(0.5, 1.0, 'Global Sales of video games')
```

```
# Bivariate Data Visualisation, N N
```

```
# Relationship between year and sales of ice hockey
ih = data.loc[data['Name']=='Ice Hockey']
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
sns.lineplot(data = ih, x="Year", y="NA_Sales")
plt.legend(["Baseball", "Ice Hockey"])
plt.xlim(left=2000, right=2020)
plt.show()
```

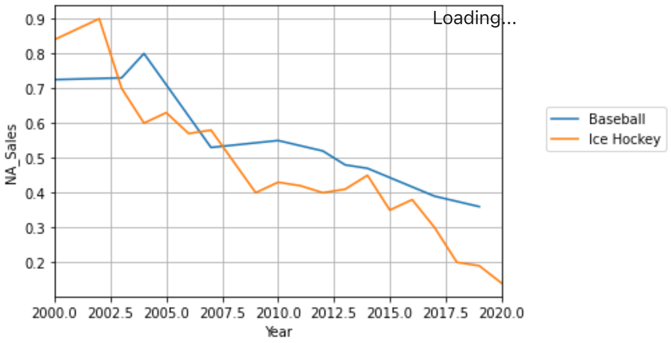


```
# Relationship between year and sales of ice hockey
ih = data.loc[data['Name']=='Ice Hockey']
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
sns.lineplot(data = ih, x="Year", y="NA_Sales")
plt.legend(["Baseball", "Ice Hockey"])
plt.xlim(left=2000, right=2020)
plt.grid()
plt.show()
```

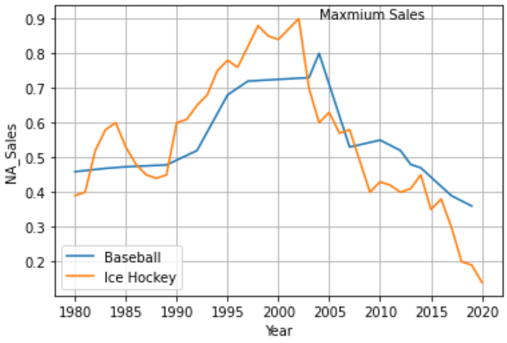


```
# Relationship between year and sales of ice hockey
ih = data.loc[data['Name']=='Ice Hockey']
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
sns.lineplot(data = ih, x="Year", y="NA_Sales")
plt.legend(["Baseball", "Ice Hockey"], loc="lower left")
plt.xlim(left=2000, right=2020)
plt.grid()
plt.show()
```

```
# Relationship between year and sales of ice hockey
ih = data.loc[data['Name']=='Ice Hockey']
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
sns.lineplot(data = ih, x="Year", y="NA_Sales")
plt.legend(["Baseball", "Ice Hockey"], loc=(1.1, 0.5))
plt.xlim(left=2000, right=2020)
plt.grid()
plt.show()
```

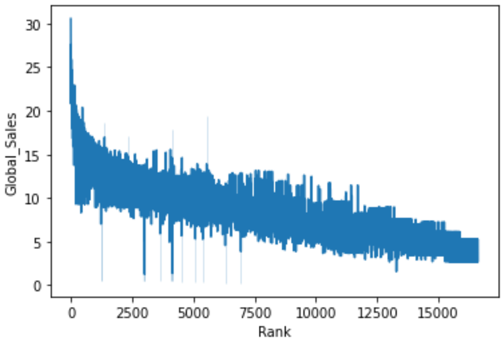


```
# Relationship between year and sales of ice hockey
ih = data.loc[data['Name']=='Ice Hockey']
baseball = data.loc[data['Name']=='Baseball']
sns.lineplot(x='Year', y='NA_Sales', data=baseball)
sns.lineplot(data = ih, x="Year", y="NA_Sales")
plt.legend(["Baseball", "Ice Hockey"], loc="lower left")
plt.text(2004, max(ih["NA_Sales"]), "Maxmium Sales")
plt.grid()
plt.show()
```



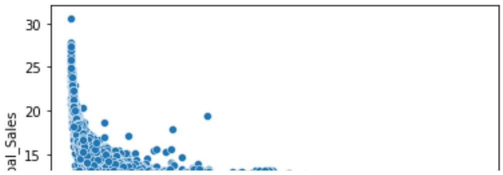
```
# Rank of the video games is associated with sales?
sns.lineplot(data=data, x="Rank", y="Global_Sales")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdd2060ffa0>
```



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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdd2070beb0>
```



✓ 1s completed at 23:03 ● ✕

Loading...