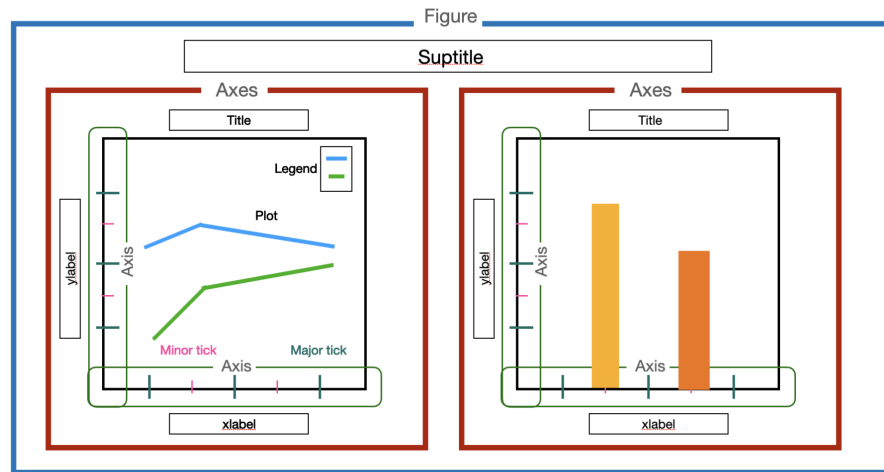


Content:

What is a plot?

a. Anatomy



b. Major components:

- **Figure:** The overall window or page that everything is drawn on.
- **Axis:** Simply the 'x-axis' and 'y-axis'
- **Axes:** It is the area on which the data is plotted
- **x-label:** Name of x-axis
- **y-label:** Name of y-axis
- **Major ticks:** subdivides the axis into major units.
- **Minor ticks:** subdivides major tick units.
- **Title:** Title of each plot (Axes)
- **Legend:** describes the elements in the plot, blue and green curves in this case
- **Suptitle:** The common title of all the plots

Figure	<code>plt.figure(figsize=(x,y))</code> E.g. <code>plt.figure(figsize=(15,10))</code>
Labels	<code>plt.xlabel('x label name')</code> <code>plt.ylabel('y label name')</code>
Title	<code>plt.title('Title of the plot')</code>
Suptitle	<code>fig.suptitle('Title of the whole figure')</code> # Used in case of subplots

Ticks	plt.xticks(rotation=90) # For x ticks plt.yticks(rotation=90) # For y ticks
Legend	plt.legend()

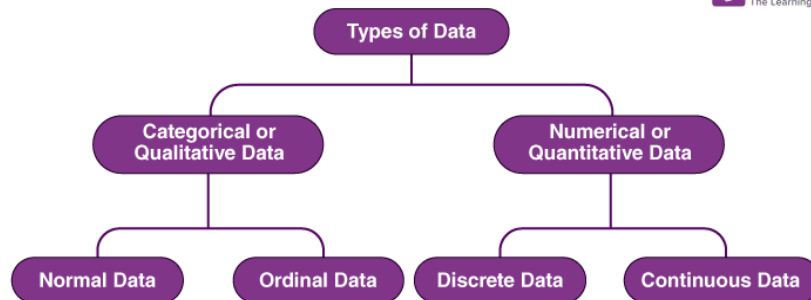
2. Types of Data:

i. Numerical data:

1. **Discrete** -> Can only take finite values (E.g. Year, 2010, 2011, etc.)
2. **Continuous** > Can take any numerical value (E.g. temperature, pressure, etc.)

ii. Categorical data:

1. **Ordinal**: Categorical Data with an order (E.g. low, medium, high)
2. **Non-ordinal/nominal**: Categorical Data without any order (example gender as Male/Female)

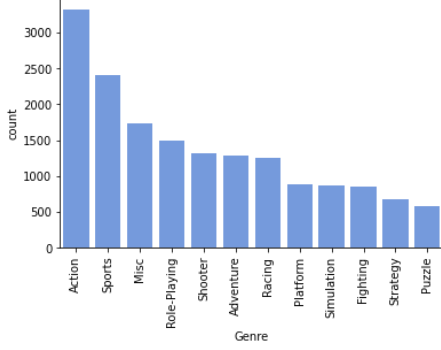
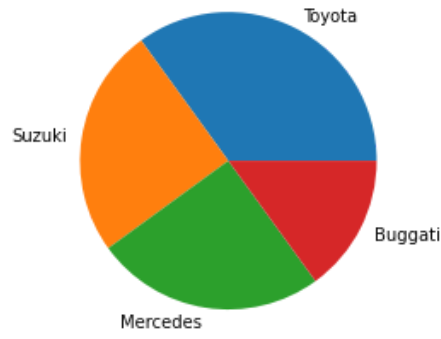


3. Installing and importing matplotlib and seaborn

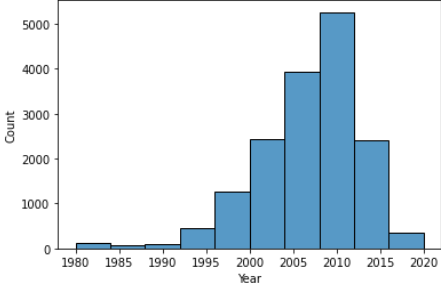
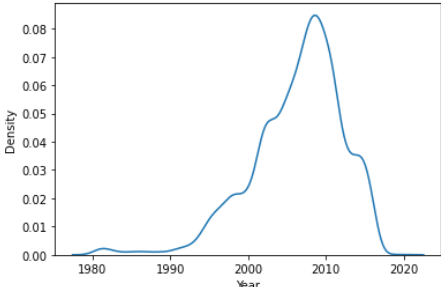
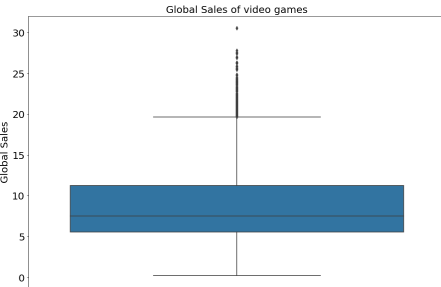
Installing	pip install matplotlib pip install seaborn
Importing convention	import matplotlib.pyplot as plt import seaborn as sns

4. Univariate Data Analysis:

a. Categorical Data

<p>Countplot (Type of barplot)</p>	<p><code>sacountplot(x='col_name', data=df)</code></p> <p>E.g. <code>sns.countplot(x='Genre', data=df)</code></p>  <table border="1"> <thead> <tr> <th>Genre</th> <th>count</th> </tr> </thead> <tbody> <tr><td>Action</td><td>3200</td></tr> <tr><td>Sports</td><td>2400</td></tr> <tr><td>Misc</td><td>1700</td></tr> <tr><td>Role-Playing</td><td>1500</td></tr> <tr><td>Shooter</td><td>1300</td></tr> <tr><td>Adventure</td><td>1250</td></tr> <tr><td>Racing</td><td>1200</td></tr> <tr><td>Platform</td><td>850</td></tr> <tr><td>Simulation</td><td>800</td></tr> <tr><td>Fighting</td><td>800</td></tr> <tr><td>Strategy</td><td>650</td></tr> <tr><td>Puzzle</td><td>550</td></tr> </tbody> </table>	Genre	count	Action	3200	Sports	2400	Misc	1700	Role-Playing	1500	Shooter	1300	Adventure	1250	Racing	1200	Platform	850	Simulation	800	Fighting	800	Strategy	650	Puzzle	550
Genre	count																										
Action	3200																										
Sports	2400																										
Misc	1700																										
Role-Playing	1500																										
Shooter	1300																										
Adventure	1250																										
Racing	1200																										
Platform	850																										
Simulation	800																										
Fighting	800																										
Strategy	650																										
Puzzle	550																										
<p>Pie chart</p>	<p><code>y = np.array([35, 25, 25, 15])</code> <code>mylabels = ["Toyota", "Suzuki", "Mercedes", "Buggati"]</code></p> <p><code>plt.pie(y, labels = mylabels)</code></p>  <table border="1"> <thead> <tr> <th>Brand</th> <th>Count</th> </tr> </thead> <tbody> <tr><td>Toyota</td><td>35</td></tr> <tr><td>Suzuki</td><td>25</td></tr> <tr><td>Mercedes</td><td>25</td></tr> <tr><td>Buggati</td><td>15</td></tr> </tbody> </table>	Brand	Count	Toyota	35	Suzuki	25	Mercedes	25	Buggati	15																
Brand	Count																										
Toyota	35																										
Suzuki	25																										
Mercedes	25																										
Buggati	15																										

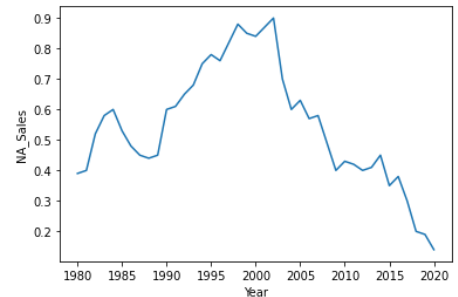
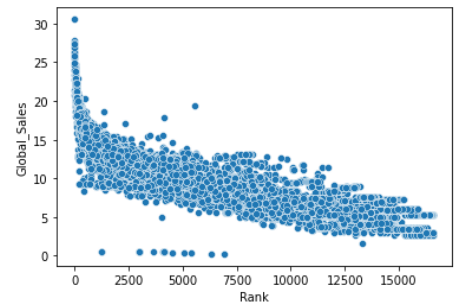
b. Numerical Data

Histogram	<p><code>sns.histplot(x='col_name', data=df)</code></p> <p>E.g. <code>sns.histplot(x='Year', data=df)</code></p> 
KDE Plot	<p><code>sns.kdeplot(x='col_name', data=df)</code></p> <p>E.g. <code>sns.kdeplot(x='Year', data=df)</code></p> 
Boxplot	<p><code>sns.boxplot(x='col_name', data=df)</code></p> <p>E.g. <code>sns.boxplot(x='Global_sales', data=df)</code></p> 

5. Bivariate

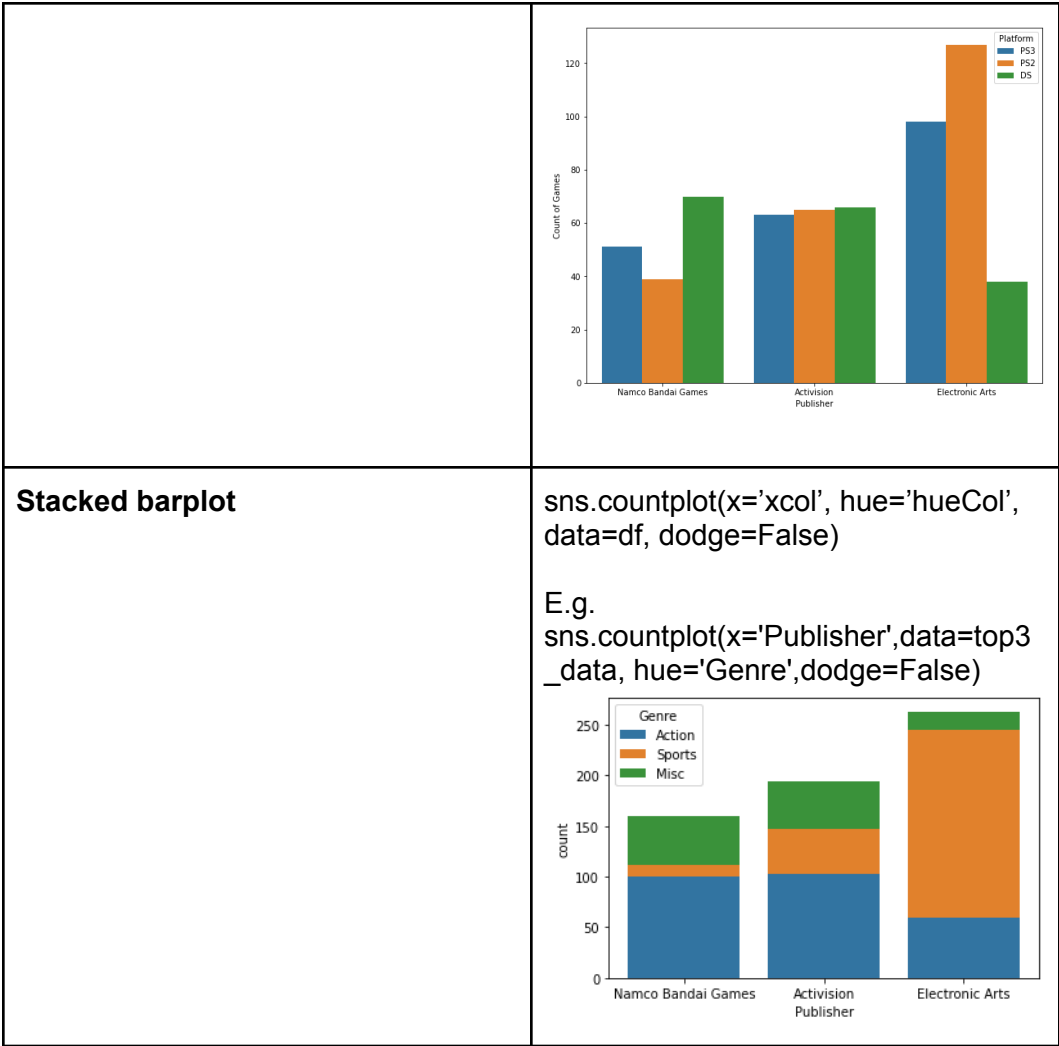
a. Numerical-Numerical

Line Plot	<p><code>sns.lineplot(x='xcol', y='ycol', data=df)</code></p>
------------------	---

	<p>E.g. <code>sns.lineplot(x='Year', y='NA_Sales', data=data)</code></p> 
<p>Scatter Plot</p>	<p><code>sns.scatterplot(x='xcol', y='ycol', data=df)</code></p> <p>E.g. <code>sns.scatterplot(x='Rank', y='Global_Sales', data=df)</code></p> 

b. Categorical-Categorical

<p>Dodged barplot (Extension of countplot)</p>	<p><code>sns.countplot(x='xcol', hue='hueCol', data=df)</code></p> <p>E.g. <code>sns.countplot(x='Publisher', hue='Platform', data=plat_data)</code></p>
--	--



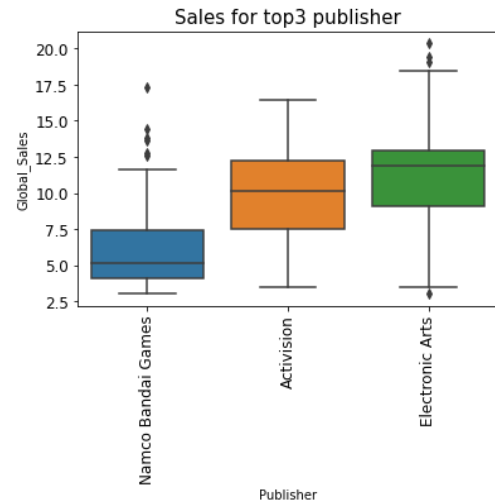
c. Categorical-Numerical

Boxplot

```
sns.boxplot(x='xcol', y='ycol', data=df)
```

E.g.

```
sns.boxplot(x='Publisher',  
y='Global_Sales', data=pub_data)
```

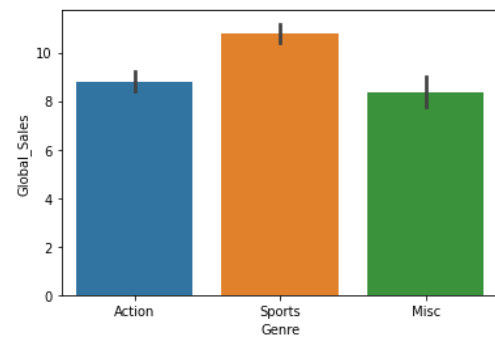


Barplot

```
sns.barplot(x='xcol', y='ycol', data=df)
```

E.g.

```
sns.barplot(x="Genre",  
y="Global_Sales", data=sales_data)
```



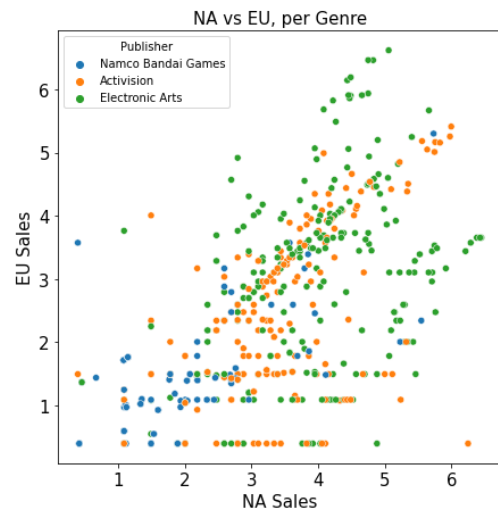
6. Multivariate

Numerical Numerical Categorical
(Scatterplot with colour)

```
sns.scatterplot(x='xcol', y='ycol',  
hue='hueCol', data=df)
```

E.g.

```
sns.scatterplot(x='NA_Sales',  
y='EU_Sales', hue='Publisher', data=top3_data)
```

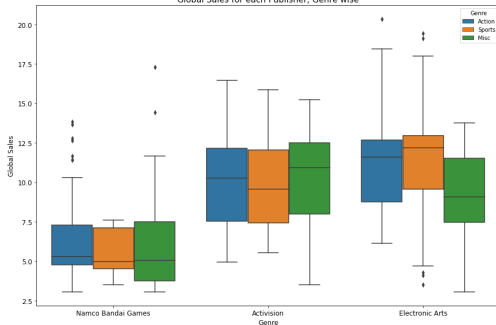
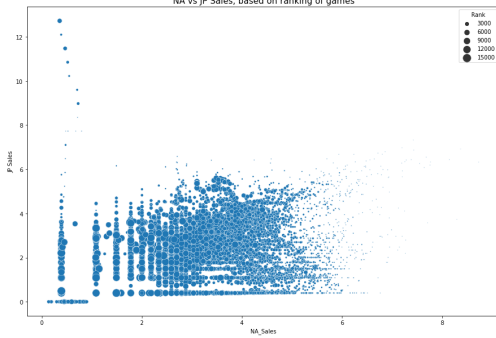


Categorical Categorical Numerical
(Dodged boxplot)

```
sns.boxplot(x='xcol', y='ycol',  
hue='hueCol', data=df)
```

Eg.

```
sns.boxplot(x='Publisher', y='Global_Sales',  
hue='Genre', data=top3_data)
```


	 <p>Global Sales for each Publisher, Genre wise</p> <p>Y-axis: Global Sales (2.5 to 20.0)</p> <p>X-axis: Publisher (Namco Bandai Games, Activision, Electronic Arts)</p> <p>Legend: Genre (Action, Sports, Misc)</p>
<p>Numerical-Numerical-Numerical (Bubble Chart)</p>	<pre>sns.scatterplot(x='xcol', y='ycol', size='rankCol', sizes=tuple_of_sizes, data=data)</pre> <p>E.g.</p> <pre>sns.scatterplot(x='NA_Sales', y='JP_Sales', size='Rank', sizes=(1, 200), data=data)</pre>  <p>NA vs JP Sales, based on ranking of games</p> <p>Y-axis: JP Sales (0 to 12)</p> <p>X-axis: NA_Sales (0 to 8)</p> <p>Legend: Rank (1000, 6000, 9000, 12000, 15000)</p>

7. Subplots

We can plot multiple plots in a single figure

```
plt.figure()
```

```
plt.subplot(row,col,1)
```

```
# plot
```

```
...
```

E.g.

```
plt.figure(figsize=(20,12)).suptitle("NA Sales vs regions",fontsize=20)
```

```
# Using a 2x3 subplot
```

```
plt.subplot(2, 3, 1)
```

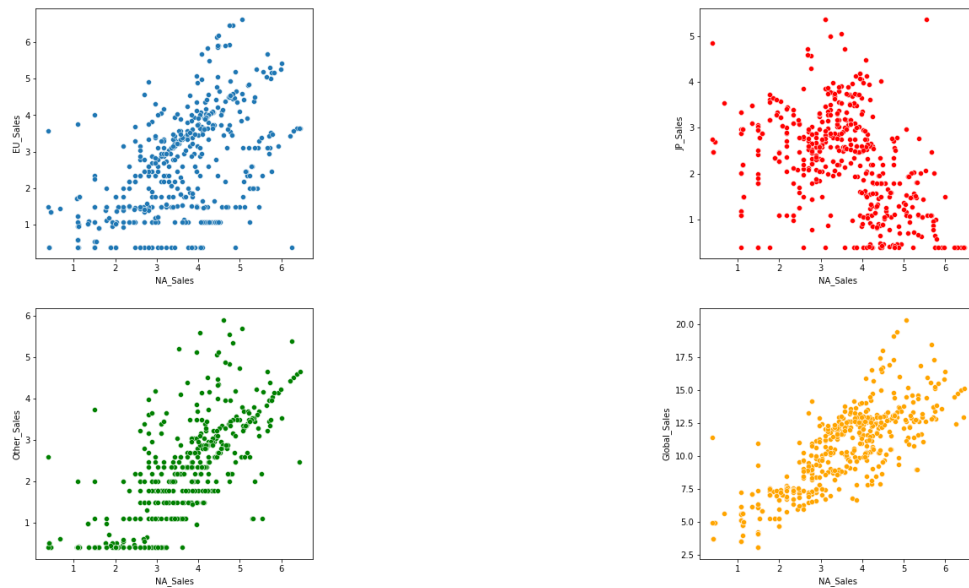
```
sns.scatterplot(x='NA_Sales', y='EU_Sales', data=top3_data)
```

```
plt.subplot(2, 3, 3)
sns.scatterplot(x='NA_Sales', y='JP_Sales', data=top3_data, color='red')

plt.subplot(2, 3, 4)
sns.scatterplot(x='NA_Sales', y='Other_Sales', data=top3_data, color='green')

plt.subplot(2, 3, 6)
sns.scatterplot(x='NA_Sales', y='Global_Sales', data=top3_data, color='orange')
```

NA Sales vs regions



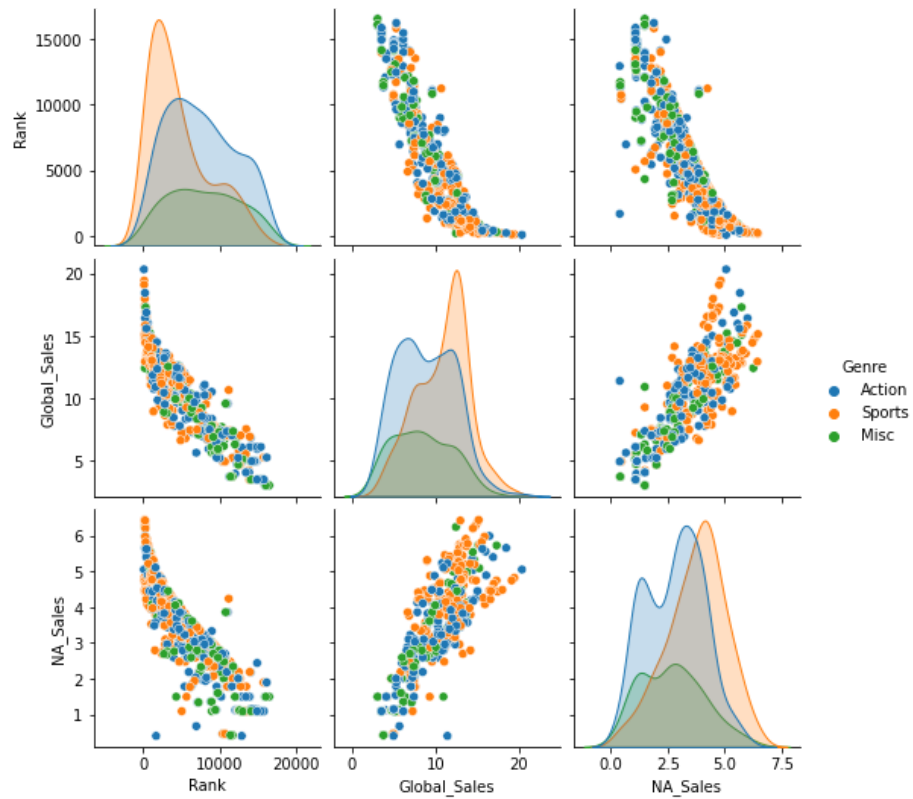
8. Pairplot

Displays a scatterplot for each pair of attributes, can provide a hue for each category too

```
sns.pairplot(data=df, hue='hueCol')
```

E.g.

```
sns.pairplot(data=df, hue='Genre')
```

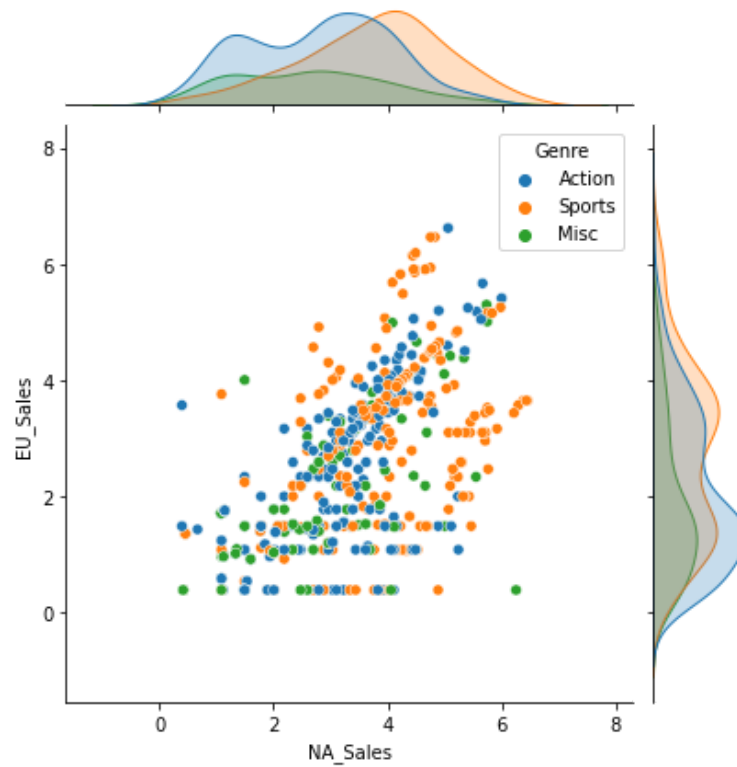


9. Jointplot

Draws multiple types of plot of two variables in the same plot
`sns.jointplot(x='xcol', y='ycol', data=data, hue='hueCol')`

E.g.

`sns.jointplot(x='NA_Sales', y='EU_Sales', data=top3_data, hue='Genre')`



10. Correlation and Heatmaps

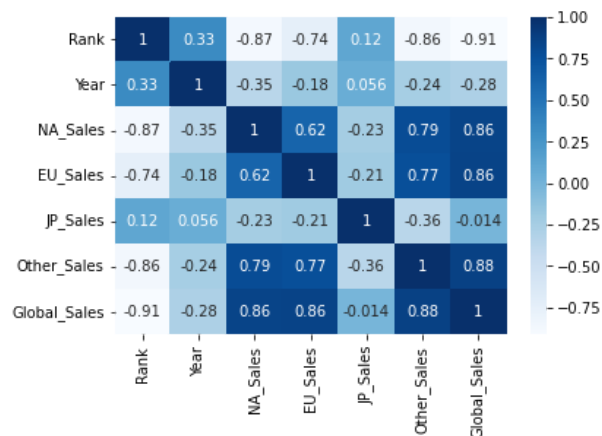
Plot a heatmap of correlation between various variables

`sns.heatmap(df.corr(), cmap='colour_map')`

E.g.

`sns.heatmap(top3_data.corr(), cmap= "Blues", annot=True)`

plots a heatmap of the data with the correlation coefficient values annotated

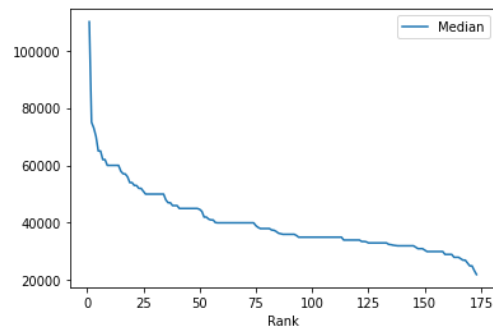


11. Plotting with Pandas

Lineplot

```
df.plot(x='col1', y='col2')
```

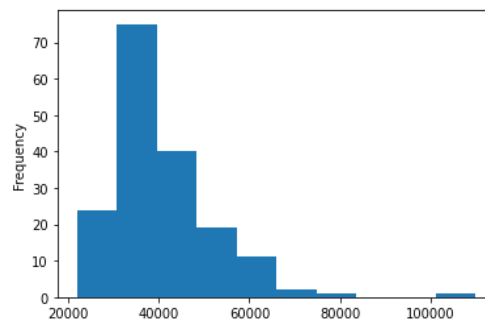
E.g.
`df.plot(x="Rank", y = "Median")`



Histogram

```
df['col'].plot(kind='hist')
```

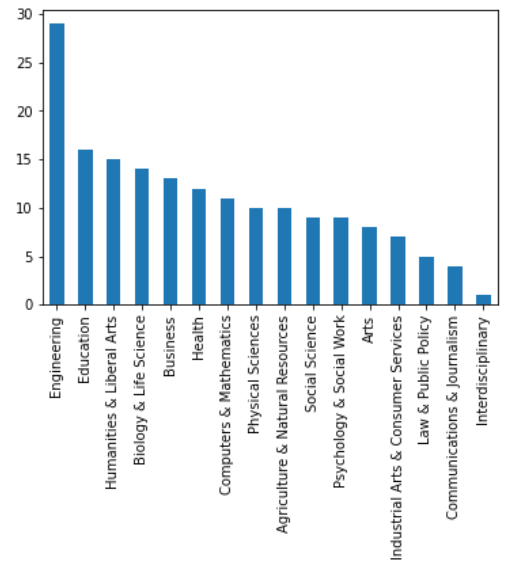
E.g.
`df["Median"].plot(kind="hist")`



Barplot

```
df['col'].value_counts().plot(kind='bar')
```

E.g.
`df["Major_category"].value_counts().plot(kind = 'bar')`



Scatterplot

```
df.plot(x='col1', y='col2',
kind='scatter')
```

E.g.

```
df.plot(x="Median",
y="Unemployment_rate",
kind="scatter")
```

