Content

- Multivariate Data Visualization
  - CCN
  - CNN
  - NNN
  - CCC
- JointPlot
- Pairplots
- Correlation and heatmap

Importing the data

```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/original/final_vg1_-_final_vg_%281%29.csv?16708
```

```
--2024-02-06 16:35:17--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/021/299/original/final_vg1_-_fin
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 13.35.37.31, 13.35.37.159, 13.35.37.102, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|13.35.37.31|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2041483 (1.9M) [text/plain]
Saving to: 'vgsales.csv'

vgsales.csv         100%[===================>]   1.95M  2.94MB/s    in 0.7s

2024-02-06 16:35:19 (2.94 MB/s) - 'vgsales.csv' saved [2041483/2041483]
```

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

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

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sal
0	2061	1942	NES	1985.0	Shooter	Capcom	4.569217	3.033887	3.439352	1.9916
1	9137	¡Shin Chan Flipa en colores!	DS	2007.0	Platform	505 Games	2.076955	1.493442	3.033887	0.3948
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339	1.493442	0.4086
3	8359	.hack//G.U. Vol.1//Rebirth	PS2	2006.0	Role-Playing	Namco Bandai Games	2.031986	1.389856	3.228043	0.3948
4	7109	.hack//G.U. Vol.2//Reminisce	PS2	2006.0	Role-Playing	Namco Bandai Games	2.792725	2.592054	1.440483	1.4934

If you remember, `Genres`, `Publisher` and `Platform` were categorical values

Hence similar to last lecture, we will use top 3 of each to make our analysis easier

```
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(top3_plat)) & (data['Genre'].isin(top3_gen)
top3_data
```

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sal
2	14279	.hack: Sekai no Mukou ni + Versus	PS3	2012.0	Action	Namco Bandai Games	1.145709	1.762339	1.4934
13	2742	[Prototype 2]	PS3	2012.0	Action	Activision	3.978349	3.727034	0.8488
16	1604	[Prototype]	PS3	2009.0	Action	Activision	4.569217	4.108402	1.1872
19	1741	007: Quantum of Solace	PS3	2008.0	Action	Activision	4.156030	4.346074	1.0879
21	4501	007: Quantum of Solace	PS2	2008.0	Action	Activision	3.228043	2.738800	2.5855
...	...	...	...	...	...	...	...	...	...
16438	14938	Yes! Precure 5 Go Go Zenin Shu Go! Dream Festival	DS	2008.0	Action	Namco Bandai Games	1.087977	0.592445	1.0879
16479	10979	Young Justice: Legacy	PS3	2013.0	Action	Namco Bandai Games	2.186589	1.087977	3.4090

16601	11802	ZhuZhu Pets: Quest for Zhu	DS	2011.0	Misc	Activision	2.340740	1.525543	3.1038
16636	9196	Zoobles! Spring to Life!	DS	2011.0	Misc	Activision	2.697415	1.087977	2.7607
16640	9816	Zubo	DS	2008.0	Misc	Electronic Arts	2.592054	1.493442	1.4934

617 rows × 11 columns

Multivariate

Let's try to add 3rd variable on the top of the plots we have seen so far

NNC

How can we visualize the correlation between NA and EU, but for different genres?

Here, we have two numerical and one categorical variable!

Numerical-Numerical → Scatterplot, need to add info about one categorical variable

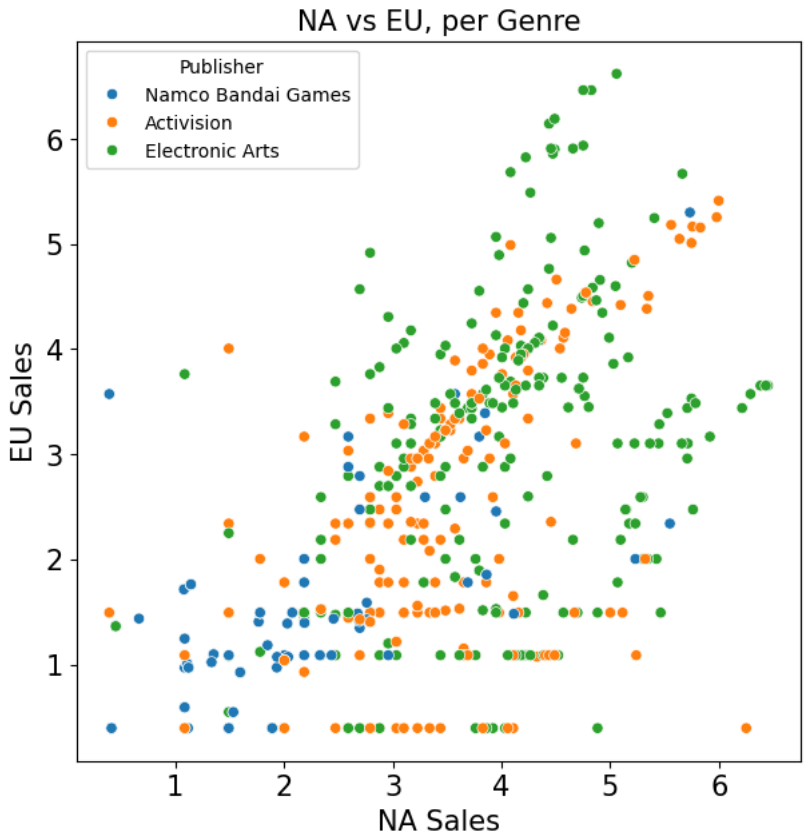
Numerical-Categorical → Boxplot, need to add info about one numerical variable

Let's ask two questions

- Is it Possible to add information about a continuous variable upon boxplots?
  - Perhaps No
- Is it Possible to add information about a categorical variable on scatterplot?
  - Yes, use colors

Solution: Scatterplot with color

```
plt.figure(figsize=(7,7))
sns.scatterplot(x='NA_Sales', y='EU_Sales',hue='Publisher',data=top3_data)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.xlabel('NA Sales',fontsize=15)
plt.ylabel('EU Sales',fontsize=15)
plt.title('NA vs EU, per Genre', fontsize=15)
plt.show()
```



Inferences:

- If we see this plot, we can notice now that Namco has lower sales correlation, while Activision has a concentrated positivee correlation
- EA also has positive correlation, but it's more spread compared to Activision

CCN

Now, how will you visualize Global Sales for each publisher, but separated by Genres?

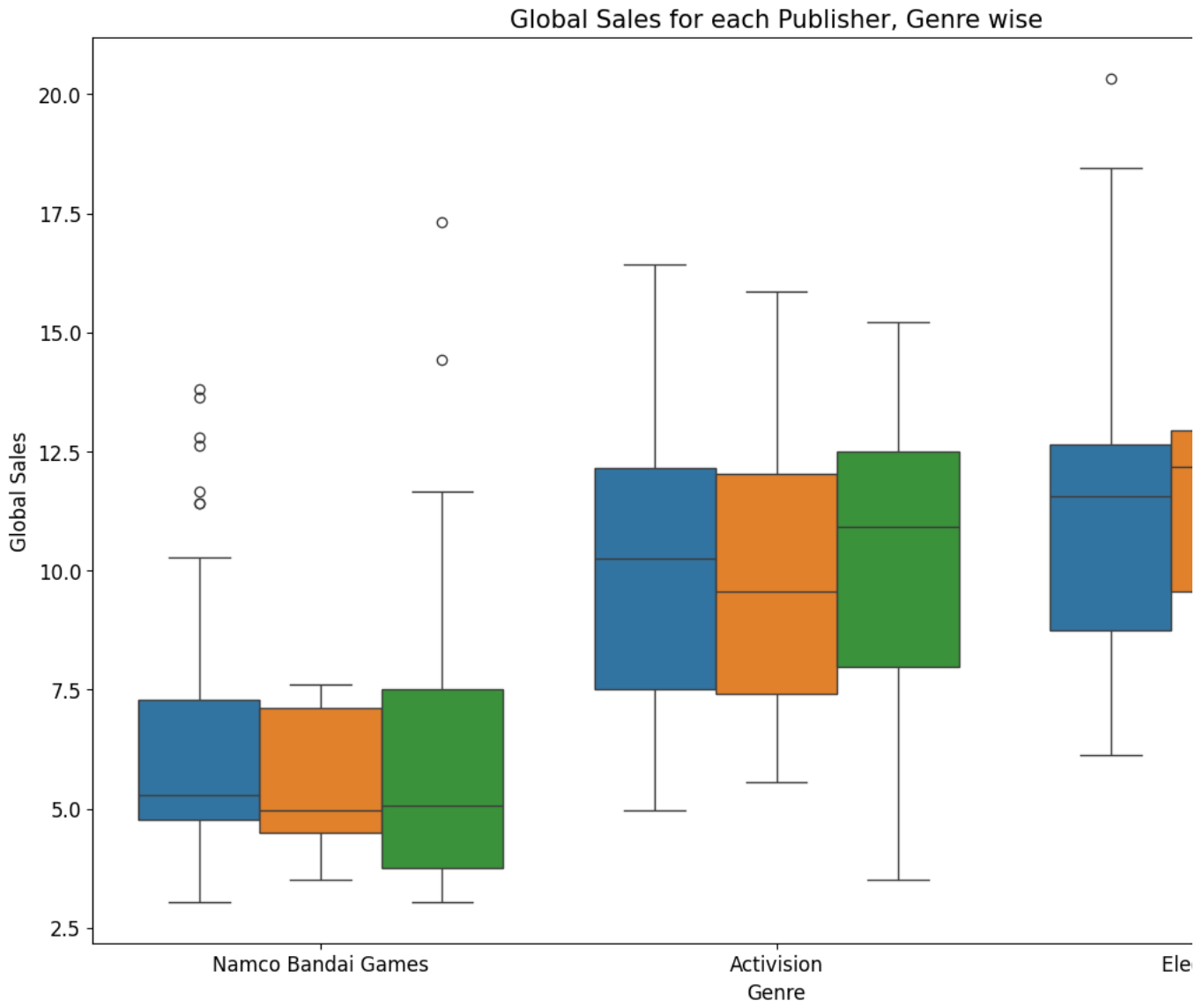
We have two categorical and one numerical data here!

- Categorical-Categorical → Stacked Barplot, need to add info about one continuous feature
- Categorical-Numerical → Boxplots, need to add categorical variable

Which one is easier and possible? We can add one categorical variable by "dodging" multiple boxplots

Solution: Dodged Boxplots

```
plt.figure(figsize=(15,10))
sns.boxplot(x='Publisher',y='Global_Sales',hue='Genre',data=top3_data)
plt.xlabel('Genre', fontsize=12)
plt.ylabel('Global Sales', fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title('Global Sales for each Publisher, Genre wise', fontsize=15)
plt.show()
```



Inferences:

- Namco has lower median sales in every Genre as compared to all publishers
- Looking at `Action` Genre, even though EA and Activision has almost similar medians, Action is more spread in EA
- An interesting thing to notice here is that, for each of the three publishers, three different genre of games have higher sales median:

- Namco: Action
- Activision: Misc
- EA: Sports

NNN

So far we have seen how NA and EU are correlated with each other.

But how can we compare the data when we have 3 numerical variables?

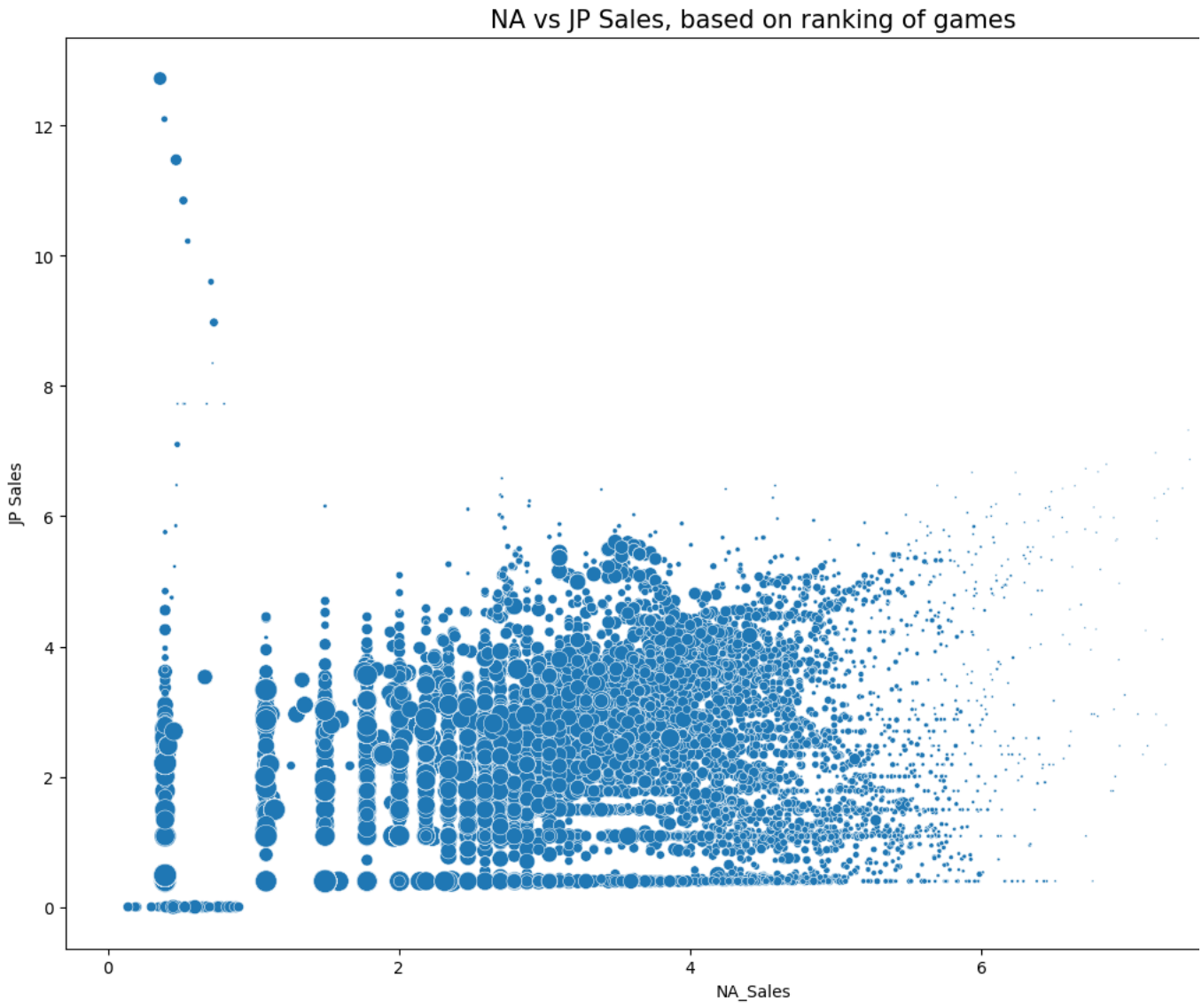
Say, the question is, how does rank affect the correlation between NA and EU Sales?

We have used scatter plot for two numerical features, we have two options here

- Make a 3D Scatterplot
  - → nice for 3D viz, but tough to report/show in static setting
- Add info about 3rd feature on the 2D scatter plot itself
  - → Bubble Chart

```
plt.figure(figsize=(15,10))
# sns.scatterplot(x=data['NA_Sales'], y=data['JP_Sales'],data=top3_data, size=data['Rank'], sizes=(1, 200))
sns.scatterplot(x='NA_Sales', y='JP_Sales', size='Rank', sizes=(1, 200), data=data)

plt.xlabel('NA_Sales',fontsize=10)
plt.ylabel('JP Sales', fontsize=10)
plt.title('NA vs JP Sales, based on ranking of games', fontsize=15)
plt.show()
```



Inferences:

- Now interestingly, we can notice that higher ranking games are actually on the lower scale of sales, while lower ranking games are high on the sales side

Joint Plot

Let's see a few more plots that we can visualize using `seaborn`

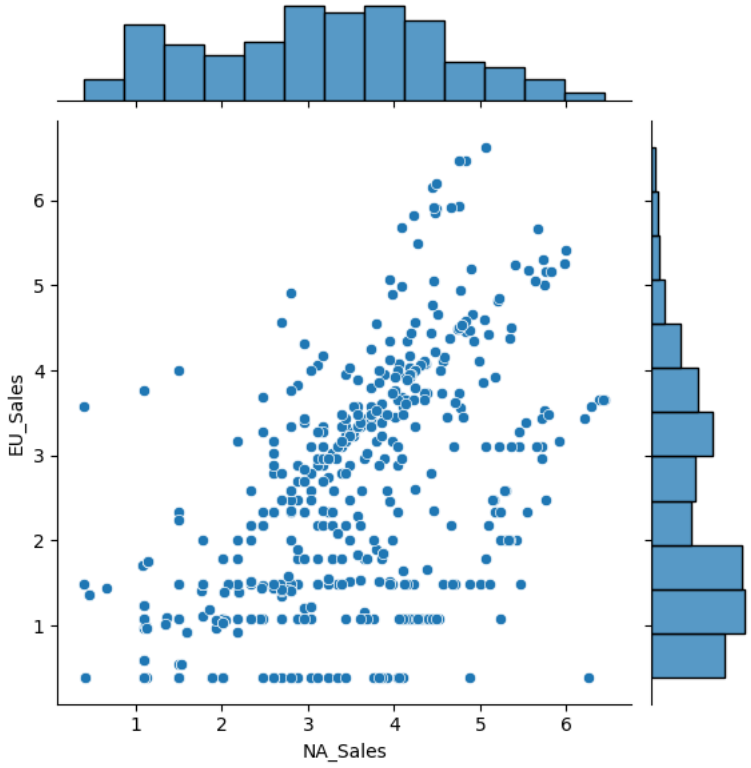
Joint Plot

- It draws a plot of two variables
- It shows scatter, histogram and KDE graphs in the same plot.

Let's check it out

- We will take **`NA_Sales` as x-coordinates** and **`EU_Sales` as y-coordinates**
- We can select from different values for **parameter `kind`** and it **will plot accordingly**
  - "scatter" | "kde" | "hist" | "hex" | "reg" | "resid"
- We will set **parameter `kind`** to **`'reg'`** here

```
sns.jointplot(x='NA_Sales', y='EU_Sales', data=top3_data)
plt.show()
```



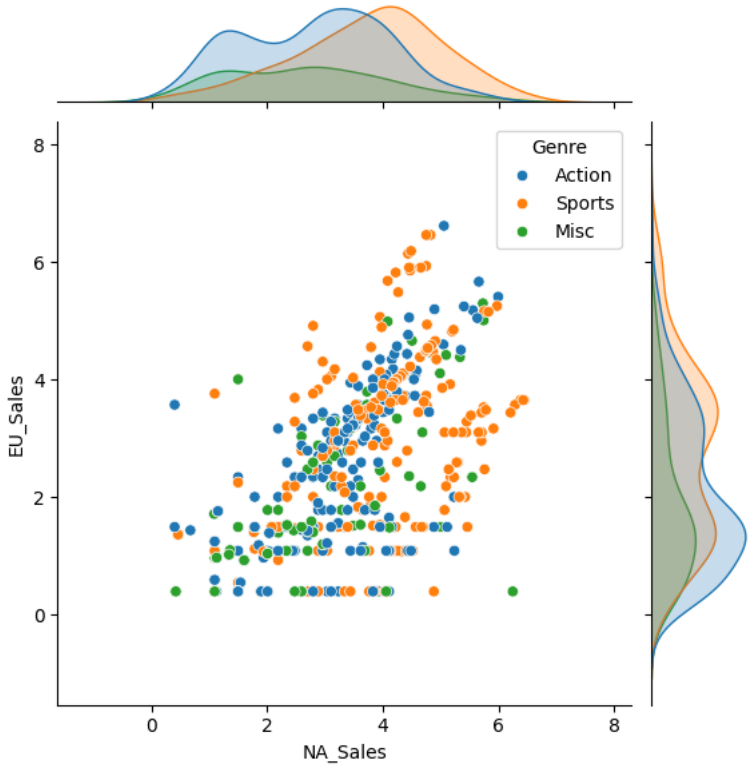
As we can see here:

- `jointplot` plots **scatter, histogram and KDE in the same graph** when we set **`kind=reg`**
- Scatter shows the **scattering of (`NA_Sales` , `EU_Sales`) pairs as (x, y) points**
- Histogram and KDE shows the separate distributions of `NA_Sales` and `EU_Sales` in the data

We can also add hue to Joint Plot

- Let's check how the 3 Genres of games are distributed in terms of `NA_Sales` and 'EU_Sales`

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



Pair Plot

- `pairplot()` in `seaborn` creates a **grid of Axes by default**
- Each numeric attribute in `data` is shared across the y-axes across a single row and the x-axes across a single column.
- It displays a **scatterplot between each pair of attributes in the data** with different **hue** for each category

Since, the diagonal plots belong to same attribute at both x and y axis, they are treated differently

- A univariate distribution plot is drawn to show the marginal distribution of the data in each column.

Let's check it out

```
sns.pairplot(data=top3_data, hue='Genre')
plt.show()
```

Notice that:

- It is **like a scatterplot of video games with `hue='Genre'`**
- But the **scatter is plotted between every pair of attributes**
- **Colour Legends** for each genre category are given on **right side**
- It shows **relation between each pair of attributes**

Diagonal plots are different from scatterplots

- Because x and y axis have same attribute
- Diagonal plots show a univariate curve category-wise for each attribute

It is also possible to show a subset of variables or plot different variables on the rows and columns

- Feel free to experiment this on your own

Finding correlations among attributes

- We can find the level of correlation b/w different attributes (variables)

But what exactly is a correlation?

- Two variables are correlated when **they change in same/opposite direction**

We can check coefficient of correlation using `corr()`

```
top3_data.corr()
```

```
<ipython-input-11-c78d7a78d920>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a
  top3_data.corr()
```

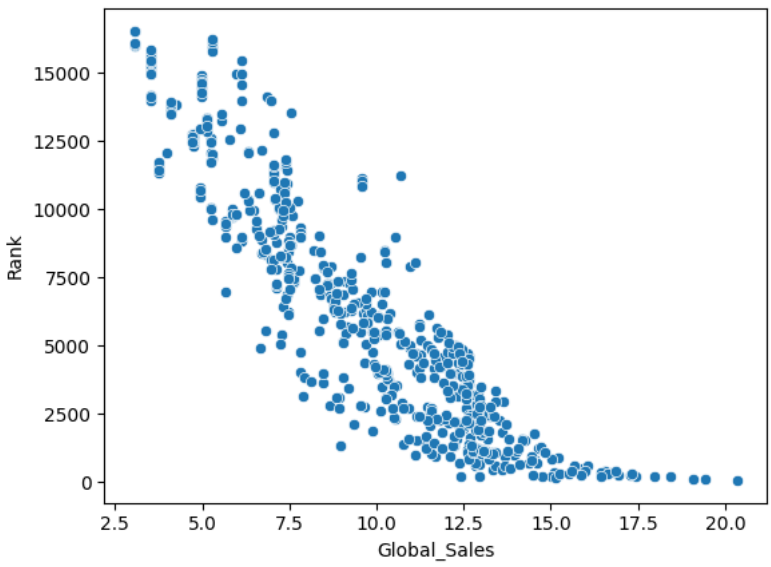
	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Rank	1.000000	0.328705	-0.873726	-0.735711	0.115459	-0.857567	-0.911721
Year	0.328705	1.000000	-0.354256	-0.178026	0.055864	-0.239876	-0.280351
NA_Sales	-0.873726	-0.354256	1.000000	0.617483	-0.233315	0.794353	0.856300
EU_Sales	-0.735711	-0.178026	0.617483	1.000000	-0.208249	0.771105	0.864147
JP_Sales	0.115459	0.055864	-0.233315	-0.208249	1.000000	-0.355825	-0.014193
Other_Sales	-0.857567	-0.239876	0.794353	0.771105	-0.355825	1.000000	0.878816
Global_Sales	-0.911721	-0.280351	0.856300	0.864147	-0.014193	0.878816	1.000000

- Higher the **MAGNITUDE** of coefficient of correlation, more the variables are **correlated**
- The **sign just determines the direction of change**
  - `+` means increase in value of one variable causes increase in value of other variable
  - `–` means increase in value of one variable causes decrease in value of other variable, and vice versa

As you can see, `Global Sales` and `Rank` have the highest correlation coeff of -0.91

Let's plot it using scatter plot

```
sns.scatterplot(x= 'Global_Sales', y= 'Rank', data = top3_data)
plt.show()
```



- When `petal_length` increases, `petal_width` also increases

But Remember

Correlation does NOT mean Causation

- We cannot conclude that change in values of a variable is causing change in values of other variable

Now, Let's look at a way to visualize correlation among variables

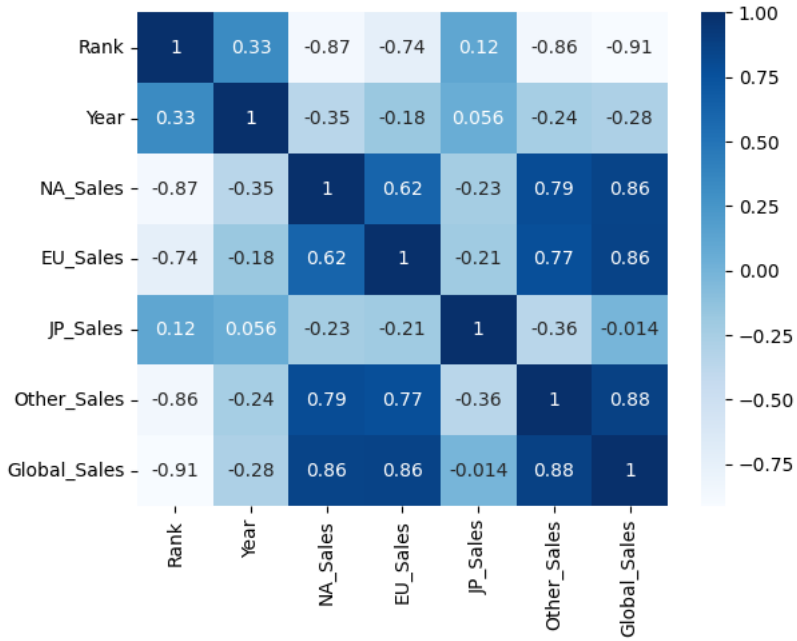
Heat Map

- A heat map plots rectangular data as a color-encoded matrix.
- **Stronger the colour, stronger the correlation b/w the variables**

Let's plot a Heat Map using correlation coefficient matrix generated using `corr()`

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

```
<ipython-input-13-4220aa71e94f>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a
  sns.heatmap(top3_data.corr(), cmap= "Blues", annot=True)
```



- **`annot=True`** is for writing correlation coeff inside each cell

You can change the colours of cells in Heat Map if you like

- There are a lot of options available!

```
print(plt.colormaps())
```

```
['magma', 'inferno', 'plasma', 'viridis', 'cividis', 'twilight', 'twilight_shifted', 'turbo', 'Blues', 'BrBG', 'BuGn', '
```

```
sns.heatmap(top3_data.corr(), cmap= "coolwarm", annot=True)
plt.show()
```

```
<ipython-input-15-72b457990d68>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In
  sns.heatmap(top3_data.corr(), cmap= "coolwarm", annot=True)
```