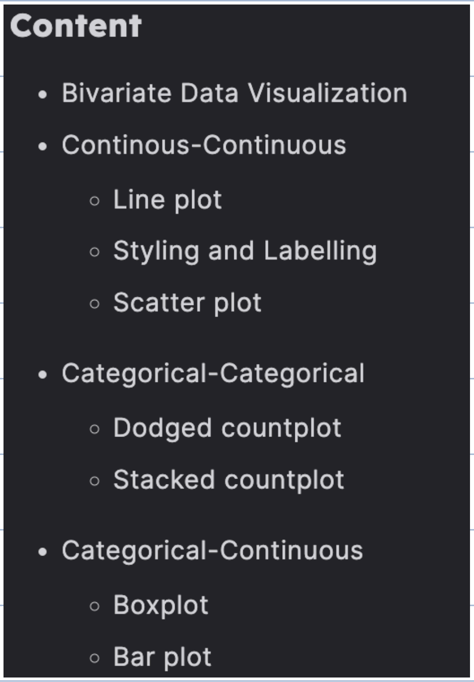
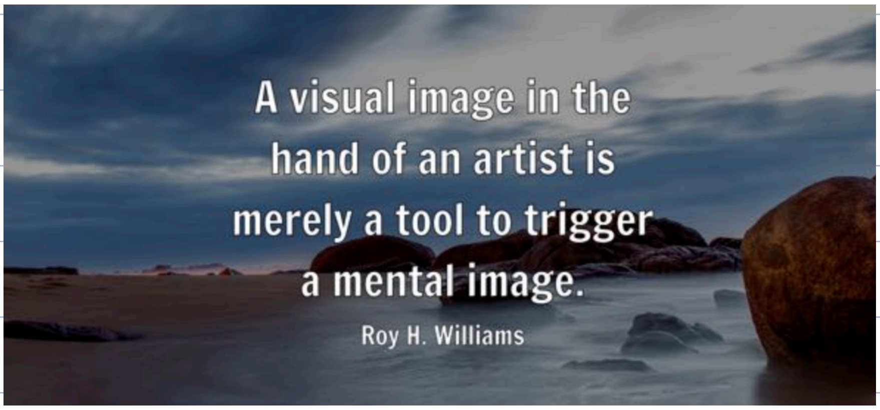


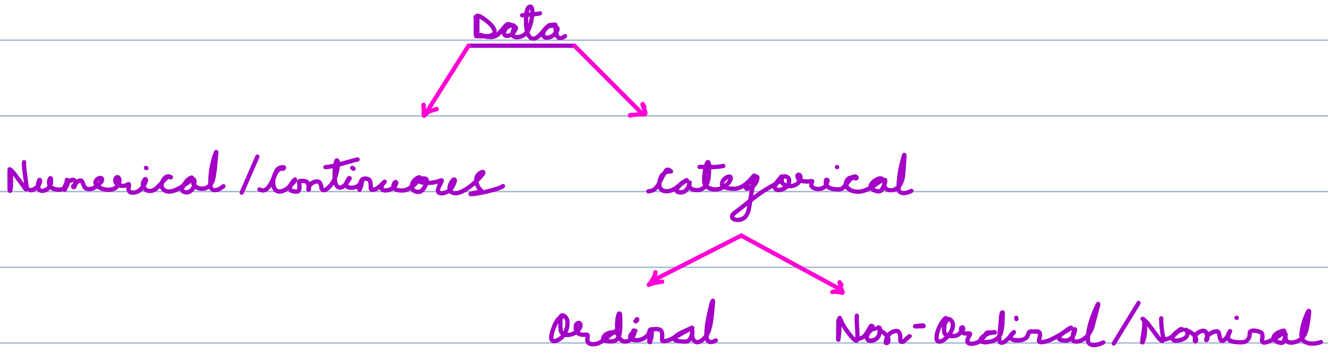
Content

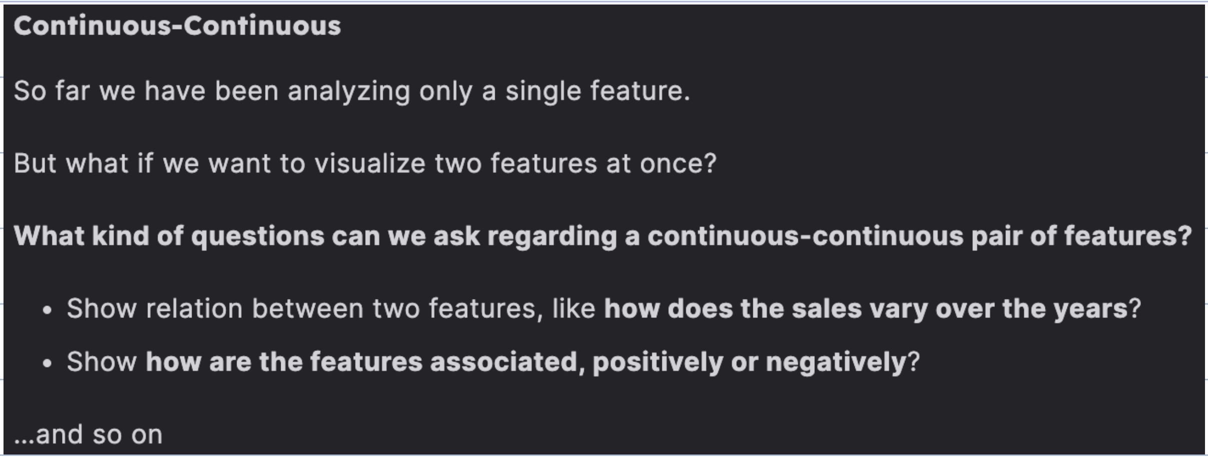
- Bivariate Data Visualization
- Continous-Continuous
  - Line plot
  - Styling and Labelling
  - Scatter plot
- Categorical-Categorical
  - Dodged countplot
  - Stacked countplot
- Categorical-Continuous
  - Boxplot
  - Bar plot



A visual image in the hand of an artist is merely a tool to trigger a mental image.

Roy H. Williams





Continuous-Continuous

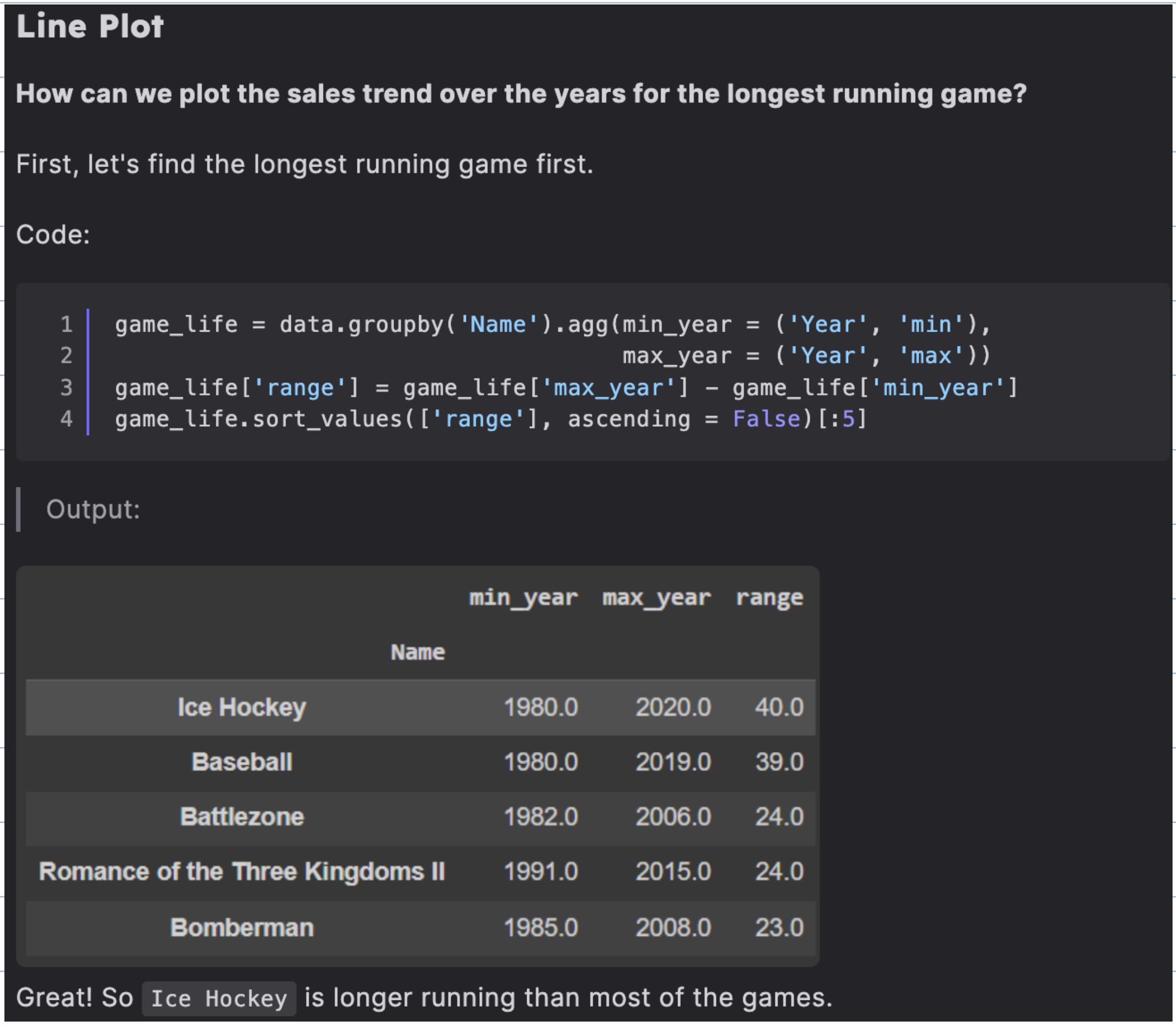
So far we have been analyzing only a single feature.

But what if we want to visualize two features at once?

What kind of questions can we ask regarding a continuous-continuous pair of features?

- Show relation between two features, like **how does the sales vary over the years**?
- Show **how are the features associated, positively or negatively**?

...and so on



Line Plot

How can we plot the sales trend over the years for the longest running game?

First, let's find the longest running game first.

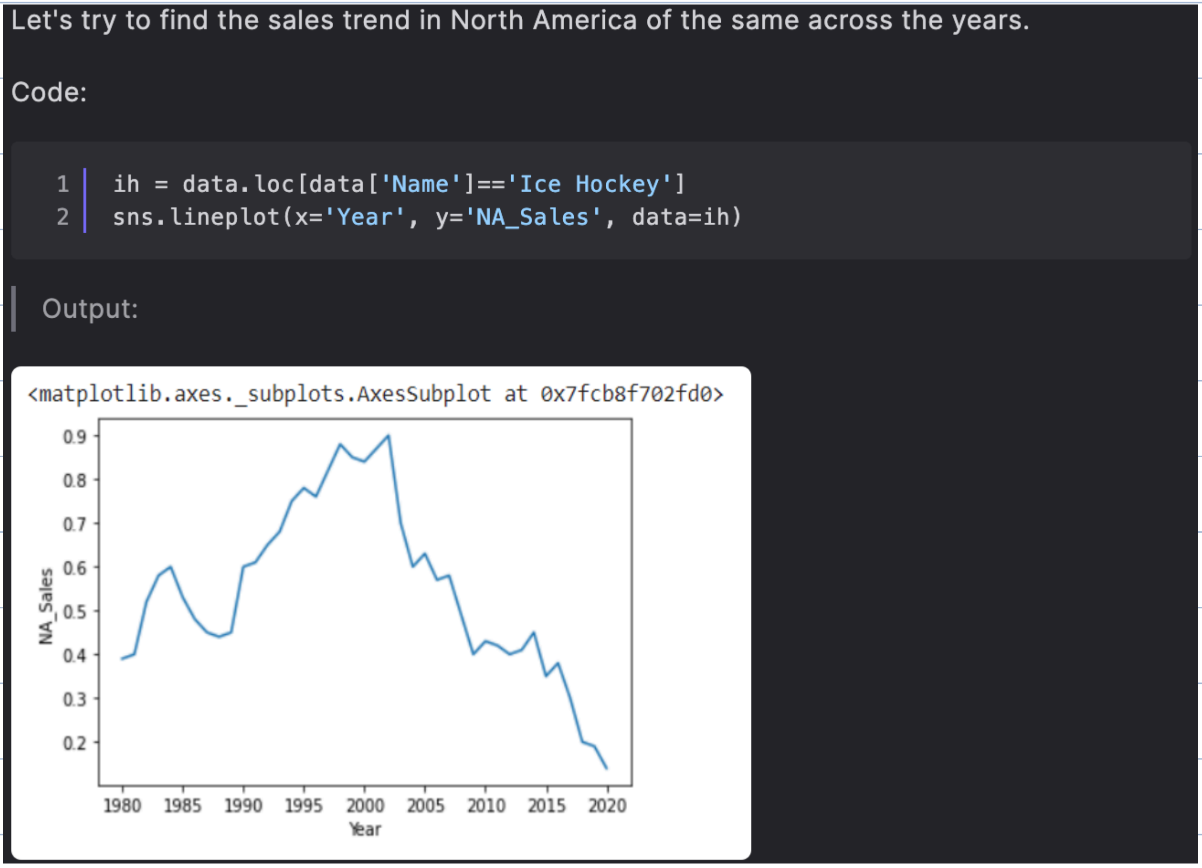
Code:

```
game_life = data.groupby('Name').agg(min_year = ('Year', 'min'),
                                     max_year = ('Year', 'max'))
game_life['range'] = game_life['max_year'] - game_life['min_year']
game_life.sort_values(['range'], ascending = False)[:5]
```

> Output:

	min_year	max_year	range
Name			
Ice Hockey	1980.0	2020.0	40.0
Baseball	1980.0	2019.0	39.0
Battlezone	1982.0	2006.0	24.0
Romance of the Three Kingdoms II	1991.0	2015.0	24.0
Bomberman	1985.0	2008.0	23.0

Great! So `Ice Hockey` is longer running than most of the games.



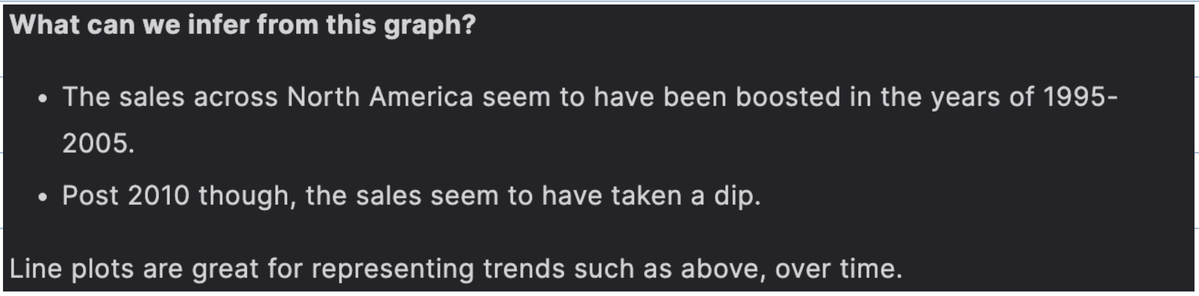
Let's try to find the sales trend in North America of the same across the years.

Code:

```
ih = data.loc[data['Name']=='Ice Hockey']
sns.lineplot(x='Year', y='NA_Sales', data=ih)
```

> Output:

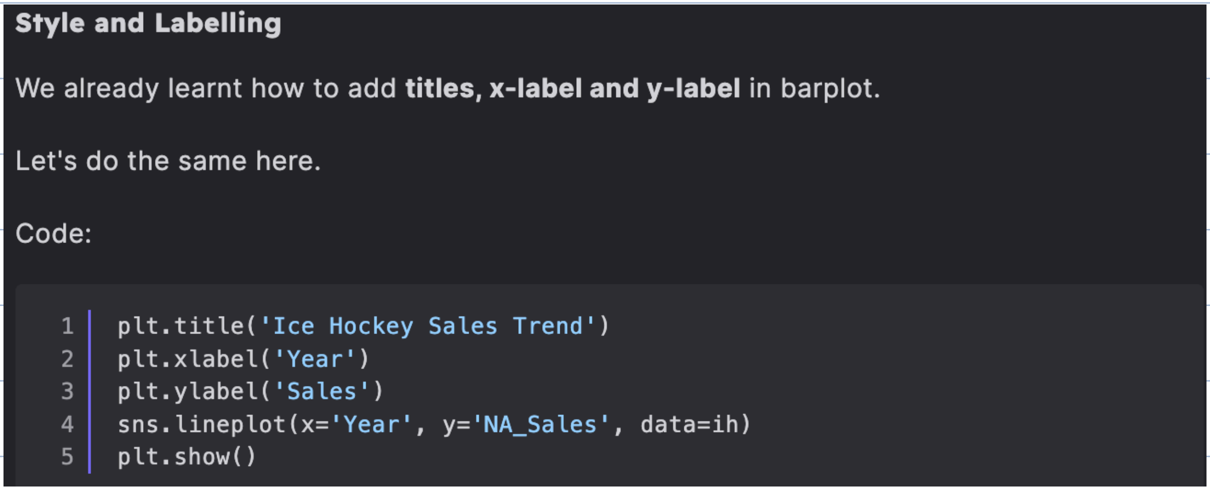
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8f702fd0>
```



What can we infer from this graph?

- The sales across North America seem to have been boosted in the years of 1995-2005.
- Post 2010 though, the sales seem to have taken a dip.

Line plots are great for representing trends such as above, over time.



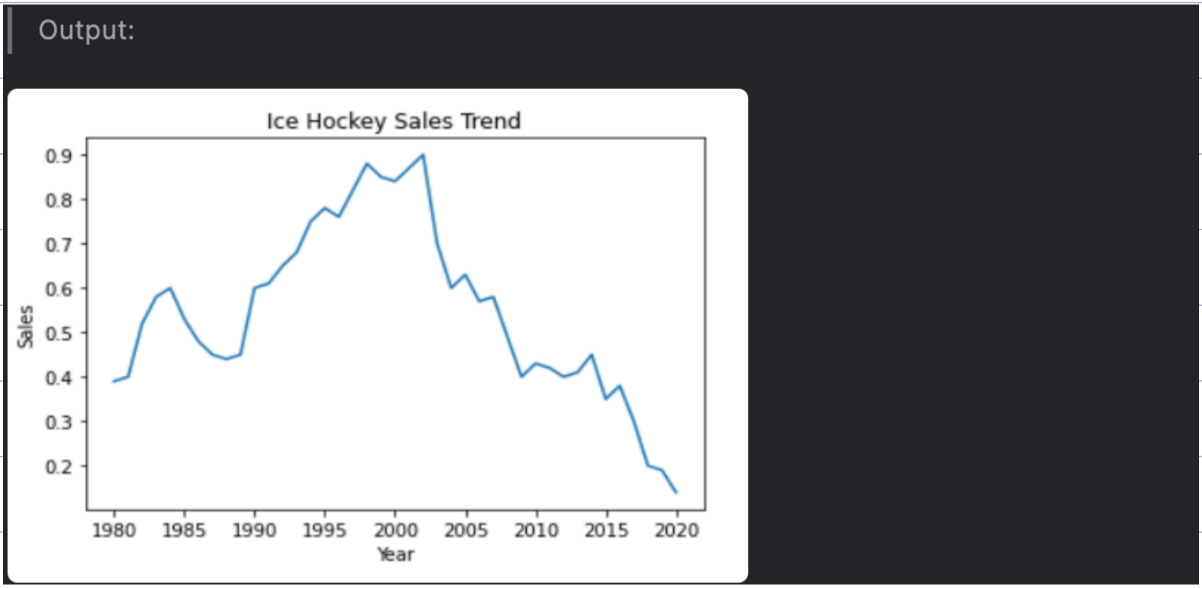
Style and Labelling

We already learnt how to add **titles, x-label and y-label** in barplot.

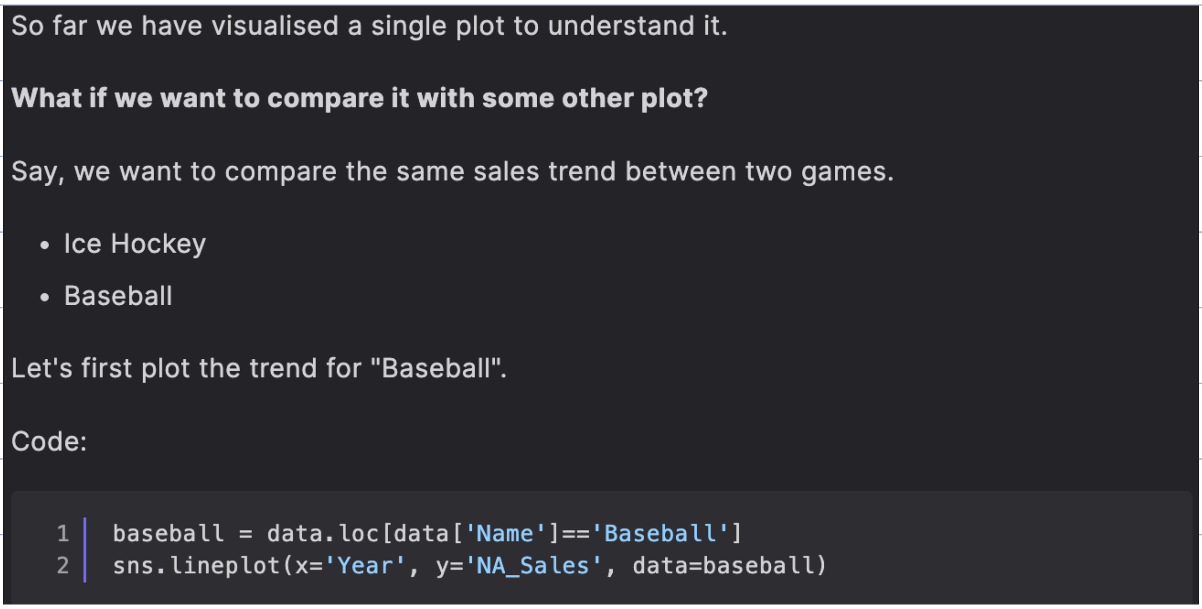
Let's do the same here.

Code:

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



Output:



So far we have visualised a single plot to understand it.

What if we want to compare it with some other plot?

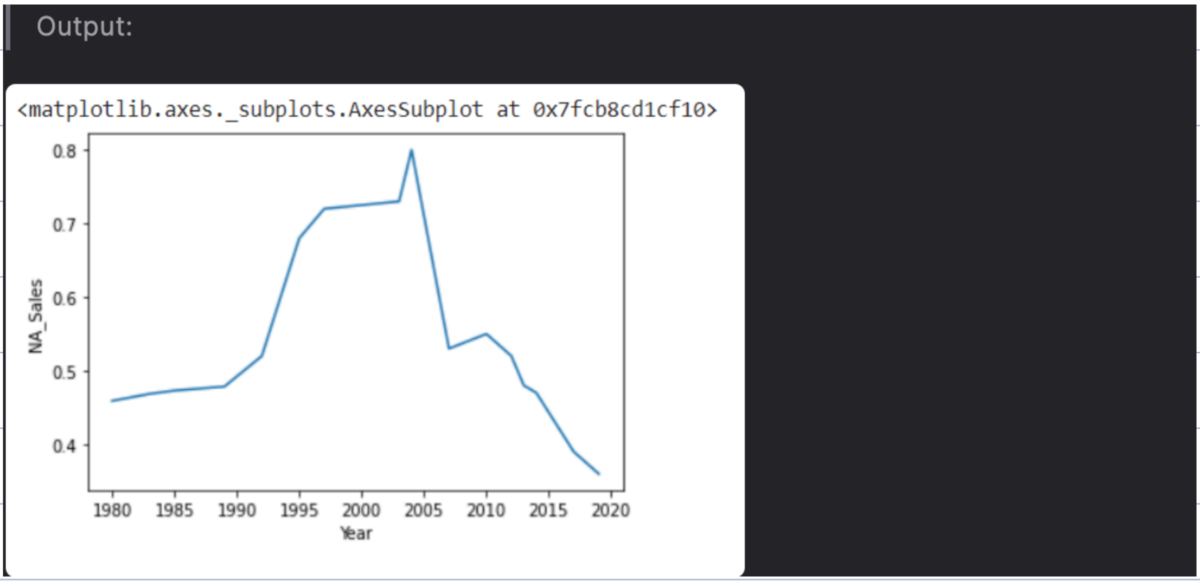
Say, we want to compare the same sales trend between two games.

- Ice Hockey
- Baseball

Let's first plot the trend for "Baseball".

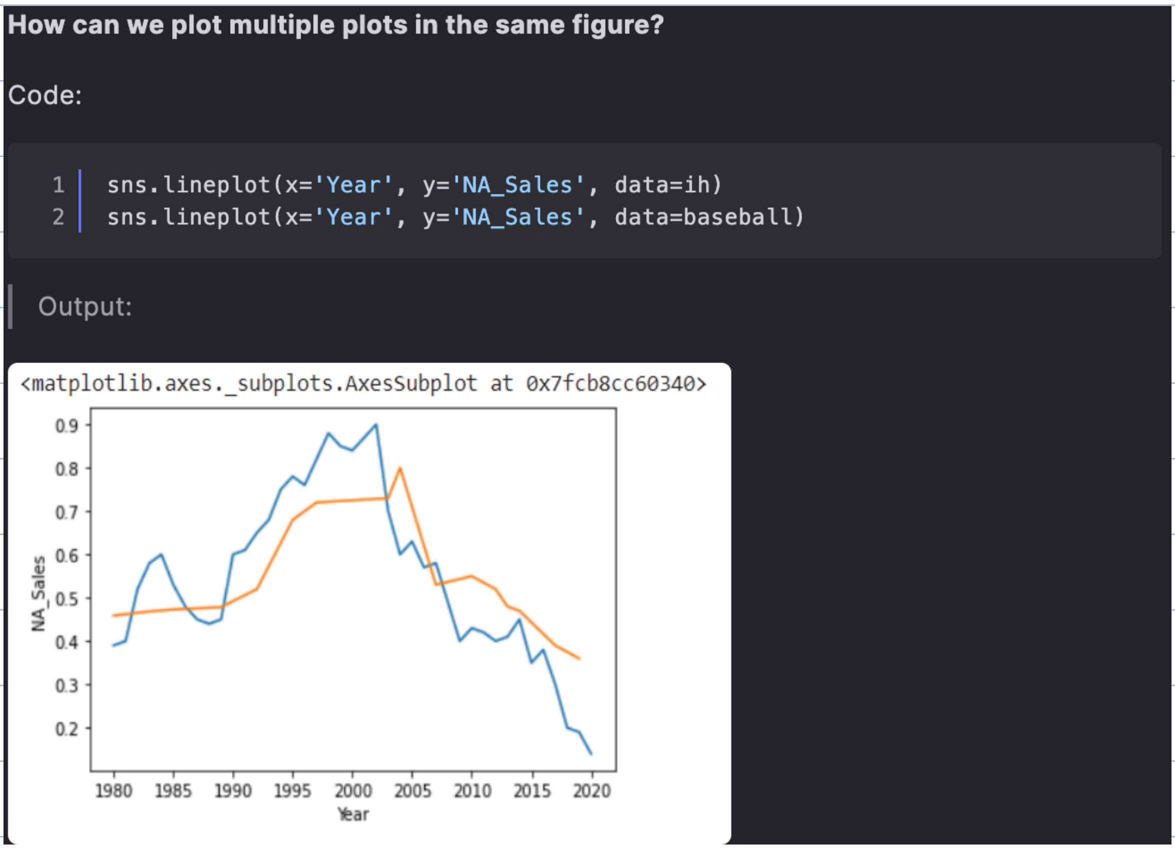
Code:

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



Output:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8cd1cf10>
```



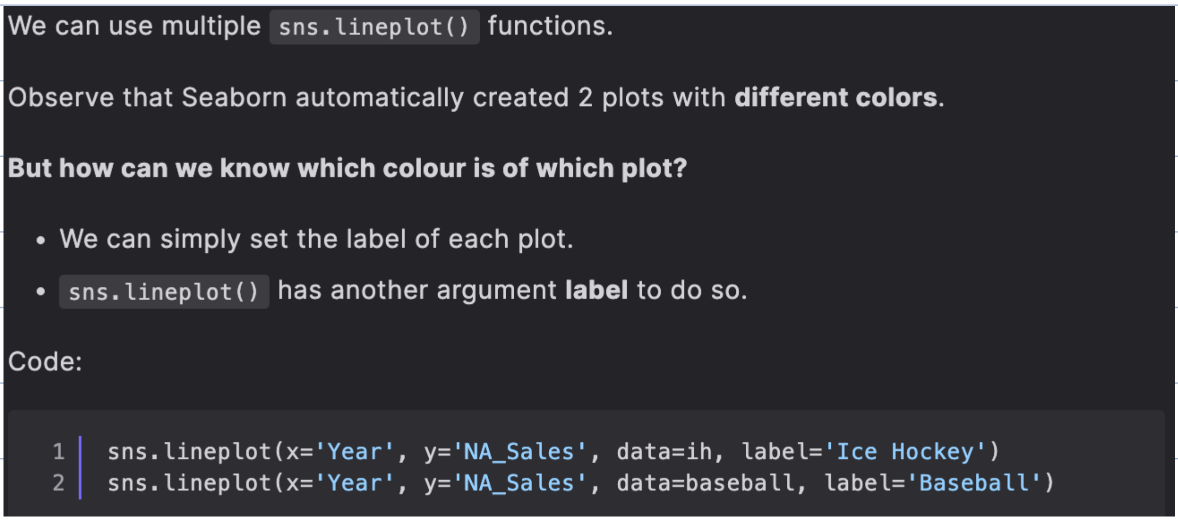
How can we plot multiple plots in the same figure?

Code:

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

> Output:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8cc60340>
```



We can use multiple `sns.lineplot()` functions.

Observe that Seaborn automatically created 2 plots with **different colors**.

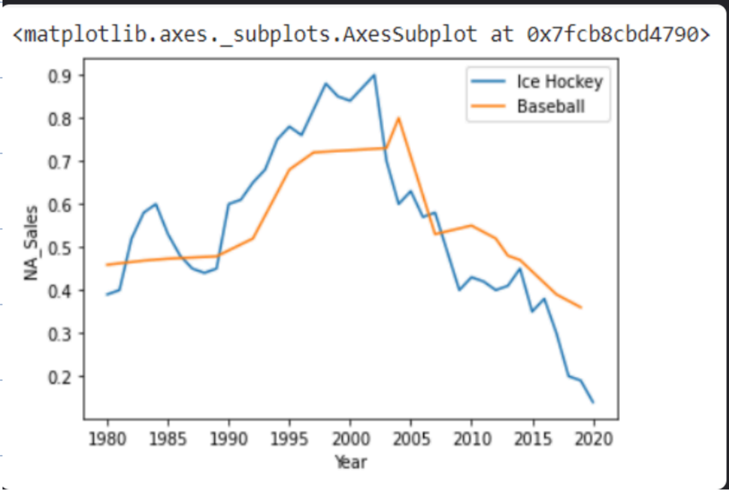
But how can we know which colour is of which plot?

- We can simply set the label of each plot.
- `sns.lineplot()` has another argument **label** to do so.

Code:

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

Output:



```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8cbd4790>
```

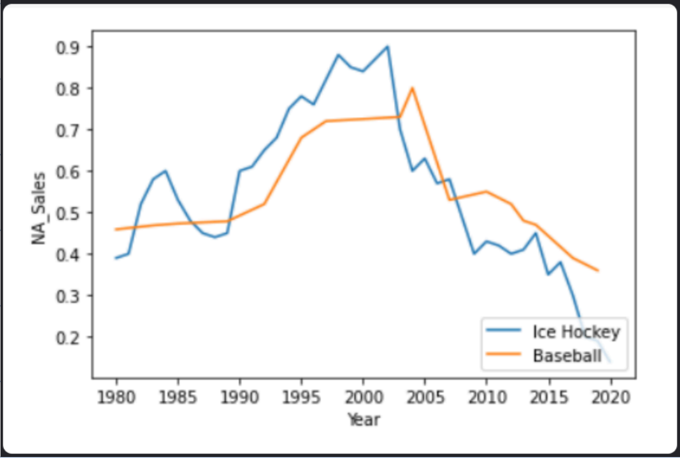
Can we change the position of the legend, say to bottom-right corner?

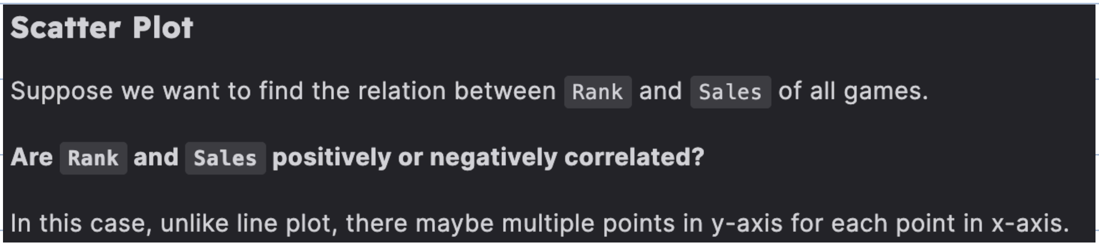
- Matplotlib automatically decides the best position for the legends.
- But we can also change it using the `loc` parameter.
- `loc` takes input as 1 of following strings:
    - upper center
    - upper left
    - upper right
    - lower right

Code:

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

Output:



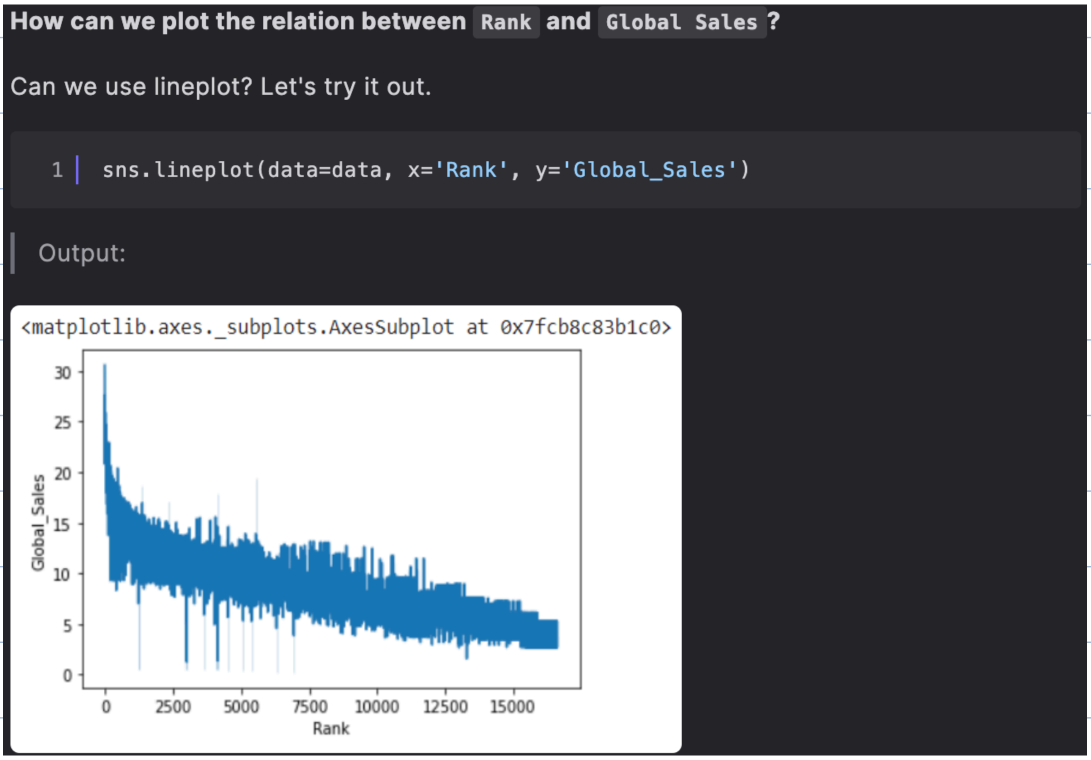


Scatter Plot

Suppose we want to find the relation between `Rank` and `Sales` of all games.

Are `Rank` and `Sales` positively or negatively correlated?

In this case, unlike line plot, there maybe multiple points in y-axis for each point in x-axis.



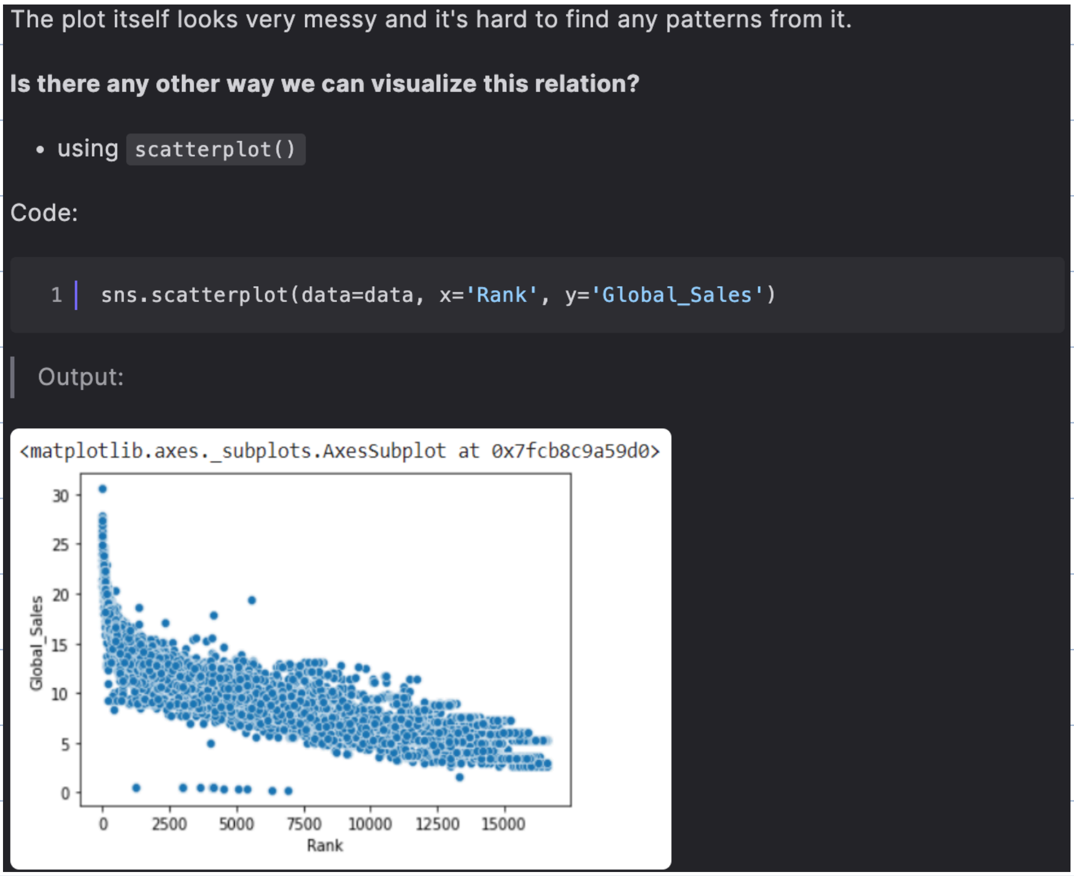
How can we plot the relation between `Rank` and `Global_Sales`?

Can we use lineplot? Let's try it out.

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

> Output:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8c83b1c0>
```



The plot itself looks very messy and it's hard to find any patterns from it.

Is there any other way we can visualize this relation?

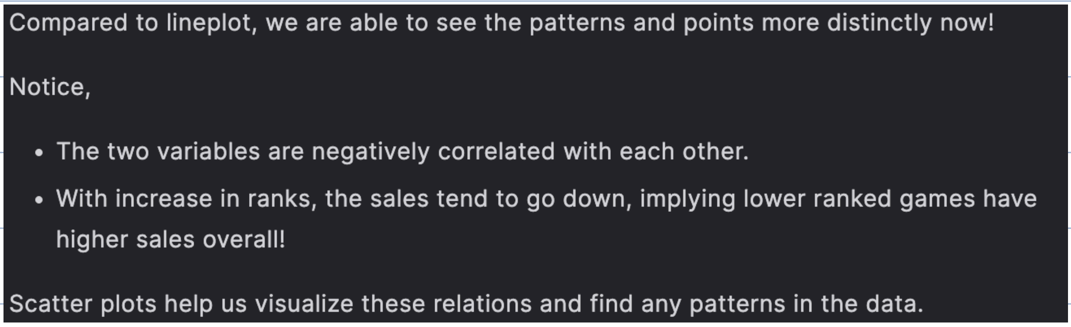
- using `scatterplot()`

Code:

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

> Output:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcb8c9a59d0>
```

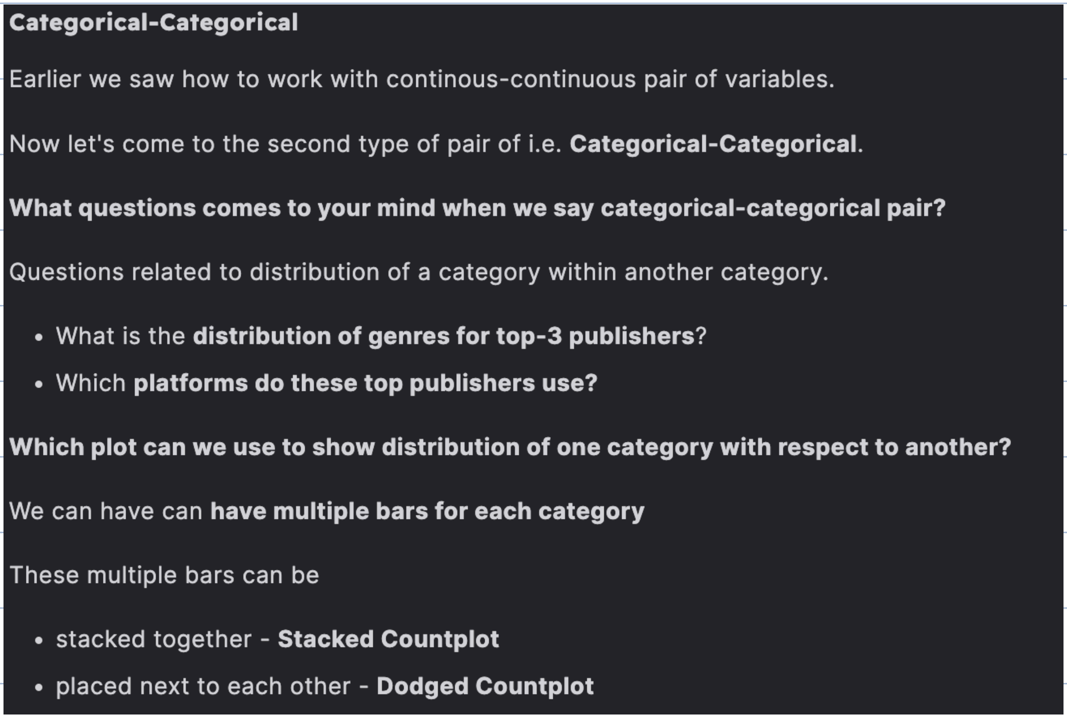


Compared to lineplot, we are able to see the patterns and points more distinctly now!

Notice,

- The two variables are negatively correlated with each other.
- With increase in ranks, the sales tend to go down, implying lower ranked games have higher sales overall!

Scatter plots help us visualize these relations and find any patterns in the data.



Categorical-Categorical

Earlier we saw how to work with continous-continuous pair of variables.

Now let's come to the second type of pair of i.e. **Categorical-Categorical**.

What questions comes to your mind when we say categorical-categorical pair?

Questions related to distribution of a category within another category.

- What is the **distribution of genres for top-3 publishers**?
- Which **platforms do these top publishers use?**

Which plot can we use to show distribution of one category with respect to another?

We can have can **have multiple bars for each category**

These multiple bars can be

- stacked together - **Stacked Countplot**
- placed next to each other - **Dodged Countplot**

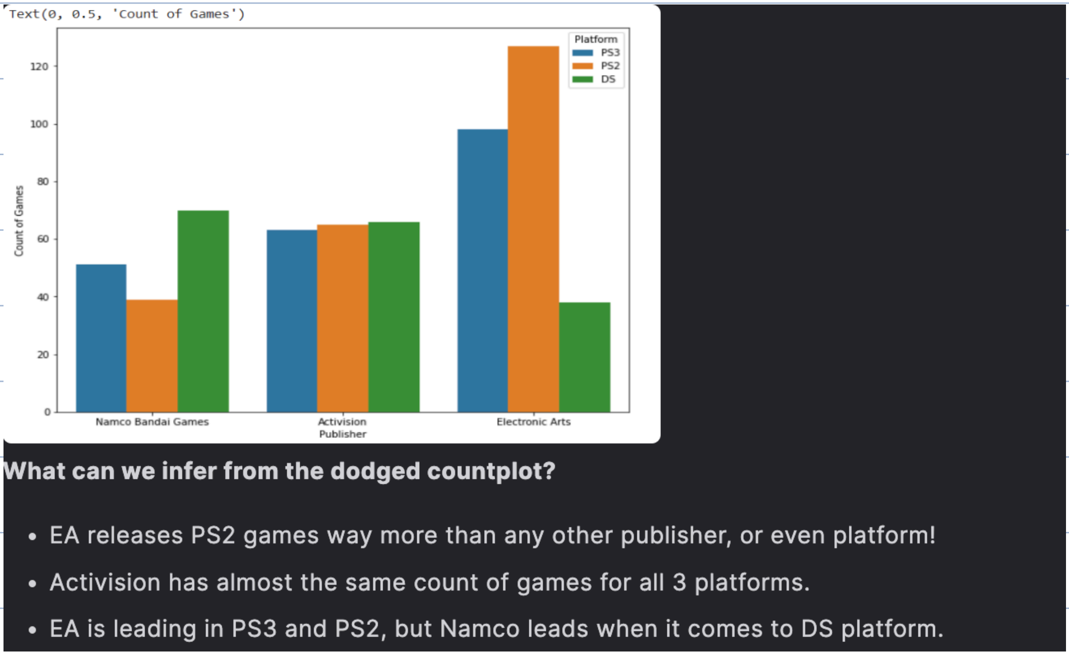
Dodged Countplot

How can we compare the top 3 platforms these publishers use?

- We can use a **dodged countplot** in this case.

Code:

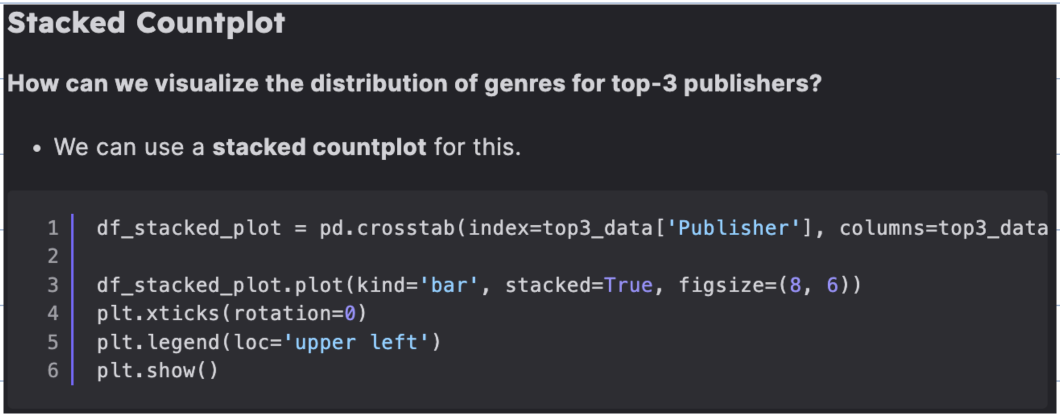
```
plt.figure(figsize=(10,8))
sns.countplot(x='Publisher', hue='Platform', data=top3_data)
plt.ylabel('Count of Games')
```



```
Text(0, 0.5, 'Count of Games')
```

What can we infer from the dodged countplot?

- EA releases PS2 games way more than any other publisher, or even platform!
- Activision has almost the same count of games for all 3 platforms.
- EA is leading in PS3 and PS2, but Namco leads when it comes to DS platform.



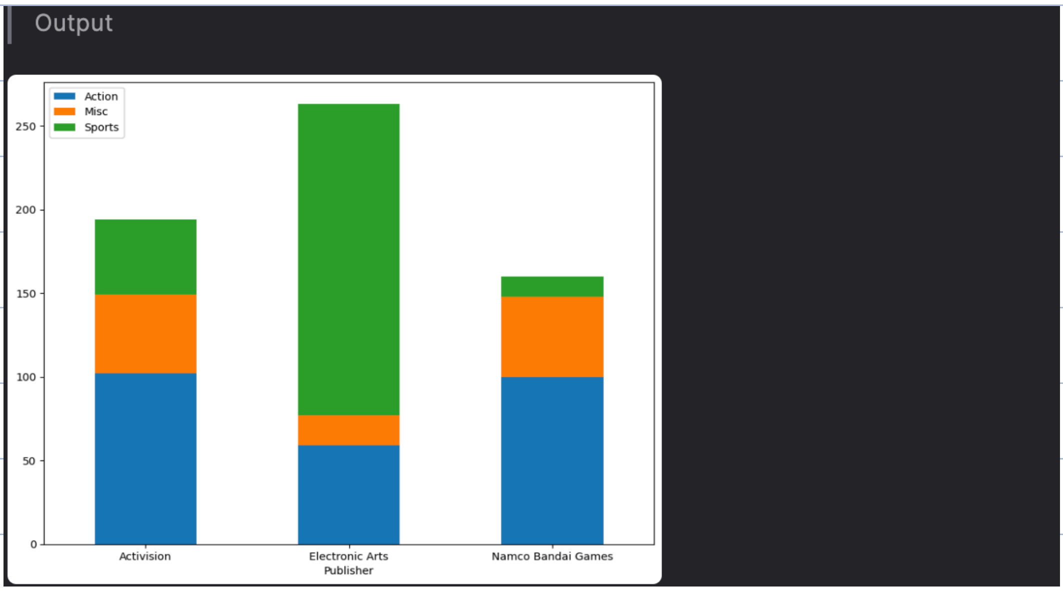
Stacked Countplot

How can we visualize the distribution of genres for top-3 publishers?

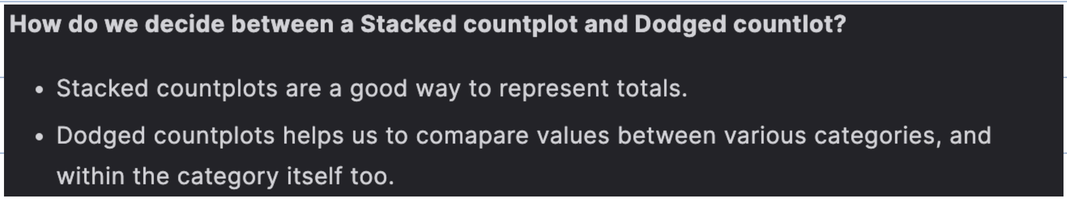
- We can use a **stacked countplot** for this.

```
df_stacked_plot = pd.crosstab(index=top3_data['Publisher'], columns=top3_data

df_stacked_plot.plot(kind='bar', stacked=True, figsize=(8, 6))
plt.xticks(rotation=0)
plt.legend(loc='upper left')
plt.show()
```

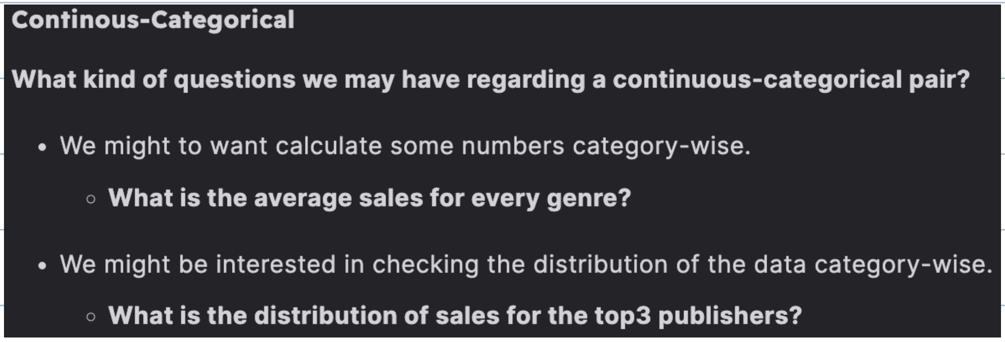


Output



How do we decide between a Stacked countplot and Dodged countlot?

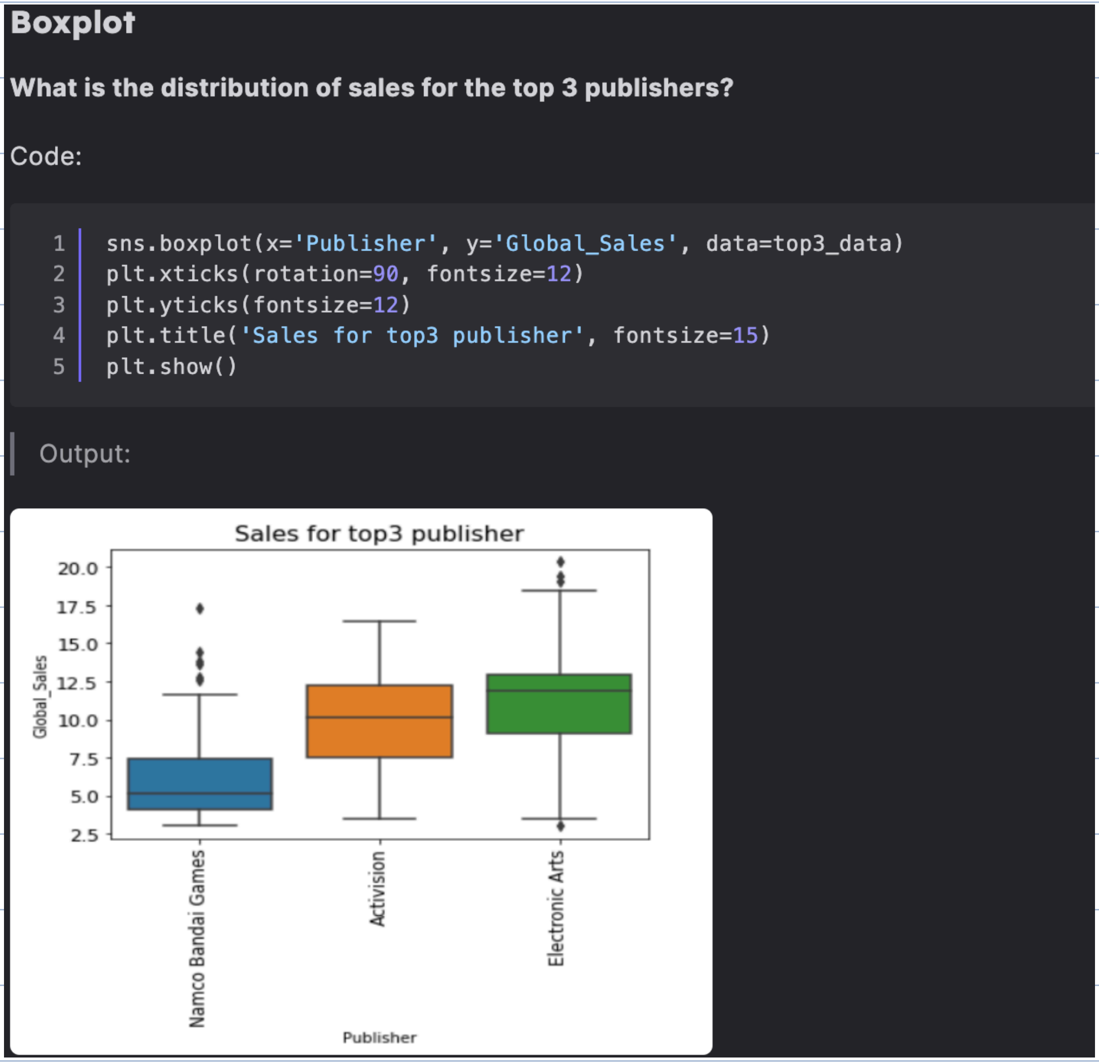
- Stacked countplots are a good way to represent totals.
- Dodged countplots helps us to comapare values between various categories, and within the category itself too.



Continous-Categorical

What kind of questions we may have regarding a continuous-categorical pair?

- We might to want calculate some numbers category-wise.
  - **What is the average sales for every genre?**
- We might be interested in checking the distribution of the data category-wise.
  - **What is the distribution of sales for the top3 publishers?**



Boxplot

What is the distribution of sales for the top 3 publishers?

Code:

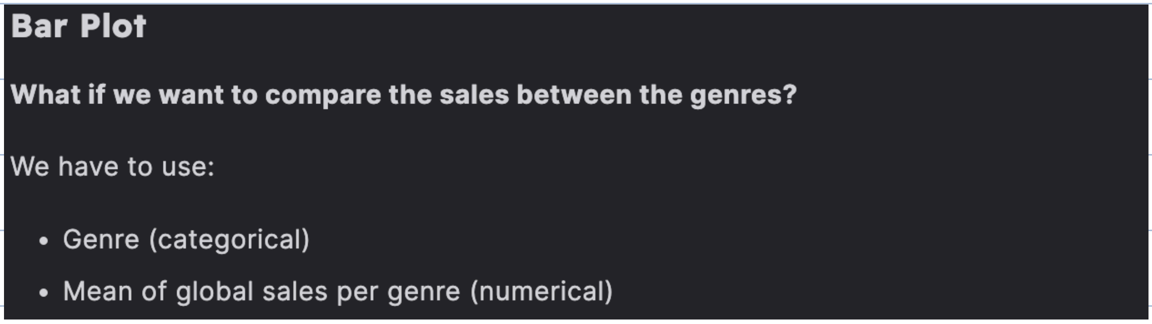
```
sns.boxplot(x='Publisher', y='Global_Sales', data=top3_data)
plt.xticks(rotation=90, fontsize=12)
plt.yticks(fontsize=12)
plt.title('Sales for top3 publisher', fontsize=15)
plt.show()
```

> Output:

Box-plot

sns. boxplot(df['Global_Sales']) → D

sns. boxplot(data = df , x = 'Genre', y = 'Global_Sales') → 2D

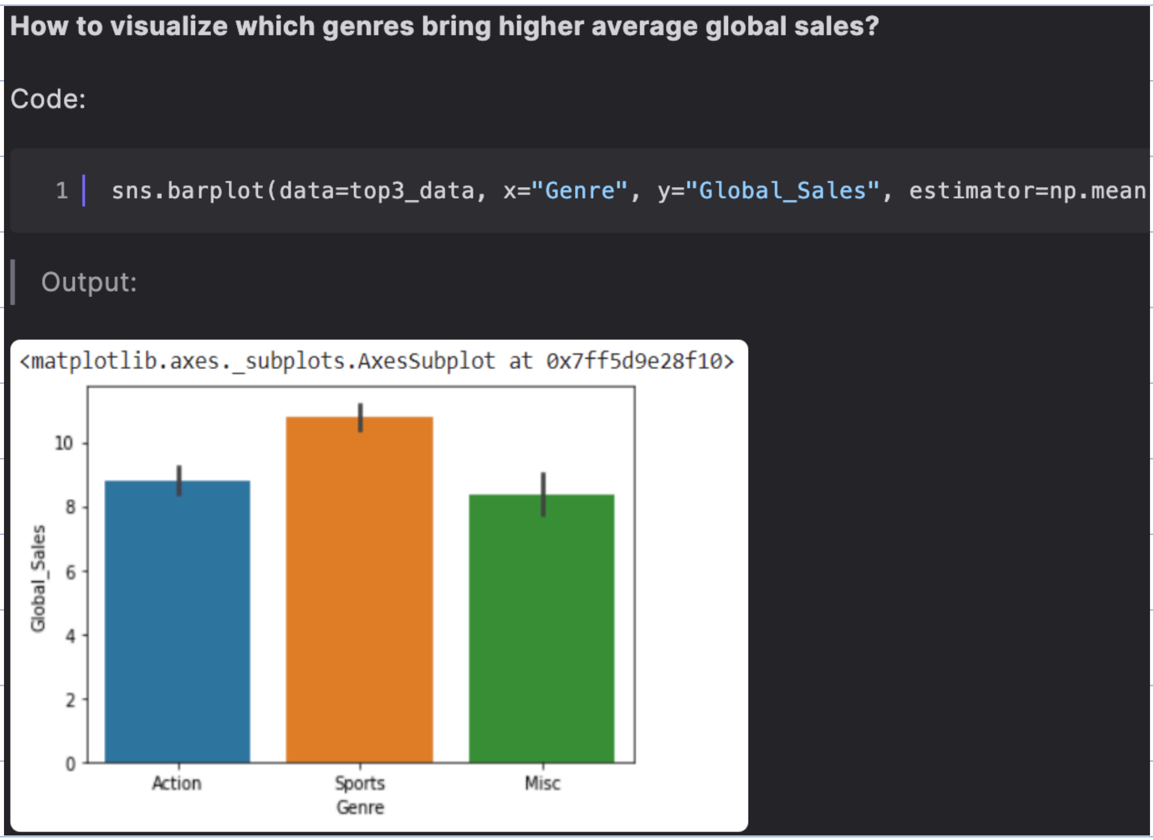


Bar Plot

What if we want to compare the sales between the genres?

We have to use:

- Genre (categorical)
- Mean of global sales per genre (numerical)



How to visualize which genres bring higher average global sales?

Code:

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

> Output:

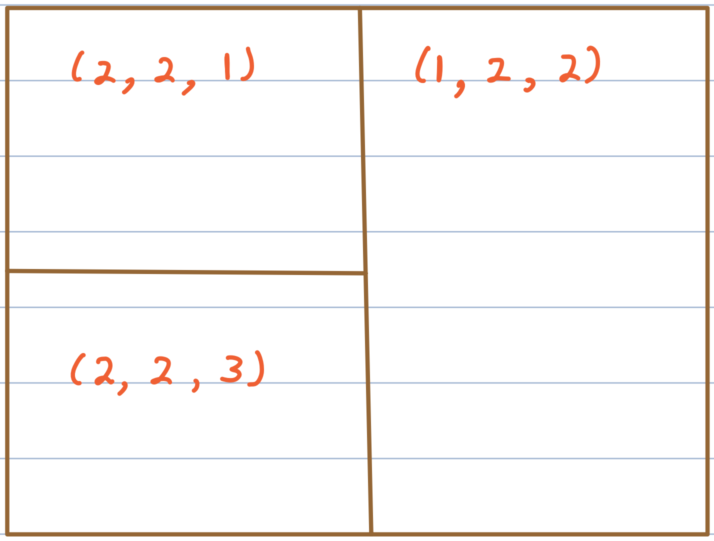
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff5d9e28f10>
```

Subplots & Axes

✓subplot → 1 axes → 0, 0	2 0, 1	3 0, 2
4 1, 0	5 1, 1	6 1, 2

2 * 3

subplot



$\underbrace{2, 2}_{4\ parts}, \_$