

Seaborn

```
import seaborn as sns
```

Read CSV files

```
sns.get_dataset_names()
```

To get a dataset provided by Seaborn

→ ['anagrams', 'anscombe', 'attention', 'brain_networks',....]

```
tips = sns.load_dataset("tips")
iris = sns.load_dataset("iris")
titanic = sns.load_dataset("titanic")
planets = sns.load_dataset("planets")
#or for a custom dataset:
data = pd.read_csv("data.csv")
```

```
tips.columns
```

→ Index(['total_bill', 'tip', 'sex', 'smoker', 'day', 'time', 'size'], dtype='object')

Scatter Plot

```
sns.scatterplot(x="tip", y="total_bill", data=tips, hue="day")
```

`x="column name", y="column name"` ⇒ sets the column name as the variable on the x and y-axis

`data="name of the dataset"`

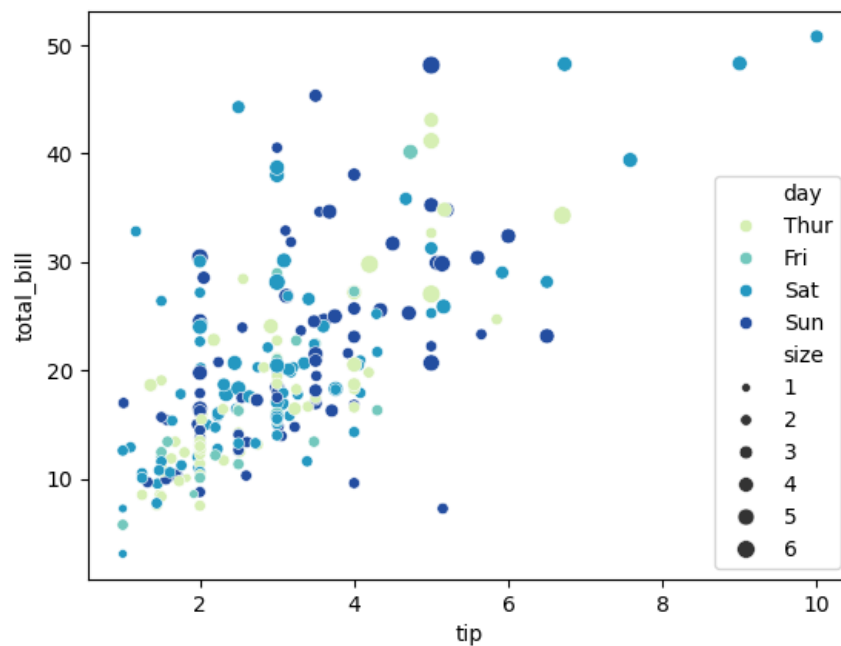
`hue="day"` ⇒ adds color to the points based on the `day` column

`size="size"` ⇒ sets the size of the points based on the `size` column.

`palette="YlGnBu"` ⇒ sets the color palette ranging from yellow to green to blue for the plot.

OR

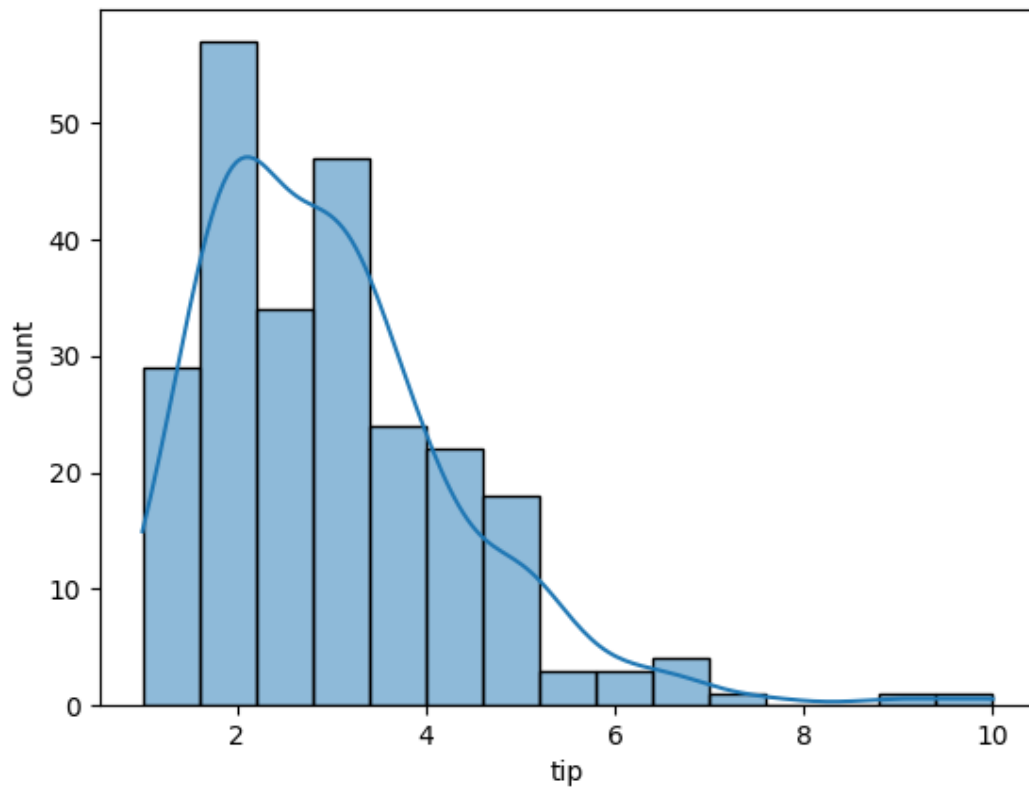
```
sns.scatterplot(data, x, y, hue, size, style, palette,...)
```



Histograms & Distribution Plots

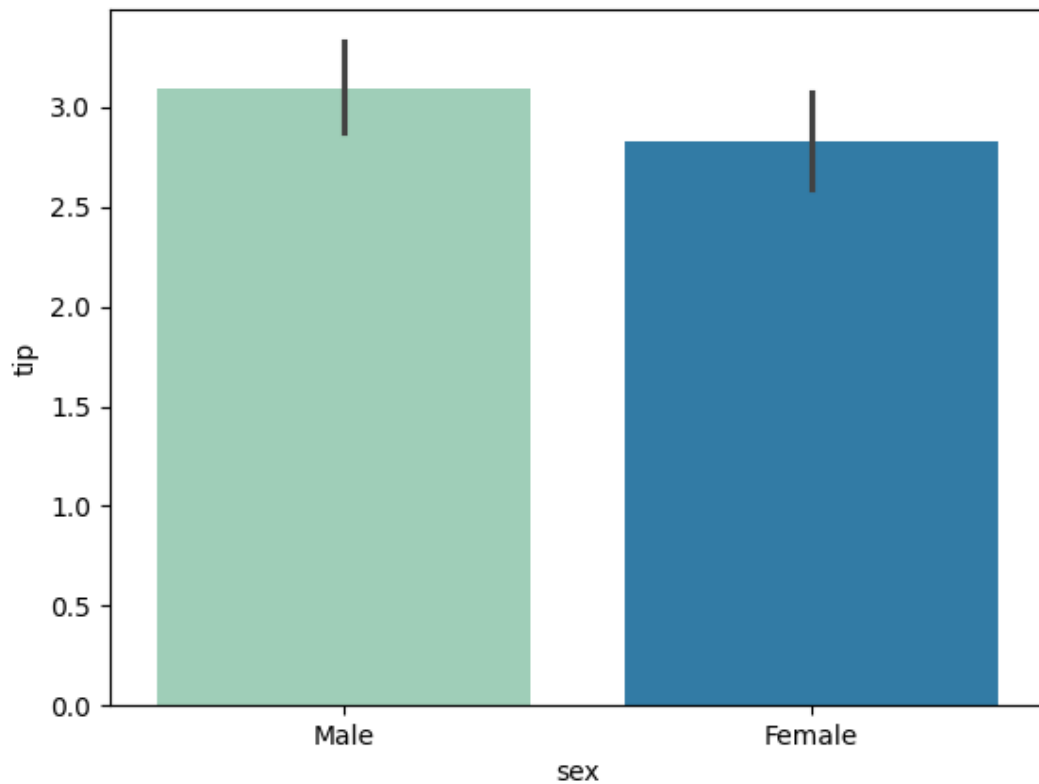
```
sns.histplot(tips['tip'], kde=True, bins=30)
# histplot is similar to displot
sns.displot(tips['tip'], kde=True, bins=30)
# but don't use distplt with t
```

`kde=True` ⇒ To get the line.



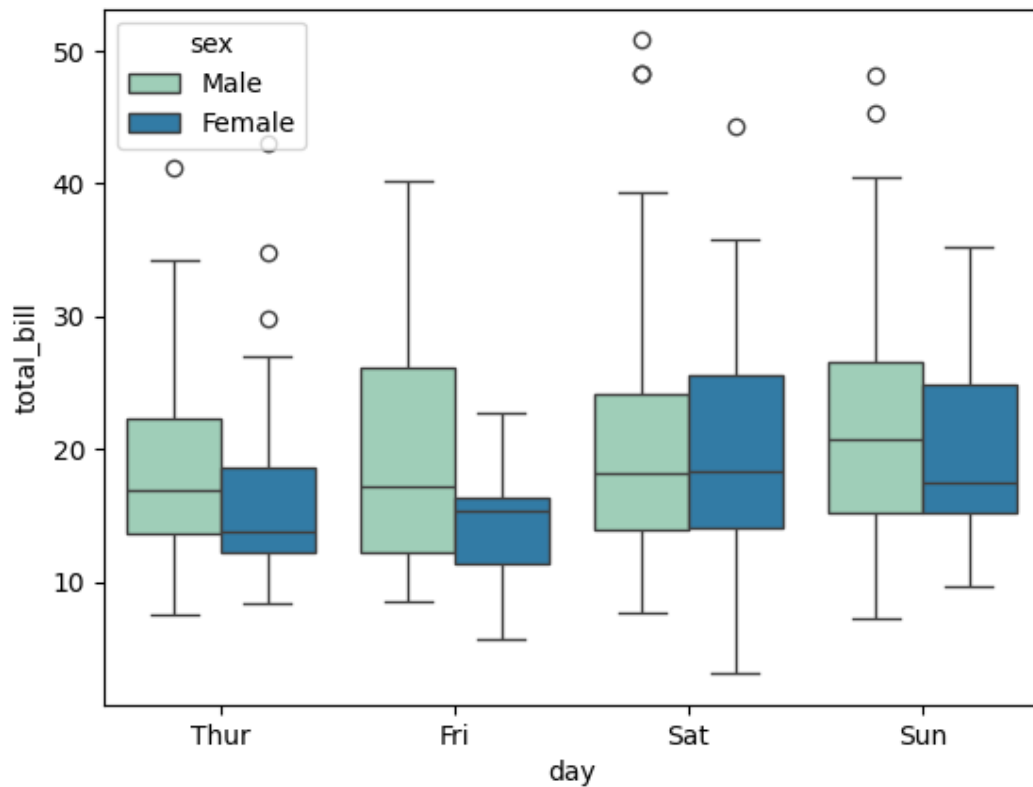
Bar Plot

```
sns.barplot(x="sex", y="tip", data=tips, palette="YlGnBu")
```



Box Plot

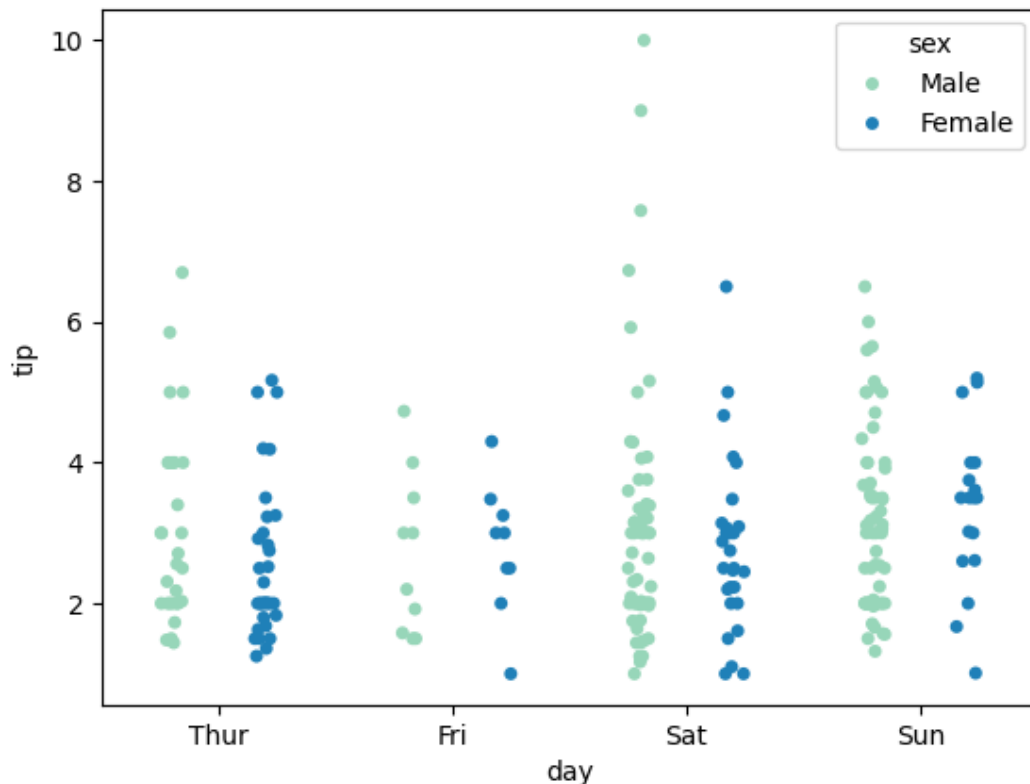
```
sns.boxplot(x="day", y="total_bill", data=tips, hue="sex", pa
```



Strip Plot

```
sns.stripplot(x="day", y="tip", data=tips, hue="sex", palette:
```

`dodge=True` ⇒ To separate the males and females in this code

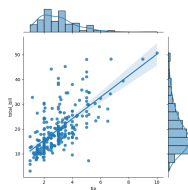


Join Plot

Join Plot ⇒ a powerful data visualization tool that combines multiple plot types. Typically, a **Scatter plot** and **Histogram**

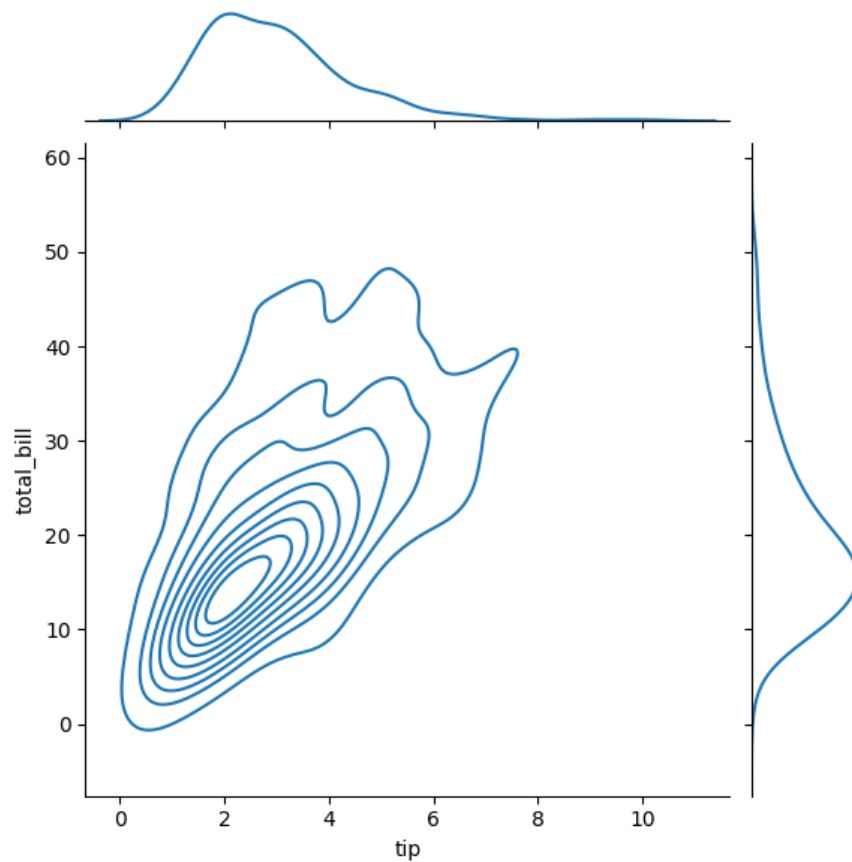
```
sns.jointplot(x="tip", y="total_bill", data=tips, kind="reg")
```

`kind="reg"` ⇒ to add a **regression** line



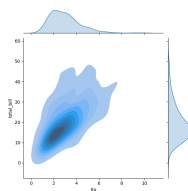
```
sns.jointplot(x="tip", y="total_bill", data=tips, kind="kde")
```

`kind="kde"` ⇒ To make it like a 3-Dimensional plot to show better where the concentration of values is.



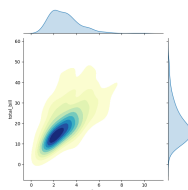
```
sns.jointplot(x="tip", y="total_bill", data=tips, kind="kde",
```

`fill=True` ⇒ To make it visually better.

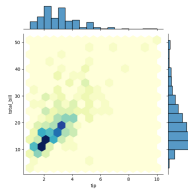


```
sns.jointplot(x="tip", y="total_bill", data=tips, kind="kde",
```

`cmap="YlGnBu"` ⇒ To make the `fill` function influenced by setting a color map

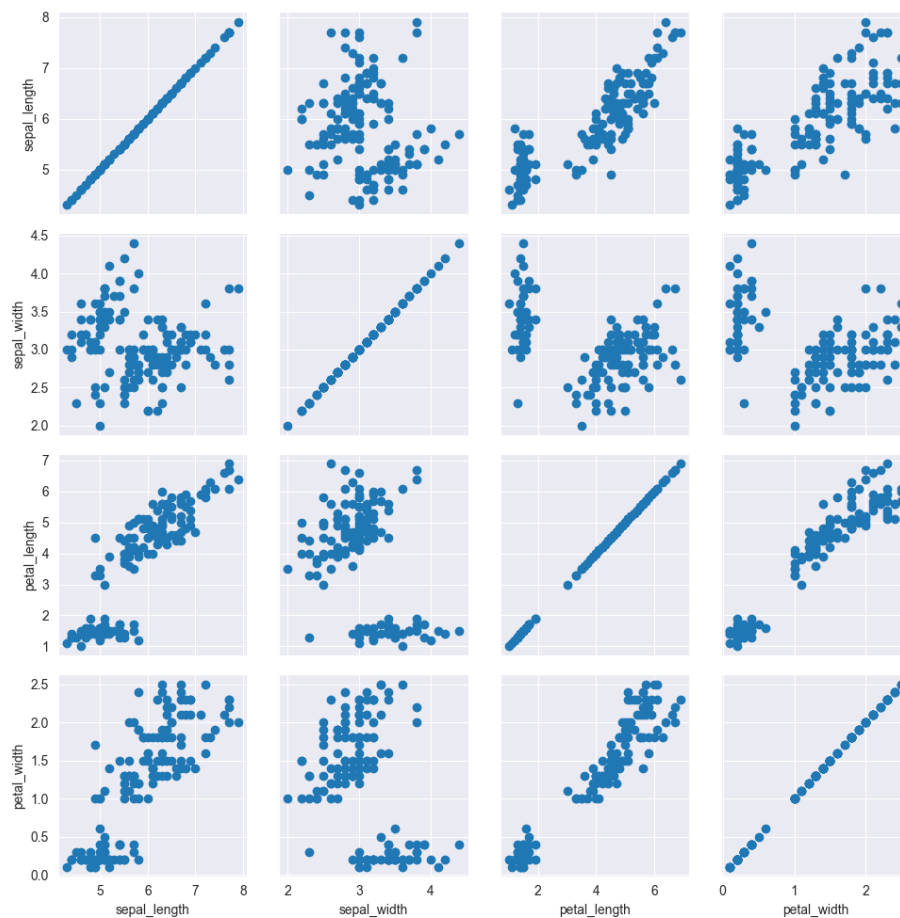


```
sns.jointplot(x="tip", y="total_bill", data=tips, kind="hex",
```



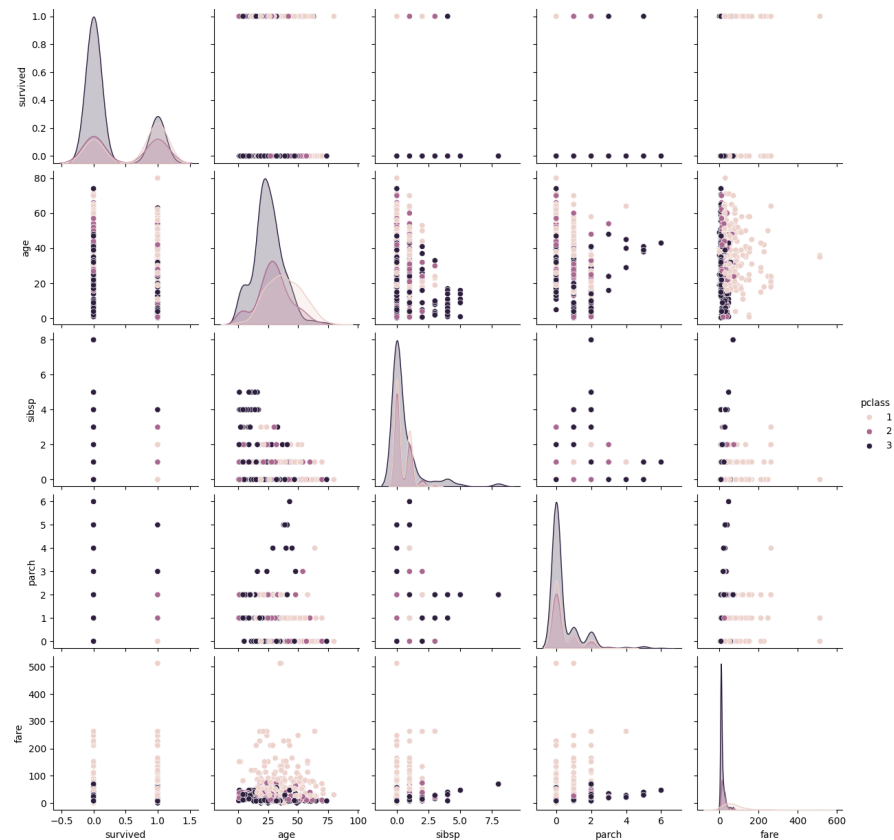
Pair Grid

```
# Create a PairGrid  
h = sns.PairGrid(iris)  
# Map a scatter plot to the grid  
h = h.map(plt.scatter)
```



Piar Plot

```
sns.pairplot(titanic.select_dtypes(['number']), hue="pclass")
```



Heat Map

Heat Map ⇒ Used in identifying patterns, trends, and correlations within the data, and making predictions.

```
sns.heatmap(titanic.corr(), annot=True, cmap="coolwarm")
```

`titanic.corr()` ⇒ To get the correlations but may not be easy to get information from.

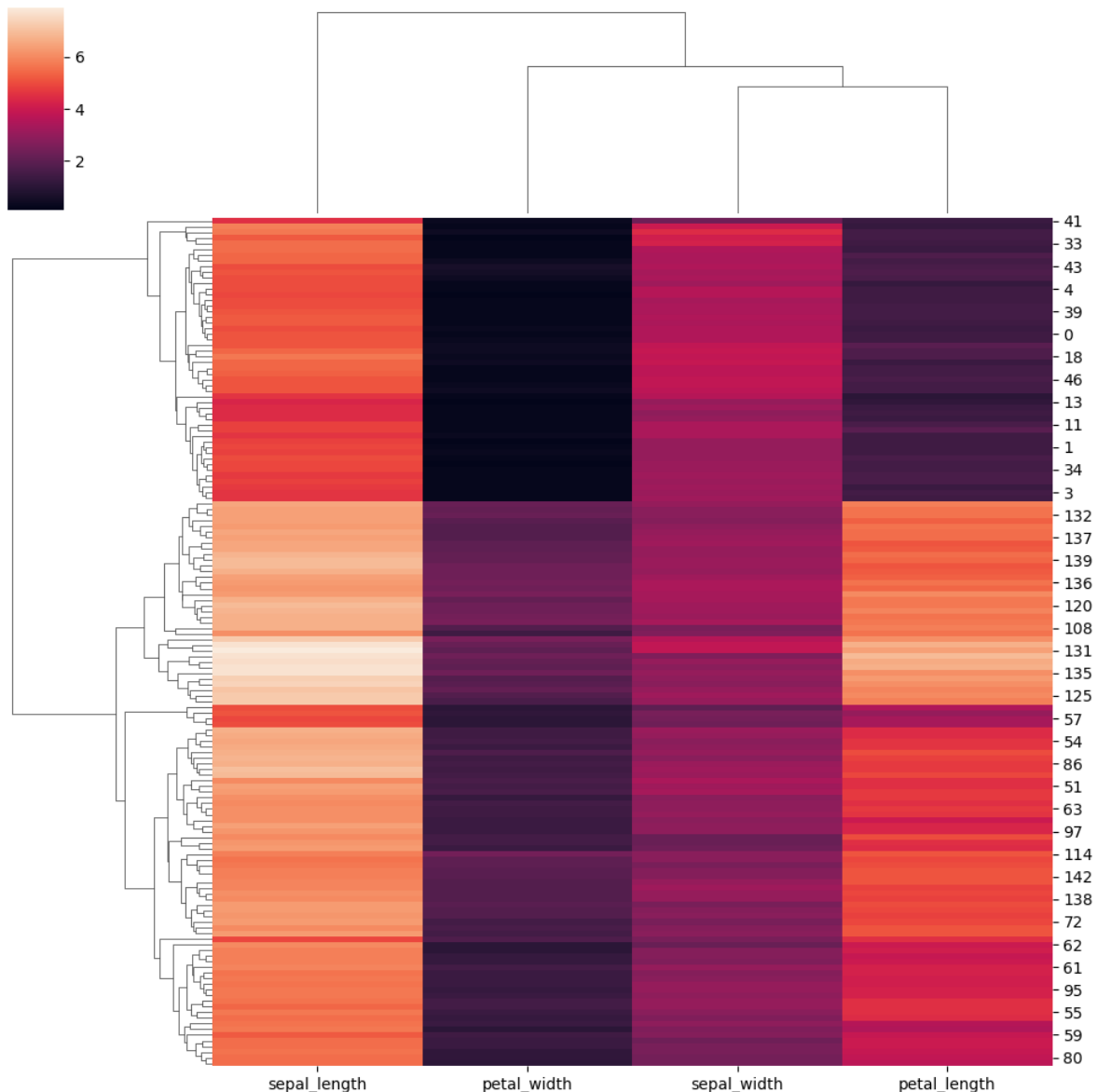
`annot=True` ⇒ to display the values of the cells in the heatmap.



Cluster Map

Cluster Map ⇒ combines **heatmaps** with **hierarchical clustering**. It displays data in a matrix format where both rows and columns are clustered to show similarities. The **dendrograms** (tree-like diagrams) on the sides of the **heatmap** indicate how data points are grouped together based on similarity ⇒ clustering the individual instances.

```
sns.clustermap(iris.drop("species", axis=1))
```

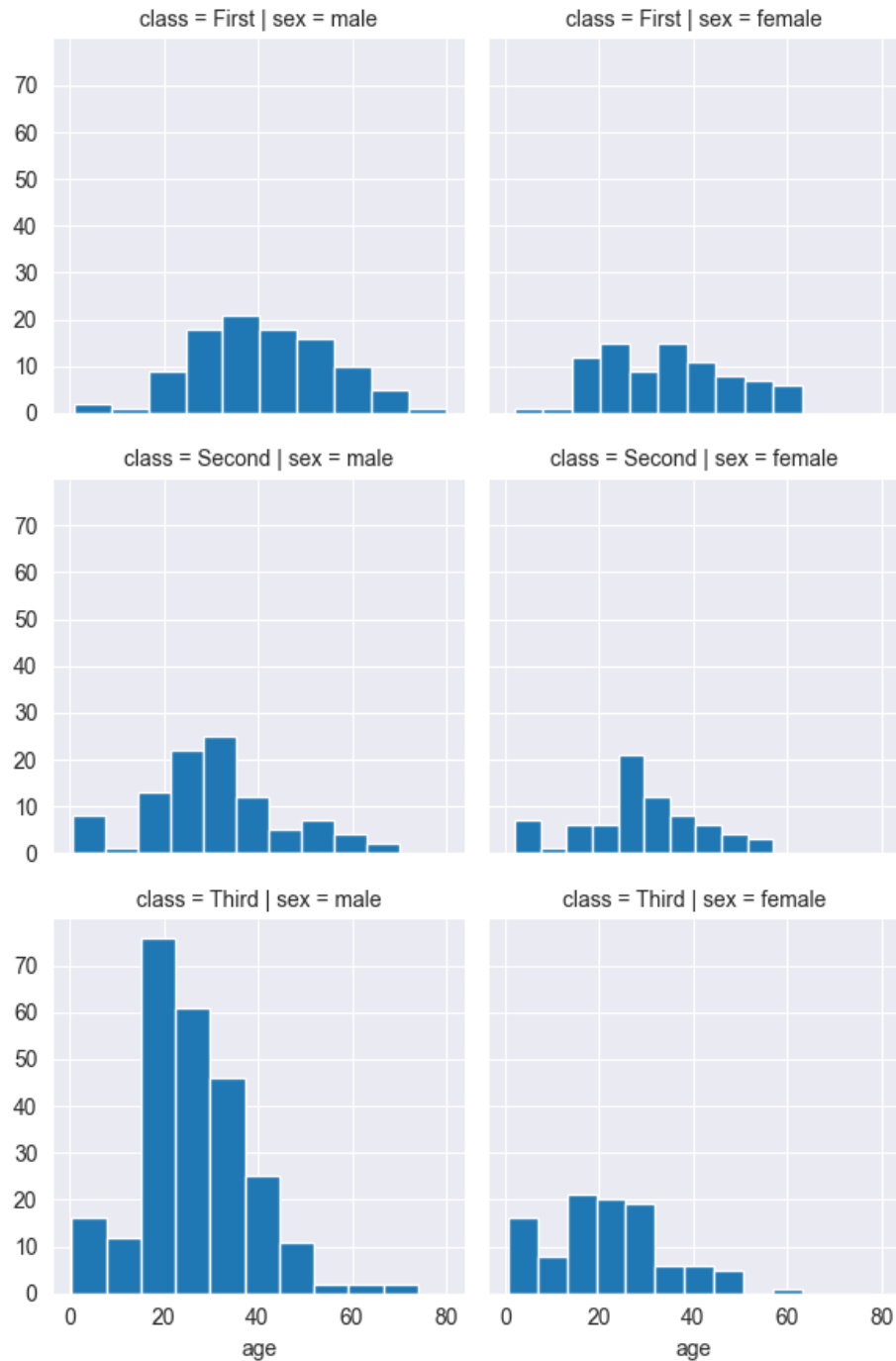


FacetGrid

FacetGrid ⇒ Creates a grid of plots based on the values of two categorical variables

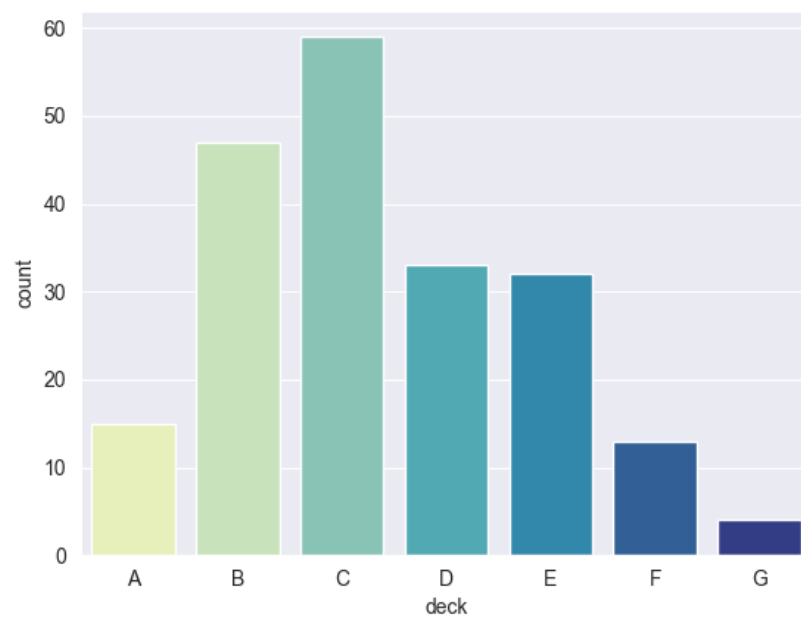
map : Maps a plotting function (here, `plt.hist`) to each subplot in the grid
 Displaying the distribution of `age` for each combination of `sex` and `class`.

```
g = sns.FacetGrid(titanic, col="sex", row="class")
g = g.map(plt.hist, "age")
```



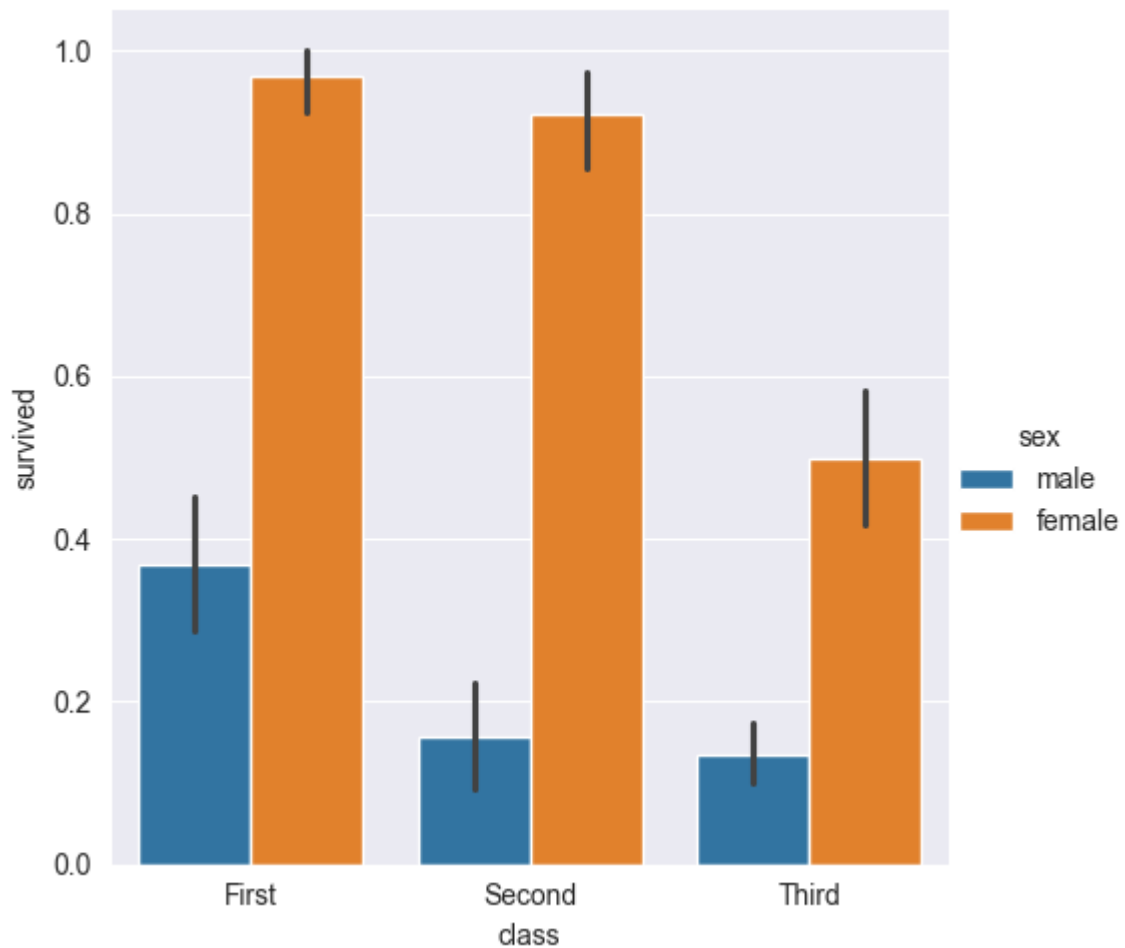
Count Plot

```
sns.countplot(x="deck", data=titanic, palette="YlGnBu")
```



Factor Plot / Cat Plot

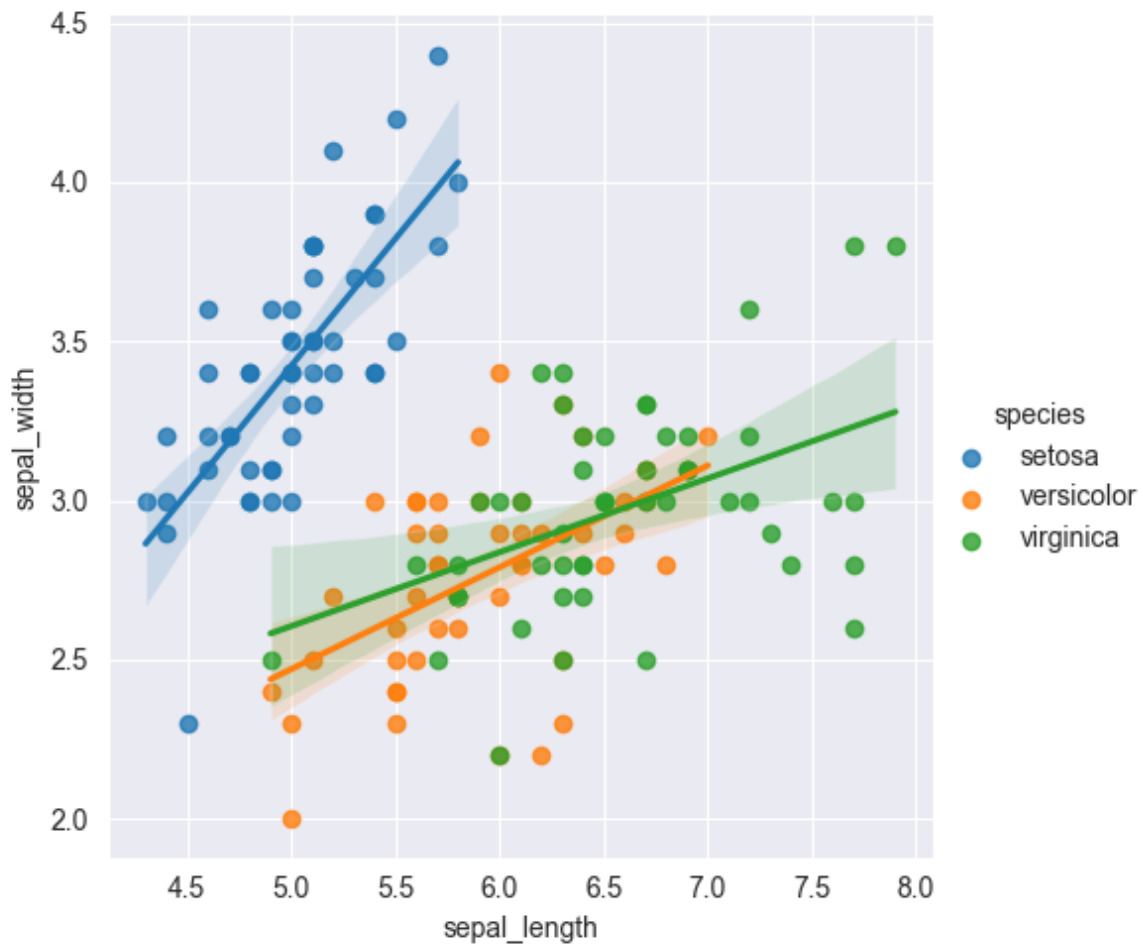
```
sns.catplot(x="class", y="survived", hue="sex", data=titanic,
```



Implot (Linear Model Plot)

Implot ⇒ Plot data and regression model fits
across a FacetGrid

```
sns.lmplot(x="sepal_length", y="sepal_width", hue="species",
```



Customizations

```
sns.set() #(Re)set the seaborn default
sns.set_style("whitegrid")
sns.set_style("ticks", {"xtick.major.size":8,"ytick.major.size":8})
sns.axes_style("whitegrid") #Return a dict of params or use w
```

The first

"whitegrid" style is set globally, affecting all subsequent plots. The second "whitegrid" style affects only the plot inside the context.

```
-> plt.despine(left=True) # Remove left spine
```

Save Plot

```
>>> plt.savefig("Plot.png") #Save the plot as a figure
>>> plt.savefig("Plot.png", transparent=True) #Save transparen
```

Close & Clear

```
>>> plt.cla() #Clear an axis
>>> plt.clf() #Clear an entire figure
>>> plt.close() #Close a window
```