## 3. Plotting With Seaborn

### Axis Grids
```python
>>> g = sns.FacetGrid(titanic,
...                    col="species",
...                    hue="sex",
...                    data=titanic)
```
Subplot grid for plotting conditional relationships

### Categorical Plots

#### Scatterplot
```python
>>> sns.scatterplot(x="sepal_length",
...                  y="sepal_width",
...                  hue="species",
...                  data=titanic)
```
Scatterplot with one categorical variable

#### Categorical scatterplot with non-overlapping points
```python
>>> sns.catplot(kind="box",
...                data=titanic)
```
Show point estimates and confidence intervals with scatterplot glyphs

#### Count Plot
```python
>>> sns.countplot(x="species",
...                data=titanic)
```
Show count of observations

#### Boxplot
```python
>>> sns.boxplot(x="species",
...              y="sepal_length",
...              hue="sex",
...              data=titanic)
```
Boxplot

#### Boxplot with wide-form data
```python
>>> sns.boxplot(data=titanic)
```
Boxplot

#### Violinplot
```python
>>> sns.violinplot(x="species",
...                 y="sepal_length",
...                 hue="sex",
...                 data=titanic)
```
Violin plot

### Scatterplot with Facetgrid
```python
>>> sns.jointplot(x="sepal_length",
...                y="sepal_width",
...                kind="hex")
```
Facetgrid

### Regression Plots

#### Regression plots
```python
>>> sns.regplot(x="sepal_length",
...              y="sepal_width",
...              data=titanic)
```
Plot data and a linear regression model fit

#### Scatterplot with residual plots
```python
>>> sns.residplot(x="sepal_length",
...                y="sepal_width",
...                data=titanic)
```
Plot data and residual plots

### Distribution Plots

#### Distribution plots with kernel density estimation
```python
>>> sns.distplot(titanic.y)
```
Distribution plots

#### Heatmap
```python
>>> sns.heatmap(uniform_data)
```
Heatmap

### Matrix Plots

#### Correlation matrix
```python
>>> sns.heatmap(titanic.corr())
```
Correlation matrix

### Further Customizations

#### Clean a figure
```python
>>> plt.cla()
```
Clean a figure

#### Show or Save Plot

#### Show the plot
```python
>>> plt.show()
```
Show the plot

#### Save the plot
```python
>>> plt.savefig("foo.png")
```
Save the plot

### Show and Save Plot

#### Show plot
```python
>>> sns.set_theme(style="ticks")
```
Show plot

### Close & Clear

#### Close a window
```python
>>> plt.close()
```
Close a window

### Data Visualization

The Python visualization library Seaborn is based on Matplotlib and provides a high-level interface for drawing attractive statistical graphics.

Make use of the following aliases to import the libraries:

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

The basic steps to creating plots with Seaborn are:

1. Prepare some data
2. Control figure aesthetics
3. Plot with Seaborn
4. Further customize your plot

### Data

```python
>>> import pandas as pd
>>> import numpy as np
>>> data = pd.DataFrame({
...     'x': np.random.normal(0,4,100),
...     'y': np.arange(1,101),
...     'z': np.random.rand(10, 12)
... })
```

### Figure Aesthetics

```python
>>> f, ax = plt.subplots(figsize=(5,6))
```
Create a figure and one subplot

### Seaborn styles

```python
>>> sns.set_theme(style="ticks")
```
Set the matplotlib parameters

### Context Functions

```python
>>> sns.set_theme(style="ticks")
```
Set the context and style

### Color Palette

```python
>>> sns.set_palette("husl", 3)
```
Define the color palette

### Show & Save Plot

#### Show the plot
```python
>>> plt.show()
```
Show the plot

#### Save the plot
```python
>>> plt.savefig("foo.png")
```
Save the plot