Visualization Tools

In this lesson, some visualization tools are explored.

We'll cover the following

This lesson reviews some of the tools and libraries that are used in this chapter to visualize various forms of data.

Matplotlib

Matplotlib is an interactive python graph-plotting library that helps us visualize data in various 2D plots. The following is an example of how `matplotlib` can be used to create a visualization of the *Sin* wave.

```
import numpy as np
import matplotlib.pyplot as plt

# Get x values of the sine wave
points = np.arange(0, 10, 0.1);

# Plot a sine wave
plt.plot(points, np.sin(points))

# Give a title for the sine wave plot
plt.title('Sine wave')

# Give x axis label for the sine wave plot
plt.xlabel('Time')

# Give y axis label for the sine wave plot
plt.ylabel('Amplitude')

plt.show()
```

Seaborn

Seaborn is an extension of the `matplotlib` package and builds on top of the already provided `matplotlib` functions. It adds more interactivity in the already present plots providing more concise information. The following code is inspired by an example from `seaborn` documentation that plots a joint plot. It can also be found here.

```
import numpy as np
import pandas as pd
import seaborn as sns

# Generate a random correlated bivariate dataset
rs = np.random.RandomState(5)
mean = [0, 0]
cov = [(.5, 1), (1, .5)]
var1, var2 = rs.multivariate_normal(mean, cov, 500).T
var1 = pd.Series(var1, name="X-axis")
var2 = pd.Series(var2, name="Y-axis")

# Show the joint distribution using kernel density estimation
sns1 = sns.jointplot(var1, var2, kind="scatter")
```

Types of plots

- Histogram
- Box
- Regression
- Heatmaps
- Scatter
- KDE

More information on plots can be found on Matplotlib and Seaborn.

We start to explain and use Histogram plots in the next lesson.