

There are many options and properties for working with **matplotlib**, **seaborn**, and **pandas.plotting** functions for data visualization, but as is the theme of this material, the goal is to keep it simple and give the reader just enough to be dangerous. Deep competency comes with experience and continuous usage. These cannot really be taught.

To begin, we will load Matplotlib by importing the **pyplot** module from the **matplotlib** package and the **seaborn** package.

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

We'll also import the **numpy** and **pandas** packages to create our datasets.

```
import pandas as pd
import numpy as np
```

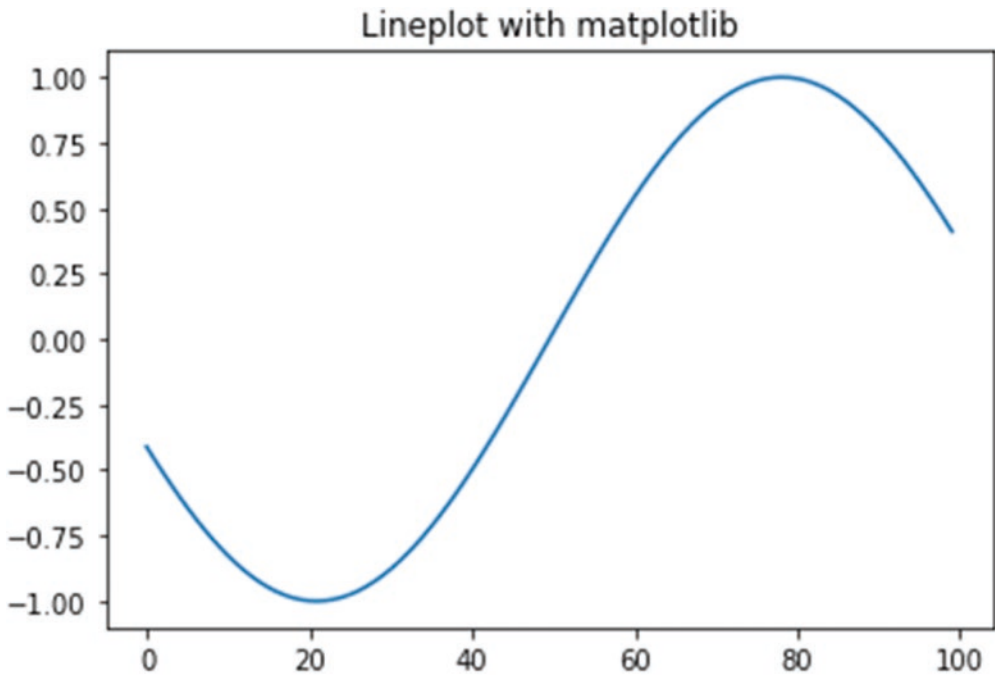
## Univariate Plots

Some common and essential univariate plots are line plots, bar plots, histograms and density plots, and the box and whisker plot, to mention just a few.

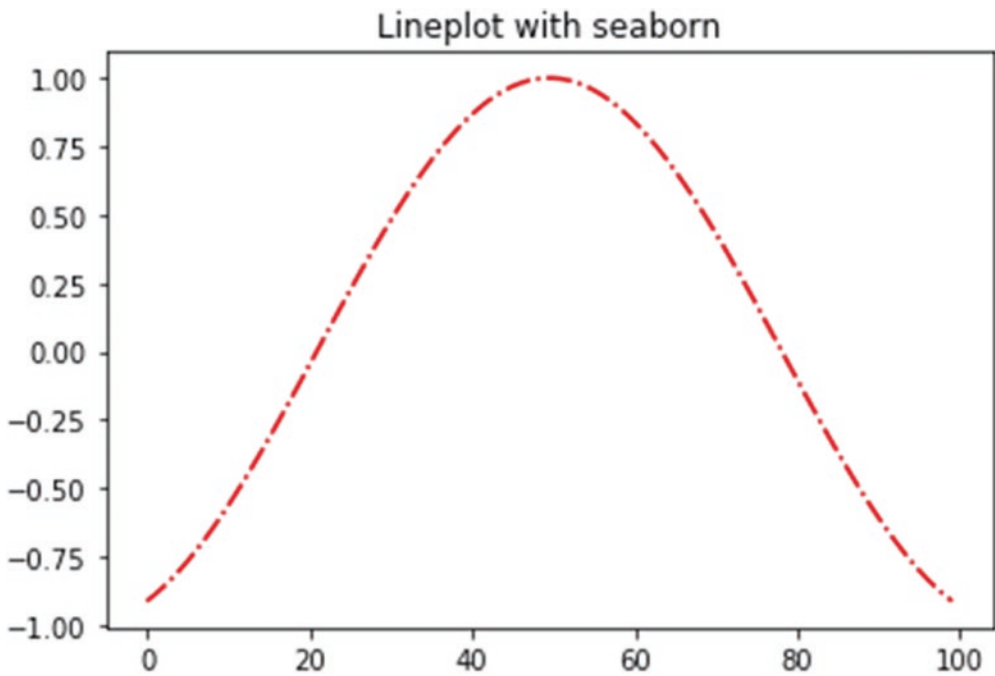
### Line Plot

Let's plot a sine graph of 100 points from the negative to positive **exponential** range. The **plot** method allows us to plot lines or markers to the figure. The outputs of the sine and cosine line plot are shown in Figure 12-1 and Figure 12-2, respectively.

```
data = np.linspace(-np.e, np.e, 100, endpoint=True)
# plot a line plot of the sine wave
plt.plot(np.sin(data))
plt.show()
# plot a red cosine wave with dash and dot markers
plt.plot(np.cos(data), 'r-.')
plt.show()
```



**Figure 12-1.** Lineplot with Matplotlib

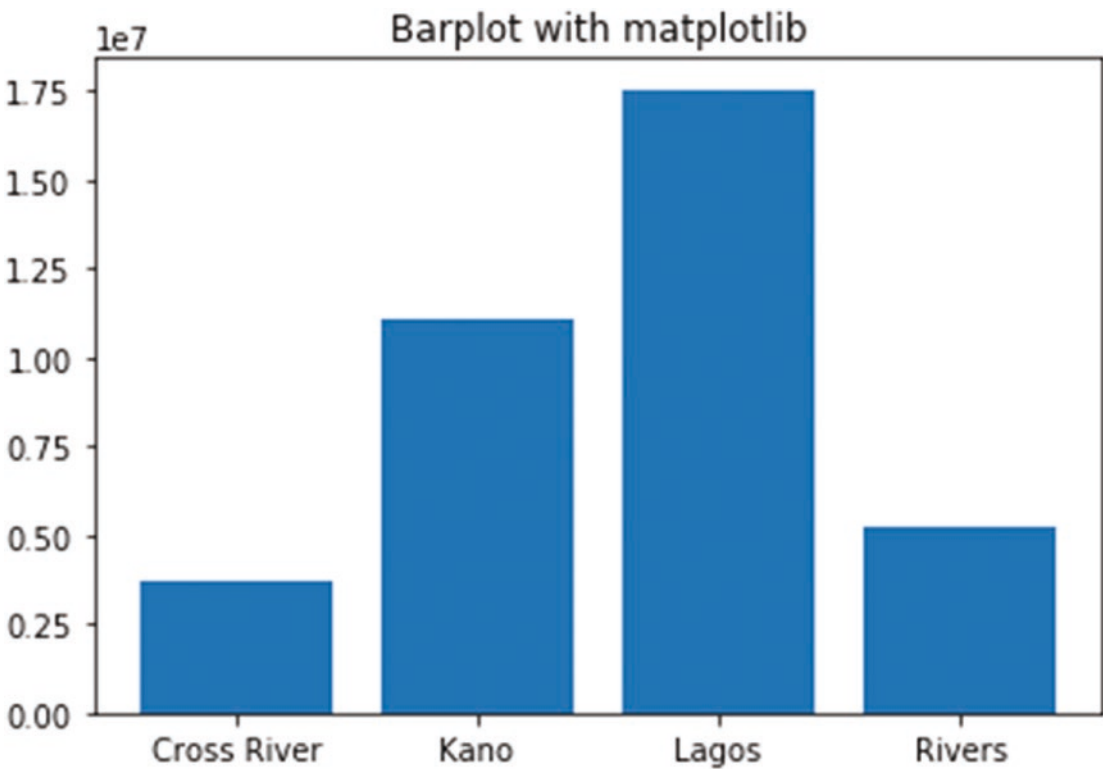


**Figure 12-2.** Lineplot with seaborn

## Bar Plot

Let's create a simple bar plot using the `bar` method. The output with `matplotlib` is shown in Figure 12-3, and the output with `seaborn` is shown in Figure 12-4.

```
states = ["Cross River", "Lagos", "Rivers", "Kano"]
population = [3737517, 17552940, 5198716, 11058300]
# create barplot using matplotlib
plt.bar(states, population)
plt.show()
# create barplot using seaborn
sns.barplot(x=states, y=population)
plt.show()
```



**Figure 12-3.** Barplot with Matplotlib