

Library	Advantages	Disadvantages
<b>Matplot</b>	<p>Relatively easy to understand for beginners.</p> <p>People who have used Matlab or other graph plotting packages before will find it a lot easier to use. Because it is built similar to MATLAB, toggling between the two is simple.</p> <p>It offers high-quality photos and plots in multiple formats, including png, pdf, etc.</p> <p>This library controls numerous aspects of an image, including image color, image size, etc.</p>	<p>Matplotlib is great for creating graphs and charts. However, it might not be ideal for time series data because it requires importing all helper classes for the year, month, week, and day formatters. It's also inconvenient when dealing with several datasets, but converting a dataset into a long format and plotting it is simple. Another significant downside is that the library is low-level and requires extra code to generate the visualization.</p>
<b>Seaborn</b>	<p>Seaborn allows a straightforward representation of your data on plots. You can use Seaborn to visualize data without worrying about the internal details.</p> <p>It allows you to simply provide our data set or data into the relplot() function, and it will compute and place the value appropriately.</p> <p>The 'kind' property inside this library enables you to switch to any other data representation format.</p> <p>It generates a dynamic and informative plot to represent your data, making it simple for the user to comprehend and view the information on the app.</p> <p>The seaborn library employs static aggregation for plot generation for <b>data visualization with Python</b>.</p>	<p>It relies significantly on other Python libraries like NumPy. Since it is not included with Python, you need to install the Seaborn library by executing several scripts before using it. In Seaborn, the customization options are limited. Interactive visualizations are rare in this library. Sometimes, users will need to use matplotlib simultaneously along with seaborn.</p>

## Box plot

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

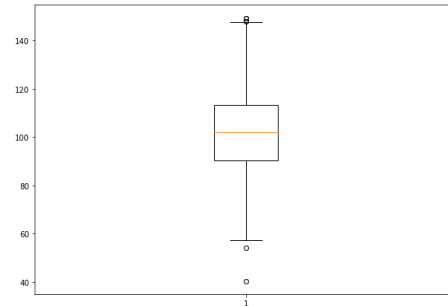
# Creating dataset
np.random.seed(10)
data = np.random.normal(100, 20, 200)

fig = plt.figure(figsize =(10, 7))

# Creating plot
plt.boxplot(data)

# show plot
plt.show()
```

### Output



## Scatter Plot

```
import matplotlib.pyplot as plt

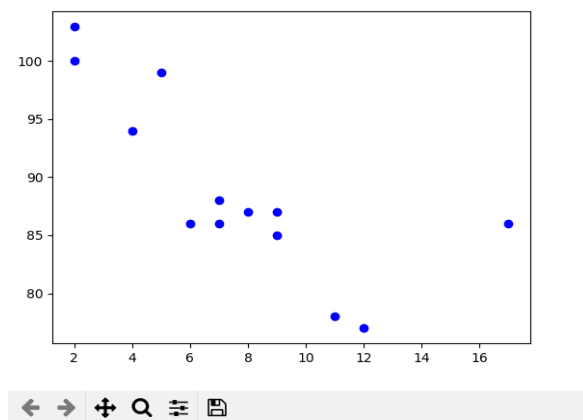
x=[5, 7, 8, 7, 2, 17, 2, 9,
  4, 11, 12, 9, 6]

y=[99, 86, 87, 88, 100, 86,
  103, 87, 94, 78, 77, 85, 86]

plt.scatter(x, y, c="blue")

# To show the plot
plt.show()
```

### Output

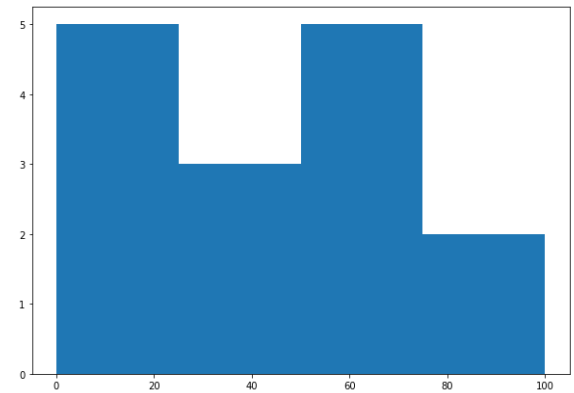


## Histogram

```
from matplotlib import pyplot as plt
import numpy as np
# Creating dataset
a = np.array([22, 87, 5, 43, 56,
              73, 55, 54, 11,
              20, 51, 5, 79, 31,
              27])

# Creating histogram
fig, ax = plt.subplots(figsize=(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])
# Show plot
plt.show()
```

### Output



## Pie chart

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

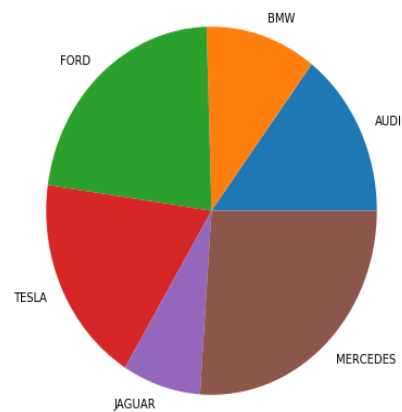
# Creating dataset
cars = ['AUDI', 'BMW', 'FORD',
        'TESLA', 'JAGUAR', 'MERCEDES']

data = [23, 17, 35, 29, 12, 41]

# Creating plot
fig = plt.figure(figsize=(10, 7))
plt.pie(data, labels = cars)

# show plot
plt.show()
```

### Output

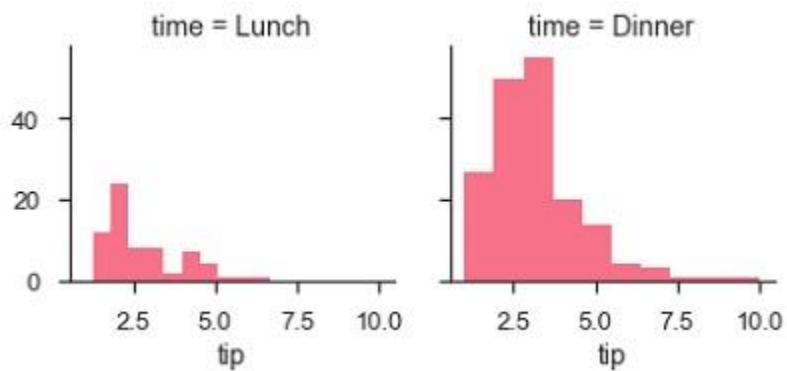


## Facet Plot

### Example

```
import pandas as pd
import seaborn as sb
from matplotlib import pyplot as plt
df = sb.load_dataset('tips')
g = sb.FacetGrid(df, col = "time")
g.map(plt.hist, "tip")
plt.show()
```

### Output



## Pair Plot

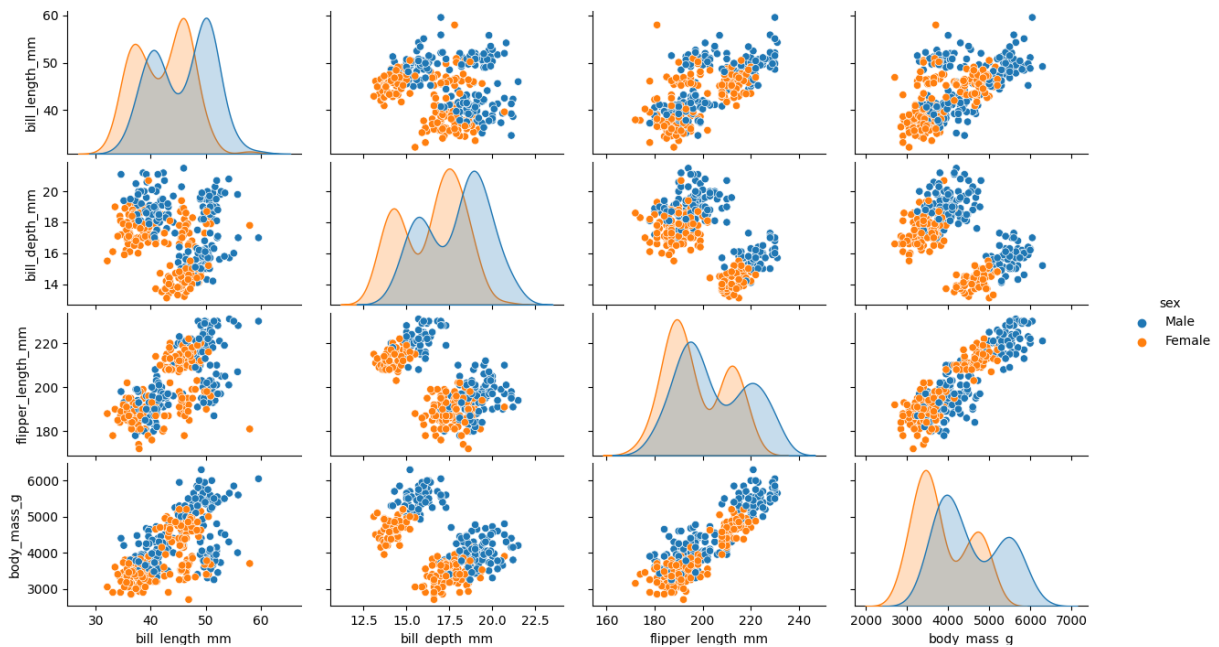
```
# importing the required libraries
import seaborn as sbn
import matplotlib.pyplot as plt

# loading the dataset using the seaborn library
mydata = sbn.load_dataset('penguins')

# pairplot with the hue = gender parameter
sbn.pairplot(mydata, hue = 'gender')

# displaying the plot
plt.show()
```

## Output



## Area Chart

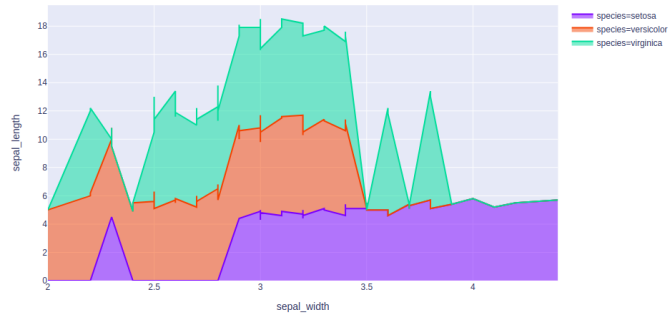
```
import plotly.express as px

df = px.data.iris()

fig = px.area(df, x="sepal_width",
              y="sepal_length",
              color="species",
              hover_data=['petal_width'],)

fig.show()
```

### Output



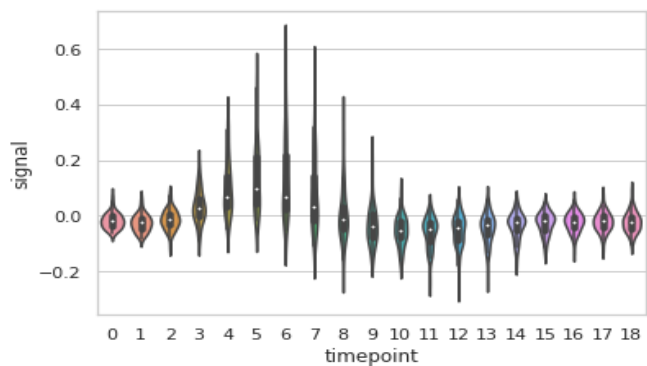
## Violin Plot

```
import seaborn

seaborn.set(style =
'whitegrid')
fmri =
seaborn.load_dataset("fmri")

seaborn.violinplot(x
="timepoint",
                    y ="signal",
                    data = fmri)
```

### Output



## Bar Chart

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

# creating the dataset
data = {'C':20, 'C++':15, 'Java':30,
        'Python':35}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(courses, values, color
       ='maroon',
       width = 0.4)

plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different
courses")
plt.show()
```

### Output

