



DATA VISUALIZATION USING MATPLOTLIB

[Leave a Comment](#) / [Neque](#) / [By shivam bhardwaj](#)

Data Visualization is an integral part of any data science project. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. For this purpose, Matplotlib provides a much easier way to visualize data using python. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

In this article some plots are discussed with codes snippet in python. For this purpose, famous Iris dataset was used. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.

Importing Libraries

Importing Libraries

```
In [1]: import numpy as np
import pandas as pd
import random
import numpy as np
import matplotlib.pyplot as plt
```

Some essential libraries which are used are imported like: Pandas, Numpy and Matplotlib.

Dataset

df

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows x 6 columns

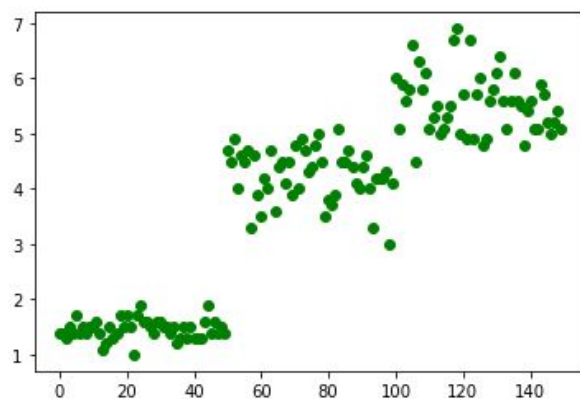
As we can see, this dataset contains 150 rows and 6 columns. There are 3 different species in the target column and four different input features.

Scatterplot

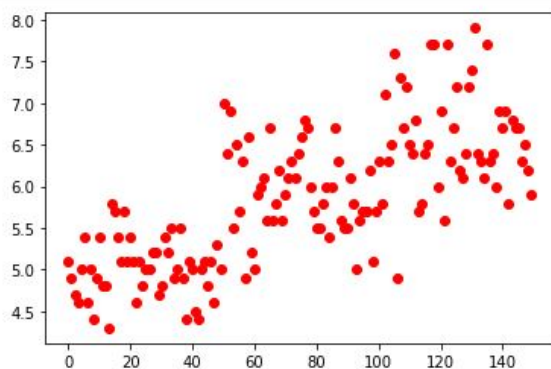
It is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data.

Scatter Plot ¶

```
In [18]: plt.plot(df['PetalLengthCm'], linestyle='none', marker='o', color='g')
plt.show()
```



```
In [7]: plt.plot(df['SepalLengthCm'], linestyle='none', marker='o', color='r')  
plt.show()
```



In these plots petal length and sepal length are displayed. We can clearly see the scatter plot of “petal length” can be grouped into three clusters.

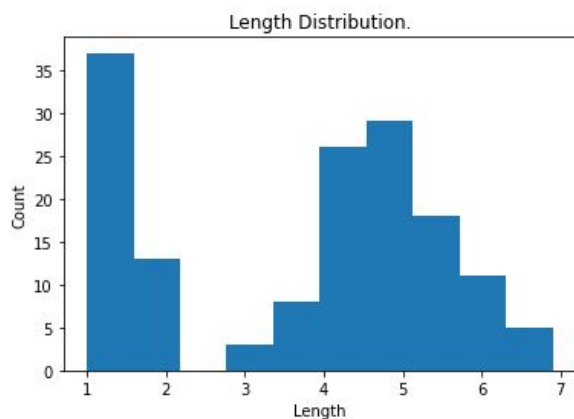
Histograms

A histogram is a graphical display of data using bars of different heights. In a histogram, each bar groups numbers into ranges.

Histogram

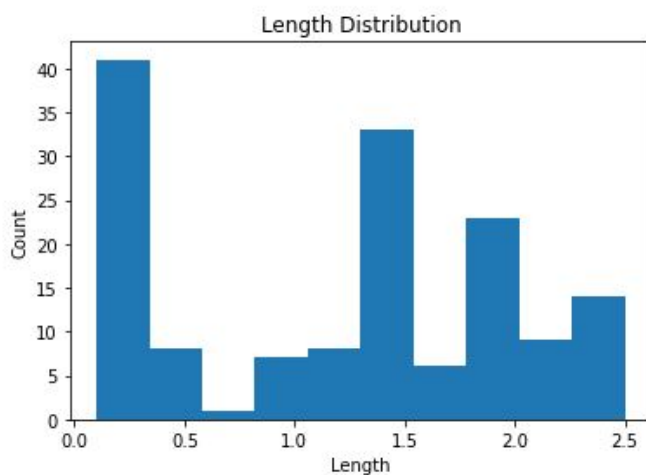
```
In [8]: plt.hist(df['PetalLengthCm'])  
plt.xlabel('Length')  
plt.ylabel('Count')  
plt.title('Length Distribution')
```

Out[8]: Text(0.5, 1.0, 'Length Distribution.')



```
In [21]: plt.hist(df['PetalWidthCm'])
plt.xlabel('Length')
plt.ylabel('Count')
plt.title('Length Distribution')
```

```
Out[21]: Text(0.5, 1.0, 'Length Distribution')
```



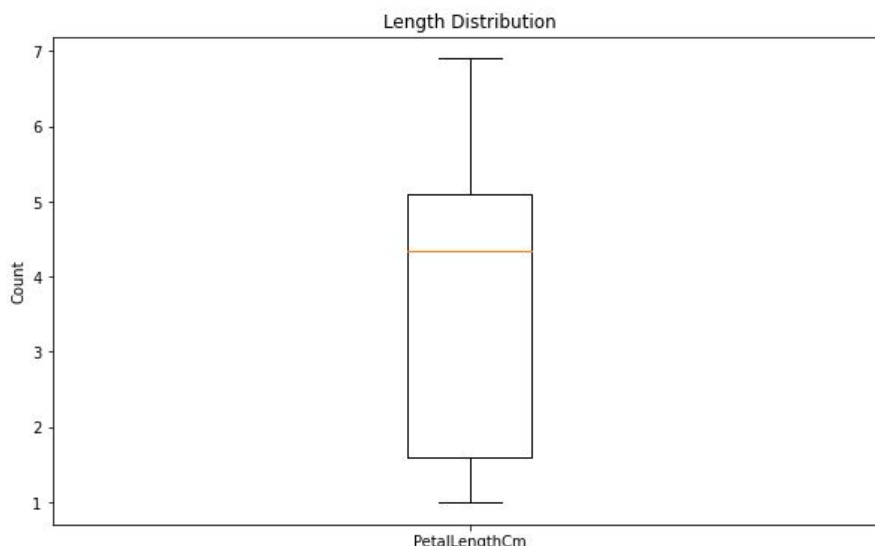
Boxplots

A boxplot is a standardized way of displaying the distribution of data based on a five number summary (minimum, first quartile (Q1), median, third quartile (Q3), and maximum).

Box Plots

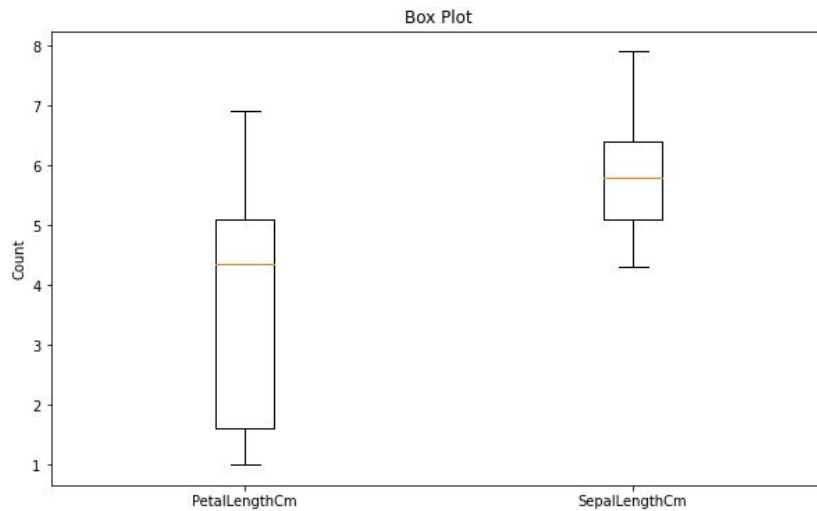
```
In [24]: plt.figure(figsize=(10,6))
plt.boxplot(df['PetalLengthCm'])
plt.ylabel('Count')
plt.title("Length Distribution")
plt.xticks([1], ['PetalLengthCm'])
```

```
Out[24]: ([<matplotlib.axis.XTick at 0x1ece0dfb640>], [Text(1, 0, 'PetalLengthCm')])
```



```
In [12]: plt.figure(figsize=(10,6))
plt.boxplot([df['PetalLengthCm'], df['SepalLengthCm']])
plt.ylabel('Count')
plt.xticks([1, 2], ['PetalLengthCm', 'SepalLengthCm'])
plt.title("Box Plot")
```

```
Out[12]: Text(0.5, 1.0, 'Box Plot')
```



So, Data Visualization helps decision makers understand how the data could be interpreted and hidden insights can be pulled out using different graphs and plots.

Thanks!!

[← Previous Post](#)

Leave a Comment

Logged in as shivam bhardwaj. [Log out?](#)

Type here..

POST COMMENT »