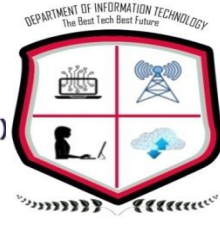




**WAVOO WAJEEHA WOMEN'S COLLEGE
OF ARTS & SCIENCE - KAYALPATNAM**
(Affiliated to Manonmanium Sundaranar University, Tirunelveli)
Run by : Wavoo SAR Educational Trust
(minority institution)



Department of Information Technology **Statistics with Python**

Module 5: Data Visualization- Box plots, Pie Charts, Histograms, Bar Charts

Visualizing Data

In addition to calculating the numerical quantities like mean, median, or variance, you can use visual methods to present, describe, and summarize data. In this section, we will learn how to present your data visually using the following graphs:

-  Box plots
-  Histograms
-  Pie charts
-  Bar charts
-  X-Y plots
-  Heatmaps

matplotlib.pyplot is a very convenient and widely-used library, though it's not the only Python library available for this purpose. You can import it like this:

```
>>> import matplotlib.pyplot as plt
>>> plt.style.use('ggplot')
```

Now, you have matplotlib.pyplot imported and ready for use. The second statement sets the style for your plots by choosing colors, line widths, and other stylistic elements.

Box Plot

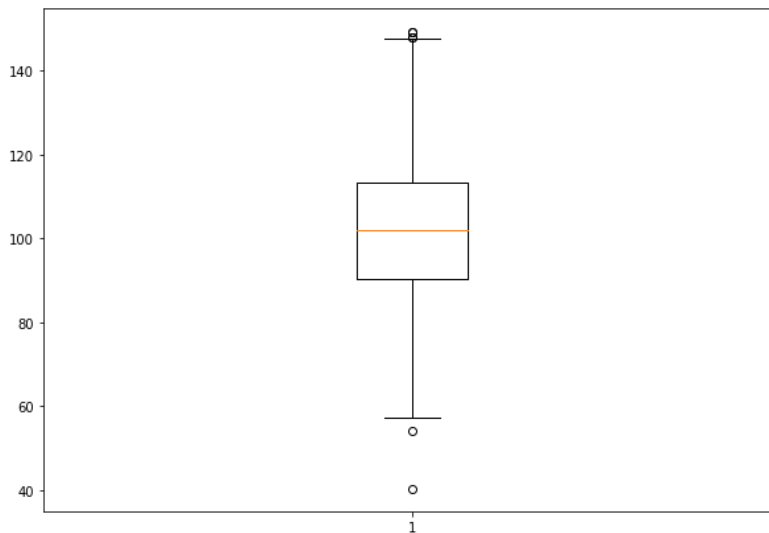
A **Box Plot** is also known as **Whisker plot** is created to display the summary of the set of data values having properties like minimum, first quartile, median, third quartile and maximum. In the box plot, a box is created from the first quartile to the third quartile, a vertical line is also there which goes through the box at the median. Here x-axis denotes the data to be plotted while the y-axis shows the frequency distribution.

Creating Box Plot-The [matplotlib.pyplot](#) module of matplotlib library provides boxplot() function with the help of which we can create box plots.

```
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:



Customizing Box Plot

The `matplotlib.pyplot.boxplot()` provides endless customization possibilities to the box plot. The `notch = True` attribute creates the notch format to the box plot, `patch_artist = True` fills the boxplot with colors, we can set different colors to different boxes. The `vert = 0` attribute creates horizontal box plot. `labels` takes same dimensions as the number data sets.

```
# Import libraries
import matplotlib.pyplot as plt
import numpy as np

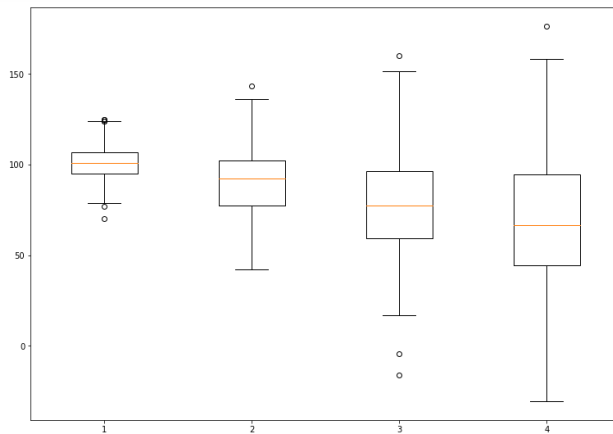
# Creating dataset
np.random.seed(10)

data_1 = np.random.normal(100, 10, 200)
data_2 = np.random.normal(90, 20, 200)
data_3 = np.random.normal(80, 30, 200)
data_4 = np.random.normal(70, 40, 200)
data = [data_1, data_2, data_3, data_4]

fig = plt.figure(figsize =(10, 7))
# Creating axes instance
ax = fig.add_axes([0, 0, 1, 1])
```

```
# Creating plot
bp = ax.boxplot(data)
# show plot
plt.show()
```

Output:



Pie Charts

With Pyplot, you can use the `pie()` function to draw pie charts:

Example

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
plt.pie(y)
plt.show()
```

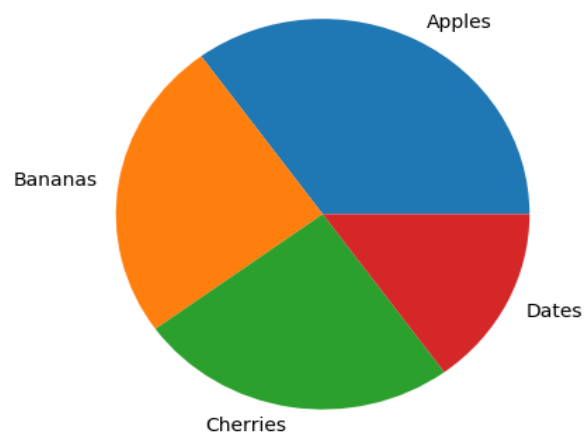


Labels to Pie chart

Add labels to the pie chart with the `label` parameter. The `label` parameter must be an array with one label for each wedge:

Example

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
plt.pie(y, labels = mylabels)
plt.show()
```

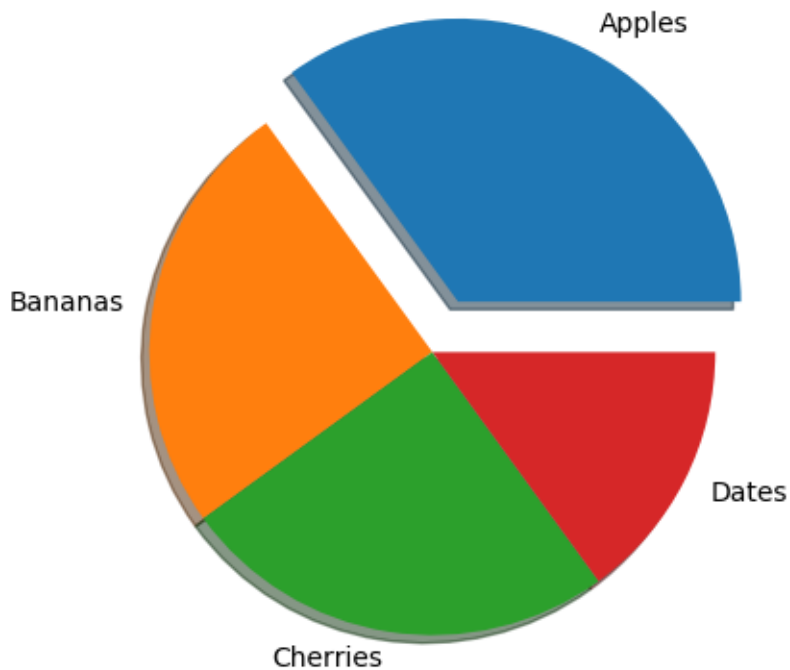


Shadow

Add a shadow to the pie chart by setting the `shadows` parameter to `True`:

Example:

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
myexplode = [0.2, 0, 0, 0]
plt.pie(y, labels = mylabels, explode = myexplode, shadow = True)
plt.show()
```



Histogram

In Matplotlib, we use the `hist()` function to create histograms.

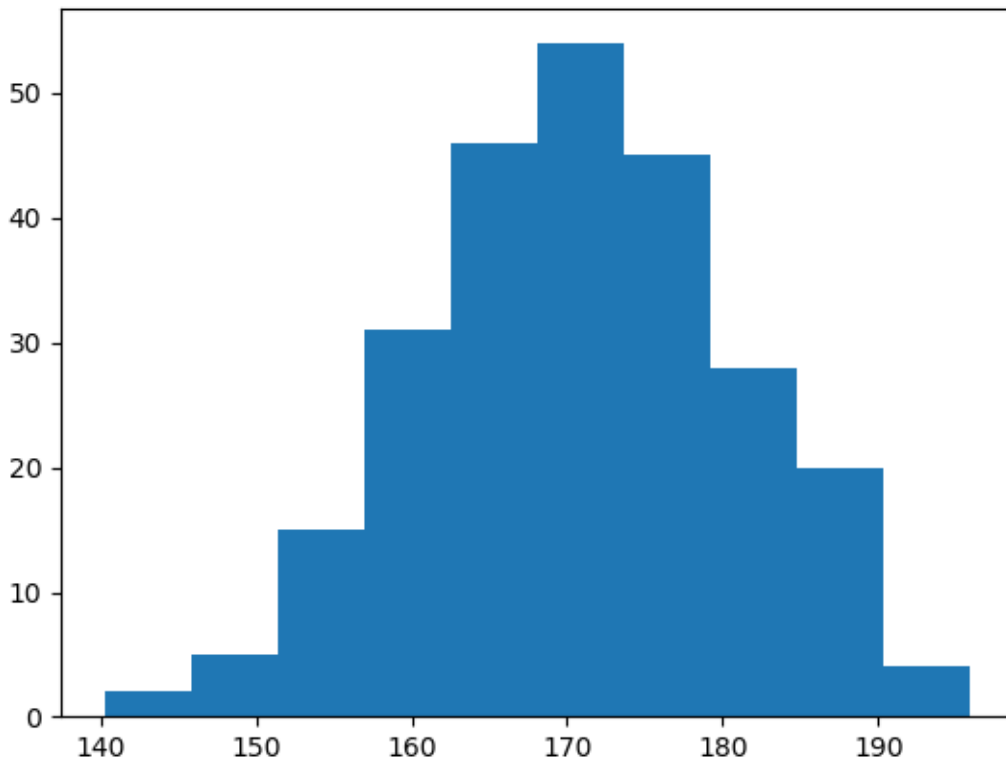
The `hist()` function will use an array of numbers to create a histogram, the array is sent into the function as an argument.

The `hist()` function will read the array and produce a histogram:

Example

A simple histogram:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.normal(170, 10, 250)
plt.hist(x)
plt.show()
```



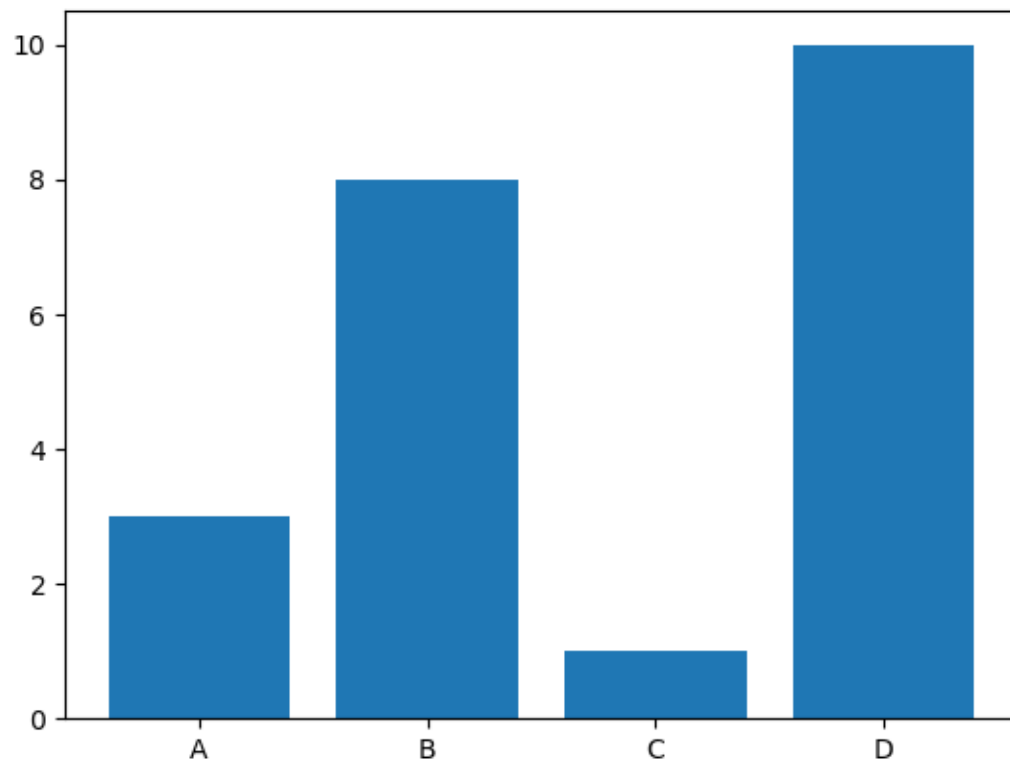
Bar Charts

With Pyplot, you can use the `bar()` function to draw bar graphs:

Example

Drawing 4 bars:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array(["A", "B", "C", "D"])
y = np.array([3, 8, 1, 10])
plt.bar(x,y)
plt.show()
```



Source : <https://realpython.com/python-statistics/#visualizing-data>