Python Programming

Data Visualization

Agenda:

- Data Visualization
- Matplotlib Installation
- Plotting
- Type of charts
- Plotting in Pandas
- Plotting in Seaborn

Data Visualization :

Data Visualization :

Data visualisation means graphical or pictorial representation of the data using graph, chart, etc.

The purpose of plotting data is to visualise variation or show relationships between variables.

Visualisation also helps to effectively communicate information to intended users.

Traffic symbols, ultrasound reports, Atlas book of maps, speedometer of a vehicle, tuners of instruments are few examples of visualisation that we come across in our daily lives.

Matplotlib library of Python used in plotting charts such as line, bar, scatter with respect to the various types of data.

Installation :

We can install matplotlib using the following pip command from the command prompt:

```
pip install matplotlib
```

For plotting using Matplotlib, we need to import its Pyplot module using the following command:

```
import matplotlib.pyplot as plt
```

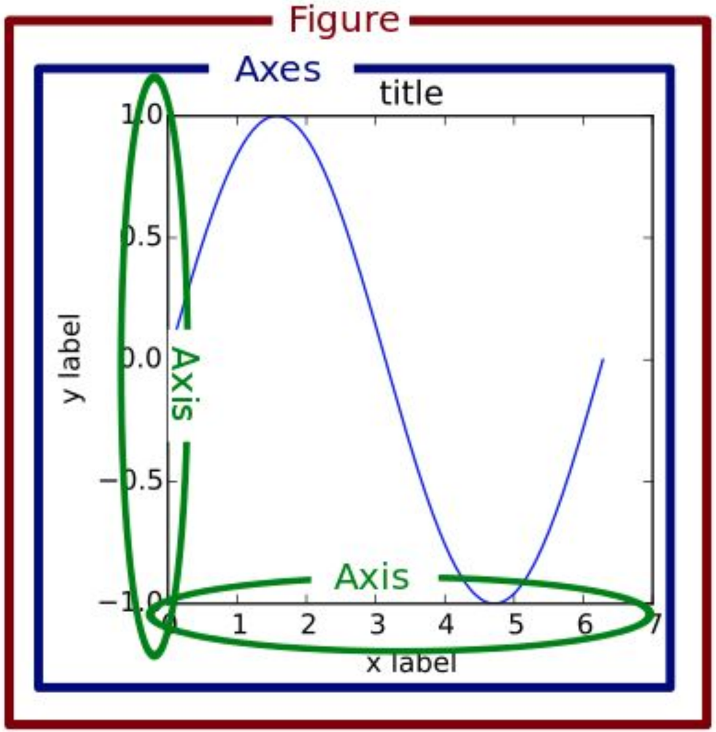
Here, plt is an alias or an alternative name for matplotlib.pyplot.

Matplotlib Objects:

Figure: Outermost container for a Matplotlib graphic. Can contain multiple Axes objects.

Axes: Actual plots. Contain smaller objects (tick marks, individual lines, etc.)

Artist: Everything that is seen on the figure is an artist.



Plotting :

The **pyplot** module of matplotlib contains a collection of functions that can be used to work on a plot.

The **plot()** function of the pyplot module is used to create a figure.

A figure is the overall window where the outputs of pyplot functions are plotted.

A figure contains a plotting area, legend, axis labels, ticks, title, etc.

Each function makes some change to a figure: example, creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

To plot x versus y, we can write **plt.plot(x,y)**.

The **show()** function is used to display the figure created using the plot() function.

Eg -

```
import matplotlib.pyplot as plt
date=["25/12","26/12","27/12"] #dates
temp=[8.5,10.5,6.8] #temp
plt.plot(date, temp) #create a figure plotting temp versus date
plt.show() #show the figure
```

Save a plot :

```
plt.savefig('x.png')
```

Pyplot functions to plot different charts :

plot() - Plot x versus y as lines and/or markers.

bar() - Make a bar plot.

boxplot() - Make a box and whisker plot.

hist() - Plot a histogram.

pie() - Plot a pie chart.

scatter() - A scatter plot of x versus y.

Bar plot :

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.

ax.bar(x, height, width, bottom, align)

X - sequence of scalars representing the x coordinates of the bars. align controls if x is the bar center (default) or left edge.

Height - scalar or sequence of scalars representing the height(s) of the bars.

Width - scalar or array-like, optional. the width(s) of the bars default 0.8

Bottom - scalar or array-like, optional. the y coordinate(s) of the bars default None.

Align - {'center', 'edge'}, optional, default 'center'

Box and Wisker plot :

- Matplotlib's boxplot mainly provides a graphical summary of a data set with features such as minimum, first quartile, median, third quartile, and maximum.
- The "Whisker Plot" is another name for the "Box Plot."
- In the box plot, the box is drawn from the first to third quartiles, with a vertical line running through it at the median.
- The x-axis of a box plot represents the data to be plotted, while the y-axis represents the frequency distribution.
- Box plots are useful for visualizing the distribution of numerical values in a field.

- **matplotlib.pyplot.boxplot(data, notch=None, vert=None, patch_artist=None, widths=None)**

- The data values given to the ax.boxplot() method can be a Numpy array or Python list or Tuple of arrays.
- 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.

Histogram :

- A histogram is basically used to represent data provided in a form of some groups.
- It is accurate method for the graphical representation of numerical data distribution.
- It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.
- To create a histogram the first step is to create bin of the ranges, then distribute the whole range of the values into a series of intervals, and count the values which fall into each of the intervals.
- Bins - Intervals
- Function - matplotlib.pyplot.hist()

Piechart :

- A Pie Chart is a circular statistical plot that can display only one series of data.
- The area of the chart is the total percentage of the given data.
- The slices of pie are called wedges
- The area of slices of the pie represents the percentage of the parts of the data.
- Pie charts are commonly used in business presentations like sales, operations, survey results, resources, etc as they provide a quick summary.

matplotlib.pyplot.pie(data, explode=None, labels=None, colors=None, autopct=None, shadow=False)

- **data** represents the array of data values
- **labels** is a list of sequence of strings which sets the label of each wedge.
- **color** attribute is used to provide color to the wedges.
- **autopct** is a string used to label the wedge with their numerical value.
- **shadow** is used to create shadow of wedge.

Scatter plot :

- Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them.
- The scatter() method in the matplotlib library is used to draw a scatter plot.
- Scatter plots are widely used to represent relation among variables and how change in one affects the other.

matplotlib.pyplot.scatter(x_axis_data, y_axis_data, s=None, c=None, marker=None, cmap=None, vmin=None, vmax=None, alpha=None, linewidths=None, edgecolors=None**)**

- **x_axis_data**- An array containing x-axis data
- **y_axis_data**- An array containing y-axis data
- **s**- marker size (can be scalar or array of size equal to size of x or y)
- **c**- color of sequence of colors for markers
- **marker**- marker style
- **cmap**- cmap name
- **linewidths**- width of marker border
- **edgecolor**- marker border color
- **alpha**- blending value, between 0 (transparent) and 1 (opaque)

Pyplot functions to customise plots

grid() - Configure the grid lines.

legend() - Place a legend on the axes.

savefig() - Save the current figure.

title() - Set a title for the axes.

xlabel() - Set the label for the x-axis.

xticks() - Get or set the current tick locations and labels of the x-axis.

ylabel() - Set the label for the y-axis.

yticks() - Get or set the current tick locations and labels of the y-axis.

The Pandas Plot function (Pandas Visualisation)

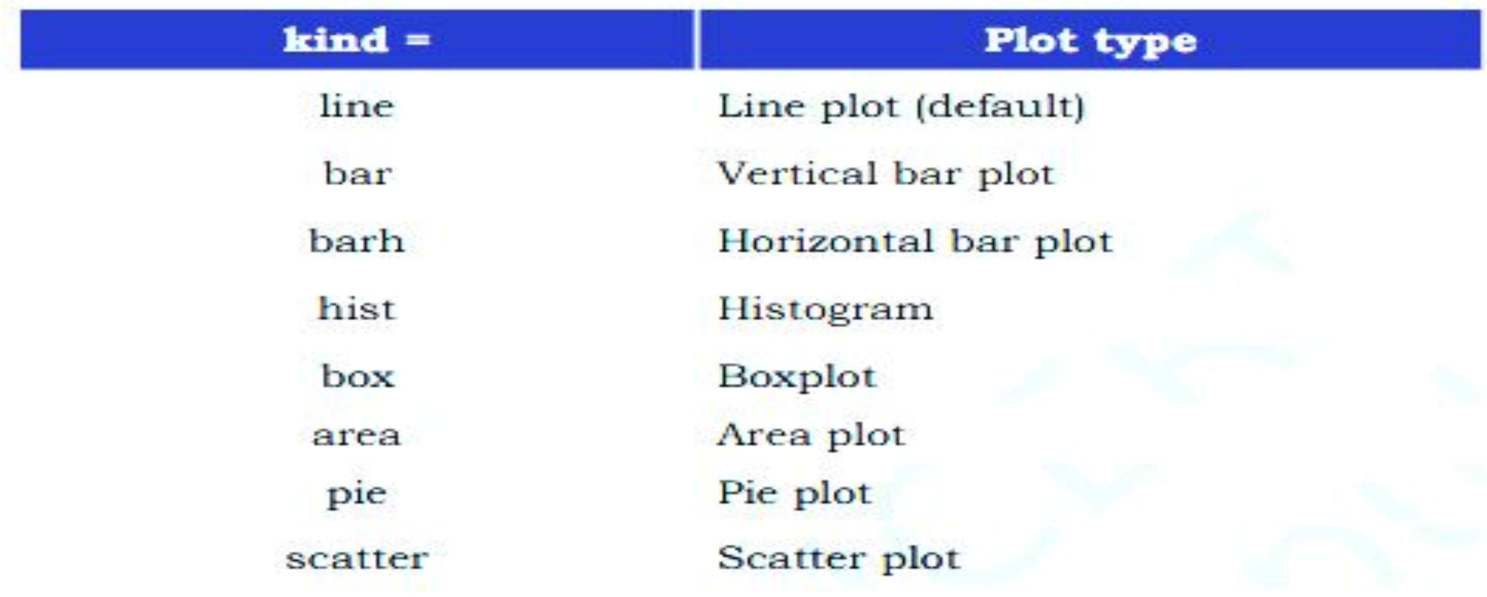
Pandas objects Series and DataFrame come equipped with their own .plot() methods.

This plot() method is just a simple wrapper around the plot() function of pyplot.

Pandas objects Series and DataFrame come equipped with their own .plot() methods.

s.plot() or df.plot()

Arguments accepted by kind for different plots :



kind =	Plot type
line	Line plot (default)
bar	Vertical bar plot
barh	Horizontal bar plot
hist	Histogram
box	Boxplot
area	Area plot
pie	Pie plot
scatter	Scatter plot

Plotting graph using Seaborn :

Seaborn is a Python data visualization library based on matplotlib.

It provides a high-level interface for drawing attractive and informative statistical graphics.

It provides beautiful default styles and color palettes to make statistical plots more attractive.

Installation - **pip install seaborn**

References and links :

Python Documentation : https://docs.python.org/3/

Matplotlib : https://matplotlib.org/stable/index.html

Pandas plot : https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html

Seaborn : https://seaborn.pydata.org/

GitHub link : https://github.com/lunatic-bot/PythonTraining

Thank You