# Plotting and Matplotlib



## What is Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Source : https://matplotlib.org/

## Pick the correct plot

- Since a graph is a visual way of representing data
- Picking correct plot is important to convey your thoughts
- Simplistic plots with no distracting elements



## Usage

- Importing

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

- Matplotlib is the base library for multiple other plotting libraries

# Simple line plot

- Use list/nparray/Series or any array like data

```
>>>plot (<xabel>, <ylabel>,  <color, symbol options>)
```

- Ex:

```
plot( [1,2,3], [1,4,9], 'ro')
```



# Controlling Multiple plots

- Multiple plot calls display on the same graph.

```
>>>plt.show()
```

- Multiple plots on same window

```
>>>plt.subplot(row, col, index)
```

- Create new figure windows using

```
>>>figure()
```

# Annotating your graphs

- Label on x,y axis

```
>>>xlabel()
>>>ylabel()
```

- Setting markers/ticks

```
>>>xticks()
>>>yticks()
```

specify markers on x and y axis, rotation etc.



- Display a legend. Works only if you have set labels for the plots

```
>>>legend()
```

- Set graph title/heading

```
>>>title()
```

- Add some text

```
>>>text( x, y, "text")
```

Inserts a single text element; requires loop otherwise

leangaurav.me@gmail.com

# Draw line and Saving

- Horizontal and vertical lines

```
>>> axvline( x )        # draw vertical line
>>> axhline( y )        # draw horizontal line
```

- Save a figure

```
>>> savefig(filename)
```



# Percentages / Comparison for categorical data

- Pie Charts

```
>>> pie(data):
shadow    :    Boolean
explode   :    [list of floats]
labels, labeldistance
```

- Horizontal or Vertical bar plots:

```
>>> bar(x, values):
label     :    Used by legend option
bottom    :    used to create stacked bar plot
```

# Histogram for distribution

- Method name

```
>>>hist()
bins      : integer
rwidth  : Width of bars; float [0 - 1.0]
```

- Frequency distribution of data grouped into ranges
- Bar like representation for non-categorical data



# Areas and stacked area

- Area Plots

```
>>>fill_between(x, y)
```

- Control alpha/transparency
- Fill between *x, y1, y2* to achieve stacked effect

# Boxplot for skewness and outliers

- Method

```
>>> boxplot(data):
data        :        can be array or a matrix for multiple plots
```



# Scatter for multiple attributes

- Multiple *y* vars for a common *x* var.

```
>>> plt.scatter()
color       :        string
s           :        integer size
alpha       :        float [0 – 1.0]
marker      :        o,_,^, $...$
```

# Image and twinx

- Display image/ plot heatmap like graph

```
>>>imshow()
```

- Different scales on same graph

```
>>>twinx()
```