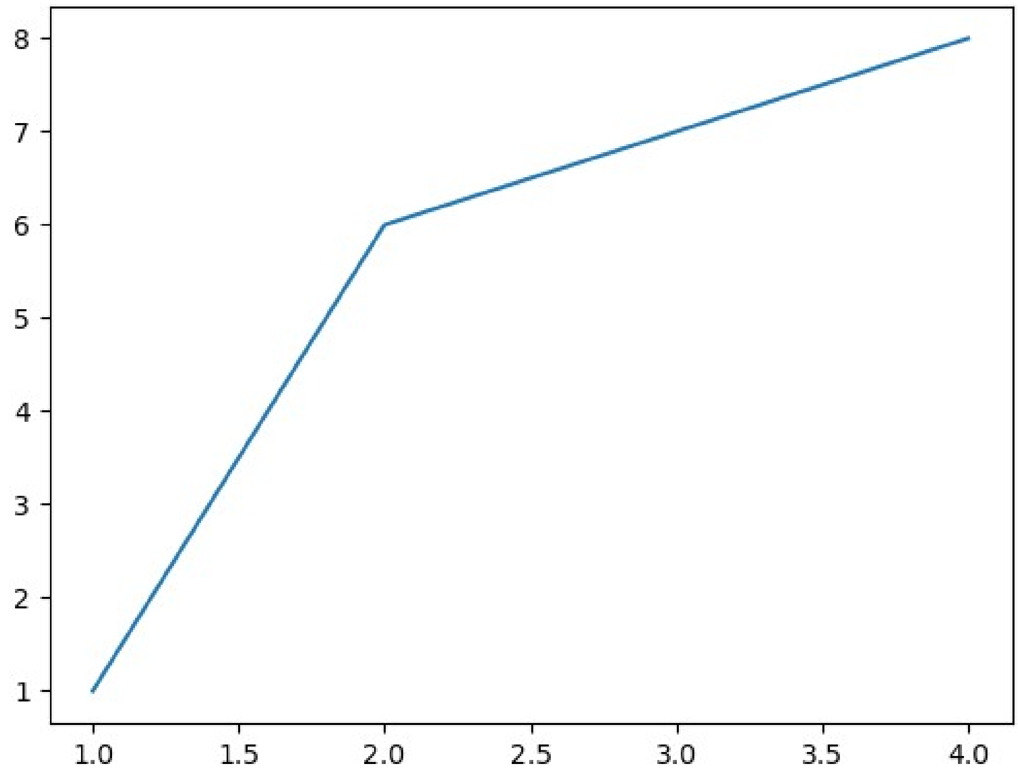
Matplotlibrary

- Matplotlib is a library used for creating visualization
- Matplotlib is a python 2D plotting
- It is a Data Science library
- Types of Plotting * Line * Bar Chart * Area plot * Scatter plot * Pie chart * Histogram

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

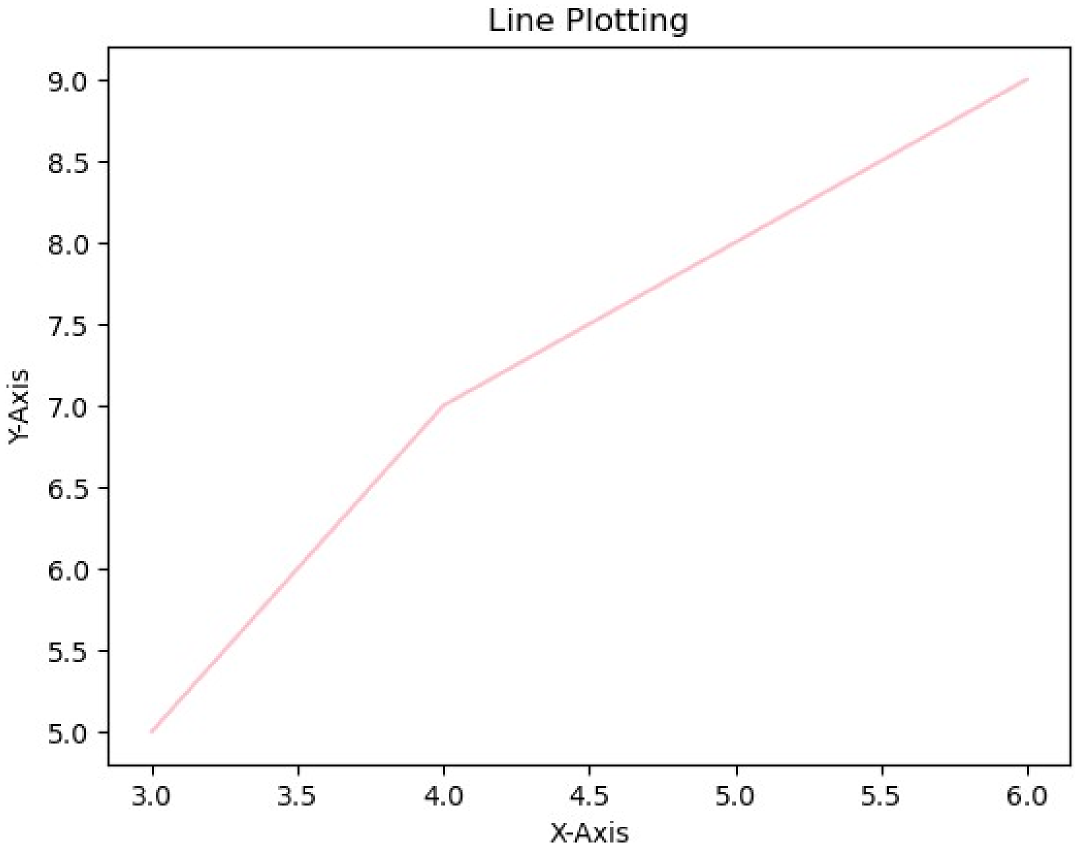
Line Plotting

```
* plt.plot()

plt.plot([1,2,3,4],[1,6,7,8])  # (1,1)(2,6)(3,7)(4,8)
plt.show()
```



```
plt.plot([3,4,5,6],[5,7,8,9],color='pink')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.title('Line Plotting')
```

```
Text(0.5, 1.0, 'Line Plotting')
```

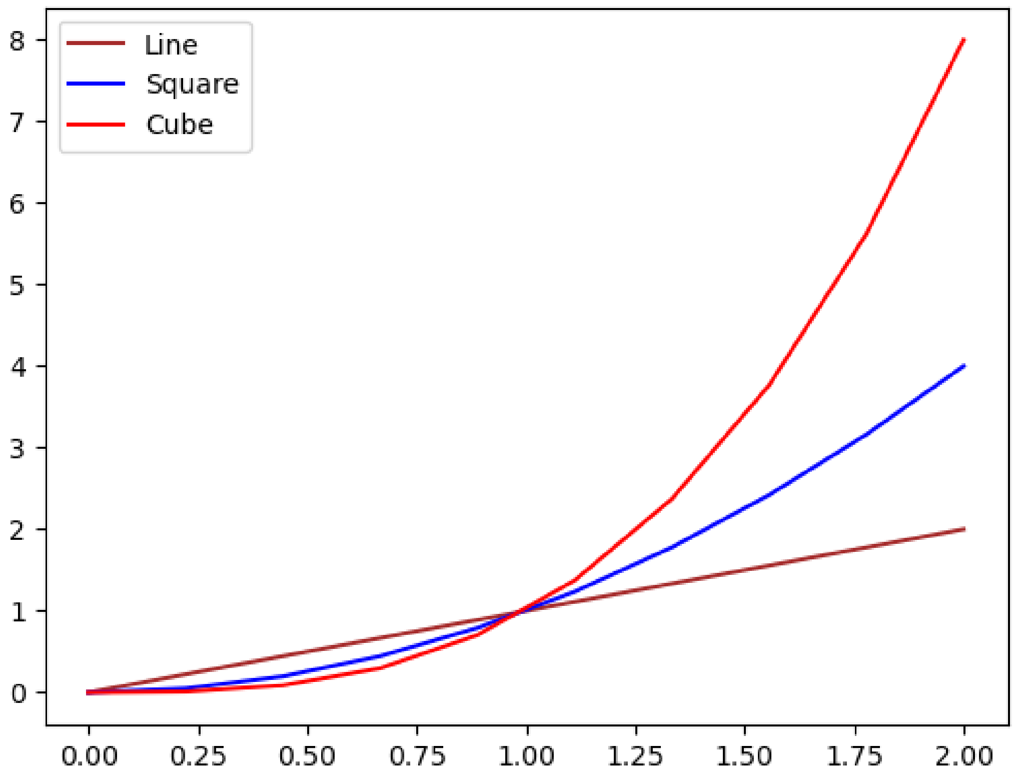


```
x = np.linspace(0,2,10)
x
```

```
array([0.        , 0.22222222, 0.44444444, 0.66666667, 0.88888889,
       1.11111111, 1.33333333, 1.55555556, 1.77777778, 2.        ])
```

```
plt.plot(x,x,label='Line',color='brown')
plt.plot(x,x**2,label='Square',color='blue')
plt.plot(x,x**3,label='Cube',color='red')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x27733f287a0>
```

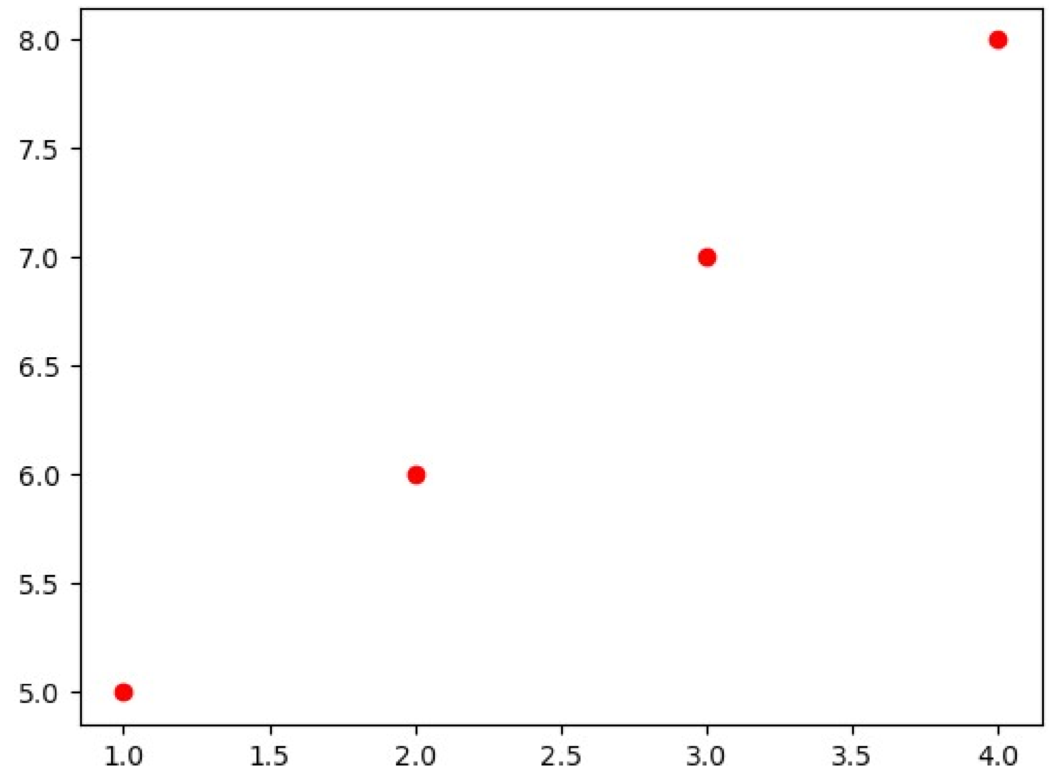


```
x1 = np.random.randint(0,10,5)
print(x1)
y1 = np.random.randint(10,50,5)
print(y1)

[3 6 6 4 6]
[24 41 45 36 11]
```

Scatter Plot

- plt.scatter()

```
plt.scatter([1,2,3,4],[5,6,7,8],color='r')

<matplotlib.collections.PathCollection at 0x2773549d550>
```

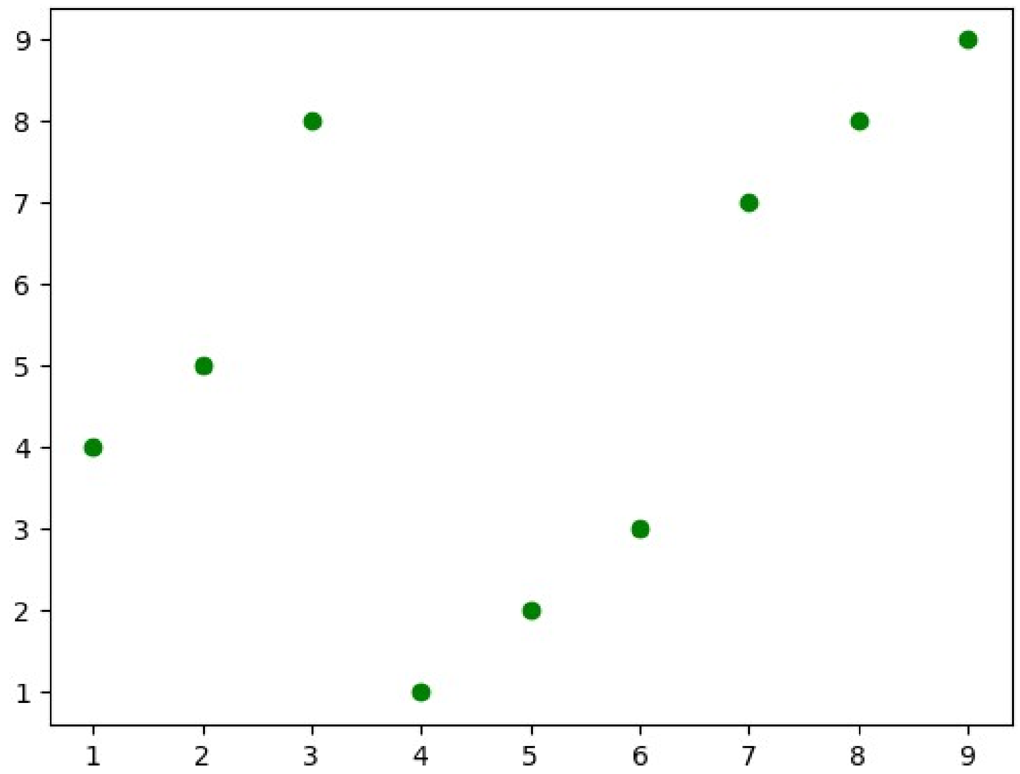


```
x1 = np.random.randint(0,10,5)
print(x1)
y1 = np.random.randint(10,50,5)
print(y1)
```

```
[7 0 5 0 1]
[39 38 36 38 26]
```

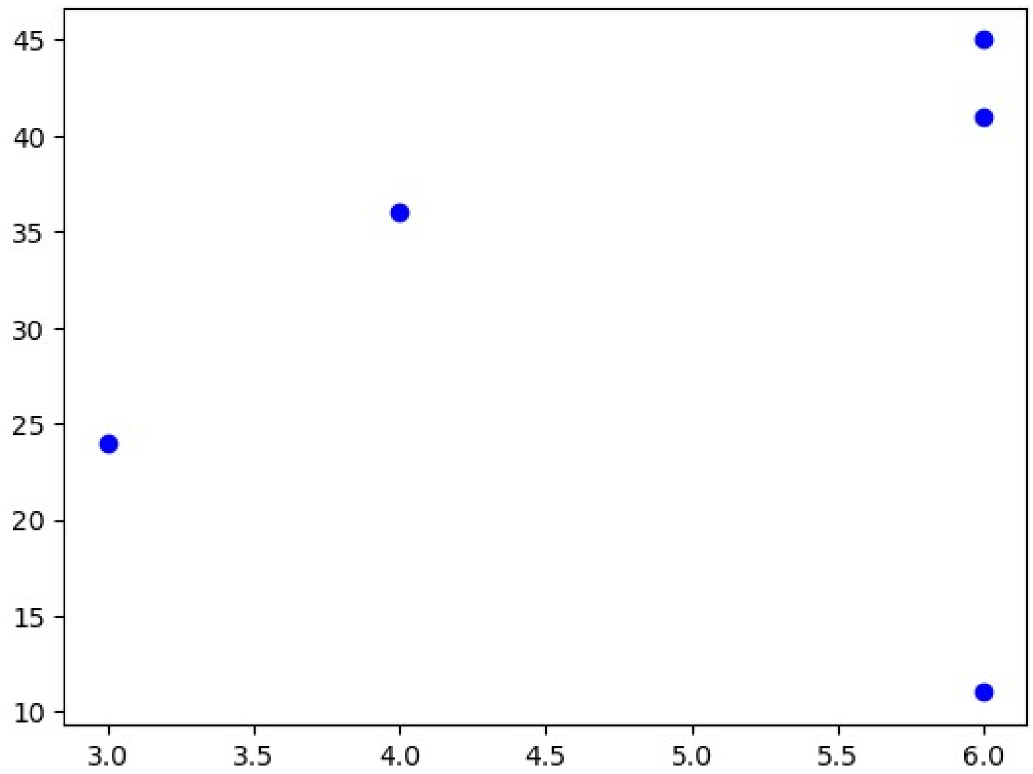
```
plt.scatter([1,2,3,4,5,6,7,8,9],[4,5,8,1,2,3,7,8,9],color='g')
```

```
<matplotlib.collections.PathCollection at 0x27733fb2cf0>
```



```
plt.scatter(x1,y1,color='blue')
```

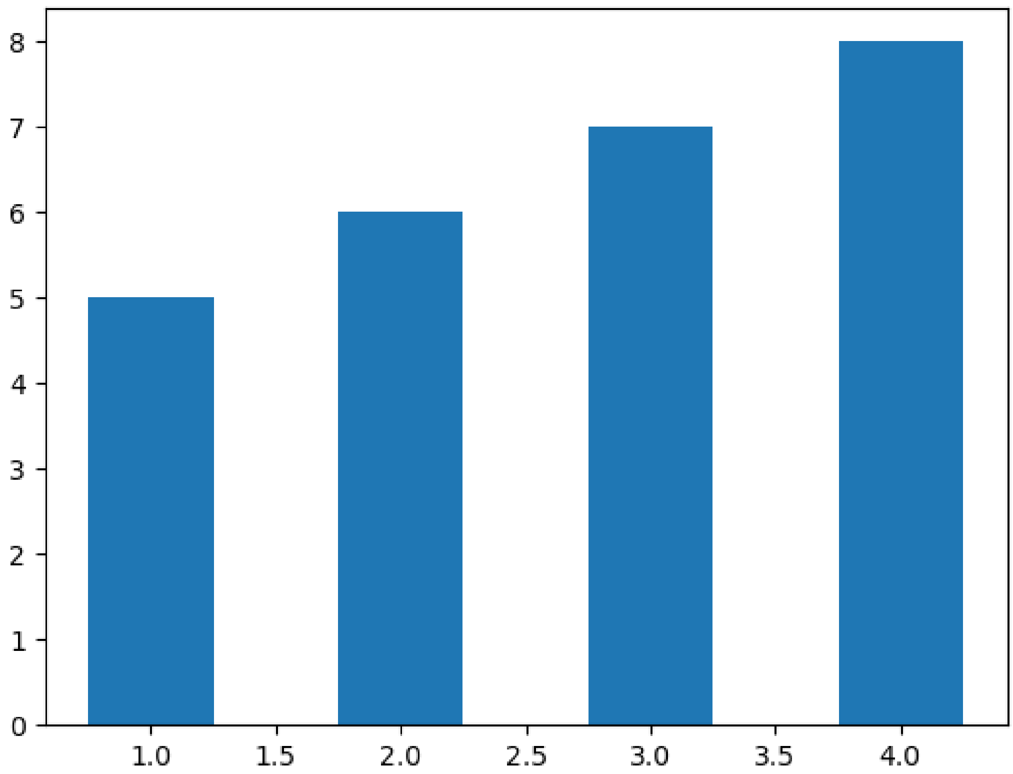
```
<matplotlib.collections.PathCollection at 0x277350285f0>
```



Bar Chart

- plt.bar()

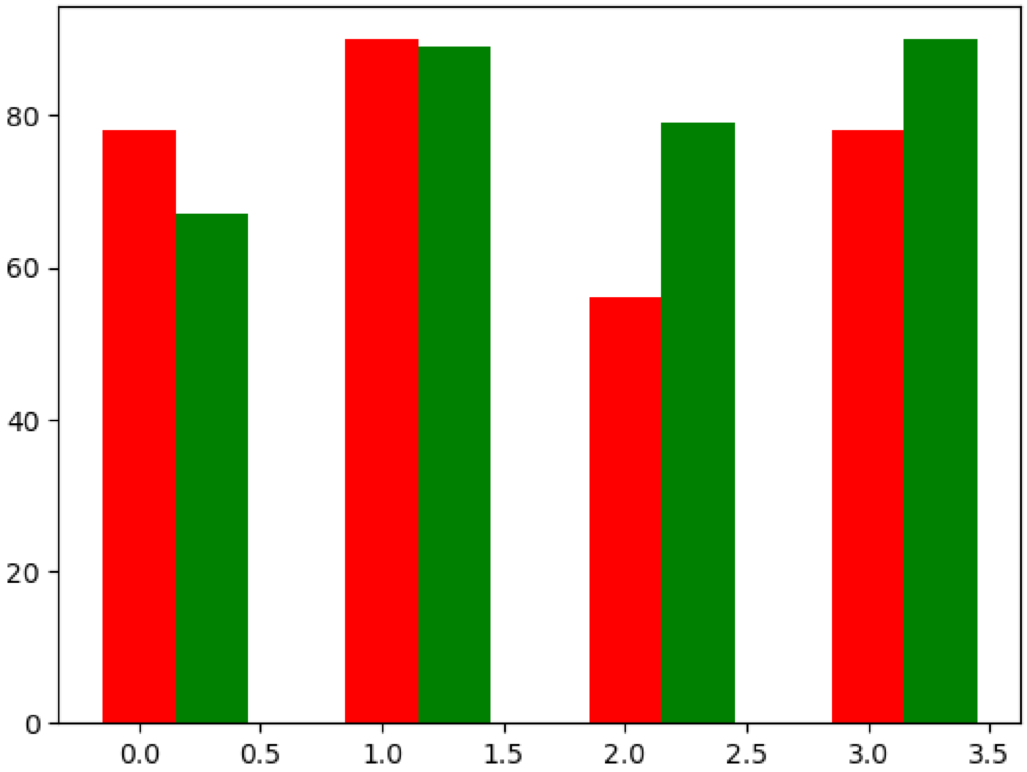
```
plt.bar([1,2,3,4],[5,6,7,8],width=0.5)

<BarContainer object of 4 artists>
```



```
x2 = np.arange(4)
print(x2)
plt.bar(x2,[78,90,56,78],width=0.3,color='red')
plt.bar(x2+0.3,[67,89,79,90],width=0.3,color='green')
```

```
[0 1 2 3]
```

```
<BarContainer object of 4 artists>
```



```
print(dir(plt))
```

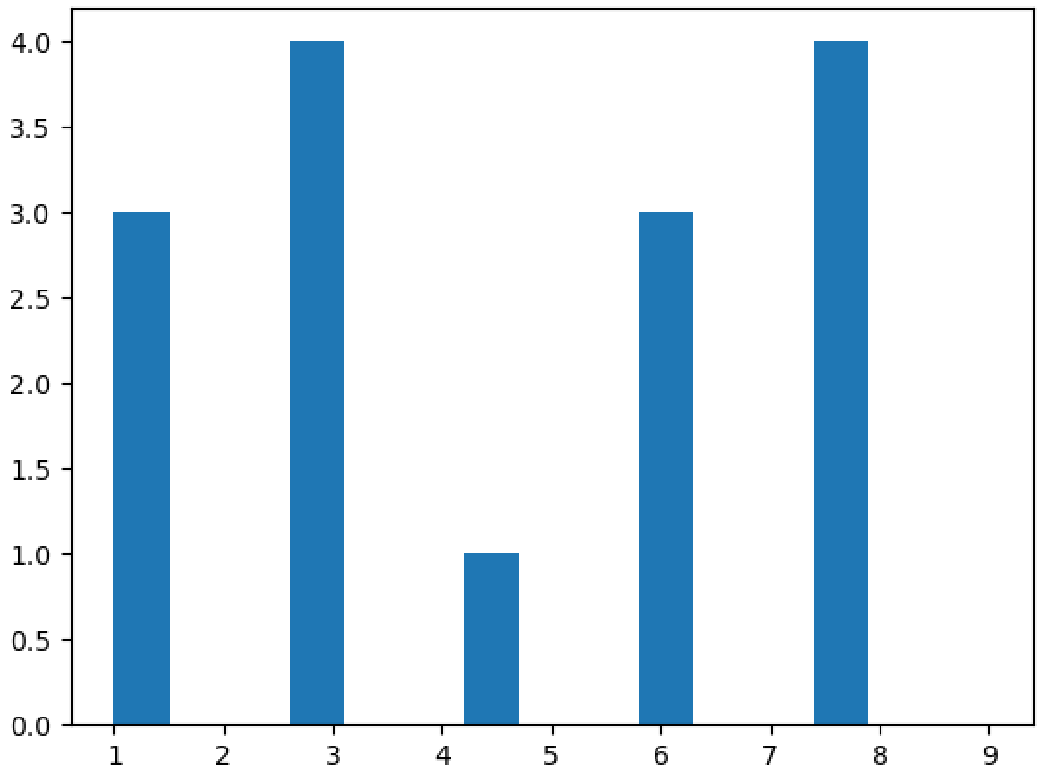
```
['AbstractContextManager', 'Annotation', 'Arrow', 'Artist',
'AutoLocator', 'AxLine', 'Axes', 'BackendFilter', 'Button', 'Circle',
'Colormap', 'Enum', 'ExitStack', 'Figure', 'FigureBase',
'FigureCanvasBase', 'FigureManagerBase', 'FixedFormatter',
'FixedLocator', 'FormatStrFormatter', 'Formatter', 'FuncFormatter',
'GridSpec', 'IndexLocator', 'Line2D', 'LinearLocator', 'Locator',
'LogFormatter', 'LogFormatterExponent', 'LogFormatterMathtext',
'LogLocator', 'MaxNLocator', 'MouseButton', 'MultipleLocator',
'Normalize', 'NullFormatter', 'NullLocator', 'PolarAxes', 'Polygon',
'Rectangle', 'ScalarFormatter', 'Slider', 'Subplot', 'SubplotSpec',
'TYPE_CHECKING', 'Text', 'TickHelper', 'Widget', '_NO_PYPLOT_NOTE',
'_REPL_DISPLAYHOOK', '_ReplDisplayHook', '__annotations__',
'__builtins__', '__cached__', '__doc__', '__file__', '__loader__',
'__name__', '__package__', '__spec__', '_add_pyplot_note', '_api',
'_auto_draw_if_interactive', '_backend_mod', '_color_sequences',
'_colormaps', '_copy_docstring_and_deprecators', '_docstring',
'_draw_all_if_interactive', '_get_backend_mod',
'_get_pyplot_commands', '_log', '_pylab_helpers',
'_warn_if_gui_out_of_main_thread', 'acorr', 'angle_spectrum',
'annotate', 'annotations', 'arrow', 'autoscale', 'autumn', 'axes',
'axhline', 'axhspan', 'axis', 'axline', 'axvline', 'axvspan',
'backend_registry', 'bar', 'bar_label', 'barbs', 'barh', 'bone',
'box', 'boxplot', 'broken_barh', 'cast', 'cbook', 'cla', 'clabel',
'clf', 'clim', 'close', 'cm', 'cohere', 'color_sequences', 'colorbar',
'colormaps', 'connect', 'contour', 'contourf', 'cool', 'copper',
'csd', 'cycler', 'delaxes', 'disconnect', 'draw', 'draw_all',
'draw_if_interactive', 'ecdf', 'errorbar', 'eventplot', 'figaspect',
'figimage', 'figlegend', 'fignum_exists', 'figtext', 'figure', 'fill',
'fill_between', 'fill_betweenx', 'findobj', 'flag', 'functools',
'gca', 'gcf', 'gci', 'get', 'get_backend', 'get_cmap',
'get_current_fig_manager', 'get_figlabels', 'get_fignums',
'get_plot_commands', 'get_scale_names', 'getp', 'ginput', 'gray',
'grid', 'hexbin', 'hist', 'hist2d', 'hlines', 'hot', 'hsv',
'importlib', 'imread', 'imsave', 'imshow', 'inferno', 'inspect',
'install_repl_displayhook', 'interactive', 'ioff', 'ion',
'isinteractive', 'jet', 'legend', 'locator_params', 'logging',
'loglog', 'magma', 'magnitude_spectrum', 'margins', 'matplotlib',
'matshow', 'minorticks_off', 'minorticks_on', 'mlab',
'new_figure_manager', 'nipy_spectral', 'np', 'overload', 'pause',
'pcolor', 'pcolormesh', 'phase_spectrum', 'pie', 'pink', 'plasma',
'plot', 'plot_date', 'polar', 'prism', 'psd', 'quiver', 'quiverkey',
'rc', 'rcParams', 'rcParamsDefault', 'rcParamsOrig', 'rc_context',
'rcdefaults', 'rcsetup', 'rgrids', 'savefig', 'sca', 'scatter', 'sci',
'semilogx', 'semilogy', 'set_cmap', 'set_loglevel', 'setp', 'show',
'specgram', 'spring', 'spy', 'stackplot', 'stairs', 'stem', 'step',
'streamplot', 'style', 'subplot', 'subplot2grid', 'subplot_mosaic',
'subplot_tool', 'subplots', 'subplots_adjust', 'summer', 'suptitle',
'switch_backend', 'sys', 'table', 'text', 'thetagrids', 'threading',
'tick_params', 'ticklabel_format', 'tight_layout', 'time', 'title',
'tricontour', 'tricontourf', 'tripcolor', 'triplot', 'twinx', 'twiny',
'uninstall_repl_displayhook', 'violinplot', 'viridis', 'vlines',
'waitforbuttonpress', 'winter', 'xcorr', 'xkcd', 'xlabel', 'xlim',
'xscale', 'xticks', 'ylabel', 'ylim', 'yscale', 'yticks']
```

Histogram

- Histogram is used to show the frequency Distribution
- plt.hist()

```
a = [2,3,4,5,6,7,8,9,1,2,3,7,8,9,3]
bins = 5
plt.hist(a,bins,width=0.5)
# bins()  - represents intervals(or) range of values on x axis
```

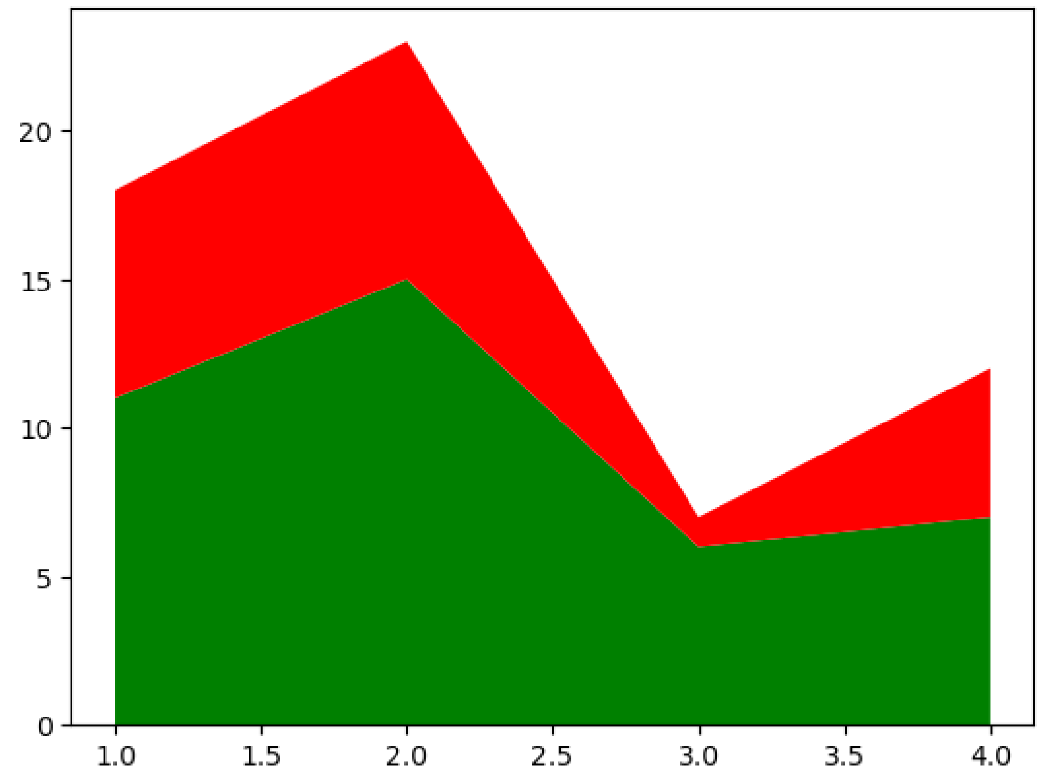
```
(array([3., 4., 1., 3., 4.]),
 array([1. , 2.6, 4.2, 5.8, 7.4, 9. ]),
 <BarContainer object of 5 artists>)
```



Area Plot

- plt.stackplot()

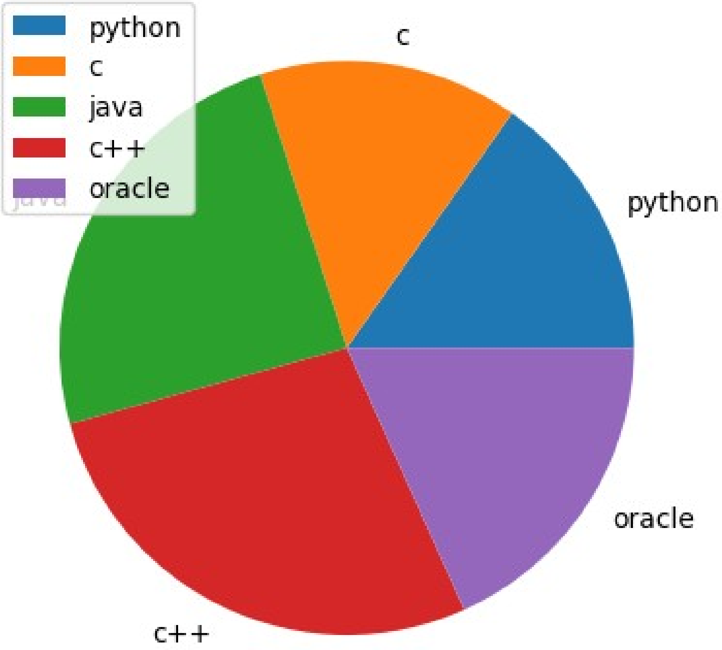
```
days = [1,2,3,4]
working = [11,15,6,7]
sleeping = [7,8,1,5]
plt.stackplot(days,working,sleeping,colors=['g','r'])
plt.show()
```



pie chart

- plt.pie()

```
rating = [5,4.8,8,9,6]
names = ['python','c','java','c++','oracle']
plt.pie(rating,labels=names)
plt.legend()
plt.show()
```


python
c
java
c++
oracle
c
python
java
oracle
c++