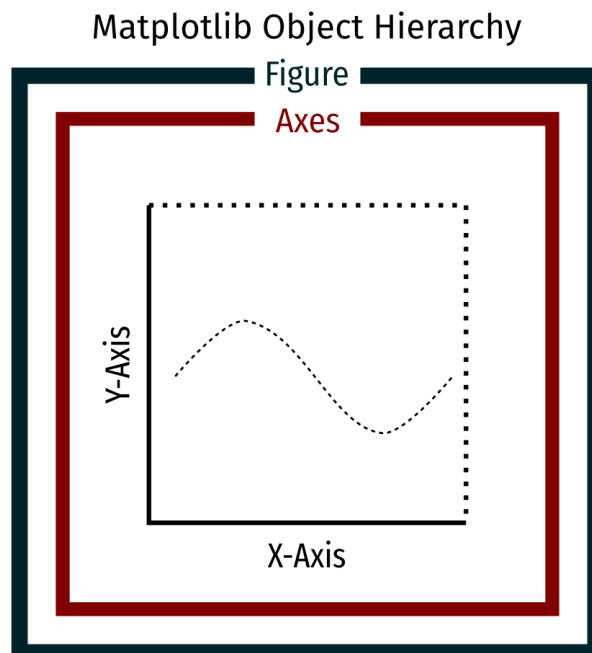


Unit 1 - Data Visualization

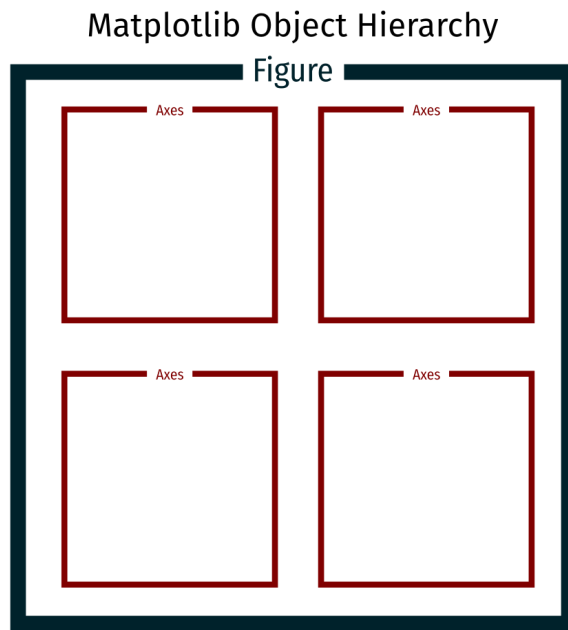


Matplotlib Object Hierarchy

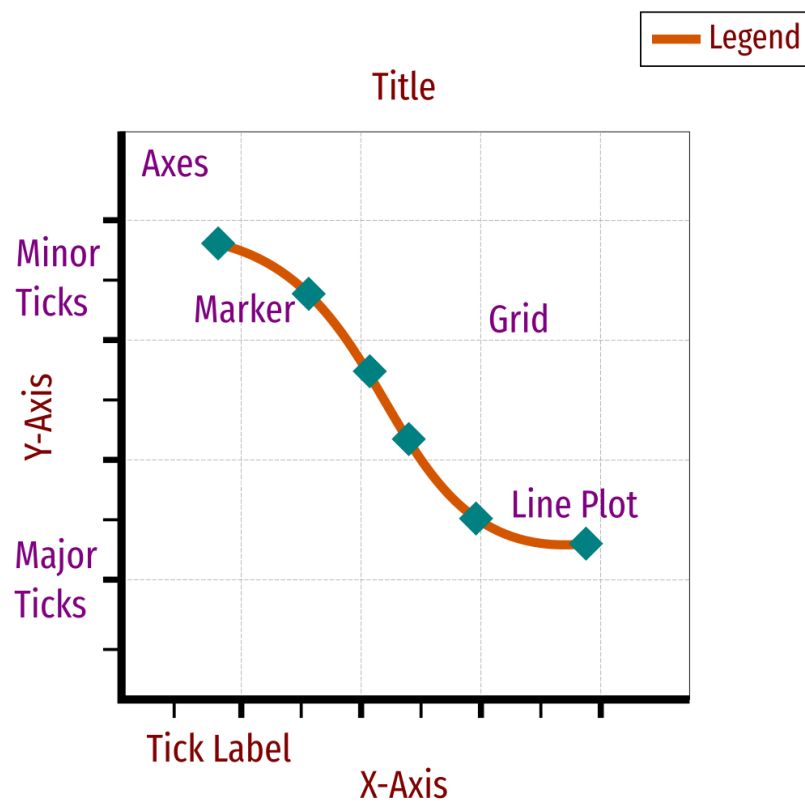
1. Single Object Hierarchy



2. Multiple Object Hierarchy



Anatomy of Figure



Installation

```
pip install matplotlib
```

OR

```
python -m pip install matplotlib
```

OR

If installing inside Jupyter Notebook

```
!pip install matplotlib
```

Import

```
from matplotlib import pyplot as plt  
%matplotlib inline
```

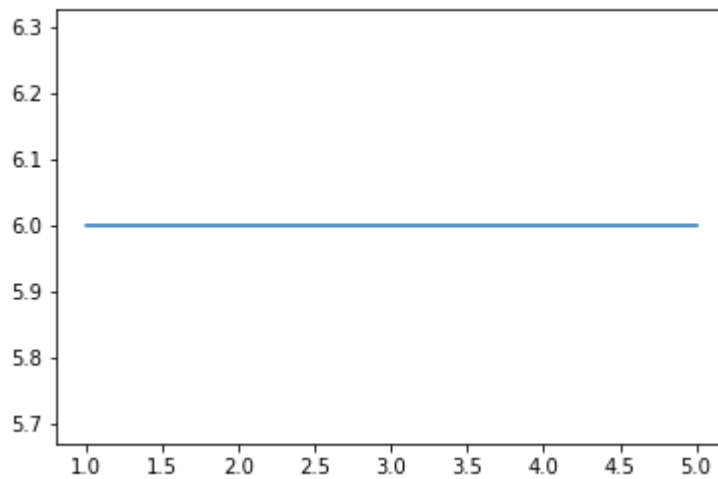
OR

```
import matplotlib.pyplot as plt  
%matplotlib inline
```

Line Plot

`pyplot.plot(x, y, color, linewidth, linestyle, marker, label)`

```
import matplotlib.pyplot as plt  
  
x = [1,2,3,4,5] # X Points  
y = [6,6,6,6,6] # Y Points  
  
# using plot() function for Line Plot  
plt.plot(x,y) # Create the Plot  
plt.show() # Show the Plot
```



Customizing Line Plot

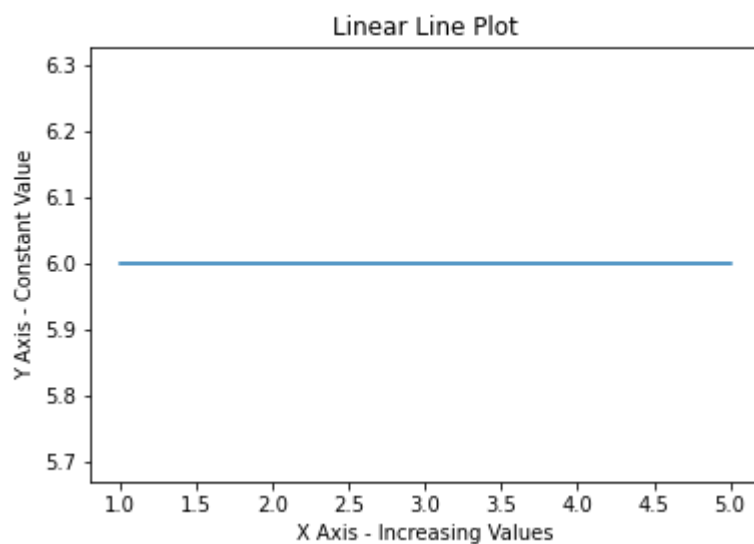
1. Adding Title and Axis Labels

1. Set title property to plot, for adding Title on top.
2. Set xlabel and ylabel property for Axis Labels

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y)
plt.title("Linear Line Plot")
plt.xlabel("X Axis - Increasing Values")
plt.ylabel("Y Axis - Constant Value")
plt.show()
```



2. Adding Legend

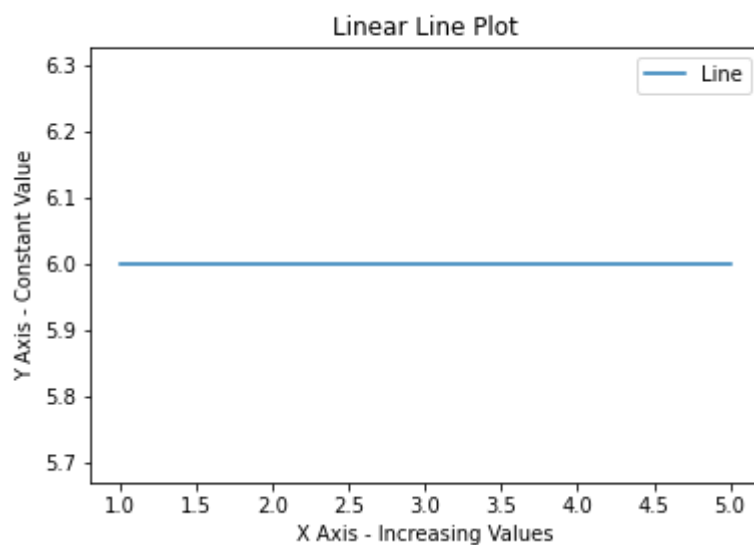
1. Specify label parameter during plot creation.
2. Set legend property to plot.
3. In loc parameter either define Location String or Location Code.

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y, label='Line')
plt.title("Linear Line Plot")
plt.xlabel("X Axis - Increasing Values")
plt.ylabel("Y Axis - Constant Value")
plt.legend(loc='best')
plt.show()
```



3. Changing Line Color

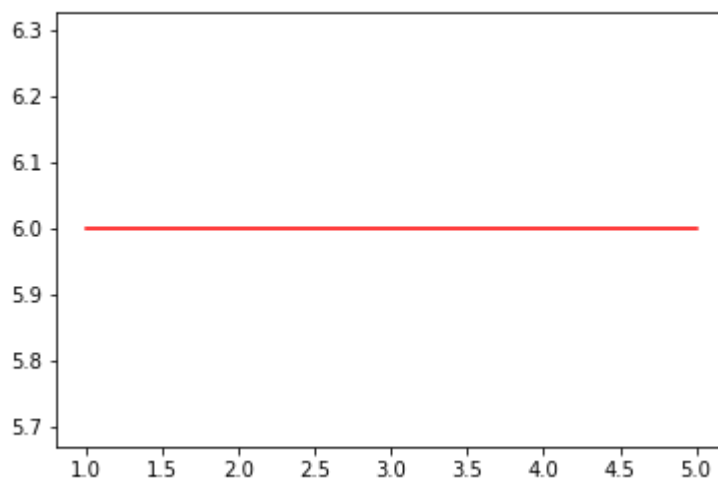
- Specify color parameter during plot creation.
- Color can be specified as a color name, color character or Hex String.

Color Name	Character
Black	'k'
White	'w'
Red	'r'
Green	'g'
Blue	'b'
Cyan	'c'
Magenta	'm'
Yellow	'y'

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y, color = 'r')
plt.show()
```



4. Changing Line Style

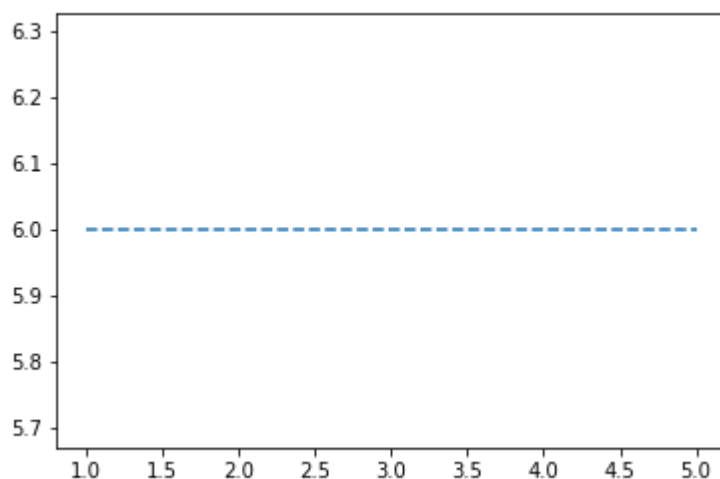
- Specify linestyle parameter during plot creation.
- Line Style can be specified as below:

Line Style	Description
'-'	Solid Line
'--'	Dashed Line
'-.'	Dash-Dot Line
':'	Dotted Line

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y, linestyle = '--')
plt.show()
```



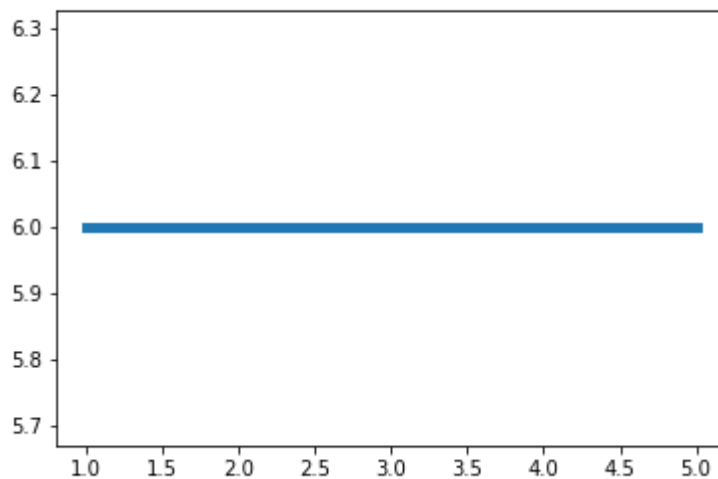
5. Changing Line Width

- Specify the linewidth parameter during plot creation.
- Line Width can be specified as integer value.


```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y, linewidth = 5)
plt.show()
```



6. Adding Marker

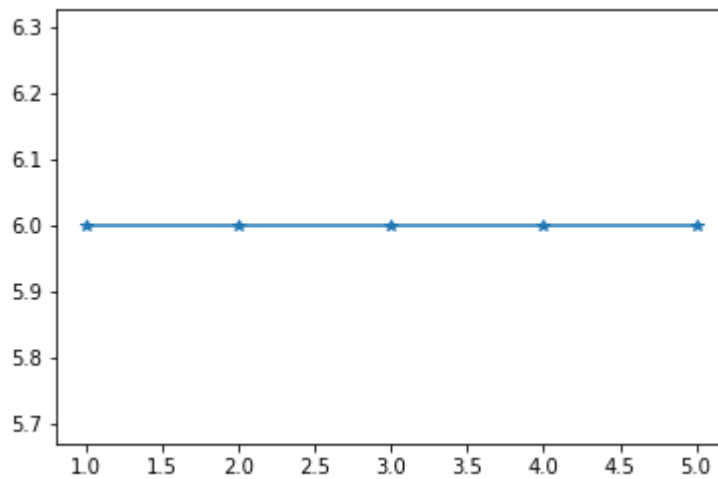
- Marker can be added by specifying marker parameter during plot creation.
- Marker Style can be specified as below:

Character	Description
'o'	Circle Marker
'v'	Triangle Down Marker
'^'	Triangle Up Marker
'<'	Triangle Left Marker
'>'	Triangle Right Marker
's'	Square Marker
'p'	Pentagon Marker
'h'	Hexagon Marker
'8'	Octagon Marker
'*'	Star Marker
'+'	Plus Marker
'x'	Cross Marker
'd'	Diamond Marker

```
import matplotlib.pyplot as plt

x = [1,2,3,4,5] # X Points
y = [6,6,6,6,6] # Y Points

# using plot() function for Line Plot
plt.plot(x,y, marker = '*')
plt.show()
```

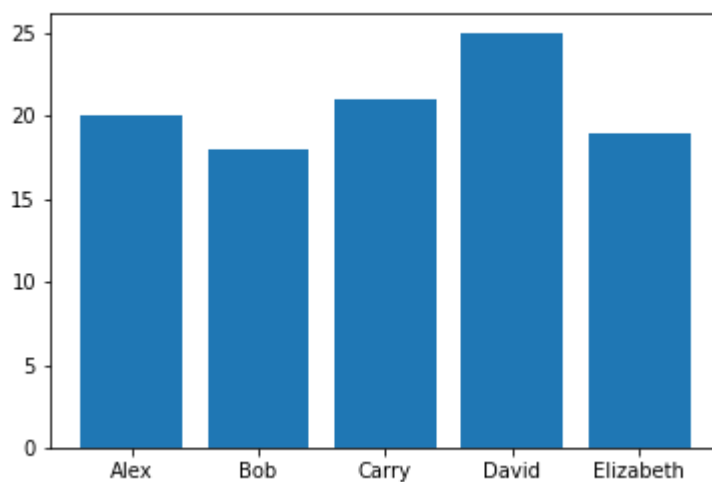


Bar Chart

`pyplot.bar(height, x, width, color, edgecolor, linewidth, label)`

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x)
plt.show()
```



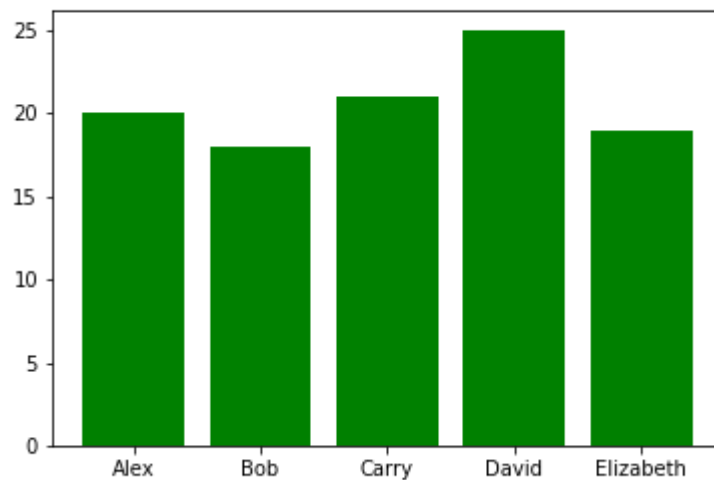
Customizing Bar Chart

1. Changing Bar Color

- Use color parameter during bar chart creation.

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x, color='g')
plt.show()
```

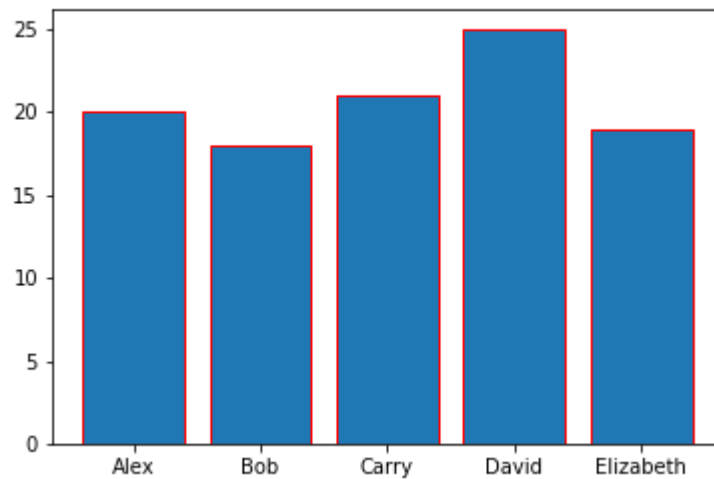


2. Changing Edge Color

- Use edgecolor parameter during bar chart creation.

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x, edgecolor='r')
plt.show()
```

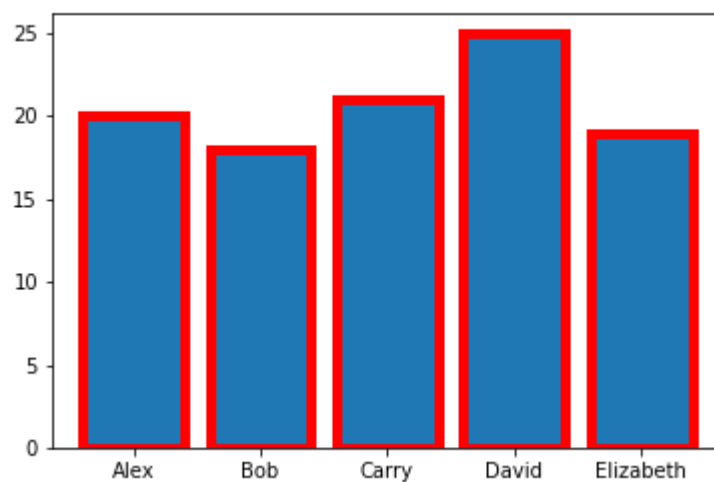


3. Changing Line Width

- Use linewidth parameter during bar chart creation

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x, edgecolor='r', linewidth=5)
plt.show()
```

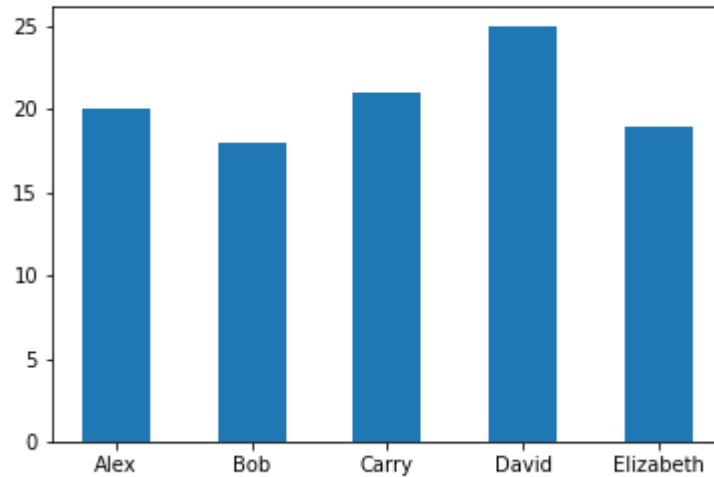


4. Changing Bar Width

- Use width parameter during bar chart creation

```
import matplotlib.pyplot as plt

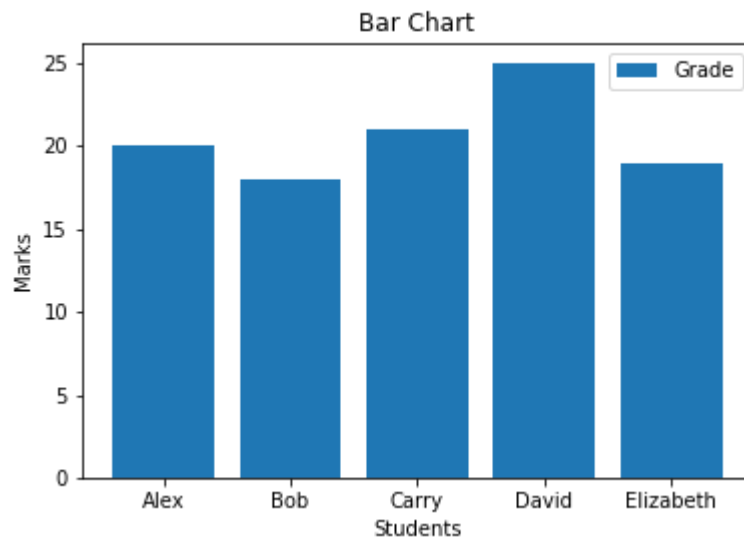
x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x, width=0.5)
plt.show()
```



5. Adding Title, Axis Labels & Legends

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using bar() function for Bar Chart
plt.bar(grades, x, label='Grade')
plt.title("Bar Chart")
plt.xlabel("Students")
plt.ylabel("Marks")
plt.legend(loc='best')
plt.show()
```

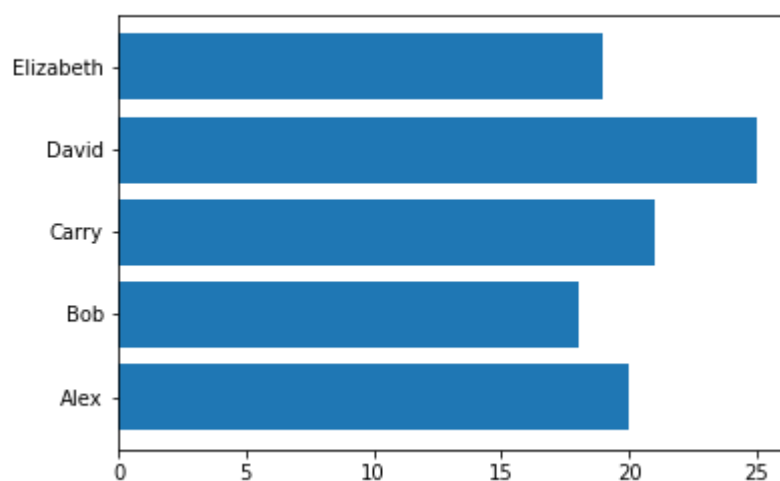


Horizontal Bar Chart

`pyplot.barh(width, x, height, color, edgecolor, linewidth, label)`

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
grades = ['Alex','Bob','Carry','David','Elizabeth']
# using barh() function for Horizontal Bar Chart
plt.barh(grades, x, height=0.8)
plt.show()
```

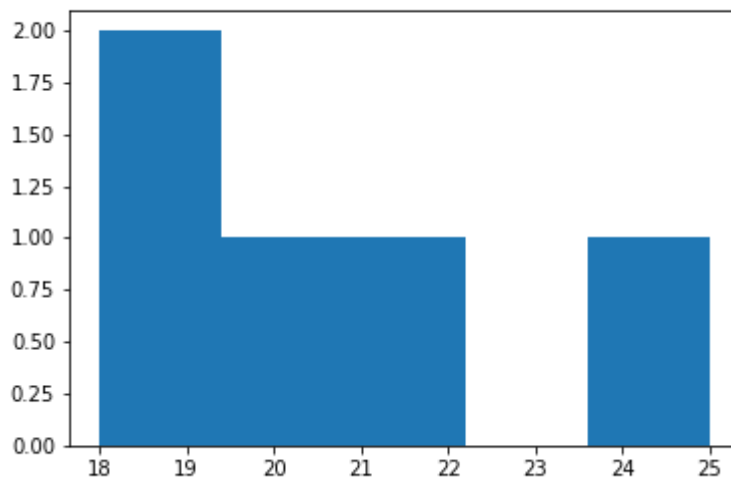


Histogram

```
pyplot.hist(x, bins, color, edgecolor, linewidth, histtype)
```

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5)
plt.show()
```

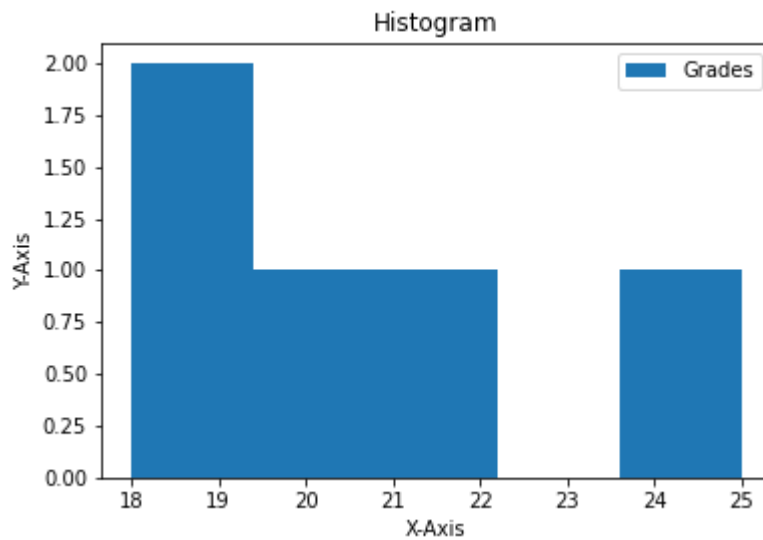


Customization

1. Adding Title, Axis-Labels & Legend

```
import matplotlib.pyplot as plt

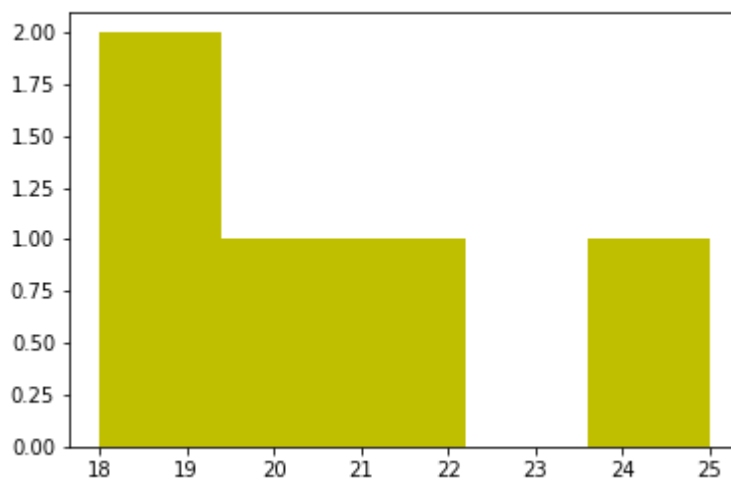
x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5,label="Grades")
plt.title("Histogram")
plt.xlabel("X-Axis")
plt.ylabel("Y-Axis")
plt.legend(loc='best')
plt.show()
```

2. Changing Face Color

```
import matplotlib.pyplot as plt

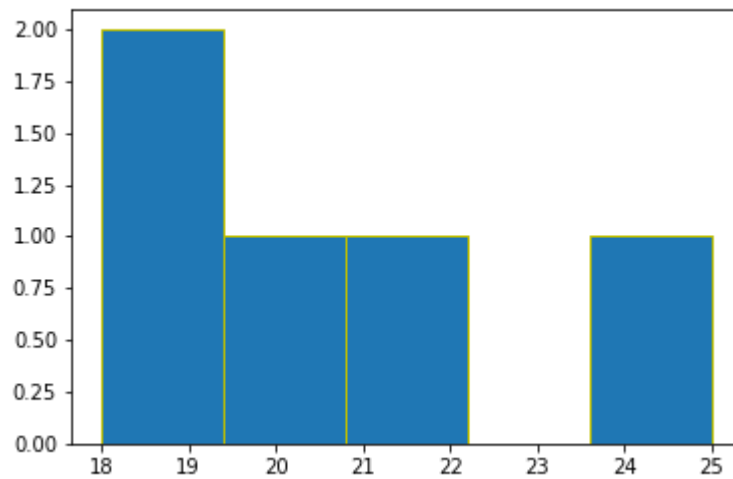
x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5, facecolor='y')
plt.show()
```



3. Changing Edge Color

```
import matplotlib.pyplot as plt

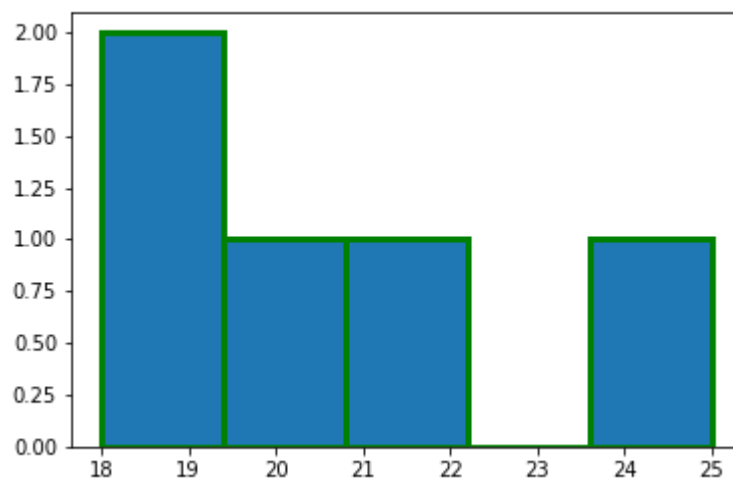
x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5, edgecolor='y')
plt.show()
```



4. Changing Line Width

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5, edgecolor='g', linewidth=3)
plt.show()
```



5. Changing Hist Type

histtypes are 'bar', 'barstacked', 'step', 'stepfilled'

```
import matplotlib.pyplot as plt

x = [20,18,21,25,19] # X Points
# using hist() function for Histogram
plt.hist(x,bins=5, histtype='step')
plt.show()
```

