

Matplotlib Graphs

Import Useful Libraries & Dataset

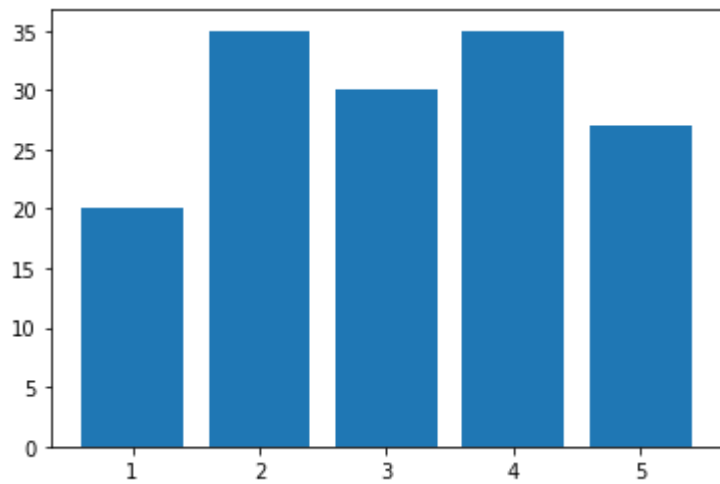
```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('E:/WORK/MACHINE LEARNING & ARTIFICIAL INTELLIGENCE/Machine Learning With Python/02 - Introduction To Machine Learning/03 - Matplotlib/iris.csv')
```

01 - Simple Bar Graph

```
In [2]: x = np.arange(1,6)
y = (20, 35, 30, 35, 27)

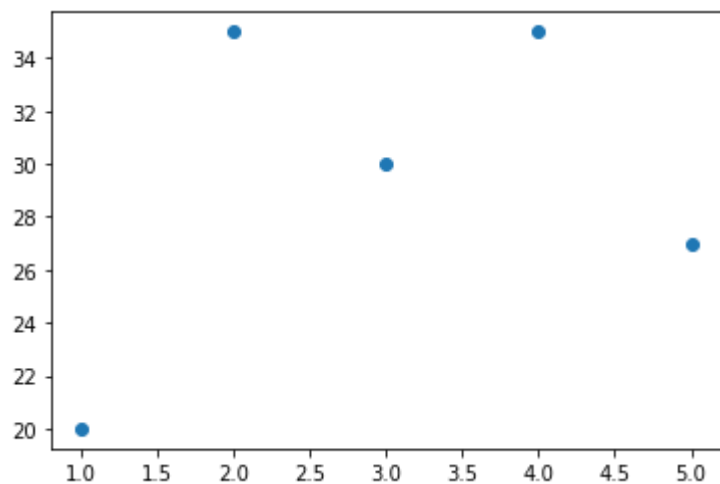
a = plt.bar(x,y)
plt.show()
```



02 - Simple Scatter Plot

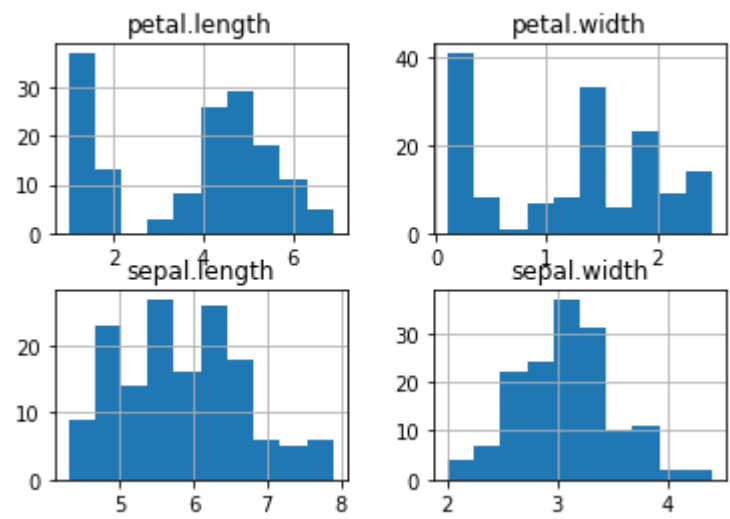
```
In [3]: x = np.arange(1,6)
y = (20, 35, 30, 35, 27)

b = plt.scatter(x,y)
plt.show()
```



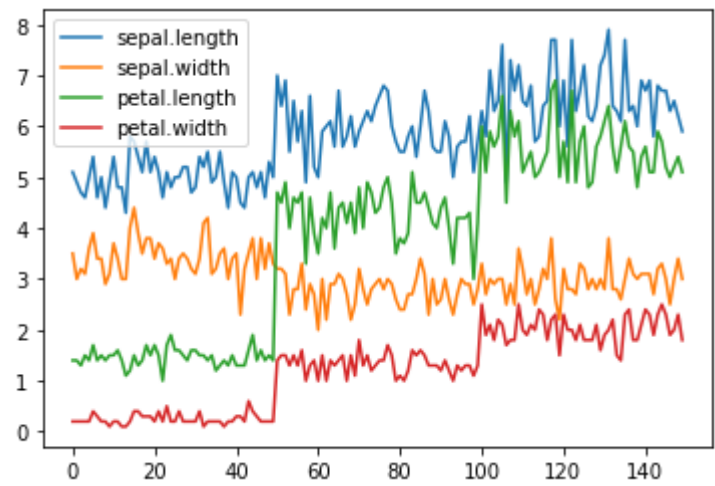
03 - Histogram

```
In [4]: df.hist()  
plt.show()
```



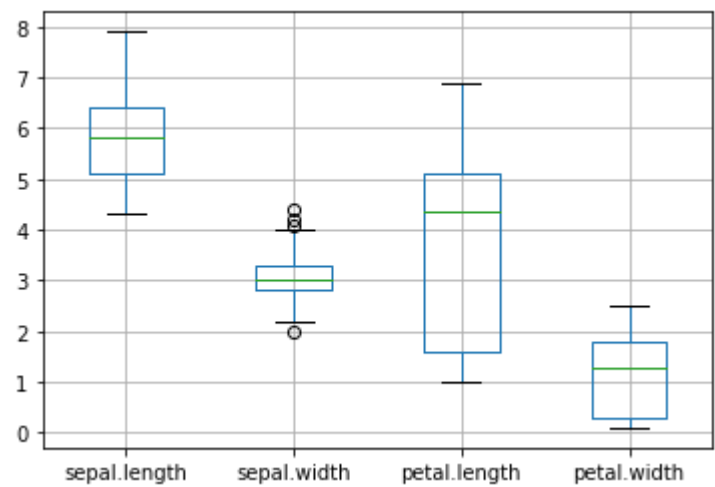
04 - Line Graph

```
In [5]: df.plot()  
plt.show()
```



05 - Box Plot

```
In [6]: df.boxplot()  
plt.show()
```



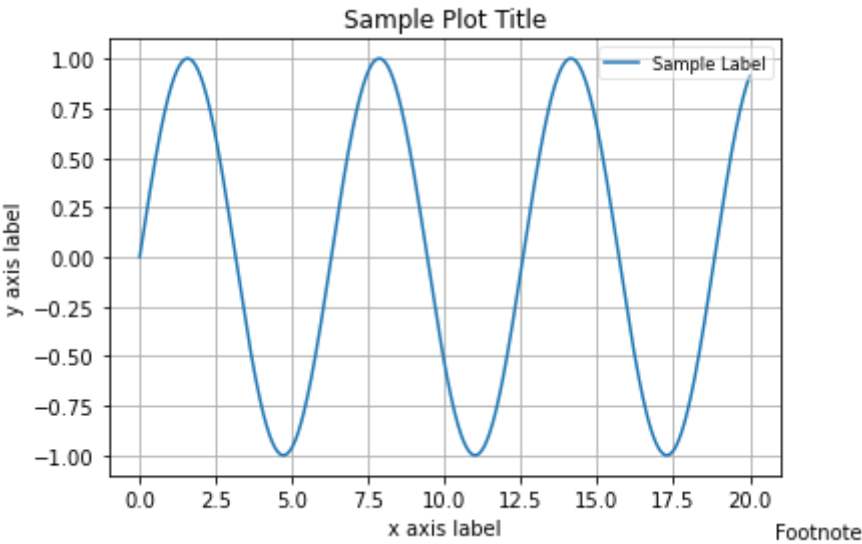
06 - Customized Plot

```
In [7]: # generate sample data
x = np.linspace(0, 20, 1000)
y = np.sin(x)

# customize axis labels
plt.plot(x, y, label = 'Sample Label')
plt.title('Sample Plot Title')
plt.xlabel('x axis label')
plt.ylabel('y axis label')
plt.grid(True)

# add footnote
plt.figtext(0.995, 0.01, 'Footnote', ha='right', va='bottom')

# add legend, location pick the best automatically
plt.legend(loc='best', framealpha=0.5, prop={'size':'small'})
plt.show()
```

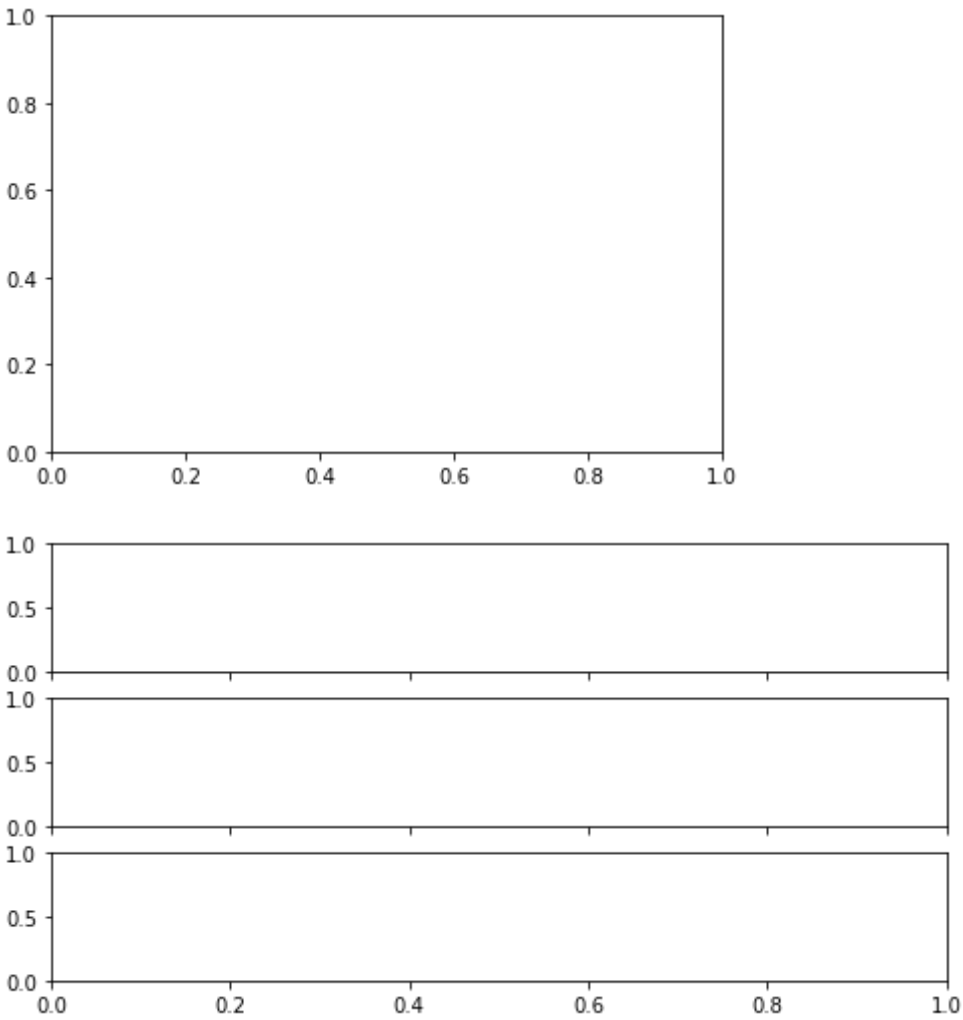


07 - Object-oriented customization

```
In [8]: fig, ax = plt.subplots()
fig,(ax1,ax2,ax3) = plt.subplots(nrows=3, ncols=1, sharex=True, sharey=False, figsize=(8,4))

# Iterating the Axes within a Figure
for ax in fig.get_axes():
    pass

plt.show()
```



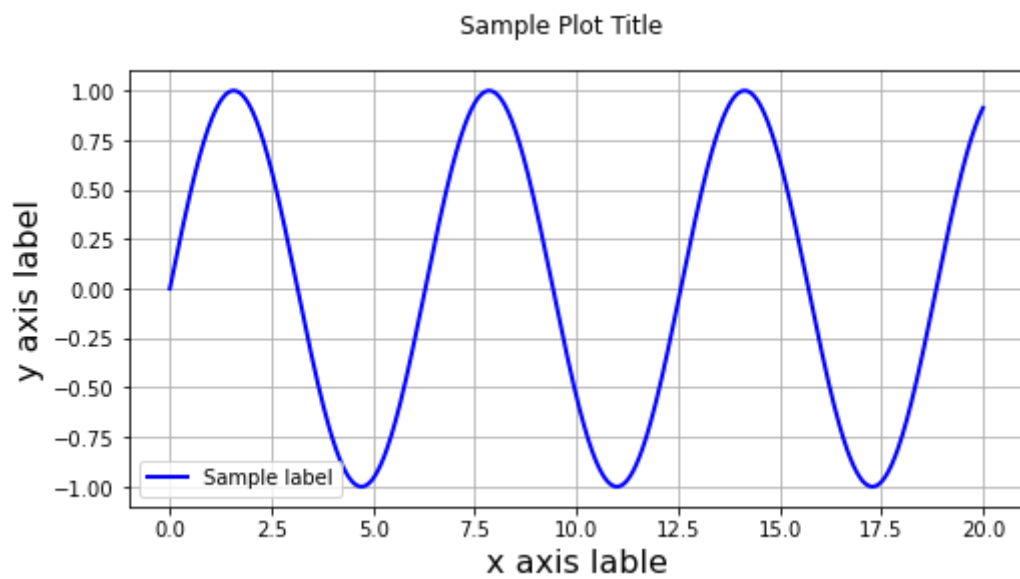
08 - Single line plot using ax.plot()

```
In [9]: x = np.linspace(0, 20, 1000)
y = np.sin(x)
fig = plt.figure(figsize=(8,4))           # get an empty figure and add an Axes

ax = fig.add_subplot(1,1,1)              # row-col-num

ax.plot(x, y, 'b-', linewidth=2, label='Sample label')           # line plot data on the Axes

ax.set_xlabel('x axis lable', fontsize=16)           # x label
ax.set_ylabel('y axis label', fontsize=16)           # y label
ax.legend(loc='best')
ax.grid(True)
fig.suptitle('Sample Plot Title')           # title
plt.show()
```



09 - Multiple line plot on same axis

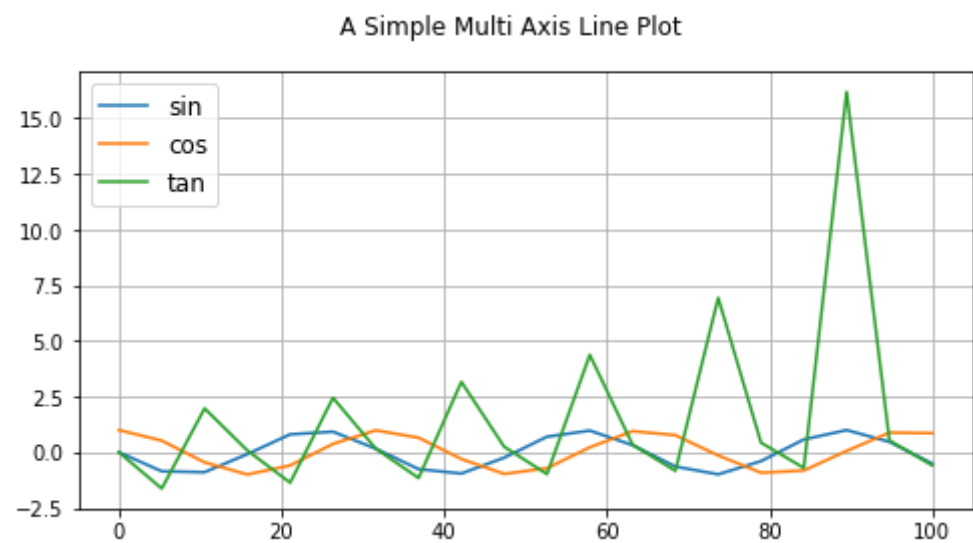
```
In [10]: fig, ax = plt.subplots(figsize=(8,4))

x1 = np.linspace(0, 100, 20)
x2 = np.linspace(0, 100, 20)
x3 = np.linspace(0, 100, 20)
y1 = np.sin(x1)
y2 = np.cos(x2)
y3 = np.tan(x3)

ax.plot(x1, y1, label='sin')
ax.plot(x2, y2, label='cos')
ax.plot(x3, y3, label='tan')

ax.grid(True)
ax.legend(loc='best', prop={'size':'large'})
fig.suptitle('A Simple Multi Axis Line Plot')

plt.show()
```



10 - Multiple lines on different axis

```
In [11]: # Changing sharex to True will use the same x axis
fig, (ax1,ax2,ax3) = plt.subplots(nrows=3, ncols=1, sharex=False, sharey = False, figsize=(8,4))

# plot some lines
x1 = np.linspace(0, 100, 20)
x2 = np.linspace(0, 100, 20)
x3 = np.linspace(0, 100, 20)

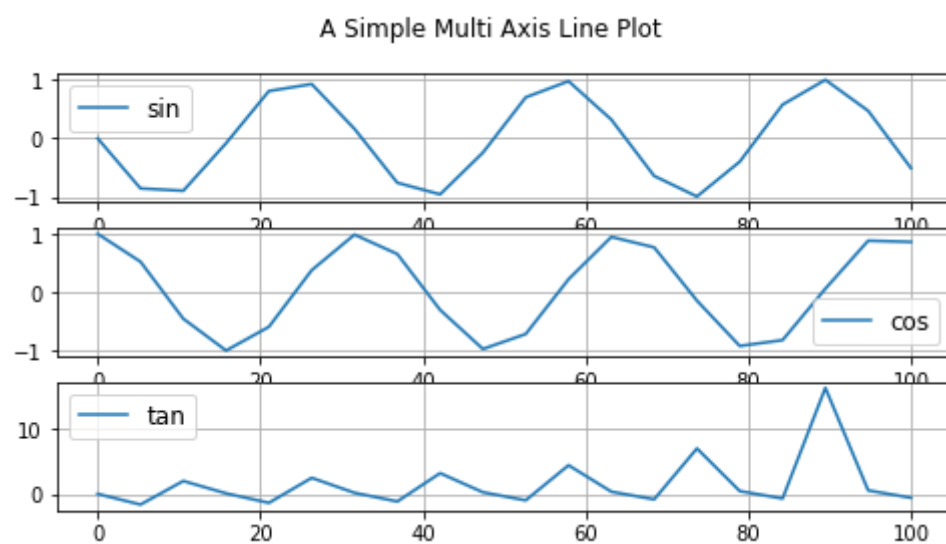
y1 = np.sin(x1)
y2 = np.cos(x2)
y3 = np.tan(x3)

ax1.plot(x1, y1, label='sin')
ax2.plot(x2, y2, label='cos')
ax3.plot(x3, y3, label='tan')

# add grid, legend, title and save
ax1.grid(True)
ax2.grid(True)
ax3.grid(True)

ax1.legend(loc='best', prop={'size':'large'})
ax2.legend(loc='best', prop={'size':'large'})
ax3.legend(loc='best', prop={'size':'large'})

fig.suptitle('A Simple Multi Axis Line Plot')
plt.show()
```



11 - Line style and marker style controls

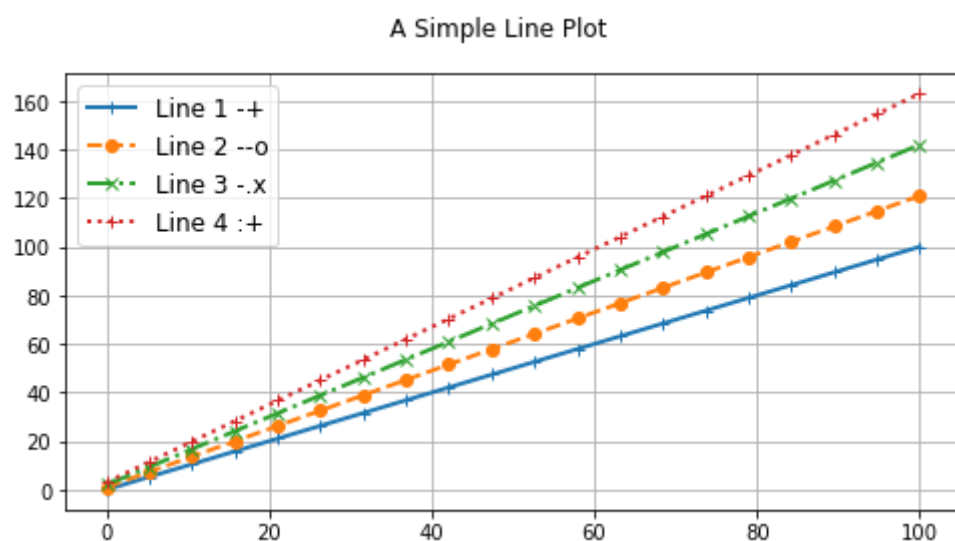
```
In [12]: fig, ax = plt.subplots(figsize=(8,4))

N = 4 # the number of lines we will plot
styles = ['- ', '--', '-.', ':']
markers = list('+ox')

x = np.linspace(0, 100, 20)

for i in range(N): # add line-by-line
    y = x + x/5*i + i
    s = styles[i % len(styles)]
    m = markers[i % len(markers)]
    ax.plot(x, y, label='Line ' + str(i+1) + ' ' + s + m, marker=m, linewidth=2, linestyle=s)

# add grid, legend, title and save
ax.grid(True)
ax.legend(loc='best', prop={'size':'large'})
fig.suptitle('A Simple Line Plot')
fig.savefig('Task08.png', dpi=125) # Dots per inch
plt.show()
```



12 - Bar plot using ax.bar()

```
In [13]: N = 4
labels = list('ABCD')
data = np.array(range(N)) + np.random.rand(N)

# Plot the data
fig, ax = plt.subplots(figsize=(8, 3.5))
width = 0.5
tickLocations = np.arange(N)
rectLocations = tickLocations-(width/2.0)

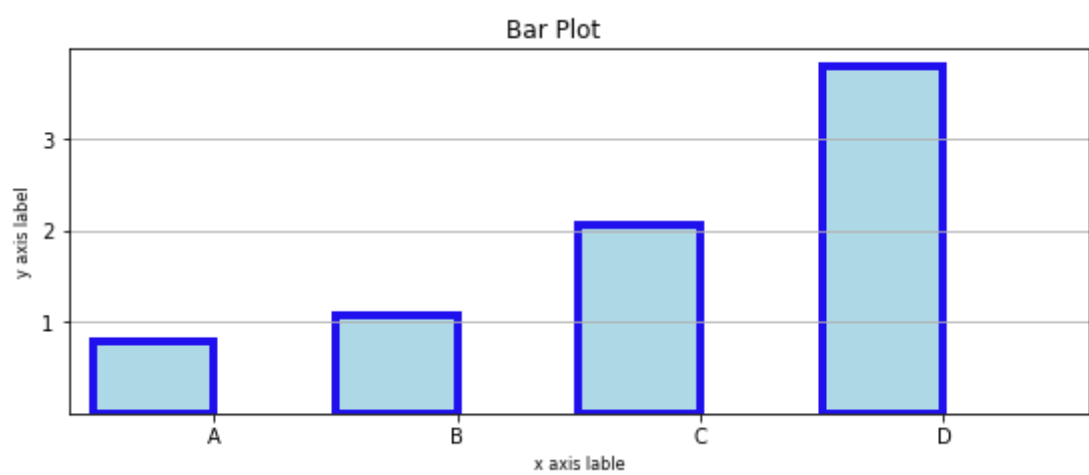
# For color either HEX value of the name of the color can be used
ax.bar(rectLocations, data, width, color='lightblue', edgecolor='#1f10ed', linewidth=4.0)

# Setting the plot
ax.set_xticks(ticks= tickLocations)      # It will set the locations on the axes
ax.set_xticklabels(labels)               # It will set the displayed text
ax.set_xlim(min(tickLocations)-0.6, max(tickLocations)+0.6)

ax.set_yticks(range(N)[1:])
ax.set_ylim((0,N))

ax.yaxis.grid(True)

ax.set_ylabel('y axis label', fontsize=8)
ax.set_xlabel('x axis lable', fontsize=8)
fig.suptitle("Bar Plot")
fig.tight_layout(pad=2)
plt.show()
```



13 - Horizontal bar charts

```
In [14]: N = 4
labels = list('ABCD')

data = np.array(range(N)) + np.random.rand(N)

# plot the data
fig, ax = plt.subplots(figsize=(8, 3.5))

width = 0.5
tickLocations = np.arange(N)
rectLocations = tickLocations-(width/2.0)

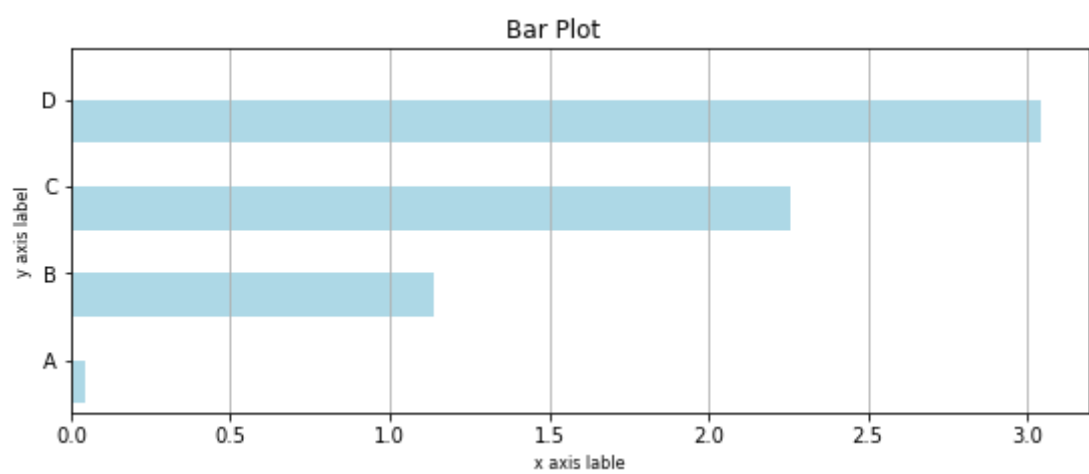
# for color either HEX value of the name of the color can be used
ax.barh(rectLocations, data, width, color='lightblue')

# Setting the plot
ax.set_yticks(ticks= tickLocations)
ax.set_yticklabels(labels)
ax.set_ylim(min(tickLocations)-0.6, max(tickLocations)+0.6)

ax.xaxis.grid(True)

ax.set_ylabel('y axis label', fontsize=8) # y Label
ax.set_xlabel('x axis lable', fontsize=8) # x Label

fig.suptitle("Bar Plot")
fig.tight_layout(pad=2)
plt.show()
```



14 - Side-by-Side Bar Chart

```
In [15]: pre = np.array([19, 6, 11, 9])
post = np.array([15, 11, 9, 8])

labels=['Survey '+x for x in list('ABCD')]

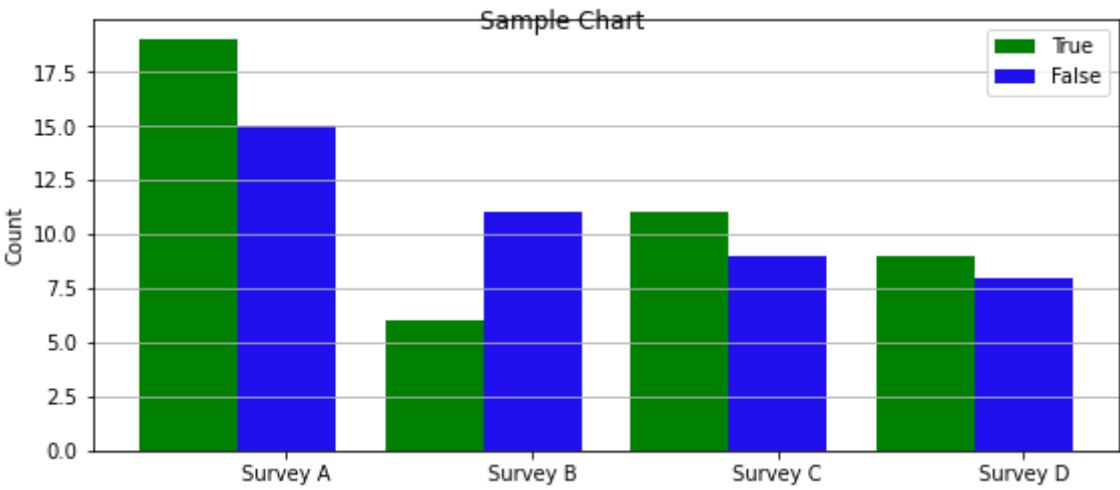
# the plot - left then right
fig, ax = plt.subplots(figsize=(8, 3.5))
width = 0.4 # bar width
xlocs = np.arange(len(pre))

ax.bar(xlocs-width, pre, width, color='green', label='True')
ax.bar(xlocs, post, width, color='#1f10ed', label='False')

ax.set_xticks(ticks=range(len(pre)))
ax.set_xticklabels(labels)

ax.yaxis.grid(True)
ax.legend(loc='best')
ax.set_ylabel('Count')

fig.suptitle('Sample Chart')
fig.tight_layout(pad=1)
plt.show()
```



15 - Stacked Bar Charts

```
In [16]: pre = np.array([19, 6, 11, 9])
post = np.array([15, 11, 9, 8])
labels=['Survey '+x for x in list('ABCD')]

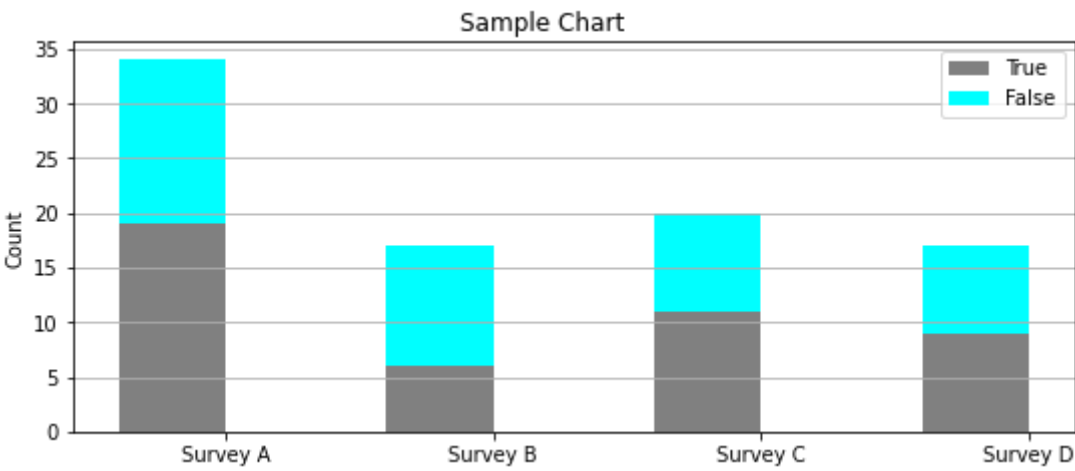
# the plot - left then right
fig, ax = plt.subplots(figsize=(8, 3.5))

width = 0.4
xlocs = np.arange(len(pre)+2)
adjlocs = xlocs[1:-1] - width/2.0

ax.bar(adjlocs, pre, width, color='grey', label='True')
ax.bar(adjlocs, post, width, color='cyan', label='False', bottom=pre)

ax.set_xticks(ticks=xlocs[1:-1])
ax.set_xticklabels(labels)
ax.yaxis.grid(True)
ax.legend(loc='best')
ax.set_ylabel('Count')

fig.suptitle('Sample Chart')
fig.tight_layout(pad=2)
plt.show()
```



16 - Pie chart

```
In [17]: data = np.array([15,8,4])

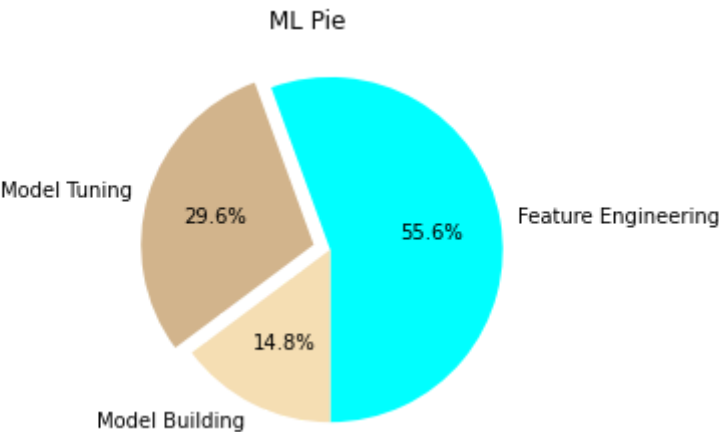
labels = ['Feature Engineering', 'Model Tuning', 'Model Building']

explode = (0, 0.1, 0)
colrs=['cyan', 'tan', 'wheat']

fig, ax = plt.subplots(figsize=(8, 3.5))

ax.pie(data, explode=explode, labels=labels, autopct='%1.1f%%', startangle=270, colors=colrs)
ax.axis('equal')           # keep it a circle

fig.suptitle("ML Pie")
plt.show()
```

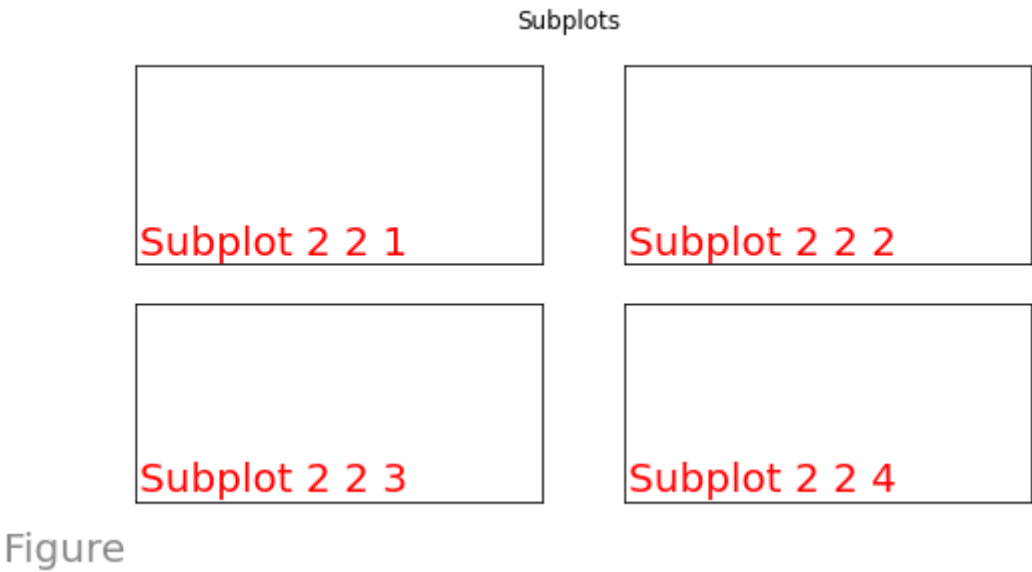


17 - Grid Creation

```
In [18]: fig = plt.figure(figsize=(8,4))
fig.text(x=0.01, y=0.01, s='Figure',color='#888888', ha='left', va='bottom', fontsize=20)

for i in range(4):
    ax = fig.add_subplot(2, 2, i+1)
    ax.text(x=0.01, y=0.01, s='Subplot 2 2 '+str(i+1), color='red', ha='left', va='bottom', fontsize=20)
    ax.set_xticks([])
    ax.set_yticks([])

fig.suptitle('Subplots')
plt.show()
```



Figure