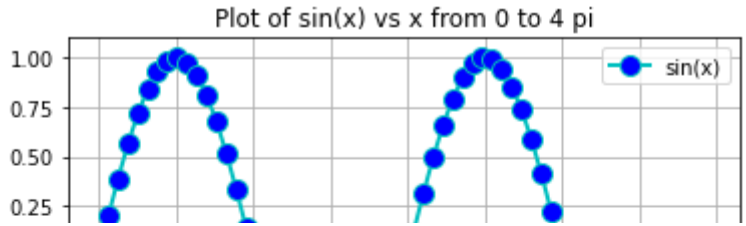
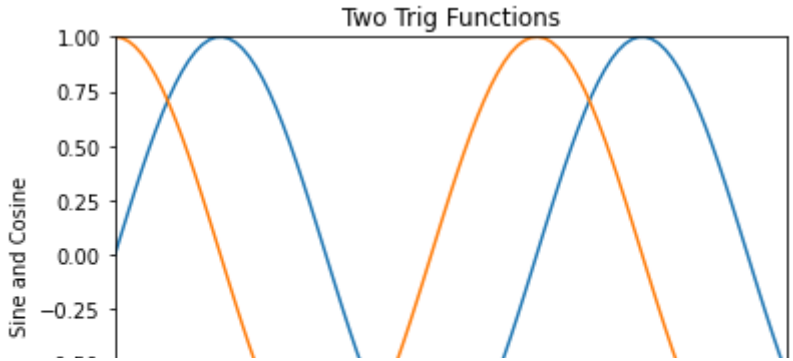
Line chart

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

x = np.arange(0, 4 * np.pi, 0.2)
y = np.sin(x)


# 3. Plot data including options
plt.plot(x, y,linewidth=2,linestyle='-',color='c',marker='o',markersize=10,markerfacecolor='blue')


# 4. Add plot details
plt.title('Plot of sin(x) vs x from 0 to 4 pi')
plt.xlabel('x (0 to 4 pi)')
plt.ylabel('sin(x)')

plt.legend(['sin(x)']) # list containing one string
plt.xticks(np.arange(0, 4*np.pi + np.pi/2, np.pi/2), ['0','pi/2','pi','3pi/2','2pi','5pi/2','3pi','7pi/2','4pi'])
plt.grid(True)


# 5. Show the plot
plt.show()
```

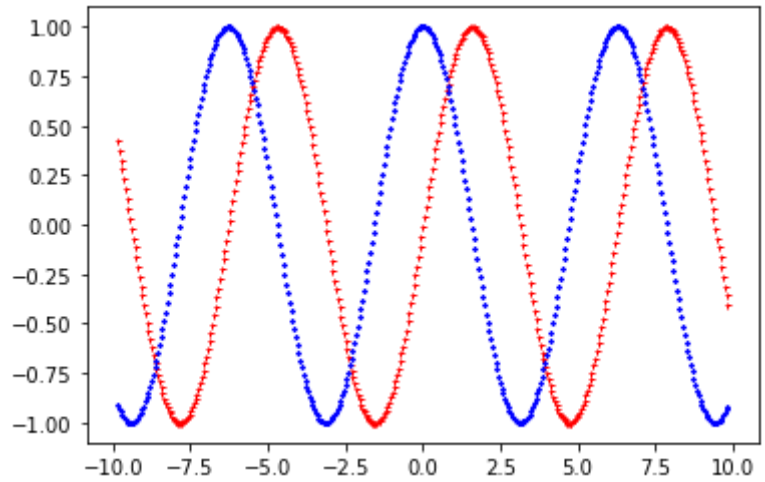



```
#multi-line plot using oo interface
x = np.arange(0,4*np.pi,0.1)
y = np.sin(x)
z = np.cos(x)


fig, ax = plt.subplots()  #create fig,ax object


ax.plot(x,y)
ax.plot(x,z)# plot


ax.set_title('Two Trig Functions')   #  plt.title()
ax.legend(['sin','cos'])             #  plt.legend()
ax.xaxis.set_label_text('Angle 0')    # plt.xlabel()
ax.yaxis.set_label_text('Sine and Cosine')

ax.set_xlim(0, 10)                    #plt.xlim
ax.set_ylim(-1, 1)

plt.savefig('plot.png', dpi=300) # save figure
plt.show()
```




```
#multi-line using plt
x1=np.arange(-3.14*3.14,3.14*3.14,.05)
y2=np.sin(x1)     # 1 waveform on x1

y1=np.cos(x1)     # 2nd waveform on x1

plt.plot(x1,y2,'r+',ms='3')

plt.plot(x1,y1,'b*',ms='2')
```

```
[<matplotlib.lines.Line2D at 0x7f0b6b569590>]
```



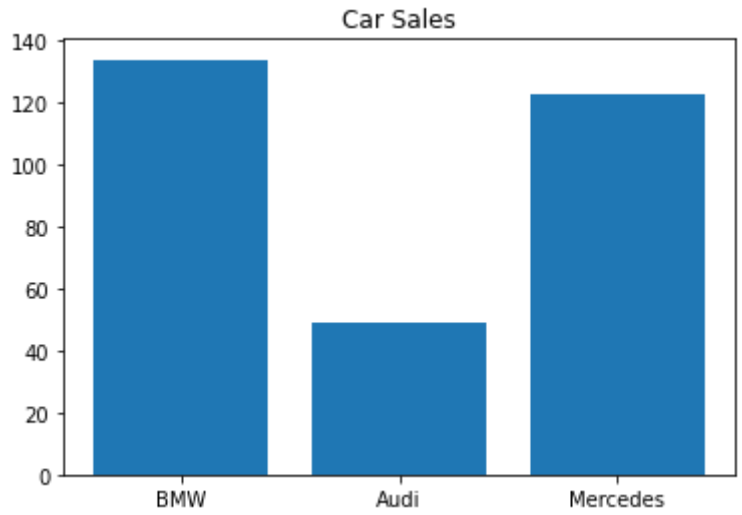
Bar chart

```
import pandas as pd
df=pd.DataFrame(data=[134,49,123],index=['BMW','Audi','Mercedes'],columns=['Number of units sold'])
df
```

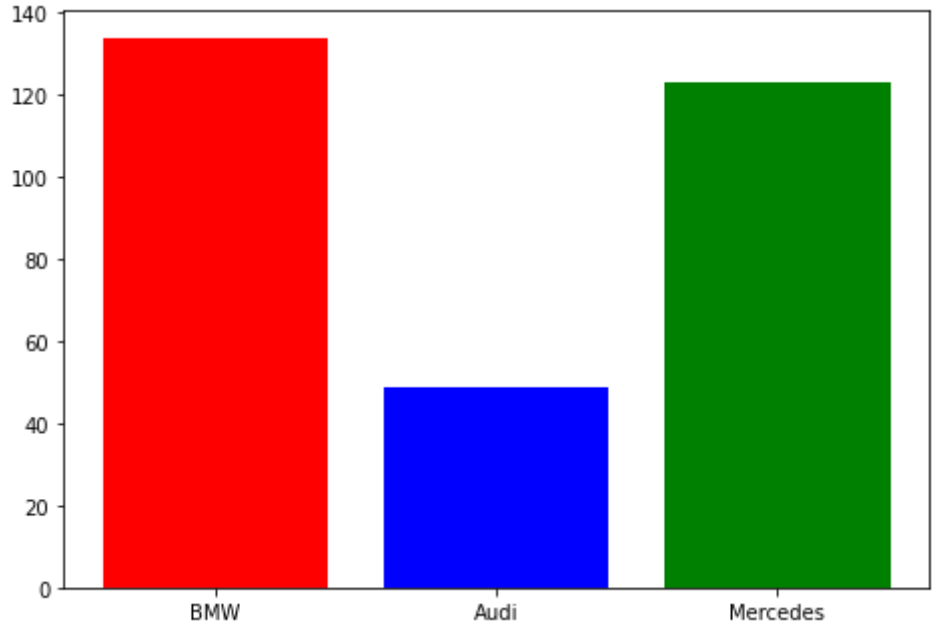
	Number of units sold
BMW	134
Audi	49
Mercedes	123

Visualize Number of units sold vs Car Sales

```
plt.bar(df.index,df['Number of units sold'])
plt.title('Car Sales')
plt.show()
```



```
fig=plt.figure()      #  figure object

axes=fig.add_axes([0,0,1,1])

axes.bar(df.index,df['Number of units sold'],color=['r','b','g'])

plt.show()
```




```
df['Percent']=df['Number of units sold']/sum(df['Number of units sold']) * 100    # sales percent
df
```

	Number of units sold	Percent
BMW	134	43.790850
Audi	49	16.013072
Mercedes	123	40.196078

```
df.sort_values('Number of units sold', ascending=False,inplace=True)   #sort and embed
df
```

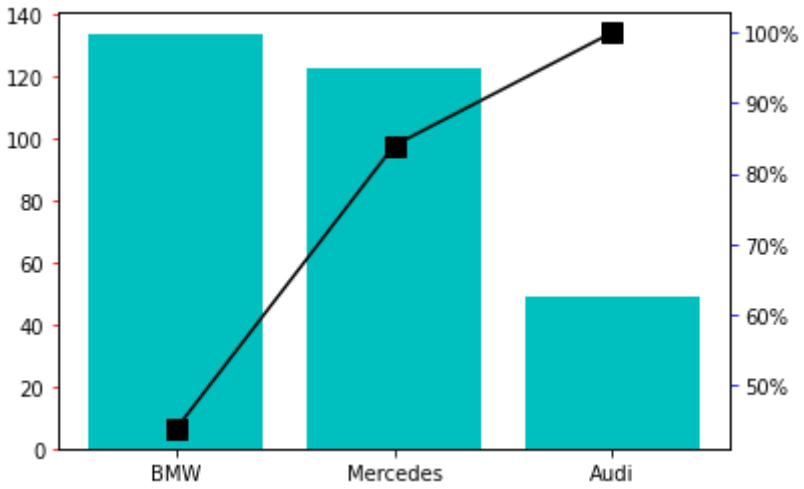
	Number of units sold	Percent
BMW	134	43.790850
Mercedes	123	40.196078
Audi	49	16.013072

```
df['cumulative percent']=(df['Number of units sold'].cumsum()/df['Number of units sold'].sum())*100
df
```

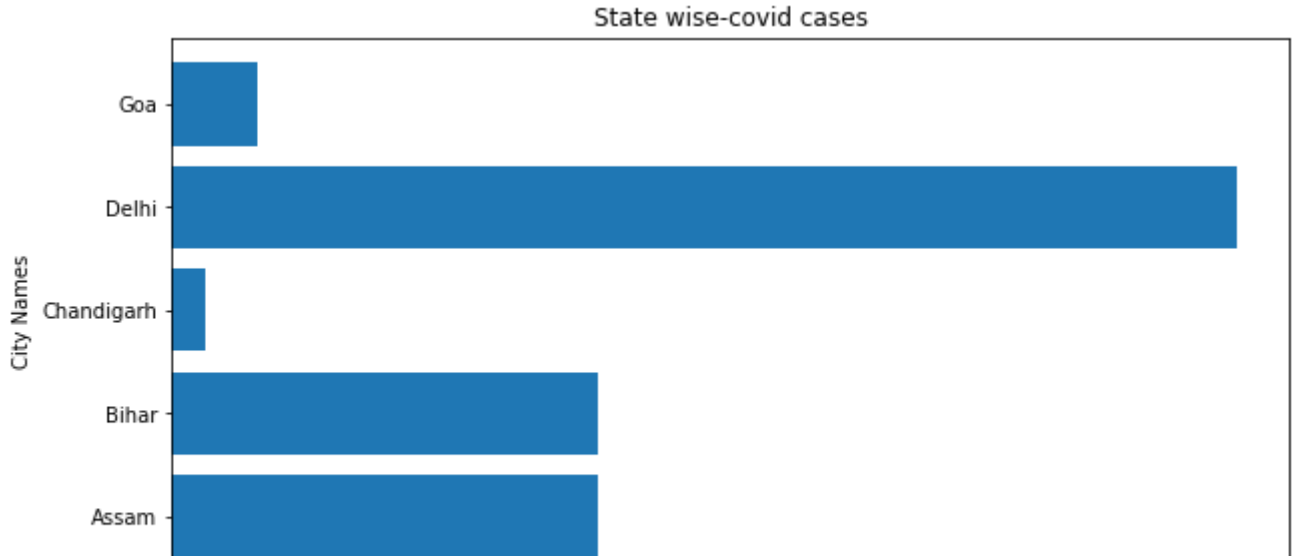
	Number of units sold	Percent	cumulative percent
BMW	134	43.790850	43.790850
Mercedes	123	40.196078	83.986928
Audi	49	16.013072	100.000000

```
from matplotlib.ticker import PercentFormatter
fig,axes=plt.subplots()

axes.bar(df.index,df['Number of units sold'],color='c')    # bar chart

axes2=axes.twinx() #create a twin Axes sharing the xaxis.

axes2.plot(df.index,df['cumulative percent'],color='k',marker='s',ms=10)      #  line chart

# markings in %
axes2.yaxis.set_major_formatter(PercentFormatter())

axes.tick_params(axis='y',color='r')
```

```
axes2.tick_params(axis='y',color='b')

plt.show()
```




```
#Horizontal bar plot

states=['Assam','Bihar','Chandigarh','Delhi','Goa']
cases=[248028,248017,19184,618747,50239]

plt.figure(figsize=(10,5))

plt.barh(states,cases)  #horizontal bar plot

plt.xlabel('Number of Covid Cases')
plt.ylabel('City Names')
plt.title('State wise-covid cases')
plt.xticks(cases)
plt.show()
```



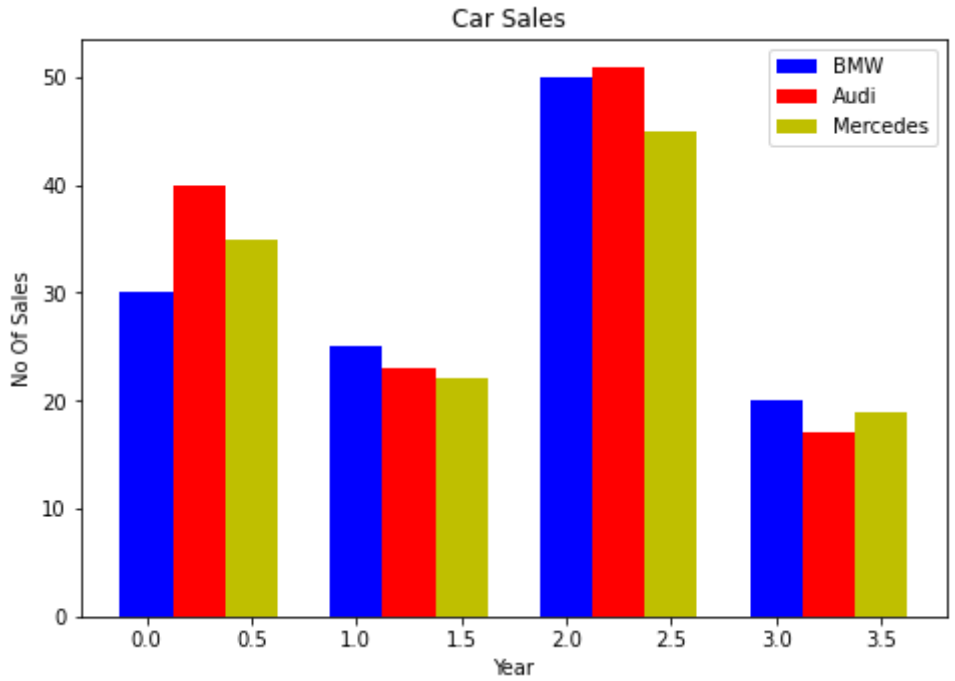

```
df_car=pd.DataFrame([[30,25,50,20],[40,23,51,17],[35,22,45,19]],index=['BMW','Audi','Mercedes'],columns=['2001','2005','2010','2020']
df_car
```

	2001	2005	2010	2020
BMW	30	25	50	20
Audi	40	23	51	17
Mercedes	35	22	45	19

```
#Grouped bar chart
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])    #axes object

x=np.arange(4)
print(x)
ax.bar(x+0.00,np.array(df_car.iloc[0:1,:]).ravel(),color='b', width=0.25)

ax.bar(x+0.25,np.array(df_car.iloc[1:2,:]).ravel(),color='r', width=0.25)
```

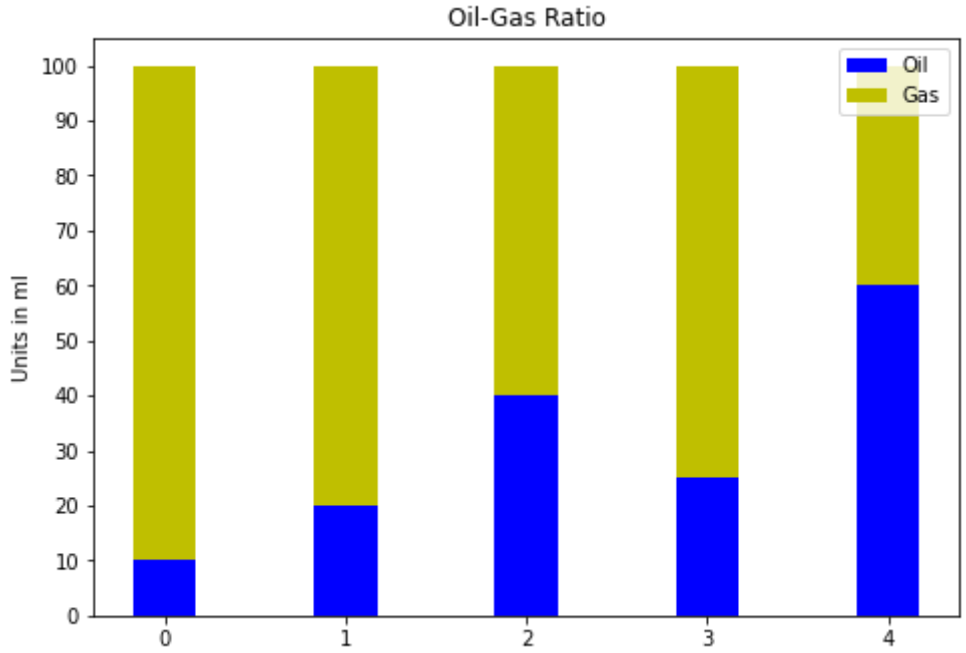
```
ax.bar(x+0.50,np.array(df_car.iloc[2:,:]).ravel(),color='y', width=0.25)

ax.set_title('Car Sales ')
ax.set_xlabel('Year')
ax.set_ylabel('No Of Sales')
ax.legend(labels=['BMW','Audi','Mercedes'])
plt.show()
```

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



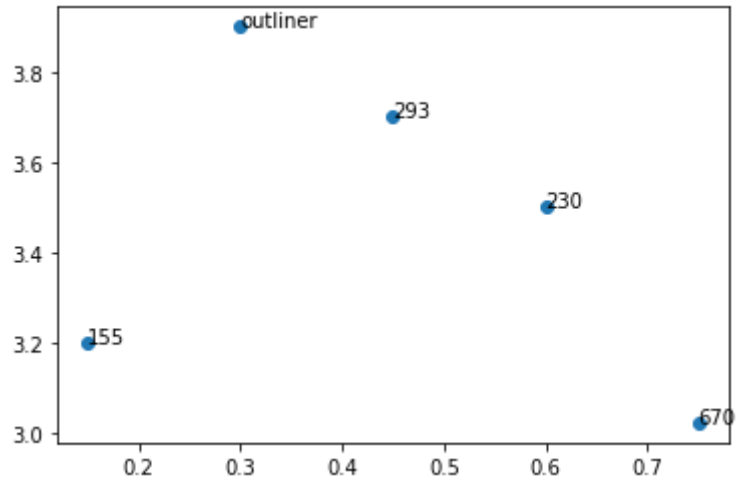

```
#Stacked bar chart
x=np.arange(5)

oilration=(10,20,40,25,60)
gasration=(90,80,60,75,40)

fig=plt.figure()

axes=fig.add_axes([0,0,1,1])    # o-o interface
```

```
axes.set_title('Oil-Gas Ratio')
axes.set_ylabel('Units in ml')
axes.set_yticks(np.arange(0,101,10))

axes.bar(x,oilration,width=0.35,color='b')
axes.bar(x,gasration,width=0.35,color='y',bottom=oilration)

axes.legend(labels=['Oil','Gas'])
plt.show()
```




```
import matplotlib.pyplot as plt
y = [3.2, 3.9, 3.7, 3.5, 3.02199]
x = [0.15, 0.3, 0.45, 0.6, 0.75]
n = [155, "outliner", 293, 230, 670]

fig, ax = plt.subplots()
ax.scatter(x, y)
```

```
for i, txt in enumerate(n):
    ax.annotate(txt, (x[i], y[i]))
plt.show()
```



PIE CHART

Civil Engineering 15,000 graduates

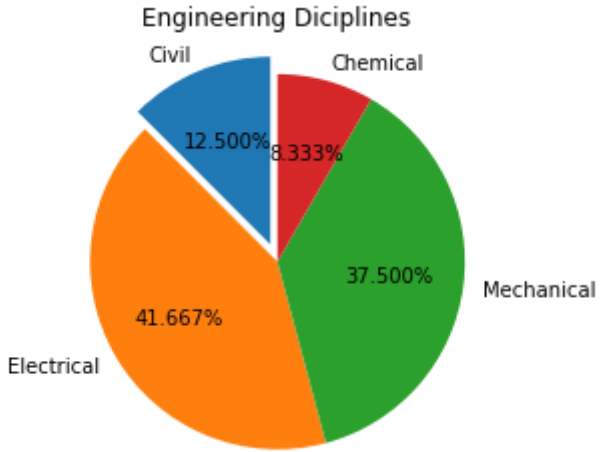
Electrical Engineering 50,000 graduates

Mechanical Engineering 45,000 graduates

Chemical Engineering 10,000 graduates

```
labels1 = ['Civil', 'Electrical', 'Mechanical', 'Chemical']
sizes = [15, 50, 45, 10]


fig, ax = plt.subplots()
explode1 = (0.1, 0, 0, 0)
```

```
ax.pie(sizes, labels=labels1, autopct='%1.3f%%',explode=explode1,startangle=90)
ax.axis('equal')  # Equal aspect ratio ensures the pie chart is circular.
ax.set_title('Engineering Diciplines')


plt.show()
```

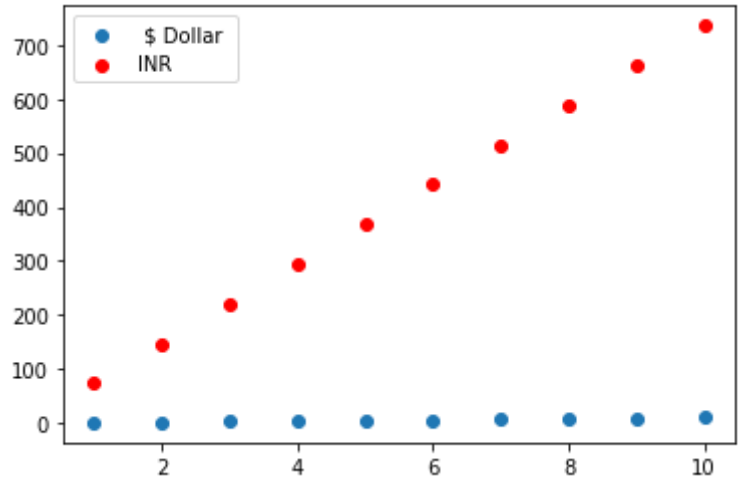



```
#scatter plot
df_curr=pd.DataFrame(np.arange(1,11),columns=['Price in $'])
df_curr['Price in INR']=df_curr['Price in $'] * 73.55
df_curr
```

	Price in $	Price in INR
0	1	73.55
1	2	147.10
2	3	220.65
3	4	294.20

```
plt.scatter(np.arange(1,11),df_curr['Price in $'],label=' $ Dollar ')
plt.scatter(np.arange(1,11),df_curr['Price in INR'],color='r',label='INR')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f0b6b446a10>
```




```
a = np.array([22, 87, 5, 43, 56,
              73, 55, 54, 11,
              20, 51, 5, 79, 31,
              27])

# Creating histogram
fig, ax = plt.subplots()
ax.hist(a, bins = [0, 25, 50, 75, 100])
```

```
# Show plot
plt.show()
```