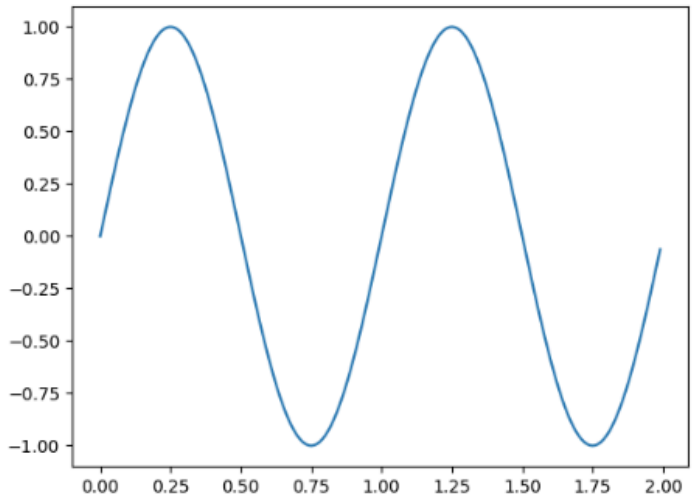
1.

```
import numpy as np
a=np.array([[1,2,3],[4,5,6],[7,8,9]])
a
```

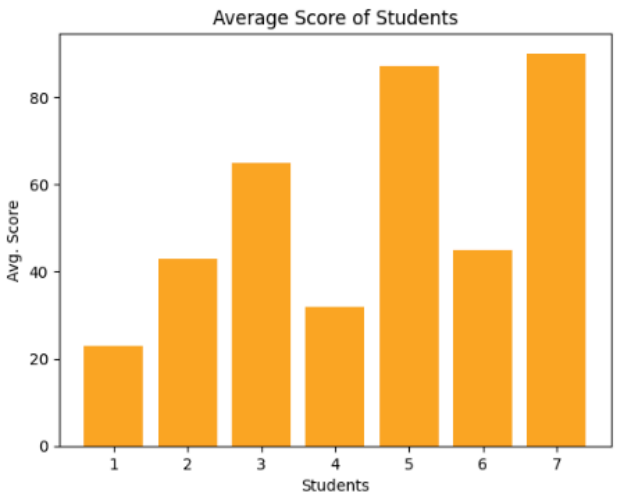
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

```
# Sinusoidal plot
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(0.0,2.0,0.01)
y=np.sin(2*np.pi*x)
plt.plot(x,y)
```

```
[<matplotlib.lines.Line2D at 0x7fe41ca37f90>]
```

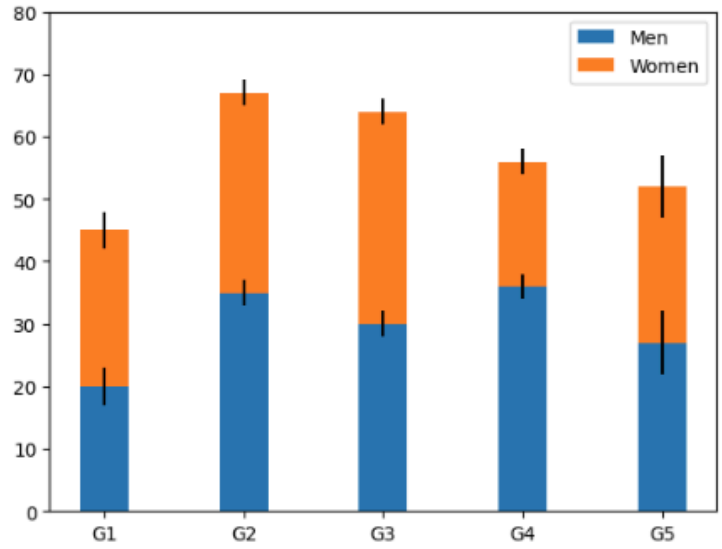



```
[3] #Barplot
    import numpy as np
    import matplotlib.pyplot as plt
    fig=plt.figure()
    x=[1,2,3,4,5,6,7]
    y=[23,43,65,32,87,45,90]
    plt.bar(x,y,color='orange')
    plt.title('Average Score of Students')
    plt.xlabel("Students")
    plt.ylabel("Avg. Score")
    fig.savefig("testing.jpg")
```

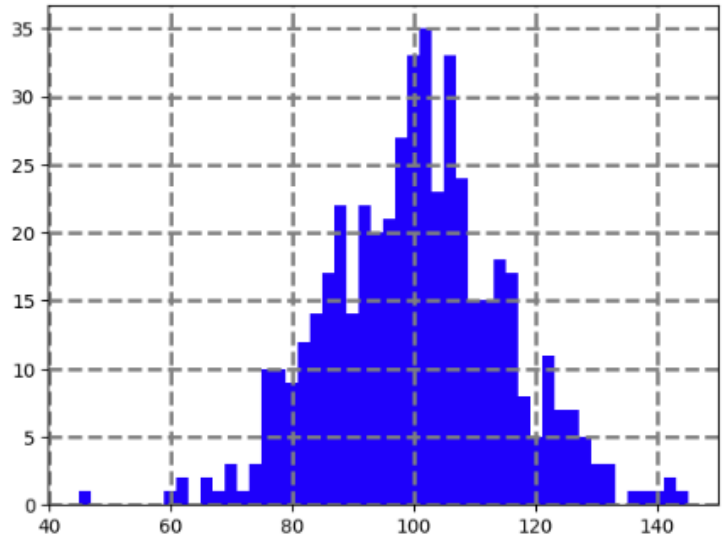


```
#single barplots with error bars
N=5
men=(20,35,30,36,27)
women=(25,32,34,20,25)
mstd=(3,2,2,2,5)
i=np.arange(N)
p1=plt.bar(i,men,width=0.35,yerr=mstd)
p2=plt.bar(i,women,width=0.35,bottom=men,yerr=mstd)
plt.xticks(i,('G1','G2','G3','G4','G5'))
plt.yticks(np.arange(0,90,10))
plt.legend((p1[0],p2[0]),('Men','Women'))
```

```
<matplotlib.legend.Legend at 0x7fe41ca34dd0>
```

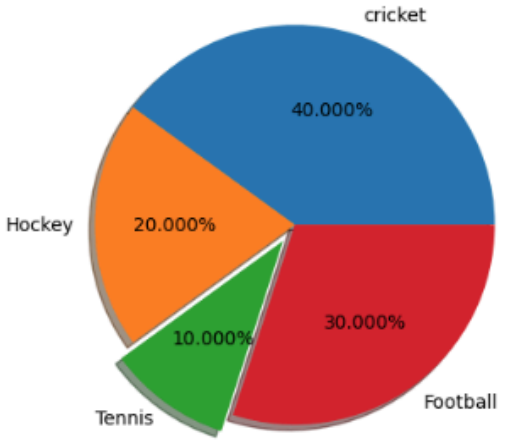



```
#Histogram
m=100
std=15
x=m+std*np.random.randn(480)
y=50
plt.hist(x,y,color='blue',density=False)
plt.grid(color='gray',linestyle='--',linewidth=2,axis='x')
plt.grid(color='gray',linestyle='--',linewidth=2,axis='y')
```



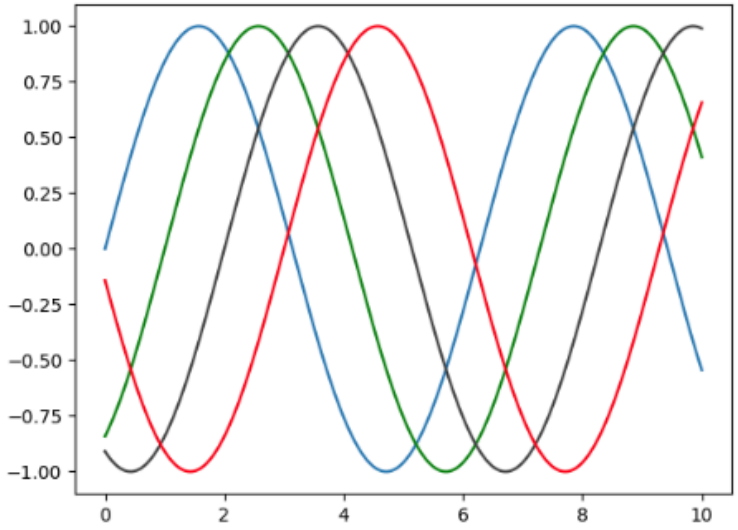
```
#Pie Chart
labels='cricket','Hockey','Tennis','Football'
v=[40,20,10,30]
e=(0,0,0.1,0)
plt.pie(v,explode=e,labels=labels,autopct='%1.3f%%',shadow=True,startangle=0)
```

```
([<matplotlib.patches.Wedge at 0x7fe41cc65c10>,
  <matplotlib.patches.Wedge at 0x7fe41cbf5dd0>,
  <matplotlib.patches.Wedge at 0x7fe41cbf7ed0>,
  <matplotlib.patches.Wedge at 0x7fe41cbe2450>],
 [Text(0.33991867422268784, 1.0461621742897658, 'cricket'),
  Text(-1.0999999999999988, -5.149471622296949e-08, 'Hockey'),
  Text(-0.7053422436693694, -0.9708204361752281, 'Tennis'),
  Text(0.6465638691739385, -0.8899186272232008, 'Football')],
 [Text(0.1854101859396479, 0.5706339132489631, '40.000%'),
  Text(-0.5999999999999993, -2.808802703071063e-08, '20.000%'),
  Text(-0.4114496421404655, -0.5663119211022163, '10.000%'),
  Text(0.35267120136760277, -0.48541016030356404, '30.000%')])
```



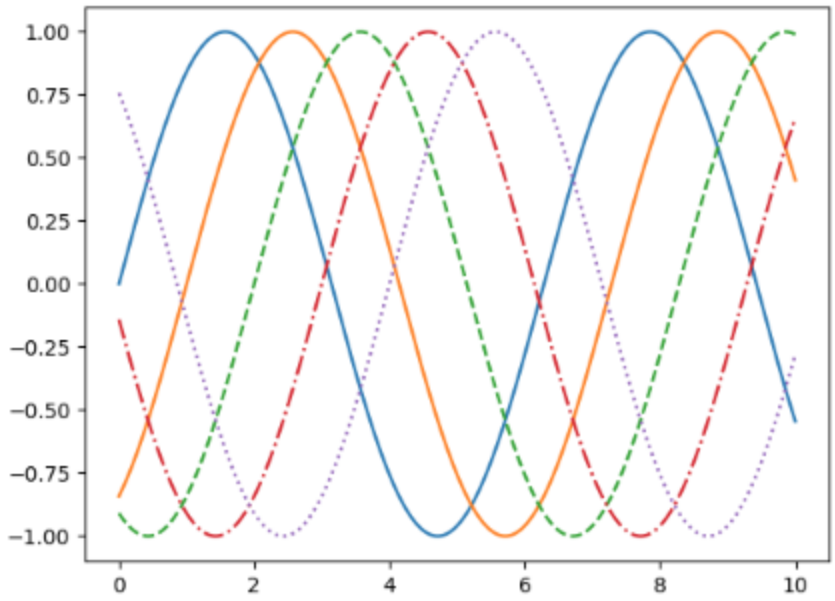

```
[7] x=np.linspace(0,10,1000) #generates linearly spaced vector with 1000 sample
    plt.plot(x,np.sin(x))
    plt.plot(x,np.sin(x-1),color='g')
    plt.plot(x,np.sin(x-2),color='0.25')
    plt.plot(x,np.sin(x-3),color='red')
```

```
[<matplotlib.lines.Line2D at 0x7fe41cbaf9d0>]
```



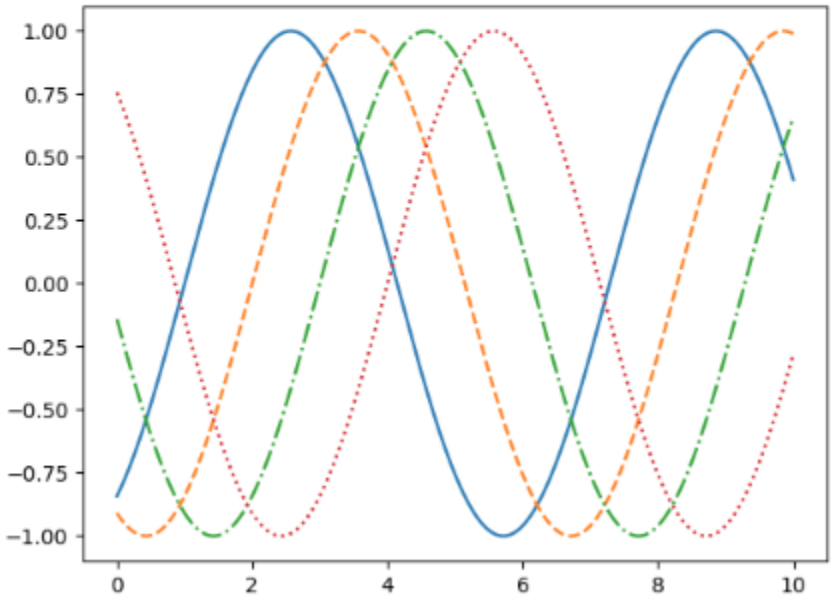
```
x=np.linspace(0,10,1000)
plt.plot(x,np.sin(x))
plt.plot(x,np.sin(x-1),linestyle='solid')
plt.plot(x,np.sin(x-2),linestyle='dashed')
plt.plot(x,np.sin(x-3),linestyle='dashdot')
plt.plot(x,np.sin(x-4),linestyle='dotted')
```

```
[<matplotlib.lines.Line2D at 0x7fe41c940290>]
```



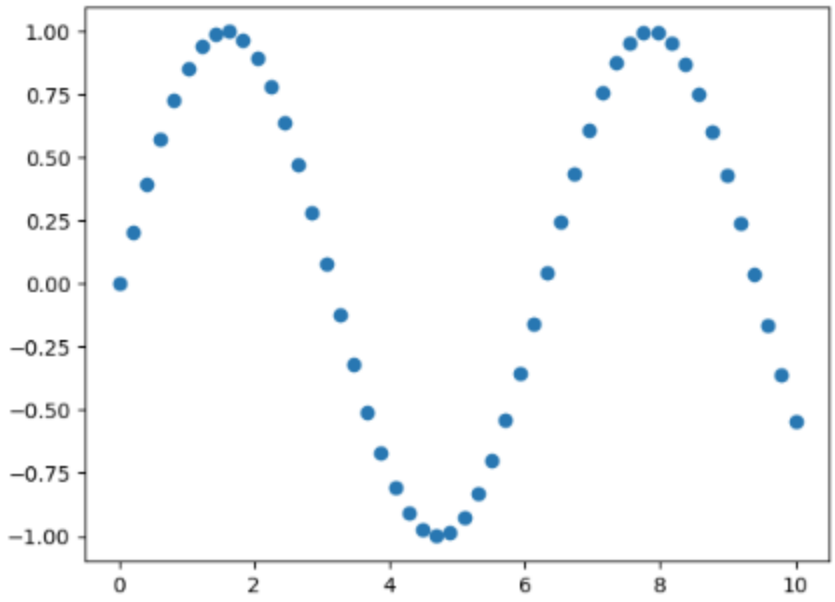

```
#plt.plot(x,np.sin(x))
plt.plot(x,np.sin(x-1),linestyle='-')
plt.plot(x,np.sin(x-2),linestyle='--')
plt.plot(x,np.sin(x-3),linestyle='-.')
plt.plot(x,np.sin(x-4),linestyle=':')
```

```
[<matplotlib.lines.Line2D at 0x7fe41c7b2090>]
```



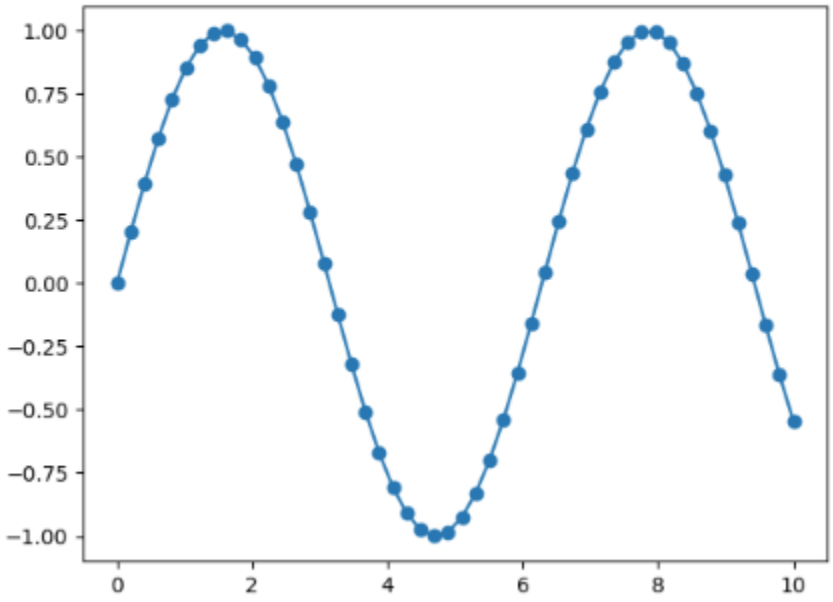
```
#Scatterplot
x=np.linspace(0,10,50)
y=np.sin(x)
plt.scatter(x,y)
```

```
<matplotlib.collections.PathCollection at 0x7fe41c7dff10>
```




```
[11] x=np.linspace(0,10,50)
plt.plot(x,np.sin(x),'-o')
```

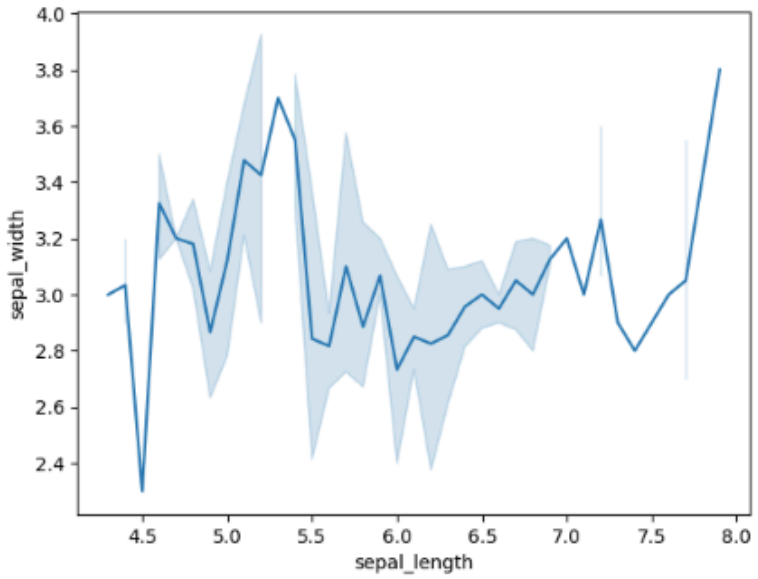
```
[<matplotlib.lines.Line2D at 0x7fe41c43d8d0>]
```



2.Iris dataset

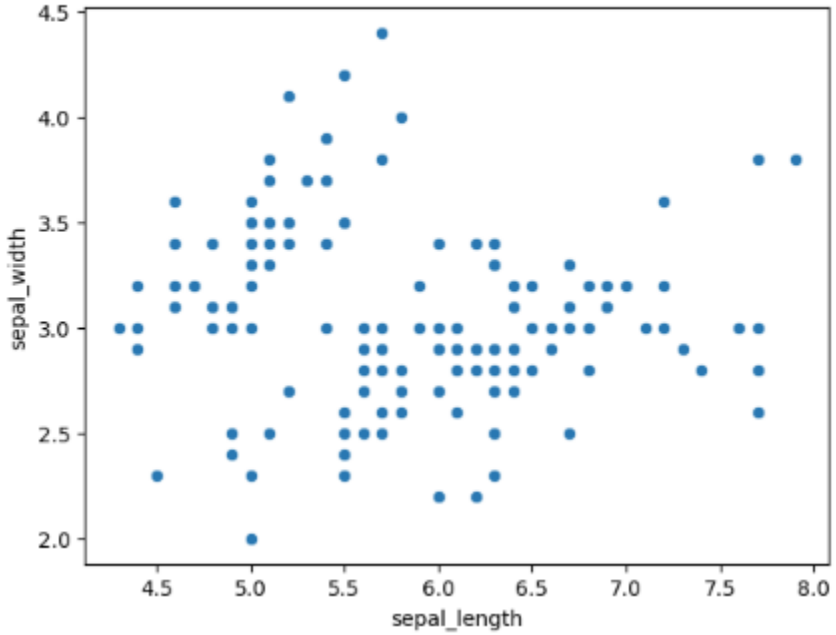
```
[8] import pandas as pd
    import seaborn as sns

    # Load the CSV file using pandas
    data = pd.read_csv("https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv")

    # Now you can use seaborn for visualization
    sns.lineplot(x="sepal_length", y="sepal_width", data=data)
```

```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```



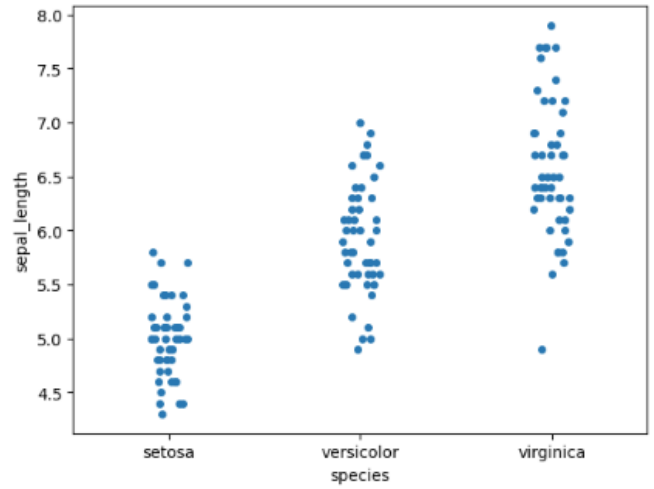

```
sns.scatterplot(x="sepal_length", y="sepal_width",data=data)
```

```
<Axes: xlabel='sepal_length', ylabel='sepal_width'>
```



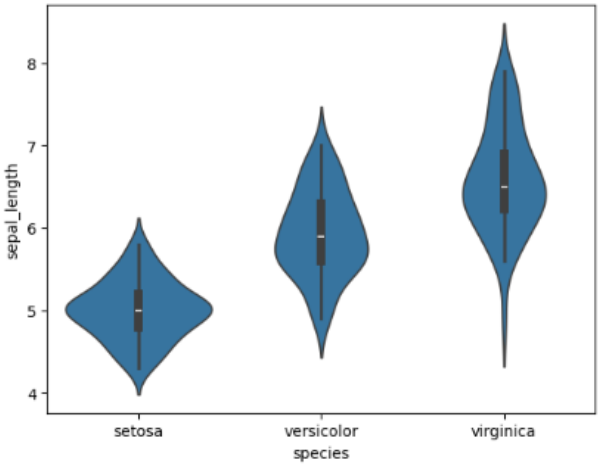
```
sns.stripplot(x="species", y="sepal_length",data=data)
```

```
<Axes: xlabel='species', ylabel='sepal_length'>
```



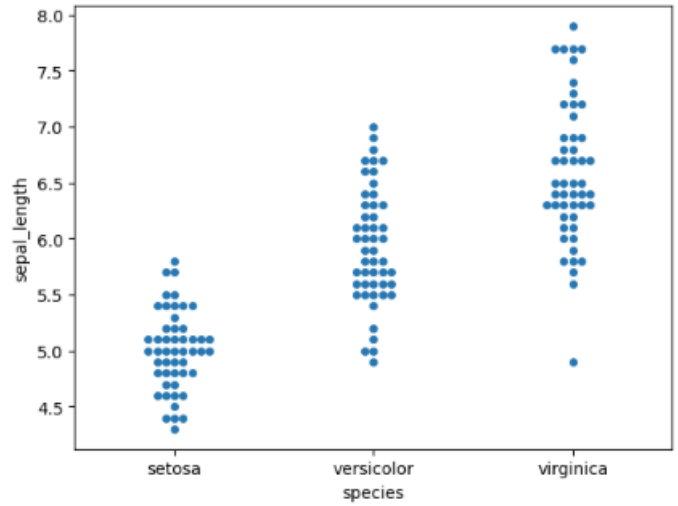

```
[11] sns.violinplot(x="species", y="sepal_length",data=data)
```

```
<Axes: xlabel='species', ylabel='sepal_length'>
```




```
sns.swarmplot(x="species", y="sepal_length",data=data)
```

```
<Axes: xlabel='species', ylabel='sepal_length'>
```



3. Automobile dataset

```
from google.colab import files


uploaded = files.upload()
```

```
Choose Files Automobile_data.csv
```

- **Automobile_data.csv**(text/csv) - 25065 bytes, last modified: 1/20/2025 - 100% done

```
Saving Automobile_data.csv to Automobile_data.csv
```

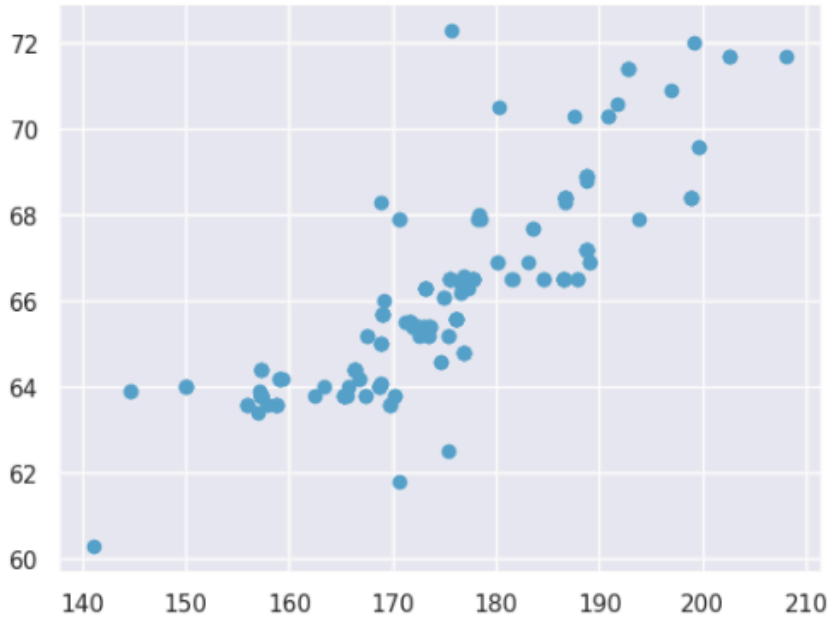
```
import pandas as pd
import io

df = pd.read_csv(io.BytesIO(uploaded['Automobile_data.csv']), encoding='utf-8')
print(df)
```

```
     symboling normalized-losses         make    fuel aspiration num-of-doors  \
0            3                 ?  alfa-romero     gas        std          two
1            3                 ?  alfa-romero     gas        std          two
2            1                 ?  alfa-romero     gas        std          two
3            2               164         audi     gas        std         four
4            2               164         audi     gas        std         four
..         ...               ...          ...     ...        ...          ...
200         -1                95        volvo     gas        std         four
201         -1                95        volvo     gas      turbo         four
202         -1                95        volvo     gas        std         four
203         -1                95        volvo  diesel      turbo         four
204         -1                95        volvo     gas      turbo         four

      body-style drive-wheels engine-location  wheel-base  ...  engine-size  \
0    convertible          rwd           front        88.6  ...          130
1    convertible          rwd           front        88.6  ...          130
2      hatchback          rwd           front        94.5  ...          152
3          sedan          fwd           front        99.8  ...          109
4          sedan          4wd           front        99.4  ...          136
..           ...          ...             ...         ...  ...          ...
200        sedan          rwd           front       109.1  ...          141
201        sedan          rwd           front       109.1  ...          141
202        sedan          rwd           front       109.1  ...          173
203        sedan          rwd           front       109.1  ...          145
204        sedan          rwd           front       109.1  ...          141

     fuel-system  bore  stroke compression-ratio horsepower  peak-rpm  \
0           mpfi  3.47    2.68               9.0        111      5000
1           mpfi  3.47    2.68               9.0        111      5000
2           mpfi  2.68    3.47               9.0        154      5000
3           mpfi  3.19     3.4              10.0        102      5500
4           mpfi  3.19     3.4               8.0        115      5500
..           ...   ...     ...               ...        ...       ...
200         mpfi  3.78    3.15               9.5        114      5400
201         mpfi  3.78    3.15               8.7        160      5300
202         mpfi  3.58    2.87               8.8        134      5500
203          idi  3.01     3.4              23.0        106      4800
204         mpfi  3.78    3.15               9.5        114      5400


     fuel-system  bore  stroke compression-ratio horsepower  peak-rpm  \
0           mpfi  3.47    2.68               9.0        111      5000
1           mpfi  3.47    2.68               9.0        111      5000
2           mpfi  2.68    3.47               9.0        154      5000
3           mpfi  3.19     3.4              10.0        102      5500
4           mpfi  3.19     3.4               8.0        115      5500
..           ...   ...     ...               ...        ...       ...
200         mpfi  3.78    3.15               9.5        114      5400
201         mpfi  3.78    3.15               8.7        160      5300
202         mpfi  3.58    2.87               8.8        134      5500
203          idi  3.01     3.4              23.0        106      4800
204         mpfi  3.78    3.15               9.5        114      5400

    city_mpg highway_mpg  price
0         21          27  13495
1         21          27  16500
2         19          26  16500
3         24          30  13950
4         18          22  17450
..       ...         ...    ...
200       23          28  16845
201       19          25  19045
202       18          23  21485
203       26          27  22470
204       19          25  22625

[205 rows x 26 columns]
```

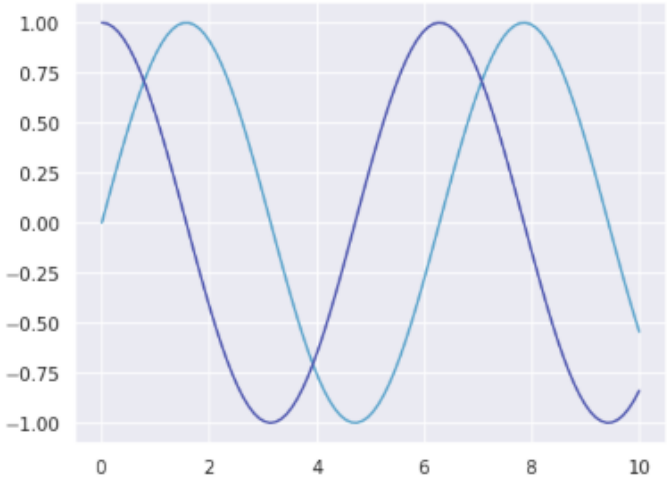
Creating scatter plot Using Matplotlib

```
[58] import numpy as np
     import matplotlib.pyplot as plt

     x = np.array([df.length])
     y = np.array([df.width])
     plt.scatter(x, y)
```

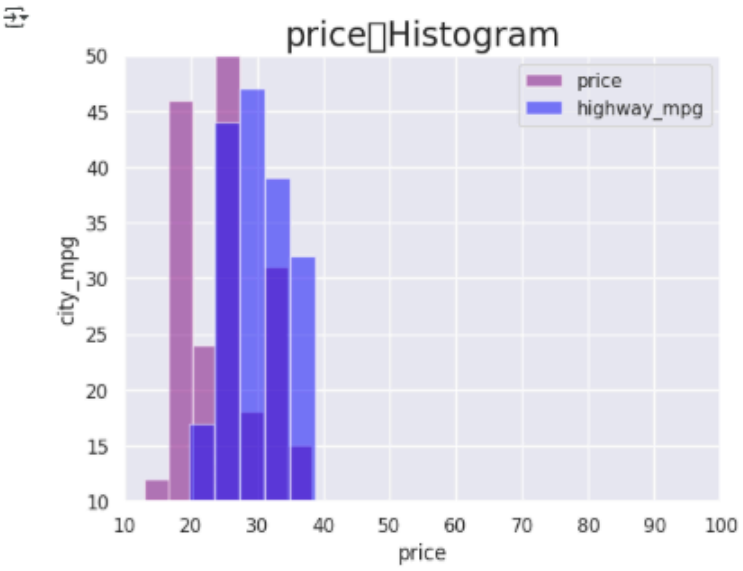
```
<matplotlib.collections.PathCollection at 0x789e875eecd0>
```




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

x = np.linspace(0, 10, 100)

plt.plot(x, np.sin(x))
plt.plot(x, np.cos(x))

plt.show()
```

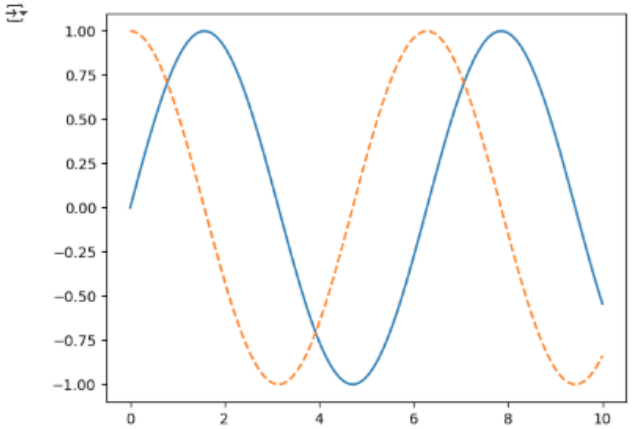


```
[59] #set  figure
     f, ax = plt.subplots(1,1)

     #graph  histogram
     plt.hist(df.city_mpg, bins=10, alpha=0.5, color='purple', label='price')
     plt.hist(df.highway_mpg, bins=10, alpha = 0.5, color='blue', label='highway_

     #set  legend
     plt.legend(loc='upper right')

     #set  title & axis  titles
     ax.set_title('price Histogram', fontsize=20)
     ax.set_xlabel('price')
     ax.set_ylabel('city_mpg')

     #set  x & y ranges
     plt.xlim(10,100)
     plt.ylim(10, 50)


     plt.show()
```



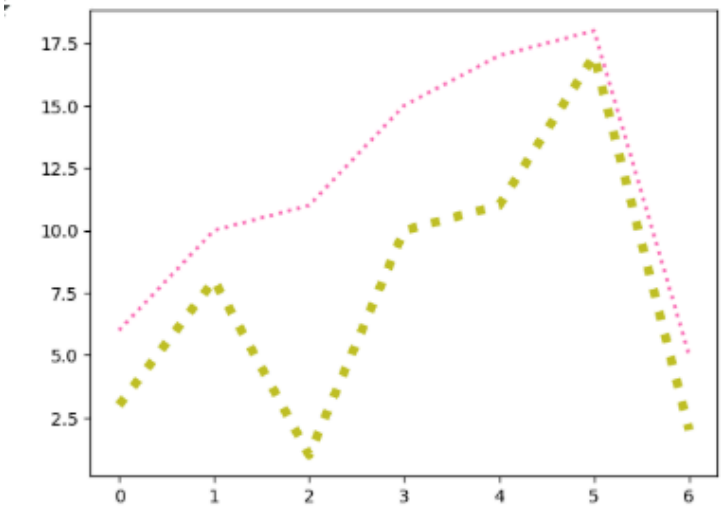

```
[15]  %matplotlib inline
```

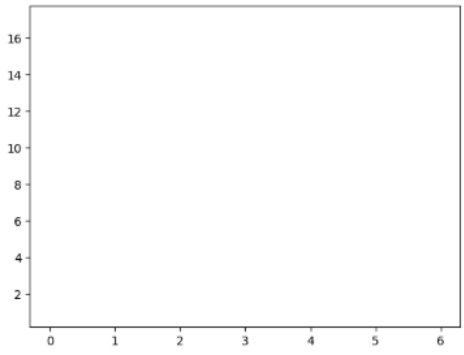
```
[16] import numpy as np
     x = np.linspace(0, 10, 100)
     fig = plt.figure()
     plt.plot(x, np.sin(x), '-')
     plt.plot(x, np.cos(x), '--');
```

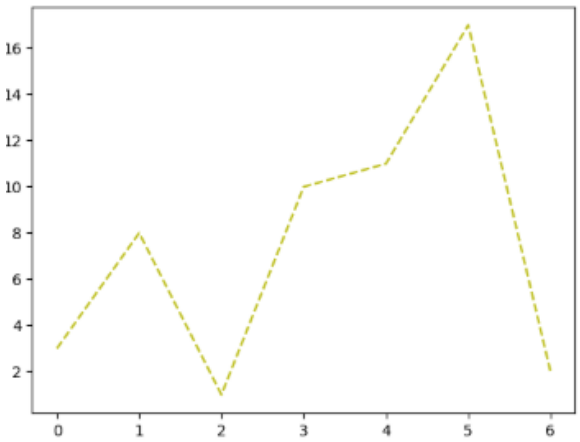


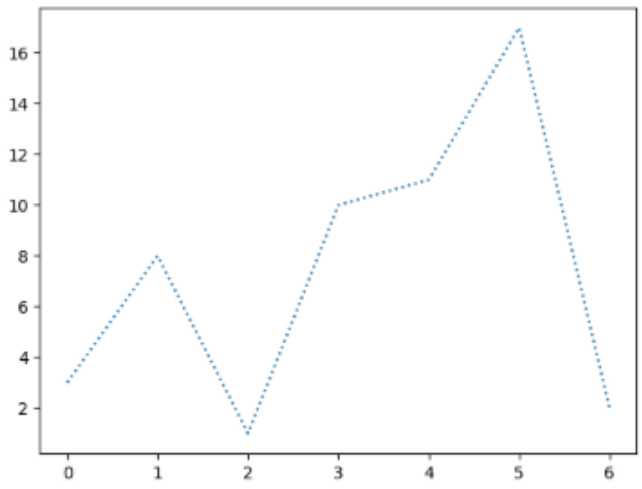
Plotting line

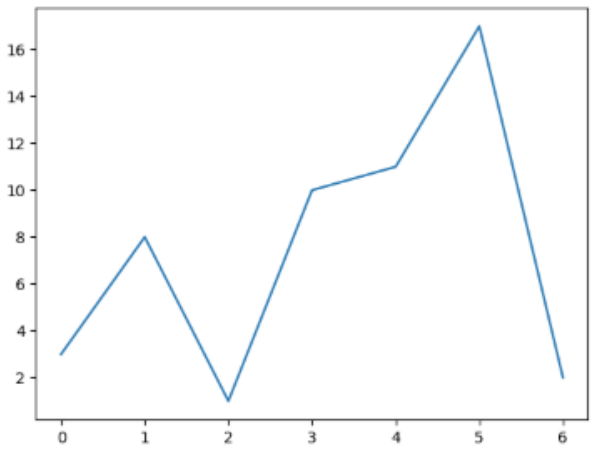
```
import matplotlib.pyplot as plt
import numpy as np

y1points = np.array([3, 8, 1,10,11,17,2])
y2points = np.array([6, 10, 11,15,17,18,5])
y1= plt.plot(y1points, linestyle = 'dotted',color ='y',linewidth=5.0)
y2= plt.plot(y2points, linestyle = 'dotted',color ='hotpink',linewidth=2.0)
plt.show(y1)
plt.show(y2)
plt.plot(y1points, linestyle = 'dashed',color='y')
plt.show()
plt.plot(y1points, ls = ':')
plt.show()
plt.plot(y1points,linestyle='solid')
plt.show()
plt.plot(y1points,linestyle='dashdot')
plt.show()
plt.plot(y1points,linestyle='None')
plt.show()
```

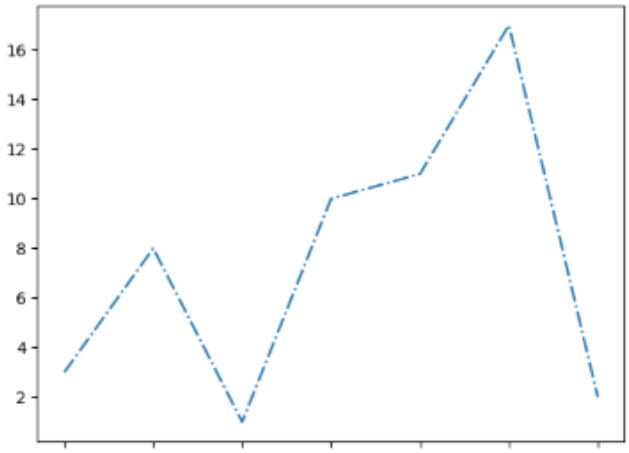













```
import matplotlib.pyplot as plt
import numpy as np

x1 = np.array([0, 1, 2, 3])
y1 = np.array([3, 8, 1, 10])
x2 = np.array([0, 1, 2, 3])
y2 = np.array([6, 2, 7, 11])

plt.plot(x1, y1, x2, y2)
plt.show()
```

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

x=np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y=np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])
plt.plot(x,y)
plt.title("Sports")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")
plt.show()
```