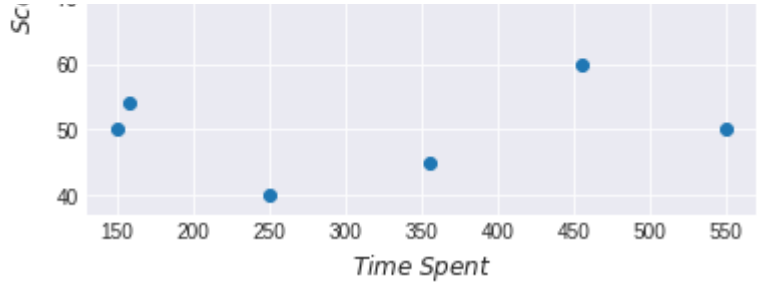
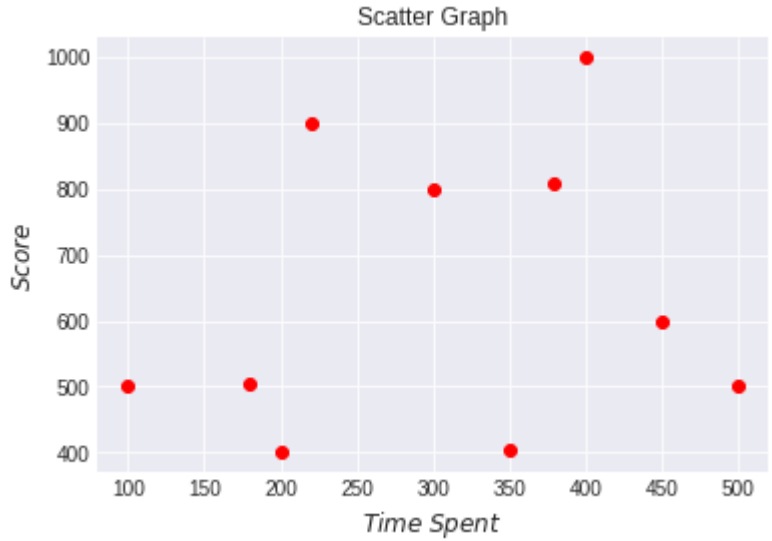
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

#Graph Styling
# https://tonysyu.github.io/raw_content/matplotlib-style-gallery/gallery.html
plt.style.use('seaborn-darkgrid')
```

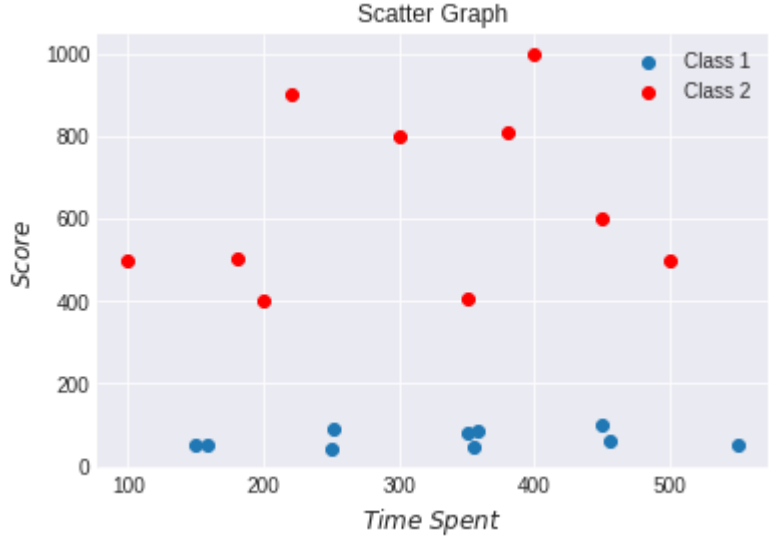
Scatter Graphs

```
x1 = np.array([250,150,350,252,450,550,455,358,158,355])
y1 =np.array([40,50,80, 90, 100,50,60,88,54,45])


x2 = np.array([200,100,300,220,400,500,450,380,180,350])
y2 = np.array([400,500,800, 900, 1000,500,600,808,504,405])

#Graph - 1
plt.scatter(x1,y1)
plt.xlabel('$Time $ $ Spent$' , fontsize = 12)
plt.ylabel('$Score$' , fontsize = 12)
plt.title ('Scatter Graph')
plt.show()

#Graph - 2
```

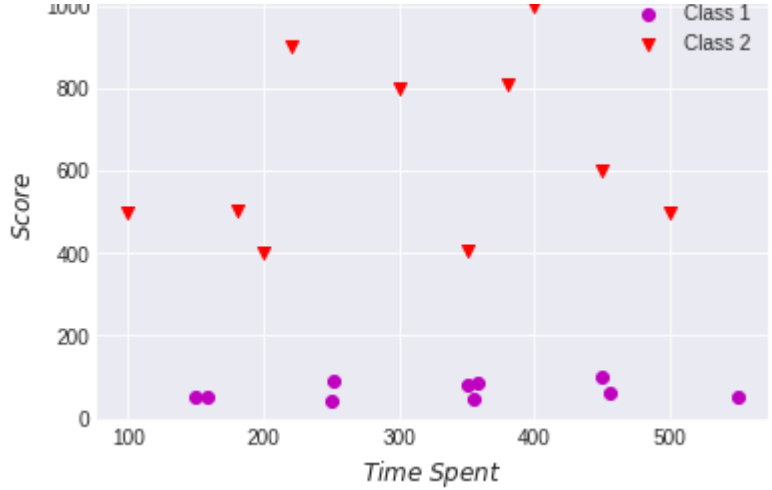
```
#Graph - 2
plt.scatter(x2,y2 ,color = 'r')
plt.xlabel('$Time $ $ Spent$' , fontsize = 12)
plt.ylabel('$Score$' , fontsize = 12)
plt.title ('Scatter Graph')
plt.show()

#Graph - 3
plt.scatter(x1,y1 ,label = 'Class 1')
plt.scatter(x2,y2 ,label = 'Class 2',color ='r')
plt.xlabel('$Time $ $ Spent$' , fontsize = 12)
plt.ylabel('$Score$' , fontsize = 12)
plt.title ('Scatter Graph')
plt.legend()
plt.show()

#Graph - 4
plt.scatter(x1,y1 ,label = 'Class 1',marker='o' , color = 'm')
plt.scatter(x2,y2 ,label = 'Class 2',marker='v',color ='r')
plt.xlabel('$Time $ $ Spent$' , fontsize = 12)
plt.ylabel('$Score$' , fontsize = 12)
plt.title ('Scatter Graph')
plt.legend()
plt.show()
```


60
50
40
150
200
250
300
350
400
450
500
550
Time Spent

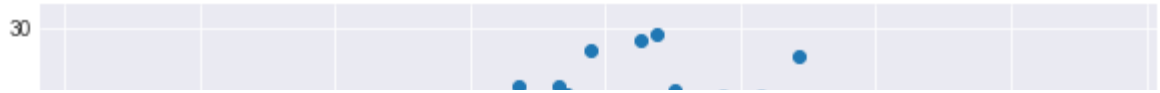

Scatter Graph
1000
900
800
700
600
500
400
Score
100
150
200
250
300
350
400
450
500
Time Spent


Scatter Graph
Class 1
Class 2
1000
800
600
400
200
0
Score
100
200
300
400
500
Time Spent

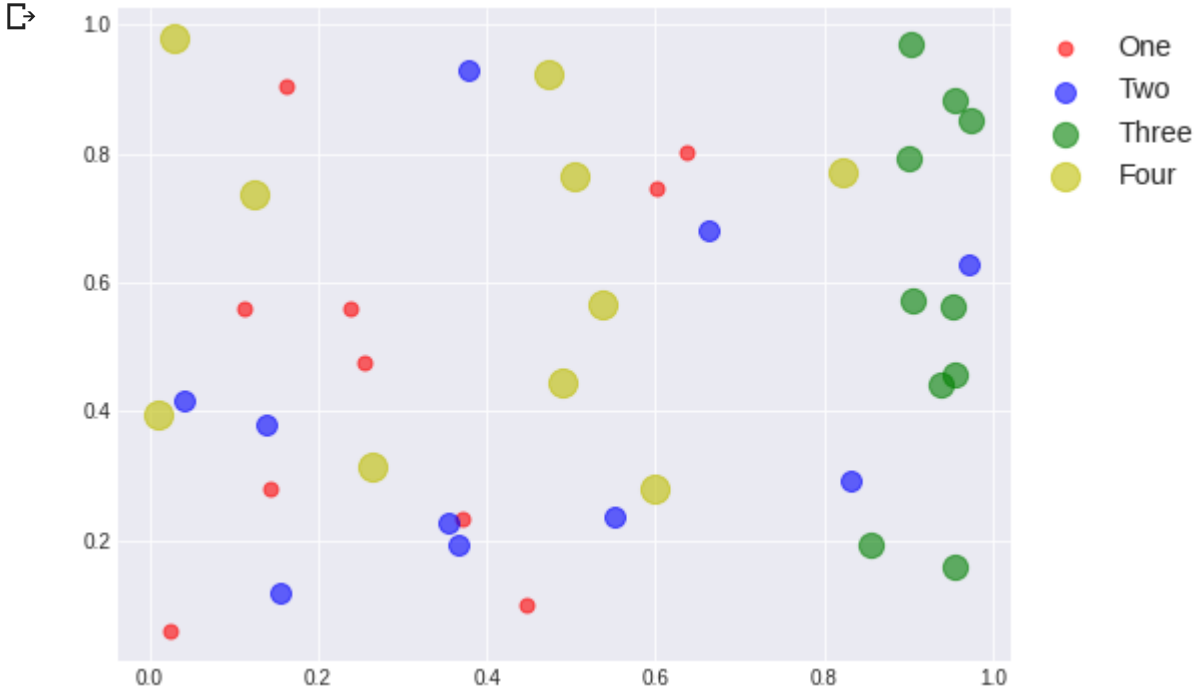

Scatter Graph




```
plt.figure(figsize=(10,6))
x = np.random.normal(0,10,1000)
y = np.random.normal(0,10,1000)
plt.scatter(x,y)
plt.show()
```




```
plt.figure(figsize=(8,6))
x = np.random.random(10)
y = np.random.random(10)
# "alpha" is used for softnening colors
plt.scatter(np.random.random(10),np.random.random(10),c='r', s=50 , alpha=0.6 , label =
plt.scatter(np.random.random(10),np.random.random(10),c='b', s=100 , alpha=0.6 , label
plt.scatter(np.random.random(10),np.random.random(10),c='g', s=150 , alpha=0.6 , label
plt.scatter(np.random.random(10),np.random.random(10),c='y', s=200 , alpha=0.6 , label
plt.legend(bbox_to_anchor=(1.0, 1.0) , shadow=True, fontsize='x-large')
plt.show()
```




```
# Changing label color
```

```
plt.figure(figsize=(8,6))
x = np.random.random(10)
y = np.random.random(10)
# "alpha" is used for softnening colors
plt.rcParams['text.color'] = 'red' # Label Color
plt.scatter(np.random.random(10),np.random.random(10),c='r', s=50 , alpha=0.6 , label =
plt.scatter(np.random.random(10),np.random.random(10),c='b', s=100 , alpha=0.6 , label
plt.scatter(np.random.random(10),np.random.random(10),c='g', s=150 , alpha=0.6 , label
plt.scatter(np.random.random(10),np.random.random(10),c='y', s=200 , alpha=0.6 , label
plt.legend(bbox_to_anchor=(1.0, 1.0) , shadow=True, fontsize='x-large')
plt.show()
```