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

In [2]:
```
d=pd.read_csv("Iris.csv")
d
```

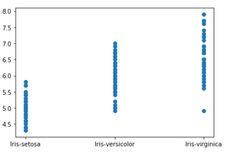
Out[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

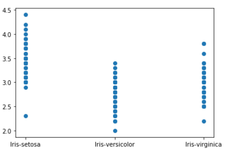
In [3]:
```
plt.scatter(d['Species'],d['SepalLengthCm'])
```

Out[3]:
```
<matplotlib.collections.PathCollection at 0x7f718180ac90>
```



In [4]:
```
plt.scatter(d['Species'],d['SepalWidthCm'])
```

Out[4]:
```
<matplotlib.collections.PathCollection at 0x7f7180560410>
```

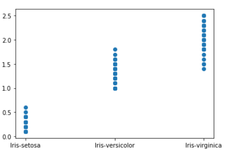


In [5]:
```
plt.scatter(d['Species'],d['PetalLengthCm'])
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x7f71804d58d0>
```

In [6]:
```
plt.scatter(d['Species'],d['PetalWidthCm'])
```

Out[6]:
```
<matplotlib.collections.PathCollection at 0x7f7180485d50>
```



In [7]:
```
#divide the dataset into clusters
k=KMeans(n_clusters=3)
k
```

Out[7]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=3, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [8]:
```
#shows which datas are under which clusters
y_predicted = k.fit_predict(d[['SepalWidthCm','SepalLengthCm']])
y_predicted
```

Out[8]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 2, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1,
       2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 1, 1, 2, 1, 1, 2, 1,
       1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 2, 1, 2, 1, 1, 2, 2, 1, 1, 1, 1,
       1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 2], dtype=int32)
```

In [9]:
```
#add cluster column
d['cluster']=y_predicted
d.head(70)
```

Out[9]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species	cluster
0	1	5.1	3.5	1.4	0.2	Iris-setosa	0
1	2	4.9	3.0	1.4	0.2	Iris-setosa	0
2	3	4.7	3.2	1.3	0.2	Iris-setosa	0
3	4	4.6	3.1	1.5	0.2	Iris-setosa	0
4	5	5.0	3.6	1.4	0.2	Iris-setosa	0
...	...	...	...	...	...	...	...
65	66	6.7	3.1	4.4	1.4	Iris-versicolor	1
66	67	5.6	3.0	4.5	1.5	Iris-versicolor	2
67	68	5.8	2.7	4.1	1.0	Iris-versicolor	2
68	69	6.2	2.2	4.5	1.5	Iris-versicolor	2
69	70	5.6	2.5	3.9	1.1	Iris-versicolor	2

70 rows × 7 columns

In [10]:
```
#find centroids location
k.cluster_centers_
```

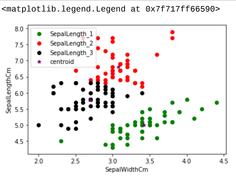
Out[10]:
```
array([[3.418    , 5.006    ],
       [3.07446809, 6.81276596],
       [2.69245283, 5.77358491]])
```

In [11]:
```
#scatter plot with clusters and centroids
d1 = d[d.cluster==0]
d2 = d[d.cluster==1]
d3 = d[d.cluster==2]

plt.scatter(d1.SepalWidthCm,d1['SepalLengthCm'],color='green',label='SepalLength_1')
plt.scatter(d2.SepalWidthCm,d2['SepalLengthCm'],color='red',label='SepalLength_2')
plt.scatter(d3.SepalWidthCm,d3['SepalLengthCm'],color='black',label='SepalLength_3')
plt.scatter(k.cluster_centers_[:,0],k.cluster_centers_[:,1],color='purple',marker='*',label=
'centroid')

plt.xlabel('SepalWidthCm')
plt.ylabel('SepalLengthCm')
plt.legend()
```

Out[11]:
```
<matplotlib.legend.Legend at 0x7f717ff66590>
```



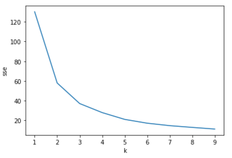
In [12]:
```
#calculate sse(Sum of Squared Error)
k_rng = range(1,10)
sse = []
for k in k_rng:
    k=KMeans(n_clusters=k)
    k.fit(d[['SepalWidthCm','SepalLengthCm']])
    sse.append(k.inertia_)
```

In [13]:
```
#showing sse values
sse
```

Out[13]:
```
[130.18093333333334,
 57.982406042078765,
 37.1237021276957,
 27.962171178088646,
 21.077101654961503,
 17.215255498972574,
 14.729199343185552,
 12.957193438914029,
 11.284309905780498]
```

In [14]:
```
#plotting elbow method to determine no. of clusters
plt.xlabel('k')
plt.ylabel('sse')
plt.plot(k_rng,sse)
```

Out[14]:
```
[<matplotlib.lines.Line2D at 0x7f717feee990>]
```



Thank You

Submitted by Goutami Dey

The Sparks Foundation

In [ ]:
```
```