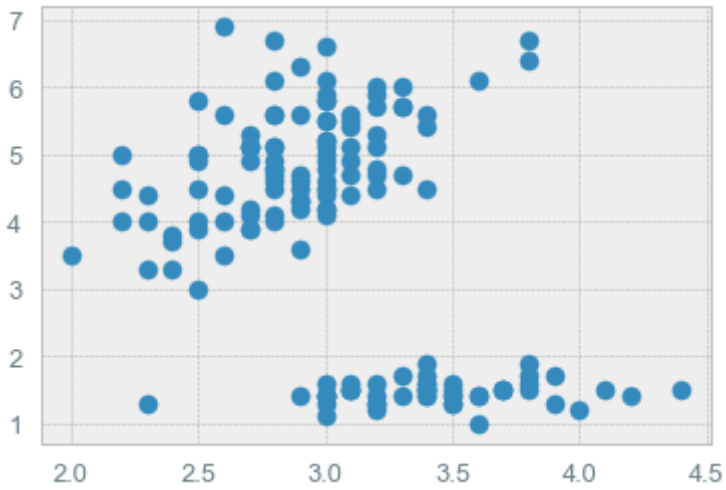
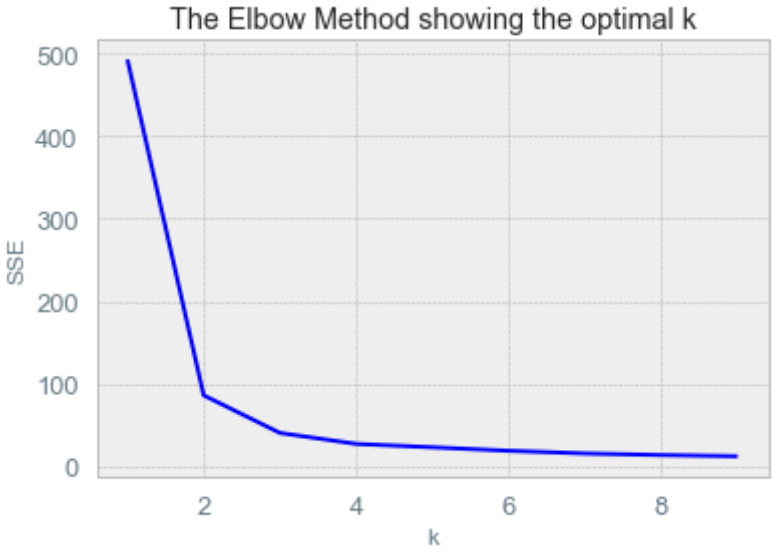
In [ ]:
```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('bmh')
import numpy as np
from sklearn.cluster import KMeans
```

In [ ]:
```
from sklearn.datasets import load_iris
iris = load_iris()
features=iris.data.T
plt.plot()
plt.scatter(features[1], features[2]) #2 fitur yang akan dipakai
plt.show()
```

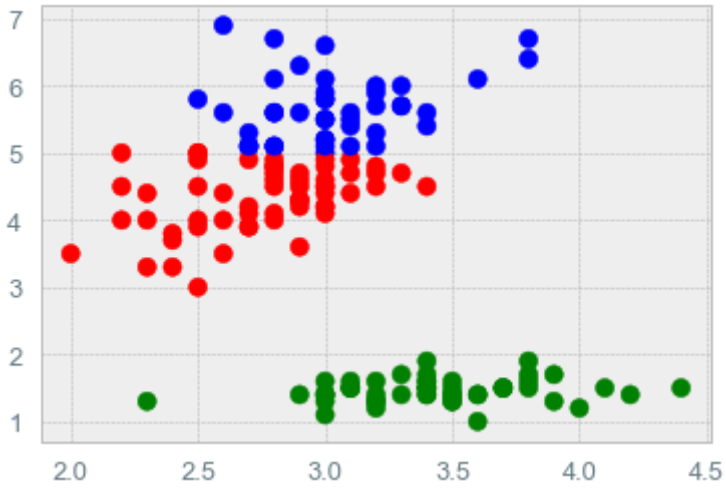


In [ ]:
```
sse = []
X = np.array(list(zip(features[1], features[2]))).reshape(len(features[1]), 2)
for k in range(1, 10):
    kmeans = KMeans(n_clusters=k).fit(X)
    sse.append(kmeans.inertia_)
plt.plot(range(1, 10), sse, 'bx-')
plt.xlabel('k')
plt.ylabel('SSE')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



In [ ]:
```
y_pred = KMeans(n_clusters=3, random_state=9).fit_predict(X)
plt.plot
```

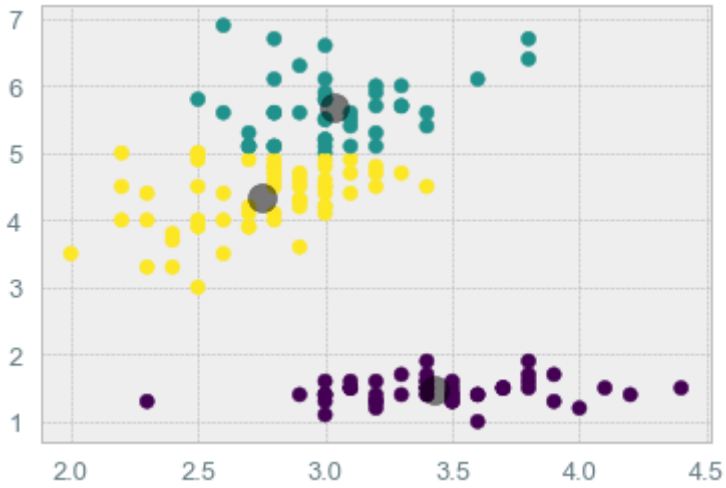
```
LABEL_COLOR_MAP = {0 : 'r', 1 : 'g', 2 : 'b'}
label_color = [LABEL_COLOR_MAP[l] for l in y_pred]
plt.scatter(features[1], features[2], c=label_color)
plt.show()
```



In [ ]:
```
# kmeans clustering
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
labels = kmeans.predict(X)
centroids = kmeans.cluster_centers_
print(centroids)

plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1], c='black', s=200, alpha=0.5);
plt.show()
```

```
[[3.428      1.462     ]
 [3.03255814 5.67209302]
 [2.75087719 4.32807018]]
```



In [ ]:
```
# evaluasi hasil clustering
for k in range(1,10):
    kmeans = KMeans(n_clusters=k, random_state=0).fit(X)
    labels = kmeans.predict(X)
    inertia = kmeans.inertia_
    print("k = ", k, " inertia = ", inertia)
```

```
k =  1  inertia =  492.6323333333333
k =  2  inertia =  86.3106476530006
k =  3  inertia =  40.737074092207266
k =  4  inertia =  27.484695238095235
k =  5  inertia =  23.365215330940988
k =  6  inertia =  19.261382975635946
k =  7  inertia =  15.867991162860129
k =  8  inertia =  14.199557292382007
k =  9  inertia =  12.41982302011714
```

In [ ]:

```python
from sklearn.metrics.cluster import silhouette_score
jumlah_cluster=range(2,10)
for k in jumlah_cluster:
    kmeans = KMeans(n_clusters=k, random_state=0).fit(X)
    labels = kmeans.predict(X)
    inertia = kmeans.inertia_
    score = silhouette_score(X, labels, metric='euclidean')
    print("k = ", k, " score = ", score)
```

```
k =  2  score =  0.7392862954615389
k =  3  score =  0.5933477102522363
k =  4  score =  0.5607669717890267
k =  5  score =  0.5106975398446402
k =  6  score =  0.3832437418032831
k =  7  score =  0.37021239543200013
k =  8  score =  0.37990058878262895
k =  9  score =  0.40148680784335655
```