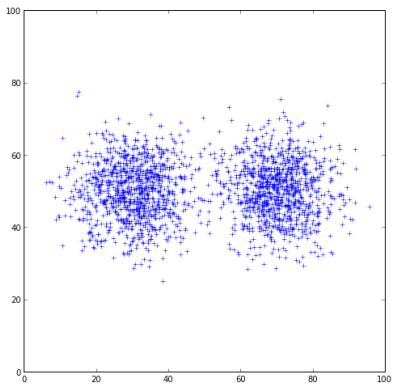



```
1 import random as pyrandom
```

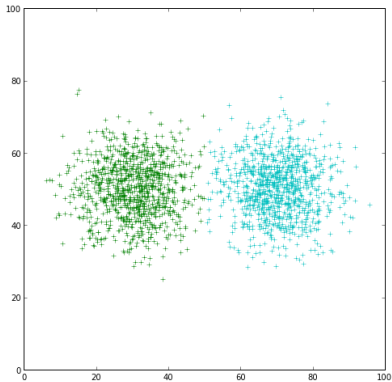
```
1 ccolors = ['c+', 'g+', 'b+', 'y+', 'r+', 'm+', 'b+']
2 def plotclusters(data, centers=None):
3     xlim([0,100]); ylim([0,100])
4     if centers is None:
5         plot(data[:,0], data[:,1], 'b+')
6     else:
7         for i in range(amax(centers)+1):
8             plot(data[centers==i,0], data[centers==i,1], ccolors[i%
                len(ccolors)])
```

K-Means Clustering

```
1 data = r_[8*randn(1000,2)+array([70,50]),  
2         8*randn(1000,2)+array([30,50])]  
3 shuffle(data)  
4 plotclusters(data)
```

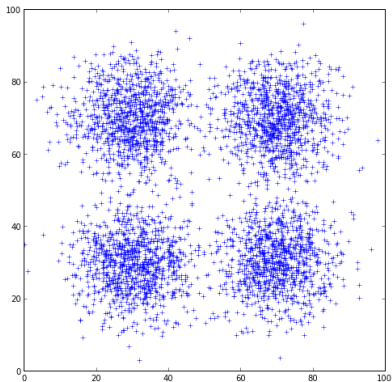


```
1 from sklearn.cluster import KMeans
2 km = KMeans(2)
3 centers = km.fit_predict(data)
4 plotclusters(data, centers)
```

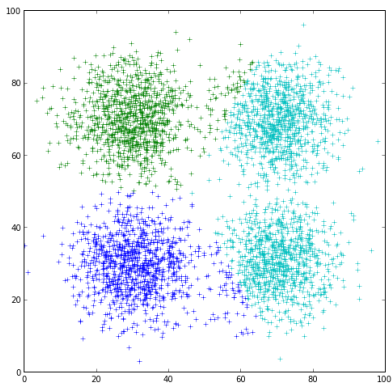


Number of Clusters

```
1 data = r_[8*randn(1000,2)+array([70,70]),
2         8*randn(1000,2)+array([30,30]),
3         8*randn(1000,2)+array([30,70]),
4         8*randn(1000,2)+array([70,30]),
5 ]
6 shuffle(data)
7 plotclusters(data)
```

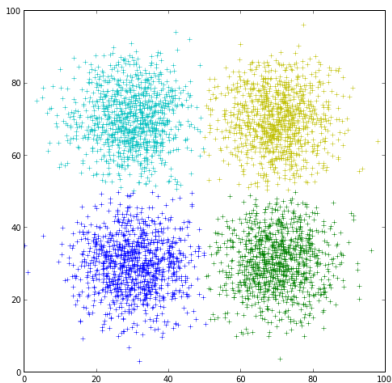



```
1 km = KMeans(3)
2 centers = km.fit_predict(data)
3 plotclusters(data, centers)
```

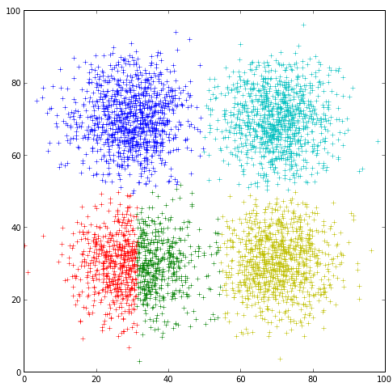


```
1 km = KMeans(3)
2 centers = km.fit_predict(data)
3 plotclusters(data,centers)
```

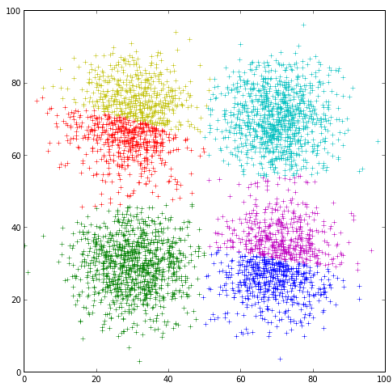
```
1 km = KMeans(4)
2 centers = km.fit_predict(data)
3 plotclusters(data, centers)
```



```
1 km = KMeans(5)
2 centers = km.fit_predict(data)
3 plotclusters(data, centers)
```



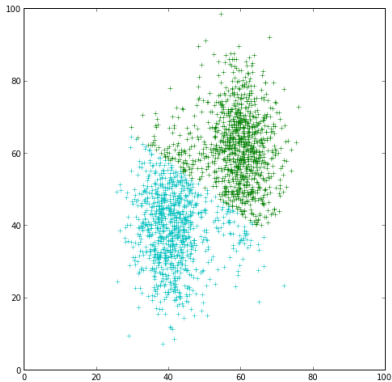
```
1 km = KMeans(6)
2 centers = km.fit_predict(data)
3 plotclusters(data, centers)
```



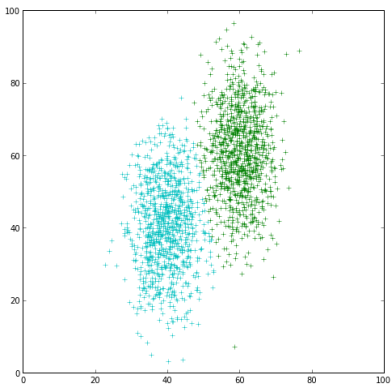
Anisotropic Clusters and Mixture Models

```
1 data = r_[dot(randn(1000,2),diag([5,12]))+array([60,60]),
2         dot(randn(1000,2),diag([5,12]))+array([40,40])]
3 shuffle(data)
```

```
1 from sklearn.cluster import KMeans
2 km = KMeans(2)
3 centers = km.fit_predict(data)
4 plotclusters(data,centers)
```




```
1 from sklearn.mixture import GMM
2 gmm = GMM(n_components=2, covariance_type='diag')
3 gmm.fit(ldata)
4 lcenters = gmm.predict(ldata)
5 plotclusters(ldata, lcenters)
```



scipy.cluster.vq

```
1 data = r_[8*randn(1000,2)+array([70,50]),  
2         8*randn(1000,2)+array([30,50])]  
3 shuffle(data)
```

```
1 from scipy.cluster import vq
2 vectors, _ = vq.kmeans(data, 2)
3 centers, _ = vq.vq(data, vectors)
4 plotclusters(data, centers)
```

