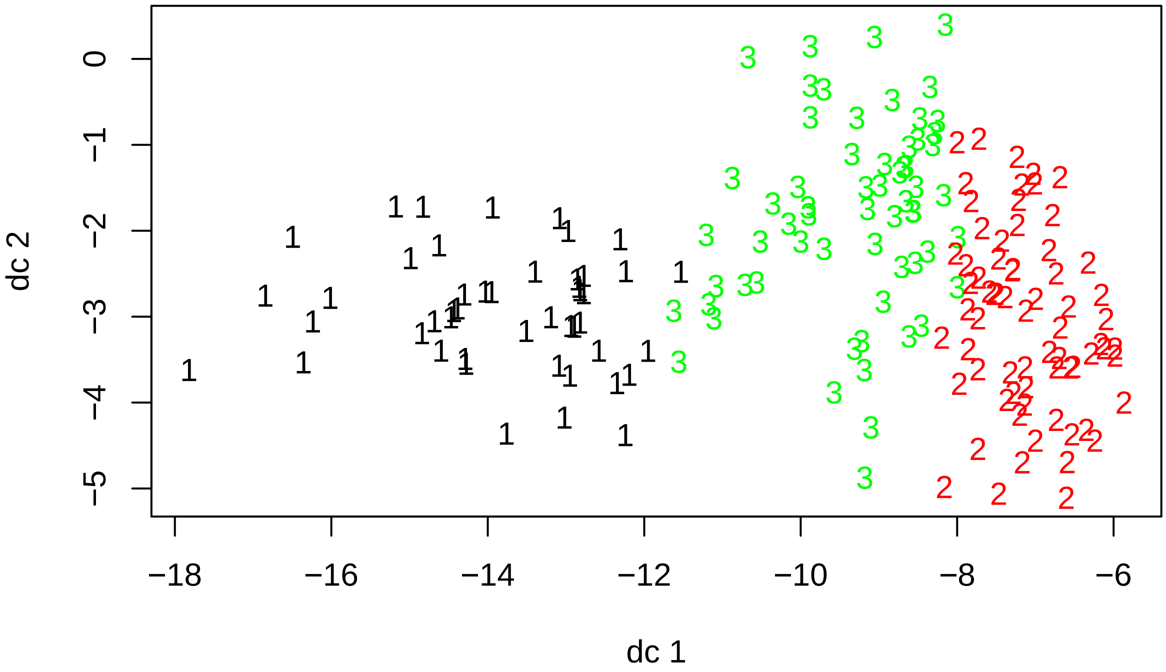
k-means

Jingting Li

Nov.01, 2015

```
data(wine, package="rattle")
data.train<-wine[-1]
fit.km<-kmeans(data.train,3)
library("fpc")
plotcluster(data.train,fit.km$cluster)
```



The clusters don't seem well-separated. There are no obvious gaps between clusters, especiallly, there is an overlapping area between cluster 1 and cluster 3. Use randIndex from flexclust to compare these two parititions – one from data set and one from result of clustering method.

```
confuseTable.km<-table(wine$Type,fit.km$cluster)
library("flexclust")
```

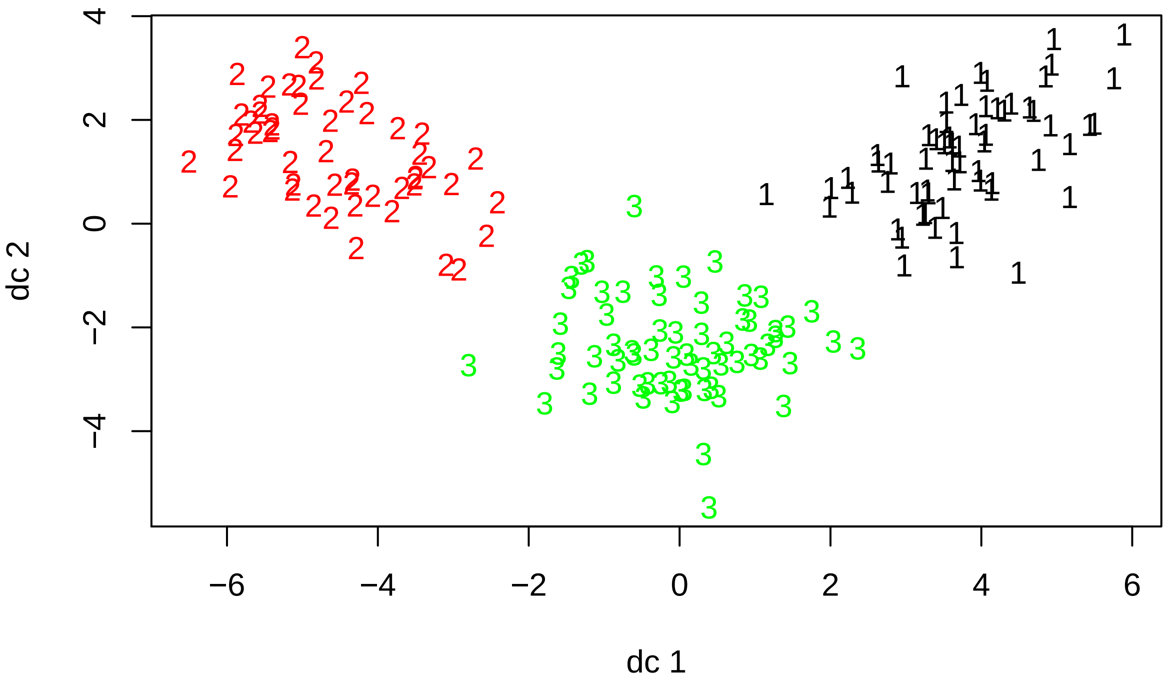
```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: modeltools
## Loading required package: stats4
```

```
randIndex(confuseTable.km)
```

```
##       ARI
## 0.3711137
```

We can observe that the result is only around 0.33, and it's far away from 1. So the algorithm doesn't work well. Use scale function for scaling and centering data.

```
data.trainnew<-scale(wine[-1])
fit.kmnew<-kmeans(data.trainnew,3)
plotcluster(data.trainnew,fit.kmnew$cluster)
```



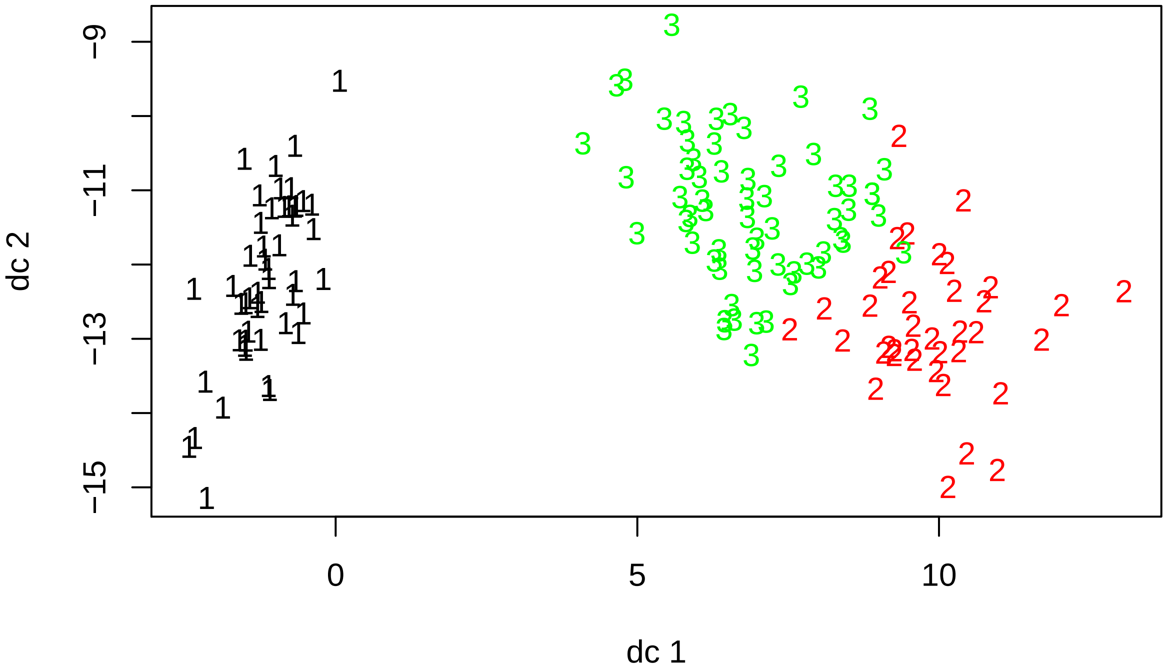
We can see the data is clustered very well, there are no collapse between clusters.

```
confuseTable.kmnew<-table(wine$Type,fit.kmnew$cluster)
randIndex(confuseTable.kmnew)
```

```
##      ARI
## 0.897495
```

The result is quite close to 1, so K-Means works well for classifying wine dataset.

```
data("iris")
data.train<-iris[-5]
fit.km<-kmeans(data.train,3)
plotcluster(data.train,fit.km$cluster)
```



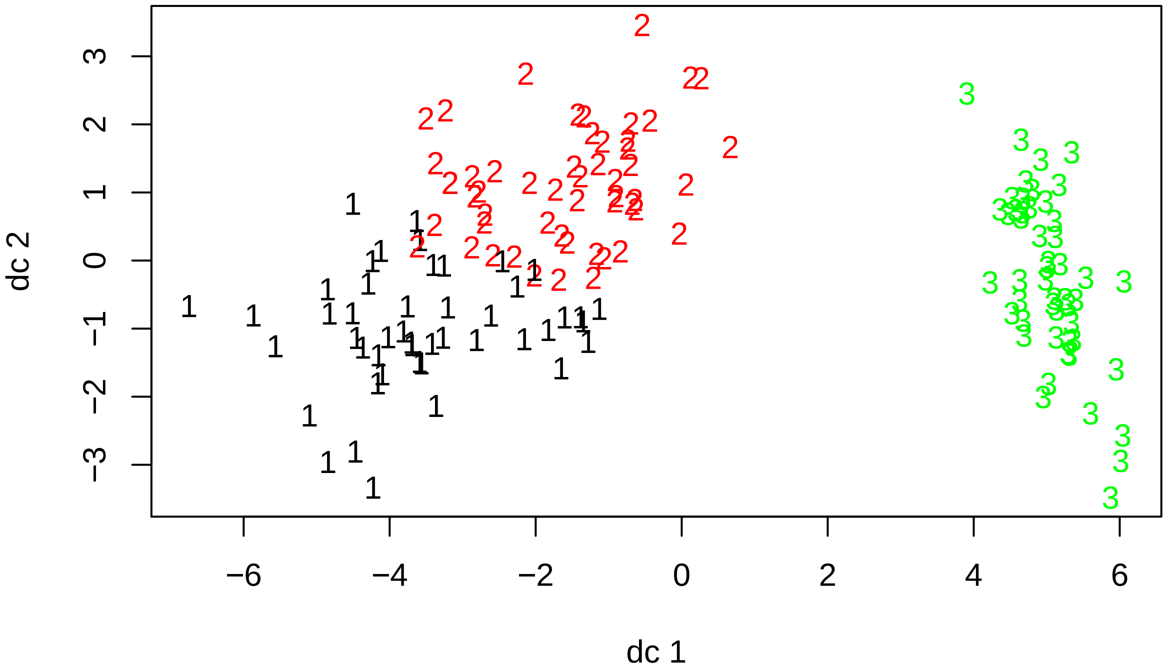
There is an overlapping area between cluster 1 and cluster 3.

```
confuseTable.km<-table(iris$Species,fit.km$cluster)
randIndex(confuseTable.km)
```

```
##       ARI
## 0.7302383
```

The result is not very close to 1. Using scaled data to cluster the iris dataset again.

```
data.trainnew<-scale(iris[-5])
fit.kmnew<-kmeans(data.trainnew,3)
plotcluster(data.trainnew,fit.kmnew$cluster)
```



There is still an overlap between cluster 1 and cluster 3.

```
confuseTable.kmnew<-table(iris$Species,fit.kmnew$cluster)
randIndex(confuseTable.kmnew)
```

```
##       ARI
## 0.6201352
```

The result is even smaller than that before scaling. So K-Means doesn't work well for classifying iris dataset. And scaling is not helpful.