Clustering - Wine

Chinpei Tang

This mini-project is based on the K-Means exercise from 'R in Action' - see http://www.r-bloggers.com/k-means-clustering-from-r-in-action/.

Exercise 0

- **Install these packages if you don't have them already**

```
library("cluster")
library("rattle")
```

```
## Rattle: A free graphical interface for data mining with R.
## Version 4.1.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library("NbClust")
library("flexclust")
```

```
## Loading required package: grid

## Loading required package: lattice

## Loading required package: modeltools

## Loading required package: stats4
```

Exercise 1

- **Remove the first column from the data and scale it using the scale() function**

Now load the data and look at the first few rows.

```
data(wine, package="rattle")
head(wine)
```

```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39
##   Nonflavanoids Proanthocyanins Color  Hue Dilution Proline
## 1          0.28            2.29  5.64 1.04     3.92    1065
## 2          0.26            1.28  4.38 1.05     3.40    1050
## 3          0.30            2.81  5.68 1.03     3.17    1185
## 4          0.24            2.18  7.80 0.86     3.45    1480
## 5          0.39            1.82  4.32 1.04     2.93     735
## 6          0.34            1.97  6.75 1.05     2.85    1450
```

```
summary(wine)
```

```
##  Type      Alcohol          Malic           Ash          Alcalinity   
##  1:59   Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60  
##  2:71   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20  
##  3:48   Median :13.05   Median :1.865   Median :2.360   Median :19.50  
##         Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49  
##         3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50  
##         Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00  
##    Magnesium         Phenols        Flavanoids    Nonflavanoids   
##  Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300  
##  1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700  
##  Median : 98.00   Median :2.355   Median :2.135   Median :0.3400  
##  Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619  
##  3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375  
##  Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600  
##  Proanthocyanins     Color             Hue             Dilution    
##  Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270  
##  1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938  
##  Median :1.555   Median : 4.690   Median :0.9650   Median :2.780  
##  Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612  
##  3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170  
##  Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000  
##     Proline      
##  Min.   : 278.0  
##  1st Qu.: 500.5  
##  Median : 673.5  
##  Mean   : 746.9  
##  3rd Qu.: 985.0  
##  Max.   :1680.0  
```

```
str(wine)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Type           : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol        : num  14.2 13.2 13.2 14.4 13.2 ...
##  $ Malic          : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
##  $ Ash            : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
##  $ Alcalinity     : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
##  $ Magnesium      : int  127 100 101 113 118 112 96 121 97 98 ...
##  $ Phenols        : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
##  $ Flavanoids     : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
##  $ Nonflavanoids  : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
##  $ Proanthocyanins: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
##  $ Color          : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
##  $ Hue            : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
##  $ Dilution       : num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
##  $ Proline        : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

There are 178 observations and 13 different chemcical measurements of each of the wines.

Remove the type of wine so that we can use clustering algorithm to cluster the types.

```
wine.noType <- wine
wine.noType$Type <- NULL
summary(wine.noType)
```

```
##     Alcohol          Malic            Ash          Alcalinity
##  Min.   :11.03   Min.   :0.740   Min.   :1.360   Min.   :10.60
##  1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210   1st Qu.:17.20
##  Median :13.05   Median :1.865   Median :2.360   Median :19.50
##  Mean   :13.00   Mean   :2.336   Mean   :2.367   Mean   :19.49
##  3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558   3rd Qu.:21.50
##  Max.   :14.83   Max.   :5.800   Max.   :3.230   Max.   :30.00
##    Magnesium         Phenols        Flavanoids    Nonflavanoids
##  Min.   : 70.00   Min.   :0.980   Min.   :0.340   Min.   :0.1300
##  1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205   1st Qu.:0.2700
##  Median : 98.00   Median :2.355   Median :2.135   Median :0.3400
##  Mean   : 99.74   Mean   :2.295   Mean   :2.029   Mean   :0.3619
##  3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875   3rd Qu.:0.4375
##  Max.   :162.00   Max.   :3.880   Max.   :5.080   Max.   :0.6600
##  Proanthocyanins     Color             Hue            Dilution
##  Min.   :0.410   Min.   : 1.280   Min.   :0.4800   Min.   :1.270
##  1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825   1st Qu.:1.938
##  Median :1.555   Median : 4.690   Median :0.9650   Median :2.780
##  Mean   :1.591   Mean   : 5.058   Mean   :0.9574   Mean   :2.612
##  3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200   3rd Qu.:3.170
##  Max.   :3.580   Max.   :13.000   Max.   :1.7100   Max.   :4.000
##     Proline
##  Min.   : 278.0
##  1st Qu.: 500.5
##  Median : 673.5
##  Mean   : 746.9
##  3rd Qu.: 985.0
##  Max.   :1680.0
```

Since the data are of different scales, use scale() function to appropriately scale the data.

```
wine.noType.scaled <- scale(wine.noType)
summary(wine.noType.scaled)
```

```
##     Alcohol             Malic              Ash
##  Min.   :-2.42739   Min.   :-1.4290   Min.   :-3.66881
##  1st Qu.:-0.78603   1st Qu.:-0.6569   1st Qu.:-0.57051
##  Median : 0.06083   Median :-0.4219   Median :-0.02375
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.83378   3rd Qu.: 0.6679   3rd Qu.: 0.69615
##  Max.   : 2.25341   Max.   : 3.1004   Max.   : 3.14745
##    Alcalinity          Magnesium          Phenols
##  Min.   :-2.663505   Min.   :-2.0824   Min.   :-2.10132
##  1st Qu.:-0.687199   1st Qu.:-0.8221   1st Qu.:-0.88298
##  Median : 0.001514   Median :-0.1219   Median : 0.09569
##  Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.600395   3rd Qu.: 0.5082   3rd Qu.: 0.80672
##  Max.   : 3.145637   Max.   : 4.3591   Max.   : 2.53237
##    Flavanoids      Nonflavanoids     Proanthocyanins        Color
##  Min.   :-1.6912   Min.   :-1.8630   Min.   :-2.06321   Min.   :-1.6297
##  1st Qu.:-0.8252   1st Qu.:-0.7381   1st Qu.:-0.59560   1st Qu.:-0.7929
##  Median : 0.1059   Median :-0.1756   Median :-0.06272   Median :-0.1588
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.8467   3rd Qu.: 0.6078   3rd Qu.: 0.62741   3rd Qu.: 0.4926
##  Max.   : 3.0542   Max.   : 2.3956   Max.   : 3.47527   Max.   : 3.4258
##       Hue               Dilution          Proline
##  Min.   :-2.08884   Min.   :-1.8897   Min.   :-1.4890
##  1st Qu.:-0.76540   1st Qu.:-0.9496   1st Qu.:-0.7824
##  Median : 0.03303   Median : 0.2371   Median :-0.2331
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.71116   3rd Qu.: 0.7864   3rd Qu.: 0.7561
##  Max.   : 3.29241   Max.   : 1.9554   Max.   : 2.9631
```

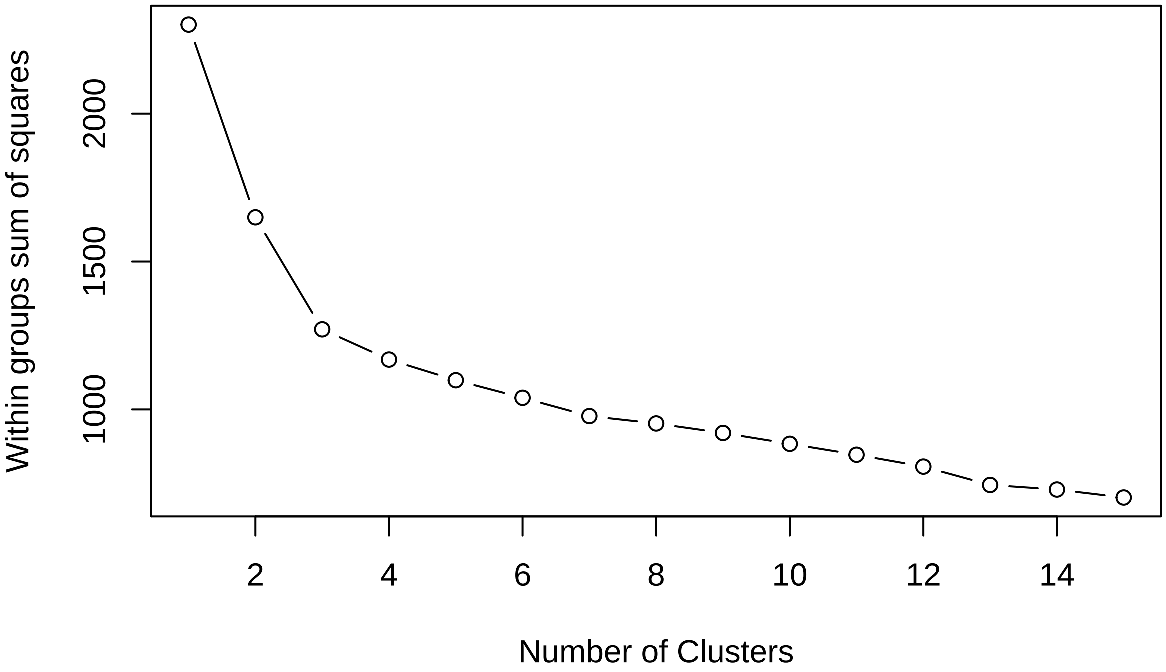
Now we'd like to cluster the data using k-means method. k-means method requires the specification of the number of clusters, so we need to first decide how many clusters to use.

Method 1

A plot of the total within-groups sums of squares against the number of clusters in a K-means solution can be helpful. A bend in the graph can suggest the appropriate number of clusters.

```
wssplot <- function(data, nc=15, seed=1234){
                    wss <- (nrow(data)-1)*sum(apply(data,2,var))
                    for (i in 2:nc){
                          set.seed(seed)
                        wss[i] <- sum(kmeans(data, centers=i)$withinss)}

                        plot(1:nc, wss, type="b", xlab="Number of Clusters",
                      ylab="Within groups sum of squares")
              }
wssplot(wine.noType.scaled)
```



Looking at the plot, since the sum of squares are significant between 1 and 2, and 2 and 3, then doesn't change much after 3, k = 3 is a good number of clusters.

Exercise 2

- **How many clusters does this method suggest?**

This method suggests k = 3 clusters.

- **Why does this method work? What's the intuition behind it?**

It looks into the sum of squares within the cluster, which is roughly how spreaded out a cluster. We want a reasonably sized cluster, so we want to reduce the sum of squares of within clusters. We can see significant improvement from 1 cluster to 2 clusters, then more improvement from 2 clusters to 3 clusters. However, the improvement from 3 clusters to 4 clusters started to decrease. This means that adding more clusters actually not distinguish too much of some of the clusters. Furthermore, if may be "overfitting" some of the features.

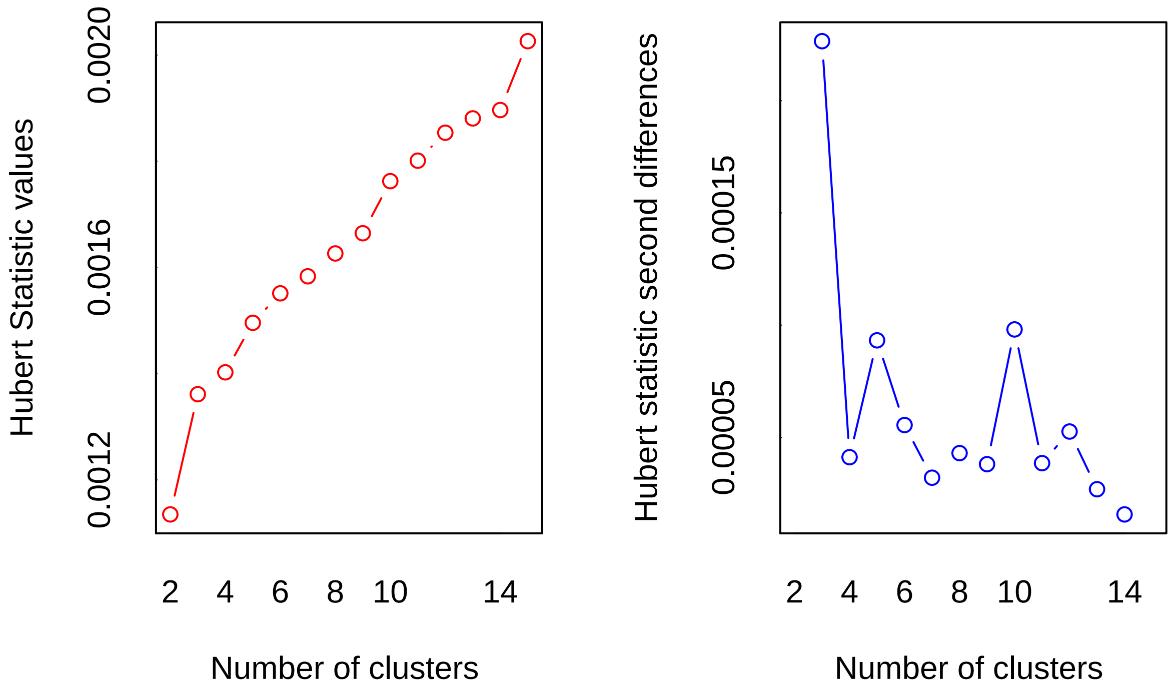
- **Look at the code for wssplot() and figure out how it works**

The wssplot functions determine sum of within-cluster sum of squares of varying number of clusters determined by a k-means method from 2 to maximum number of nc. The nc is the maximum number of clusters to consider, which is 15 in this case. The seed is the random-number seed to ensure reproducible result since k-means require initial random guess of centroids. It plots the sum of the within-cluster sum of squares over the number of clusters k tried.

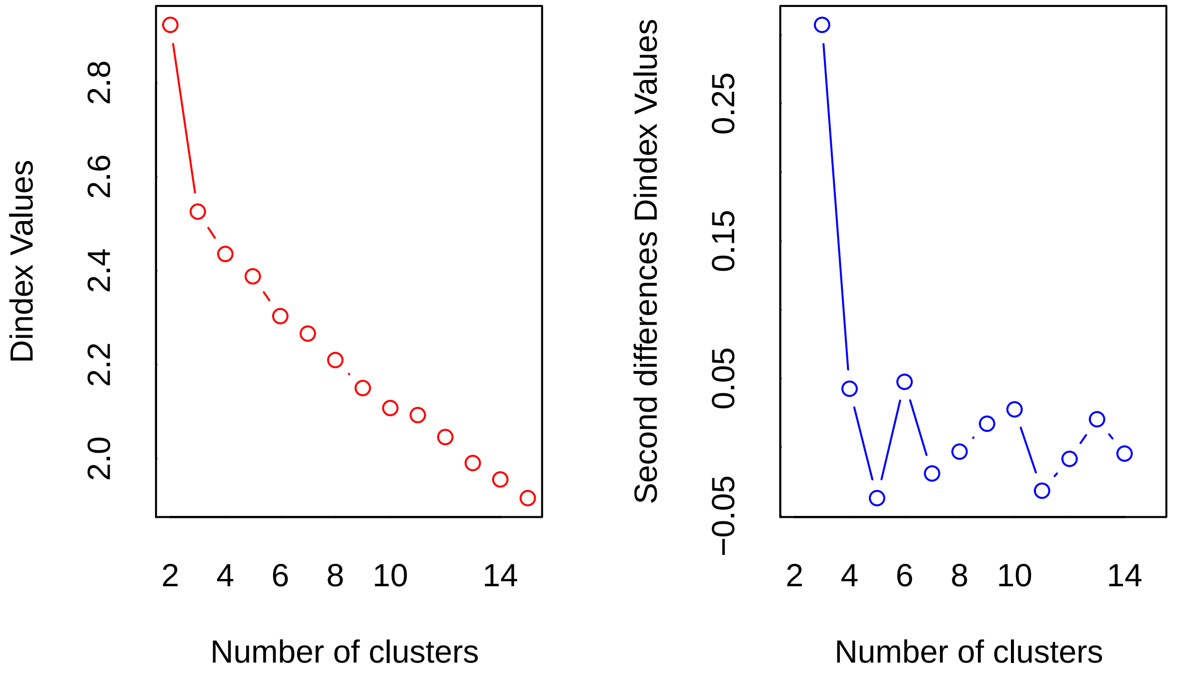
Method 2

Use the NbClust library, which runs many experiments and gives a distribution of potential number of clusters.

```r
set.seed(1234)
nc <- NbClust(wine.noType.scaled, min.nc=2, max.nc=15, method="kmeans")
```

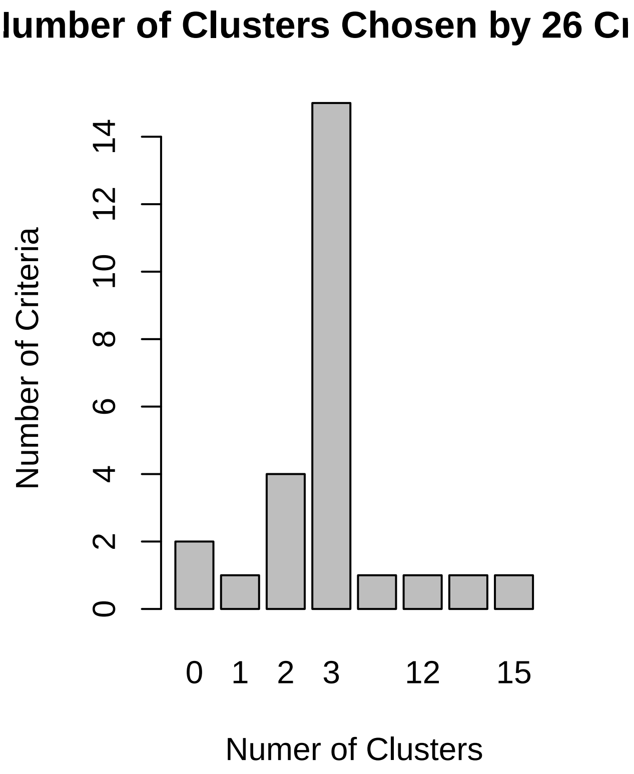


```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```



```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 4 proposed 2 as the best number of clusters
## * 15 proposed 3 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
## * 1 proposed 12 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                    ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

```
barplot(table(nc$Best.n[1,]),
              xlab="Numer of Clusters", ylab="Number of Criteria",
                    main="Number of Clusters Chosen by 26 Criteria")
```



Exercise 3

- **How many clusters does this method suggest?**

The NbClust method clearly suggested k = 3 clusters.

Exercise 4

- **Once you've picked the number of clusters, run k-means using this number of clusters. Output the result of calling kmeans() into a variable fit.km**

```
fit.km <- kmeans(wine.noType.scaled, centers = 3)
fit.km
```

```
## K-means clustering with 3 clusters of sizes 51, 65, 62
##
## Cluster means:
##      Alcohol      Malic        Ash Alcalinity   Magnesium     Phenols
## 1  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
## 2 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
## 3  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
##    Flavanoids Nonflavanoids Proanthocyanins      Color        Hue
## 1 -1.21182921    0.72402116     -0.77751312  0.9388902 -1.1615122
## 2  0.02075402   -0.03343924      0.05810161 -0.8993770  0.4605046
## 3  0.97506900   -0.56050853      0.57865427  0.1705823  0.4726504
##     Dilution    Proline
## 1 -1.2887761 -0.4059428
## 2  0.2700025 -0.7517257
## 3  0.7770551  1.1220202
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [36] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2
##  [71] 2 2 2 3 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2
## [106] 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 3 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
## [141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [176] 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 326.3537 558.6971 385.6983
##  (between_SS / total_SS =  44.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

Now we want to evaluate how well this clustering does.

Exercise 5

- **Using the table() function, show how the clusters in ct.km compares to the actual wine types. Would you consider this a good clustering?**

```
ct.km <- table(wine$Type, fit.km$cluster)
ct.km
```

```
##
##      1  2  3
##   1  0  0 59
##   2  3 65  3
##   3 48  0  0
```

```
randIndex(ct.km)
```

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

It predicts 89.75% accuracy, which is pretty good.

Exercise 6

- **Visualize these clusters using function clusplot() from the cluster library.**

clusplot() can only be used for Partitioning Around Medoids (PAM), Clustering Large Applications (CLARA) and Fuzzy Analysis Clustering (FANNY) methods. So these methods are tried here with the same selection of k = 3. However, only PAM and CLARA works since k = 3 is too small for the FANNY method.

For PAM method, it predicts about 74.11% accuracy.

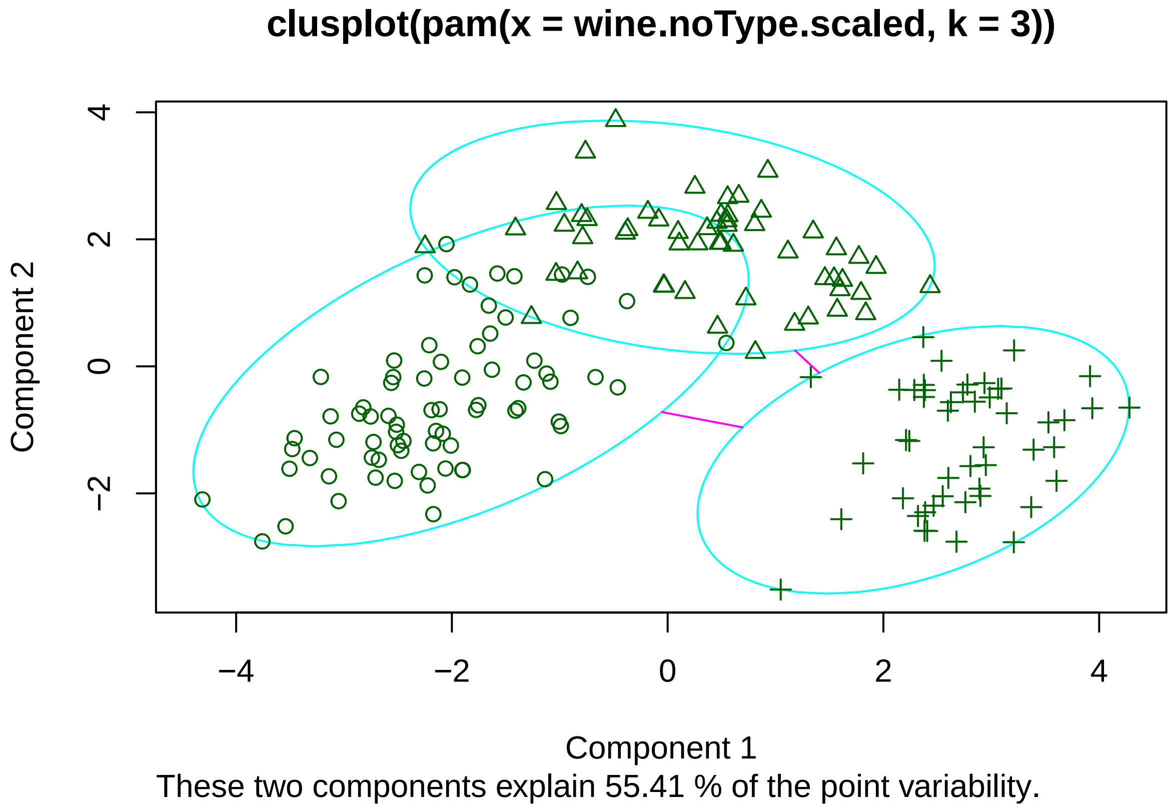
```
fit.pam <- pam(wine.noType.scaled, k = 3)
ct.pam <- table(wine$Type, fit.pam$clustering)
ct.pam
```

```
##
##      1  2  3
##   1 59  0  0
##   2 15 55  1
##   3  0  0 48
```

```
randIndex(ct.pam)
```

```
##       ARI
## 0.7411365
```

```
clusplot(fit.pam)
```



For CLARA method, it predicts about 81.42% acccuracy.

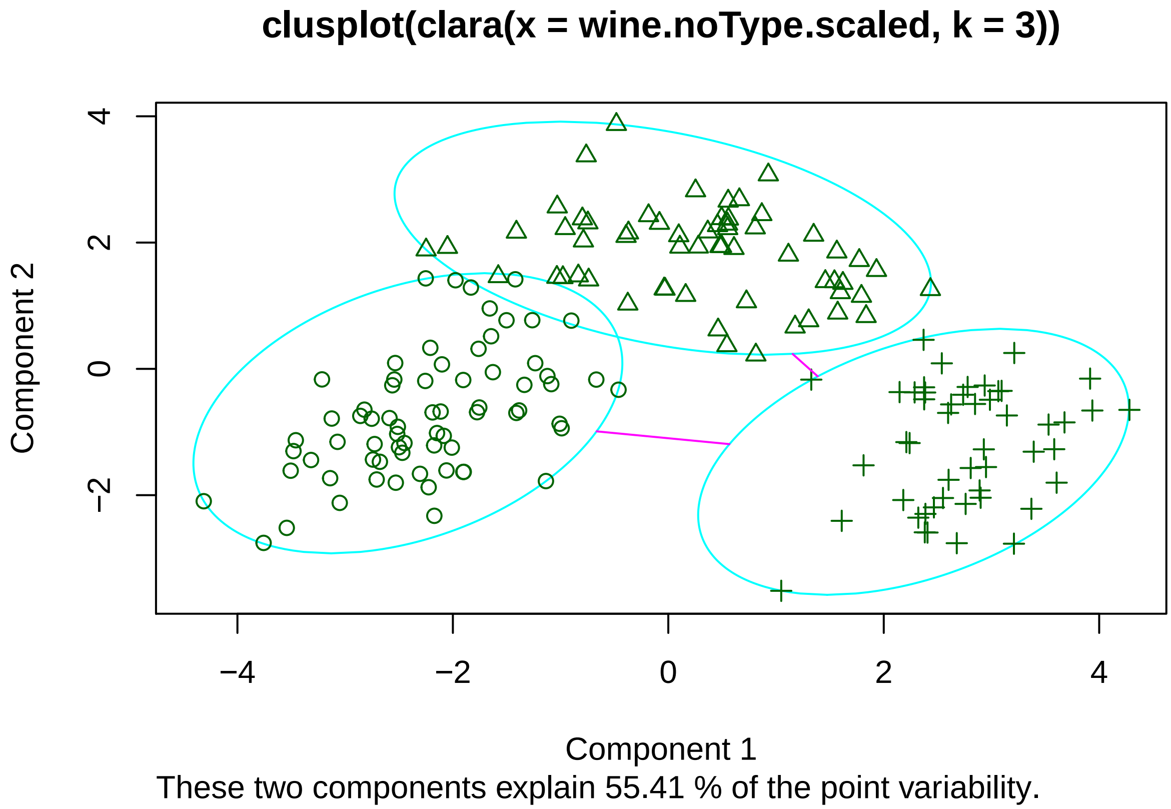
```
fit.clara <- clara(wine.noType.scaled, k = 3)
ct.clara <- table(wine$Type, fit.clara$clustering)
ct.clara
```

```
##
##       1  2  3
##    1 59  0  0
##    2 10 60  1
##    3  0  0 48
```

```
randIndex(ct.clara)
```

```
##       ARI
## 0.8141769
```

```
clusplot(fit.clara)
```



- **Would you consider this a good clustering?**

We can see that k-means is the best clustering method at 89.75% accuracy. Next best is CLARA at 81.24% accuracy. Worst is PAM at 74.11% accuracy.