Cluster Analysis in R

Archit Gupta

16 October 2016

Cluster analysis is a data - reduction technique designed to uncover subgroups of observation within a dataset. A cluster is defined as a group of observations that are more similar to each other than they are to the observation of the other group.

Common steps in cluster analysis:

```
* Choose appropriate attributes
* Scale the data
* Screen the outliers
* Calculate the distances
* Select a clustering Algorithm
* Obtain one or more cluster solutions
* Determine number of clusters present
* Obtain a final Clustering Solution
* Visualize the results
* Interpret the Clusters
* Validate the results
```

Performing Cluster Analysis on nutrient data of flexclust package:

```
data(nutrient, package = "flexclust")
head(nutrient)
```

```
##                 energy protein fat calcium iron
## BEEF BRAISED       340      20  28       9  2.6
## HAMBURGER          245      21  17       9  2.7
## BEEF ROAST         420      15  39       7  2.0
## BEEF STEAK         375      19  32       9  2.6
## BEEF CANNED        180      22  10      17  3.7
## CHICKEN BROILED    115      20   3       8  1.4
```

Finding euclidean distance between observations:

Euclidean distance are usually used as distance measure in case of continuous data.

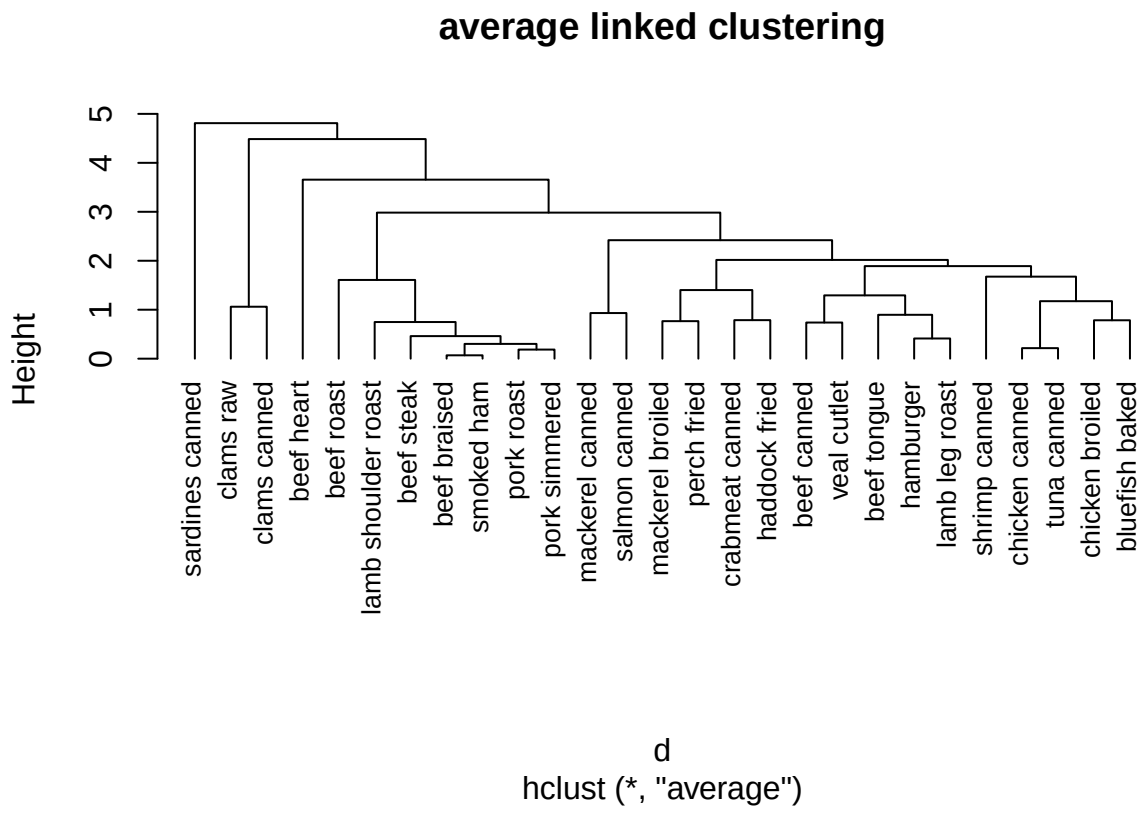
```
d <- dist(nutrient)
as.matrix(d)[1:4,1:4]
```

```
##              BEEF BRAISED HAMBURGER BEEF ROAST BEEF STEAK
## BEEF BRAISED      0.00000   95.6400   80.93429   35.24202
## HAMBURGER        95.64000    0.0000  176.49218  130.87784
## BEEF ROAST       80.93429  176.4922    0.00000   45.76418
## BEEF STEAK       35.24202  130.8778   45.76418    0.00000
```

Hierarchichal clustering - average linked

In Agglomerative hierarchical clustering , each case or observation starts as its own cluster. Clusters are then combined two at a time until all clusters are merged into a single cluster.

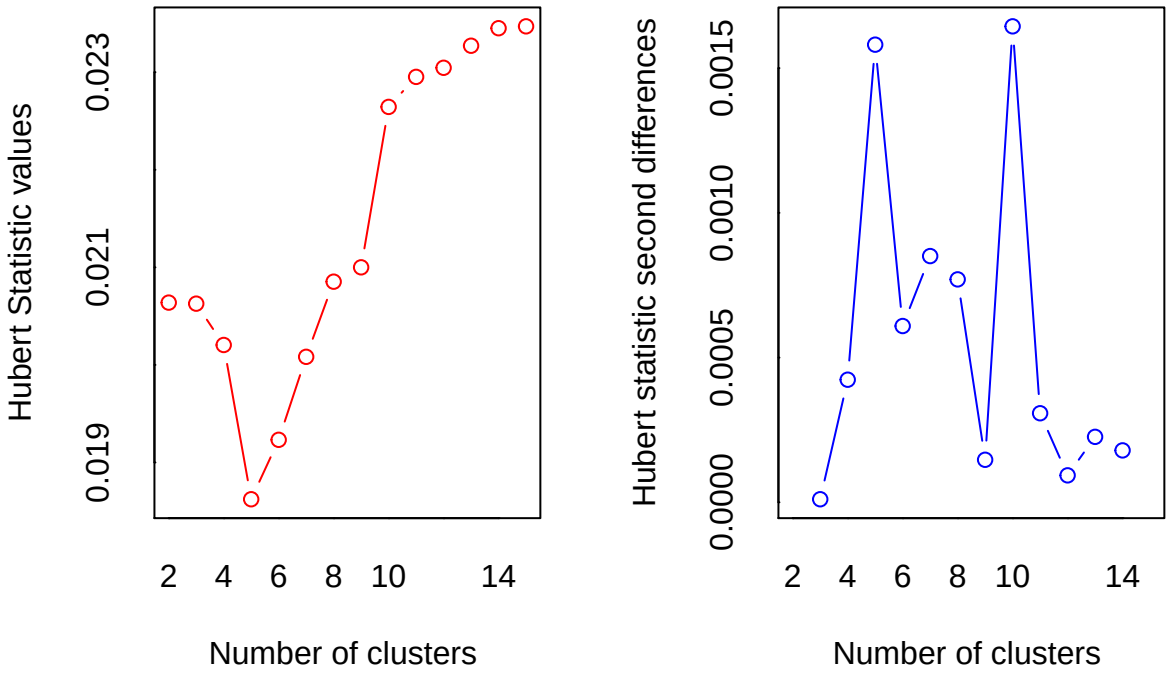
```
row.names(nutrient) <- tolower(row.names(nutrient))
nutrient.scaled <- scale(nutrient)
d <- dist(nutrient.scaled)

fit.average <- hclust(d, method = "average")
plot(fit.average, hang = -1, cex = .8, main = "average linked clustering")
```



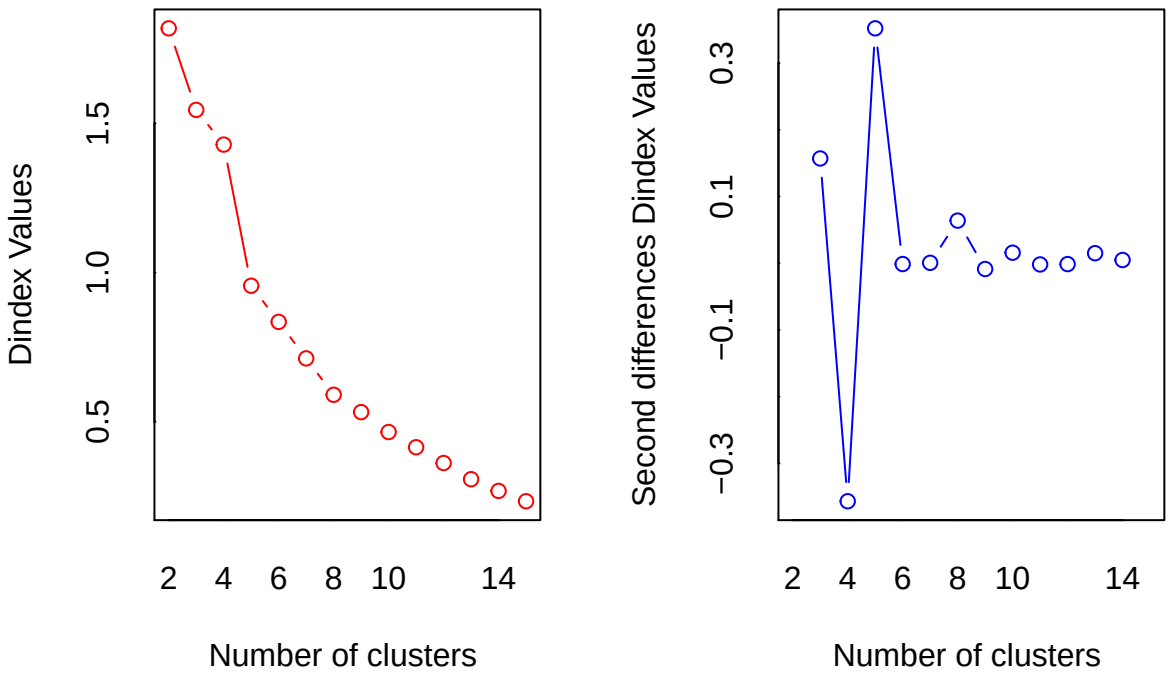
selecting the number of clusters

```
number_of_cluster <- NbClust(nutrient.scaled
                             , distance = "euclidean"
                             , min.nc = 2
                             , max.nc = 15
                             , method = "average")
```

```
## Warning in pf(beale, pp, df2): NaNs produced

## Warning in pf(beale, pp, df2): NaNs produced
```




```
## *** : 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.
##
```



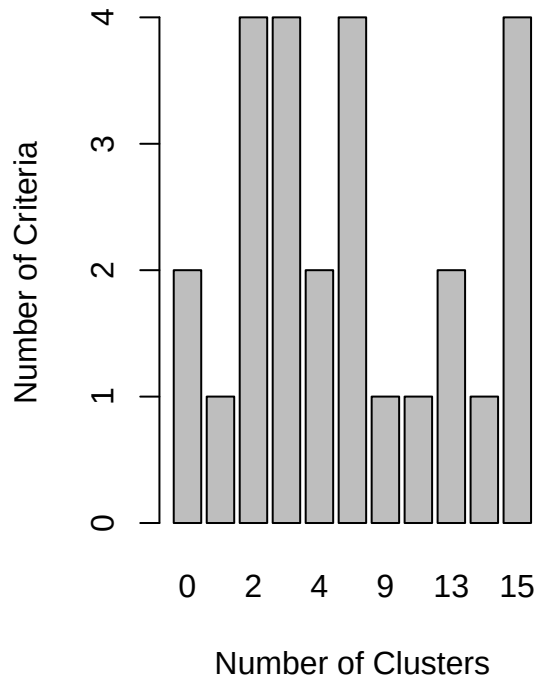

```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 4 proposed 2 as the best number of clusters
## * 4 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 4 proposed 5 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
## * 2 proposed 13 as the best number of clusters
## * 1 proposed 14 as the best number of clusters
## * 4 proposed 15 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
table(number_of_cluster$Best.n[1,])
```

```
##
```

```
##  0  1  2  3  4  5  9 10 13 14 15
##  2  1  4  4  2  4  1  1  2  1  4
```

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



Obtaining the final cluster solution

Assigning classes

```
clusters <- cutree(fit.average, k = 5)

table(clusters)
```

```
## clusters
##  1  2  3  4  5
##  7 16  1  2  1
```

Describes clusters

```
aggregate(nutrient, by = list(cluster= clusters), median)
```

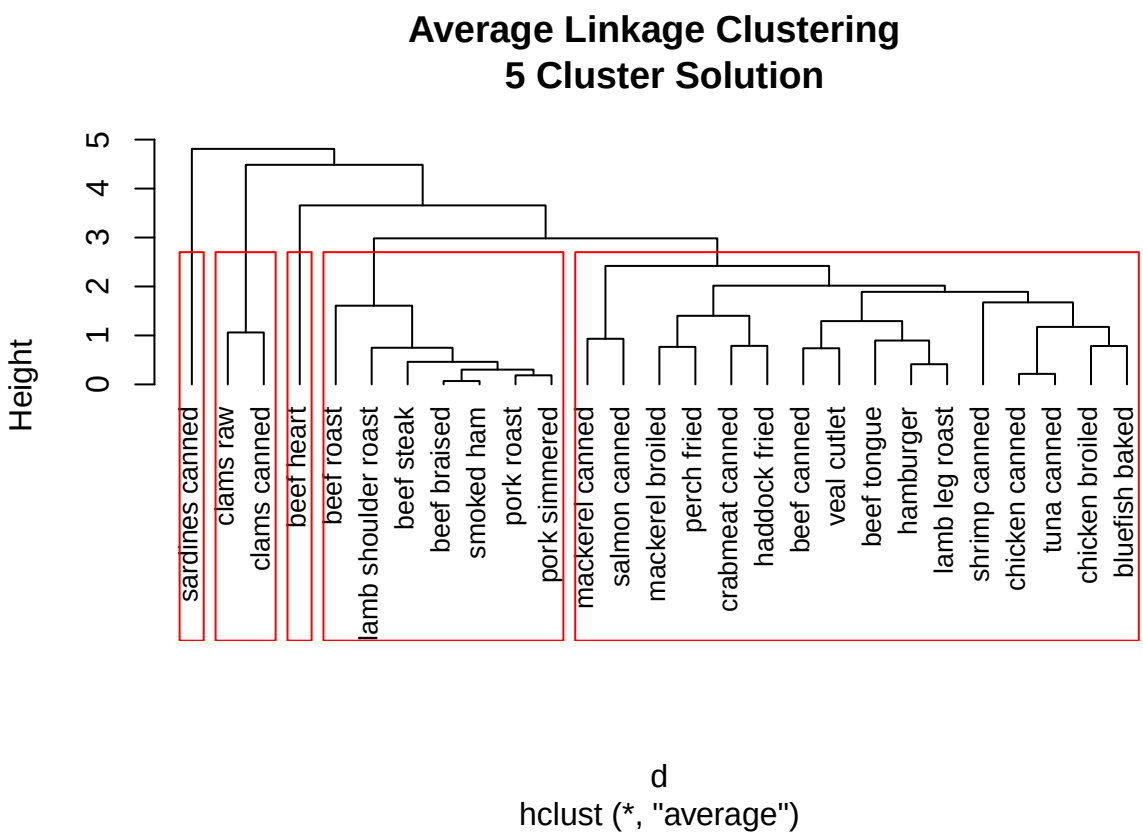
```
##   cluster energy protein fat calcium iron
## 1       1  340.0      19  29       9 2.50
## 2       2  170.0      20   8      13 1.45
## 3       3  160.0      26   5      14 5.90
## 4       4   57.5       9   1      78 5.70
## 5       5  180.0      22   9     367 2.50
```

```
aggregate(as.data.frame(nutrient.scaled), by = list(cluster= clusters), median)
```

```
##   cluster     energy    protein        fat    calcium        iron
## 1       1  1.3101024  0.0000000  1.3785620 -0.4480464  0.08110456
## 2       2 -0.3696099  0.2352002 -0.4869384 -0.3967868 -0.63743114
## 3       3 -0.4684165  1.6464016 -0.7534384 -0.3839719  2.40779157
## 4       4 -1.4811842 -2.3520023 -1.1087718  0.4361807  2.27092763
## 5       5 -0.2708033  0.7056007 -0.3981050  4.1396825  0.08110456
```

Plotting

```
plot(fit.average, hang = -1, cex = .8, main = "Average Linkage Clustering \n 5 Cluster Solution")
rect.hclust(fit.average, k = 5)
```



Partitioning Cluster Analysis

In partitioning Cluster analysis , observation are divided into K groups and reshuffled to form the most cohesive cluster possible according to a given criterion.

k-means clustering

```
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
```

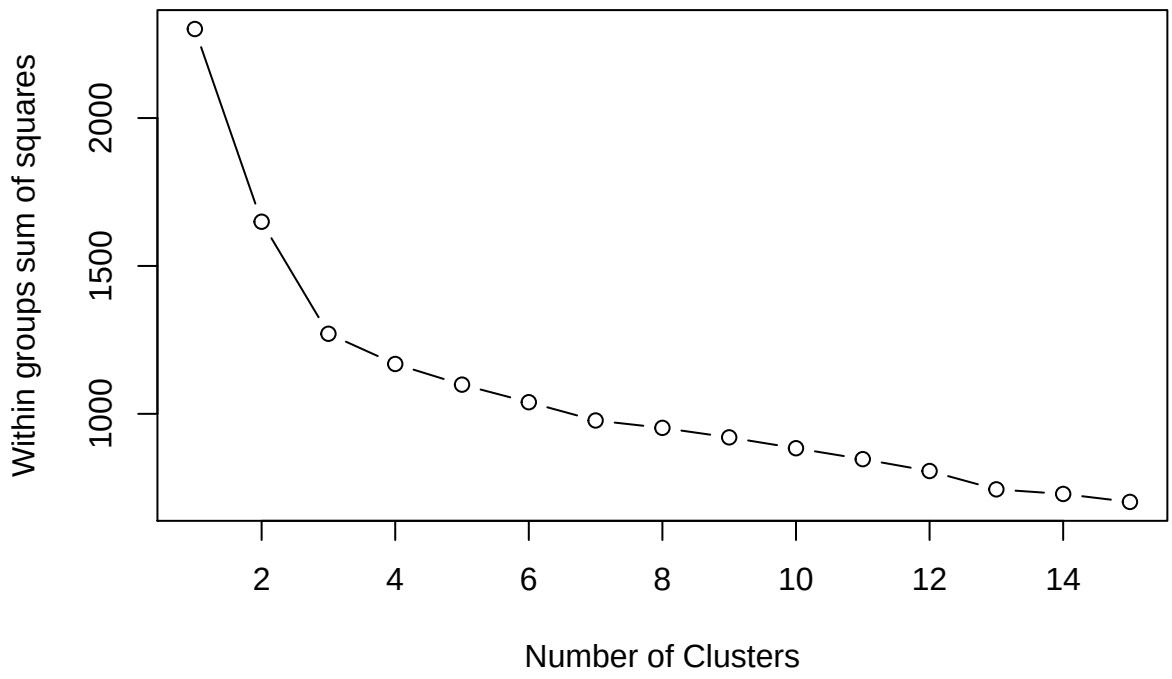
```
df <- scale(wine[-1])
head(df)
```

```
##        Alcohol       Malic        Ash Alcalinity  Magnesium   Phenols
## [1,] 1.5143408 -0.56066822  0.2313998 -1.1663032 1.90852151 0.8067217
## [2,] 0.2455968 -0.49800856 -0.8256672 -2.4838405 0.01809398 0.5670481
## [3,] 0.1963252  0.02117152  1.1062139 -0.2679823 0.08810981 0.8067217
## [4,] 1.6867914 -0.34583508  0.4865539 -0.8069748 0.92829983 2.4844372
## [5,] 0.2948684  0.22705328  1.8352256  0.4506745 1.27837900 0.8067217
## [6,] 1.4773871 -0.51591132  0.3043010 -1.2860793 0.85828399 1.5576991
##      Flavanoids Nonflavanoids Proanthocyanins      Color        Hue
## [1,]  1.0319081    -0.6577078       1.2214385  0.2510088  0.3611585
## [2,]  0.7315653    -0.8184106      -0.5431887 -0.2924962  0.4049085
## [3,]  1.2121137    -0.4970050       2.1299594  0.2682629  0.3174085
## [4,]  1.4623994    -0.9791134       1.0292513  1.1827317 -0.4263410
## [5,]  0.6614853     0.2261576       0.4002753 -0.3183774  0.3611585
## [6,]  1.3622851    -0.1755994       0.6623487  0.7298108  0.4049085
##       Dilution     Proline
## [1,] 1.8427215  1.01015939
## [2,] 1.1103172  0.96252635
## [3,] 0.7863692  1.39122370
## [4,] 1.1807407  2.32800680
## [5,] 0.4483365 -0.03776747
## [6,] 0.3356589  2.23274072
```

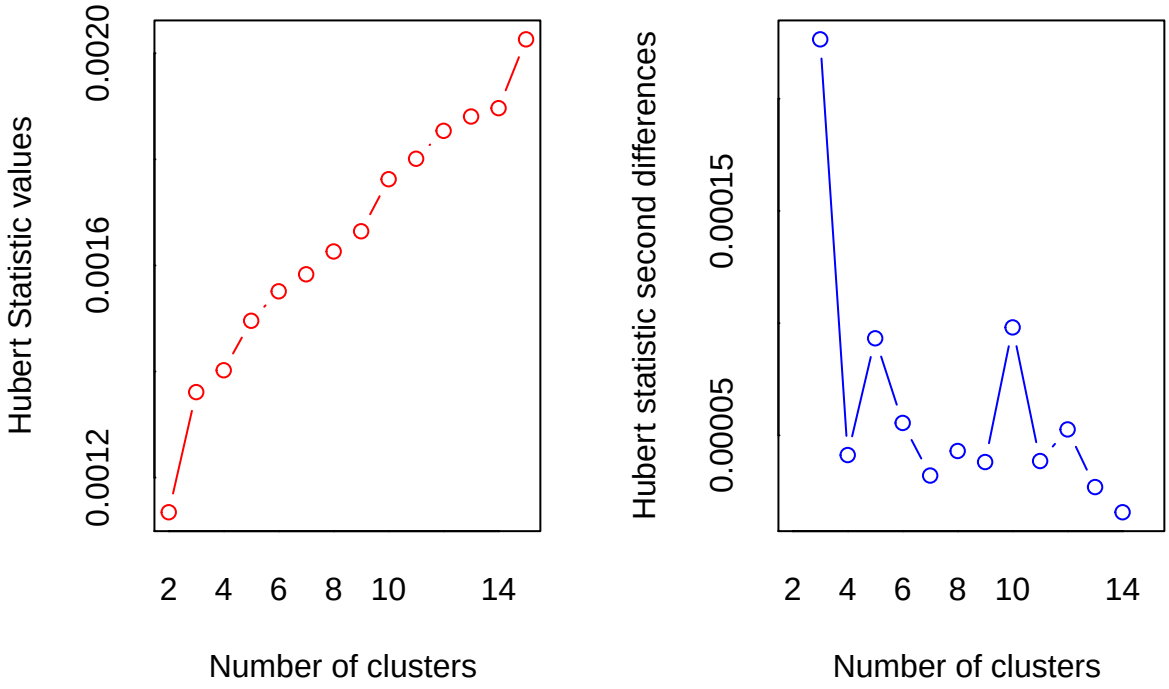
creating a function for plotting total within-groups sum of squares against number of cluster

To determine the value of the parameter kk. If we looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion".

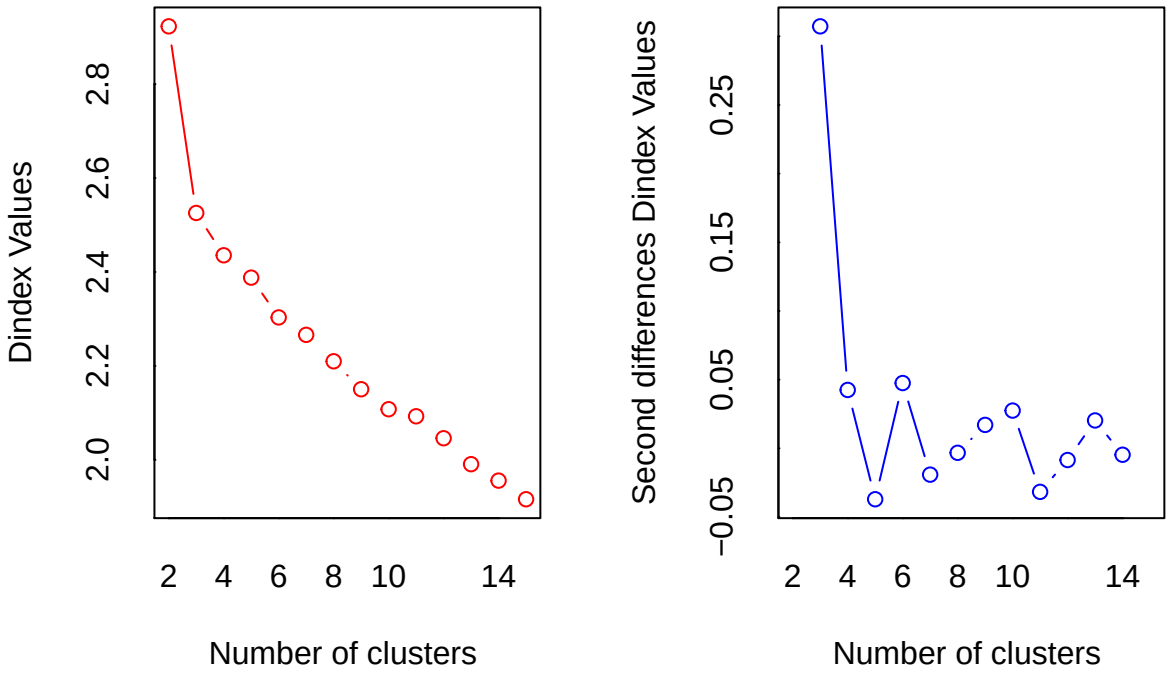
```
wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(df)
```




```
library(NbClust)
set.seed(1234)
devAskNewPage(ask = TRUE)
nc <- NbClust(df, 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
##
##
## *******************************************************************
```

```
table(nc$Best.n[1,])
```

```
##
##  0  1  2  3 10 12 14 15
##  2  1  4 15  1  1  1  1
```

```r
barplot(table(nc$Best.n[1,]),
        xlab = "Number of Clusters",
        ylab = "Number of Criteria",
        main = "Number of Clusets chosen over \n 26 criteria")
set.seed(1234)
fit.km <- kmeans(df, 3, nstart = 25)
fit.km
```

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

```r
fit.km$size
```

```
## [1] 62 65 51
```

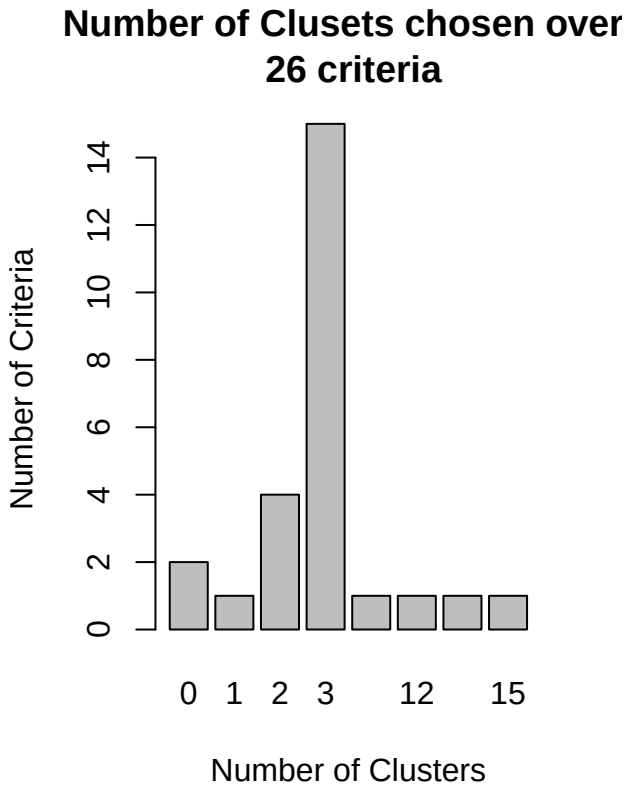
```r
fit.km$centers
```

```
##      Alcohol      Malic        Ash Alcalinity   Magnesium     Phenols
## 1  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
## 2 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
## 3  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
##    Flavanoids Nonflavanoids Proanthocyanins      Color        Hue
## 1  0.97506900   -0.56050853      0.57865427  0.1705823  0.4726504
## 2  0.02075402   -0.03343924      0.05810161 -0.8993770  0.4605046
```

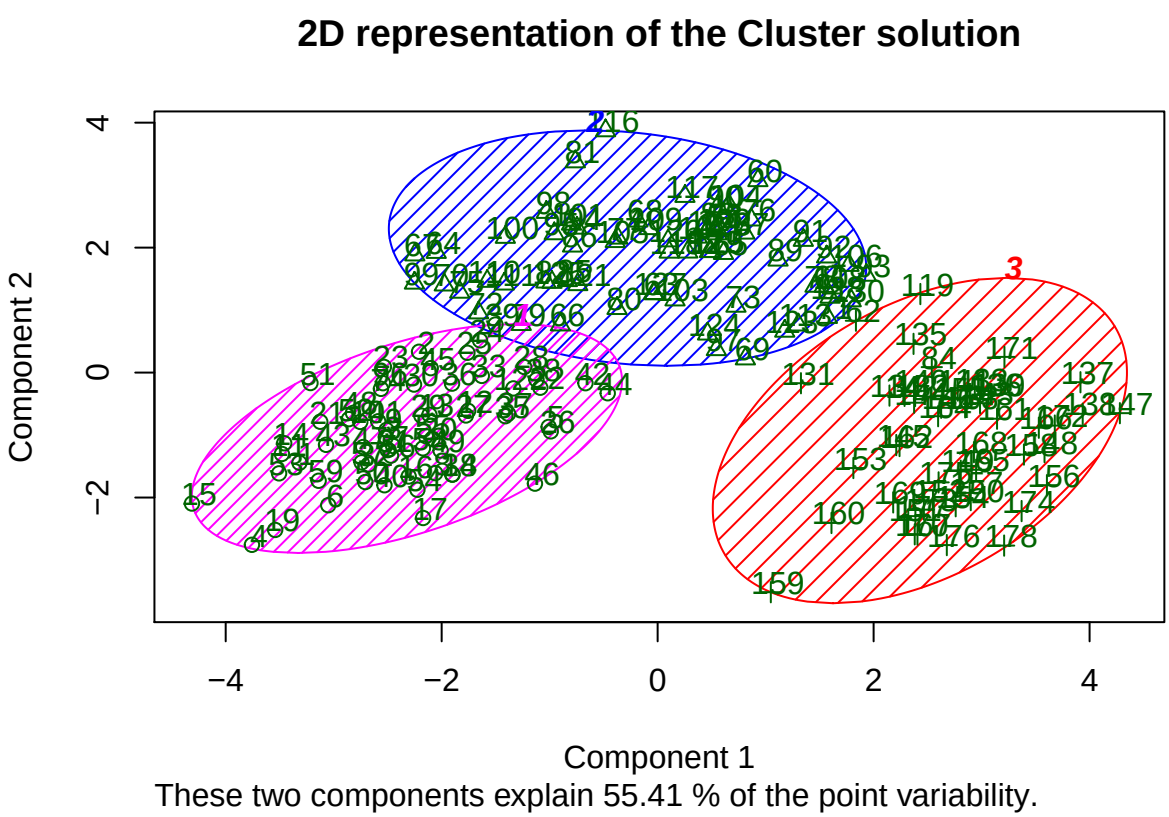
```
## 3 -1.21182921    0.72402116     -0.77751312  0.9388902 -1.1615122
##     Dilution    Proline
## 1  0.7770551  1.1220202
## 2  0.2700025 -0.7517257
## 3 -1.2887761 -0.4059428
```

```
aggregate(wine[-1], by = list(clusters = fit.km$cluster), mean)
```

```
##   clusters  Alcohol    Malic      Ash Alcalinity Magnesium  Phenols
## 1        1 13.67677 1.997903 2.466290   17.46290 107.96774 2.847581
## 2        2 12.25092 1.897385 2.231231   20.06308  92.73846 2.247692
## 3        3 13.13412 3.307255 2.417647   21.24118  98.66667 1.683922
##   Flavanoids Nonflavanoids Proanthocyanins    Color       Hue Dilution
## 1  3.0032258     0.2920968        1.922097 5.453548 1.0654839 3.163387
## 2  2.0500000     0.3576923        1.624154 2.973077 1.0627077 2.803385
## 3  0.8188235     0.4519608        1.145882 7.234706 0.6919608 1.696667
##     Proline
## 1 1100.2258
## 2  510.1692
## 3  619.0588
```




```
library(cluster)
clusplot(df, fit.km$cluster, main='2D representation of the Cluster solution',
         color=TRUE, shade=TRUE,
         labels=2, lines=0)
```



How well did k-means clustering uncover the actual structure of the data contained in the Type variable ? A cross tabulation of Type(wine varietal) and cluster membership is given by:

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

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

To quantify the agreement between type and cluster using an adjusted Rand index, provided by the flexclust package:

```
library(flexclust)
randIndex(ct.km)
```

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

The adjusted Rand index provides a measure of the agreement between two partitions, adjusted for chance. It ranges from -1(no agreement) to 1(perfect agreement) , Agreement between the wine variatal type and the cluster solution is 0.9.

References:

* R in action - Robert I. Kabacoff