clustering_3

Uohna Thiessen

February 17, 2017

Exercise 0: Install packages "cluster", "rattle", and "NbClust".

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

```
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)
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
```

Exercise 1: Remove the first column from the data and scale it using the scale() function

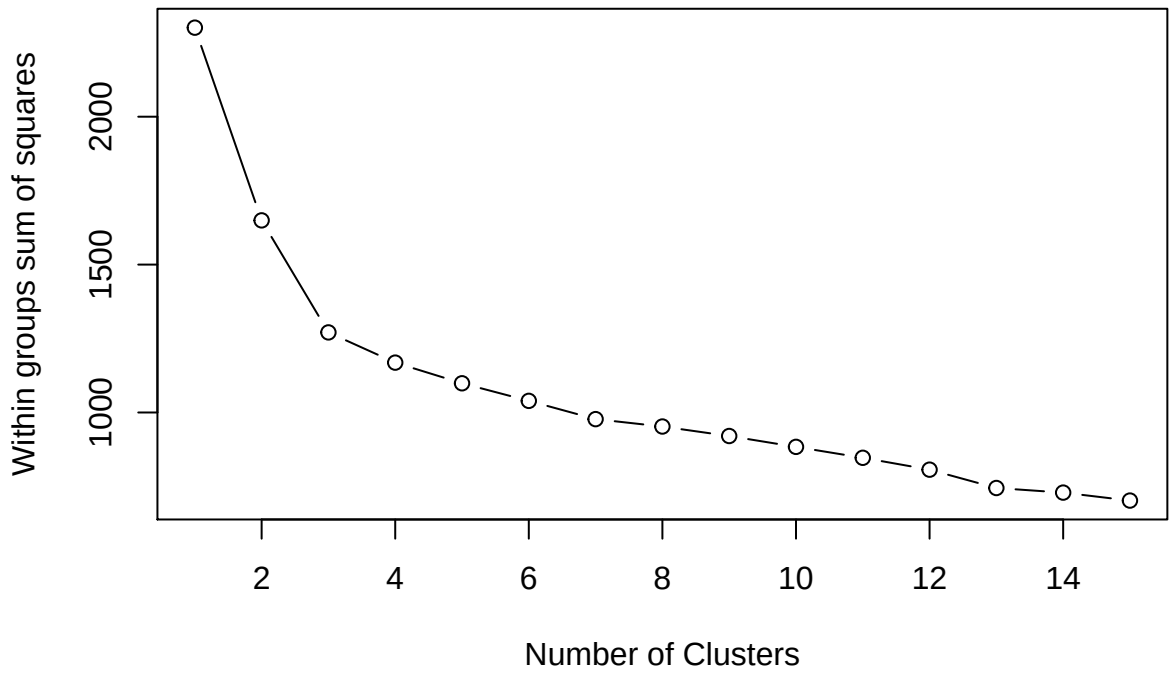
```
df <- scale(wine[-1])
```

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(df)
```



Exercise 2:

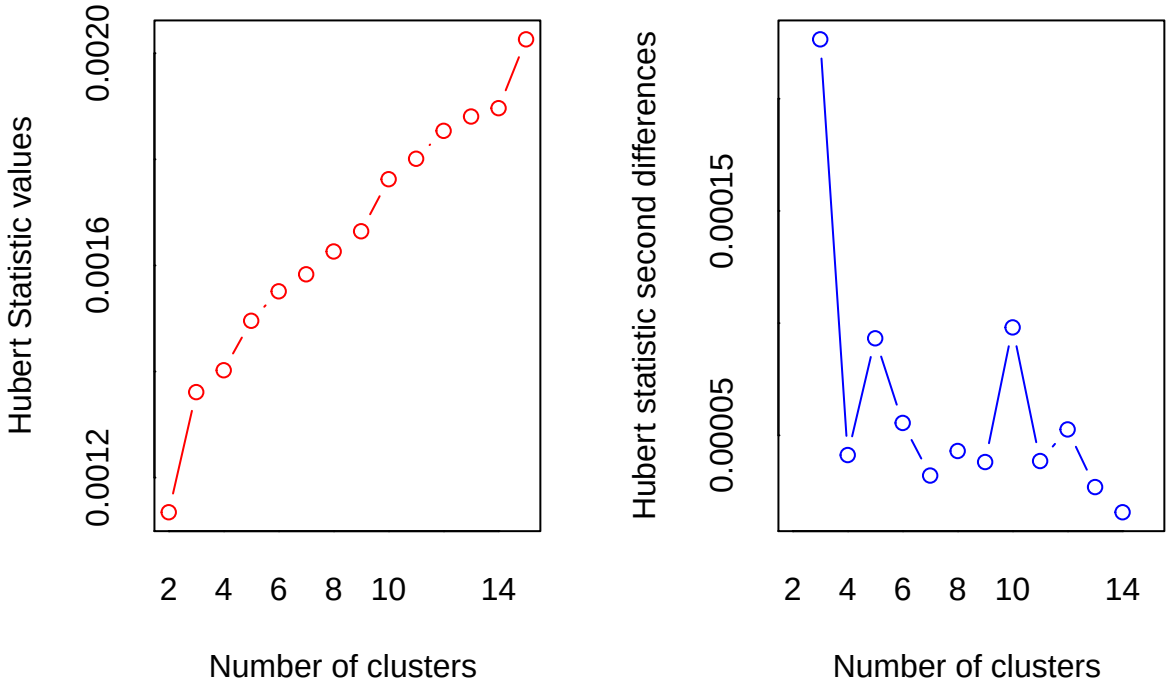
How many clusters does this method suggest? 3

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

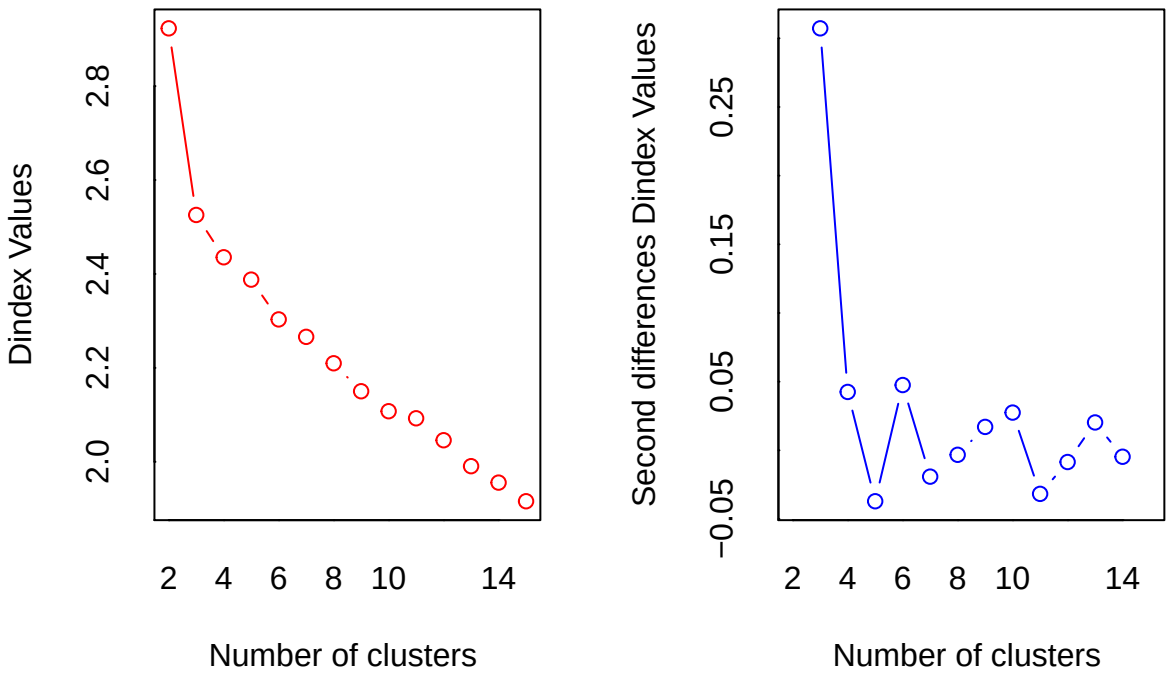
The plot shows a distinct drop in the within groups sum of squares when moving from 1 to 3 clusters. After three clusters, the plot levels off, indicating that a 3-cluster solution is the best fit to the data.

Method 2: Use the NbClust library, which runs many experiments and gives a distribution of potential number of clusters.

```
library(NbClust)
set.seed(1234)
nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans")
```

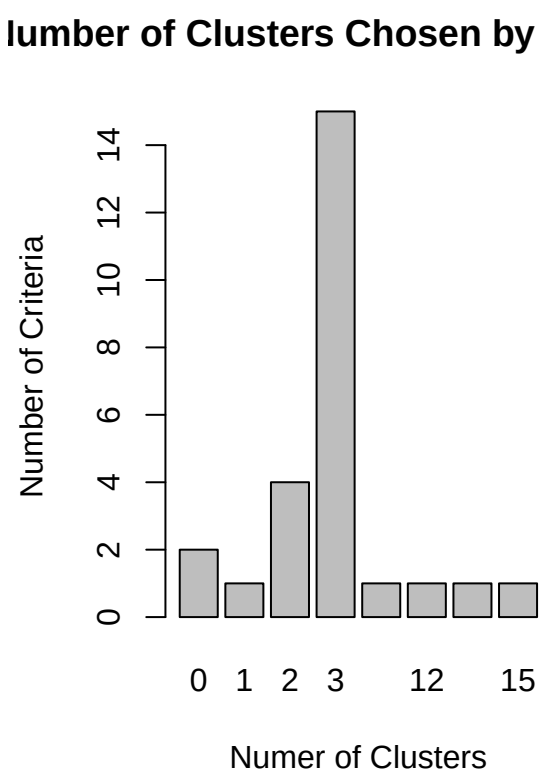



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##       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

```
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?

This method also suggest three(3) clusters is the best fit.

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(df,3)
```

Exercise 5: using the table() function, show how the clusters in fit.km compare to the actual wine types in wine$Type. Would you consider this a good clustering?

Yes, the table show an effective partiioning off into defined groups.

```
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
```

Exercise 6:

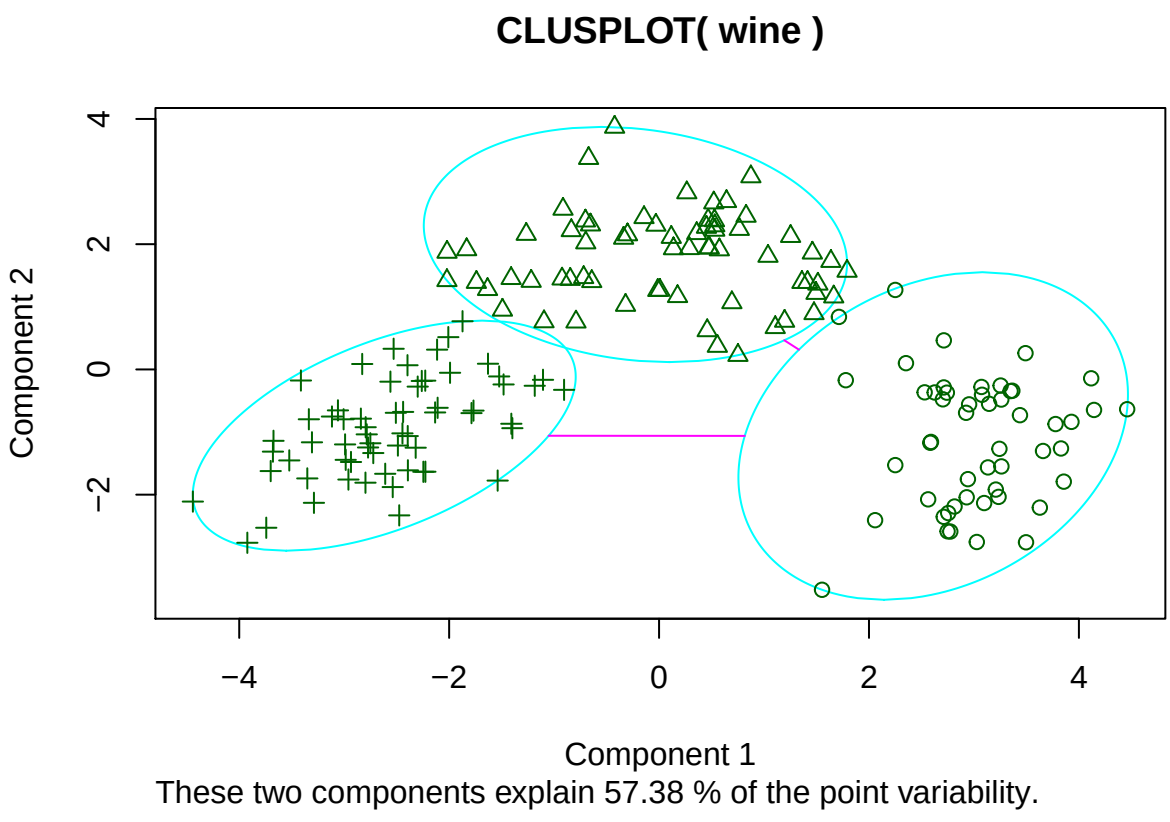
Visualize these clusters using function clusplot() from the cluster library. Would you consider this a good clustering?

The grouping in the plot is visually effective and the adjusted Rand index(ARI) is almost 1.

```
library(cluster)
library(flexclust)
```

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

```
clusplot(wine, fit.km$cluster)
```




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

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