clustering-R-Exercise

Giollamhir

July 13, 2017

This mini-project is based on the K-Means exercise from 'R in Action' Go here for the original blog post and solutions http://www.r-bloggers.com/k-means-clustering-from-r-in-action/

Exercise 1: Remove the first column from the data and scale it using the scale() function *Note. I did this instruction at first using wine$Type = NULL because $Type is the first column, but the last exercise requires using the Type column so I removed that code.*

Now we'd like to cluster the data using K-Means. How do we decide how many clusters to use if you don't know that already? We'll try two methods.

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.

```
wine2 <- wine
wine2$Type = NULL   # that's how I did it, I see on the blog you can also do it
#  by omitting it from the new definition all as one step df <- scale(wine[-1])


str(wine2)
```

```
## 'data.frame':    178 obs. of  13 variables:
##  $ 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 ...
```

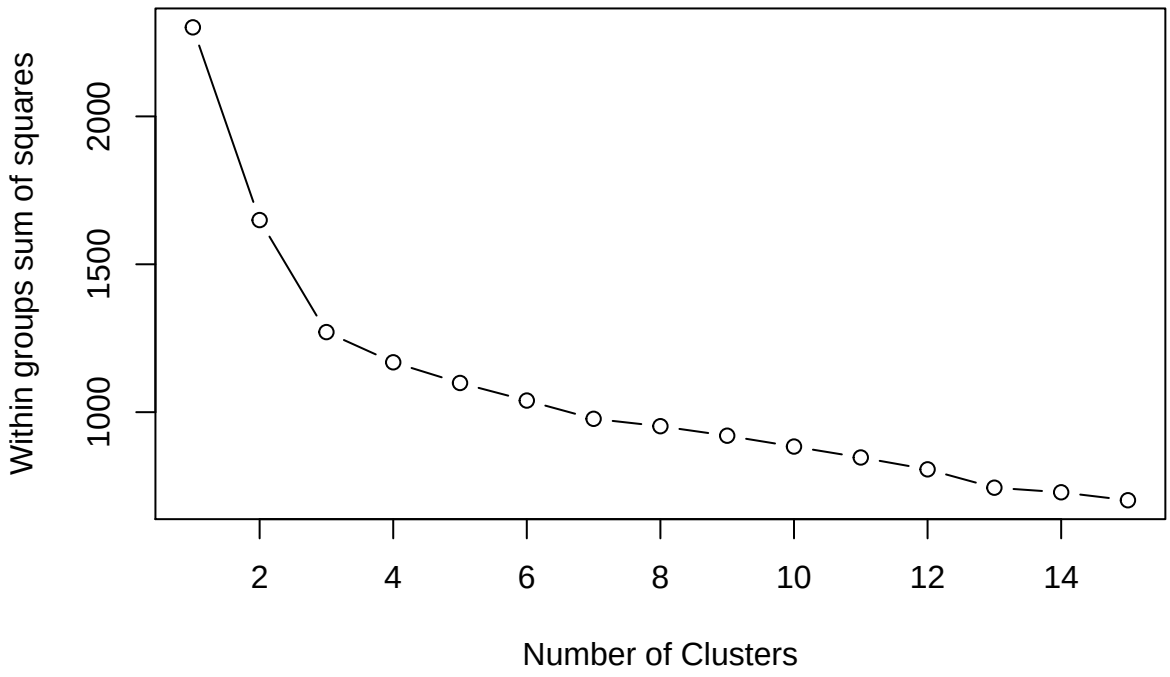
```
# scale  it using the scale() function
wine2 <- scale(wine2)

# Now cluster the data
wssplot <- function(data, nc=15, seed=1234){
                  wss <- (nrow(data)-1)*sum(apply(data,2,var)) # sums variances across
                  # 2nd column of the df in this case wine2
                      for (i in 2:nc){   # here, passing  15 as the limit
                          # as the max. number of clusters to calculate and graph
                set.seed(seed)
                      wss[i] <- sum(kmeans(data, centers=i)$withinss)} # kmeans provides
                      # withinss a vector listing within-cluster sum of squares, one component
                      # targeted number of clusters 1 - 15

              plot(1:nc, wss, type="b", xlab="Number of Clusters",
```

```
                              ylab="Within groups sum of squares")
      }

wssplot(wine2)
```



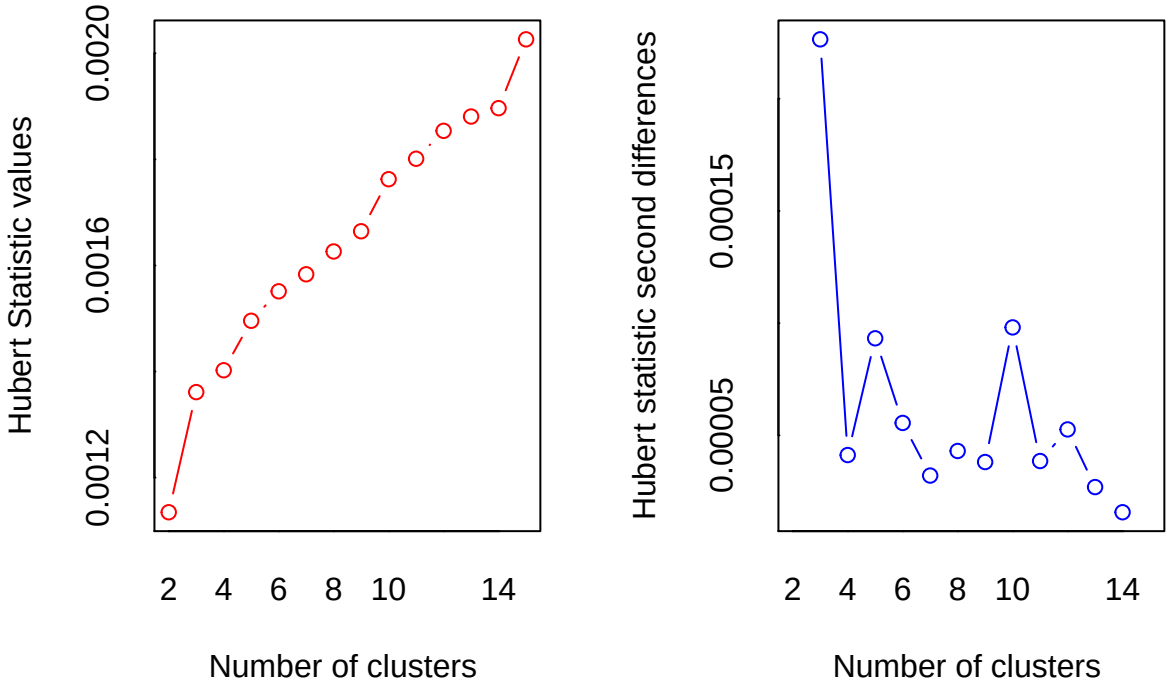
Exercise 2: **Questions.**

- How many clusters does this method suggest?
- Why does this method work? What's the intuition behind it?
- Look at the code for wssplot() and figure out how it works

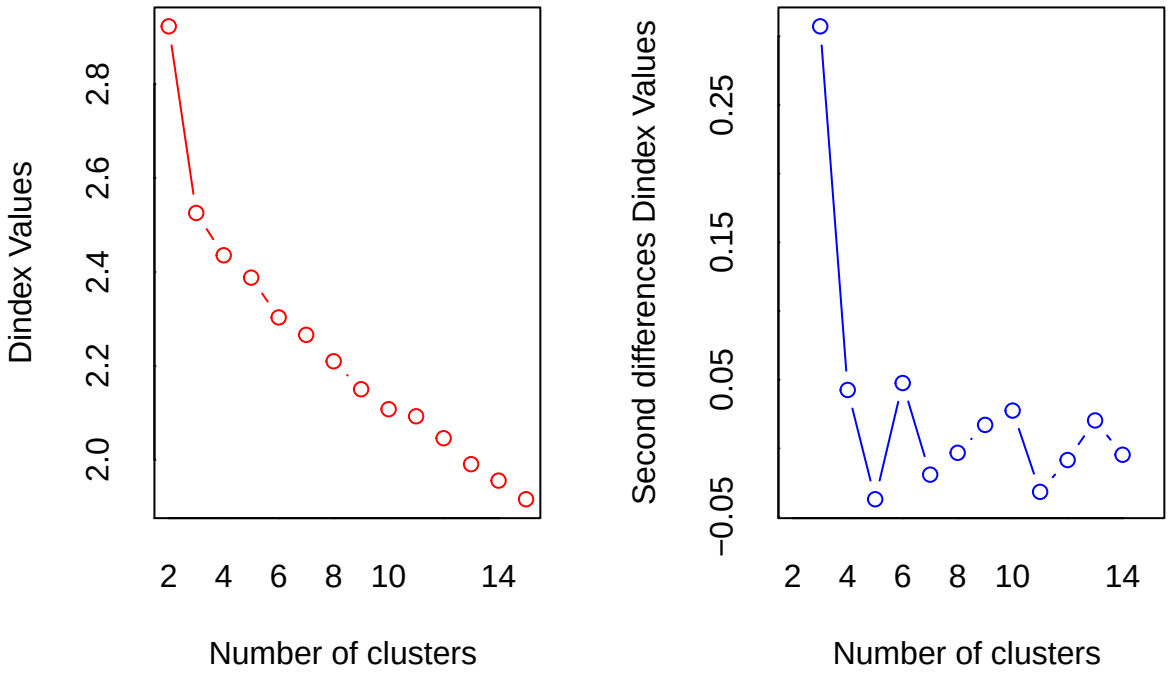
Answers.

- Two or three clusters would be most efficent, depending on the application.
- The sum of sqares indicates the difference between the points in the clusters. Where the tail is flat would indicate not that much differentiation between some of the clusters; where the curve is steep at the start would indicate a great deal of difference contained between the first and second clusters, so the clusters are less meaningful, or a bit eclectic.
- For explanation of the wssplot() code see comments in the code chunk above.

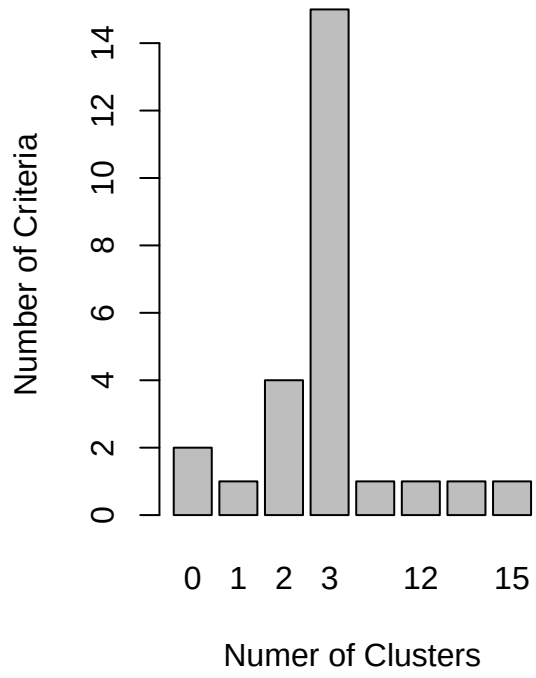
Method 2: Use the NbClust library, which runs many experiments and gives a distribution of potential number of clusters.




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



Exercise 3: How many clusters does this method suggest?

Answer. This method suggests three 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(wine2, centers=3, iter.max=1000)
```

Now we want to evaluate how well this clustering does.

Exercise 5: using the table() function, show how the clusters in fit.km*clusterscomparestotheactualwinetypesinwine*Type. Would you consider this a good clustering?

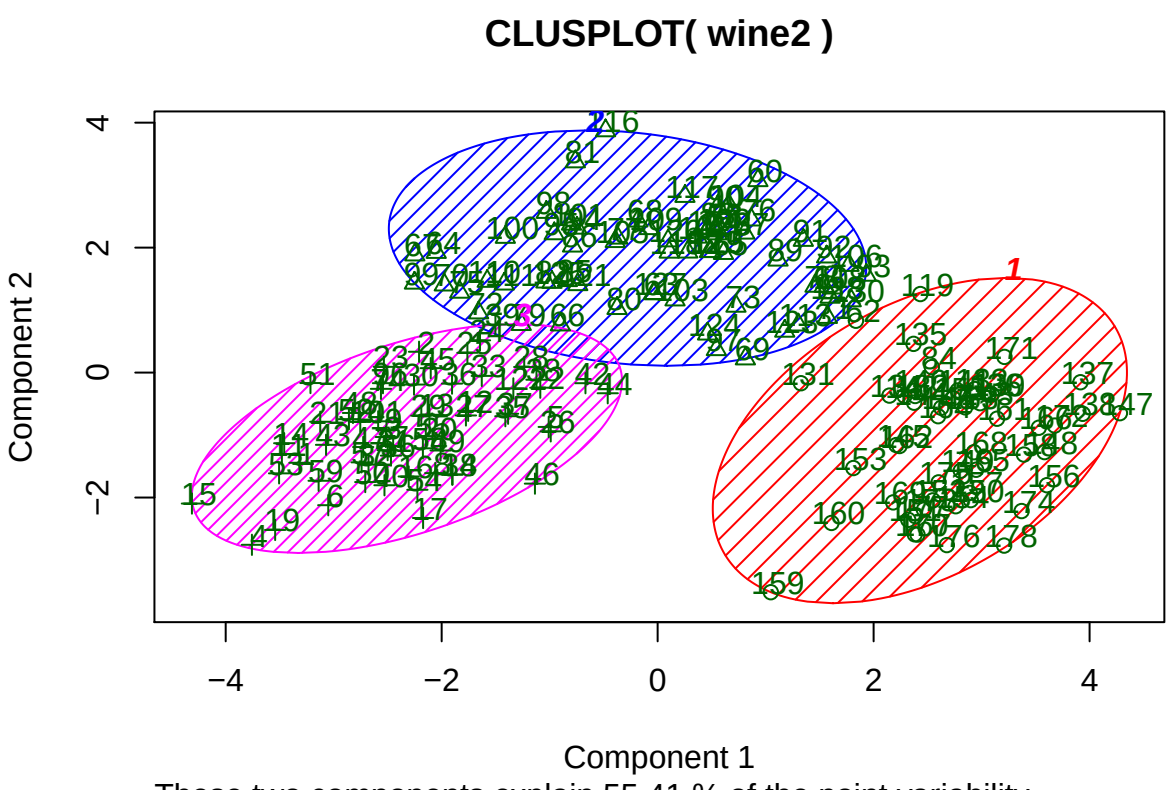
```
##
##      1  2  3
##   1  0  0 59
##   2  3 65  3
##   3 48  0  0
```

Answer. Yes. I'd say the three types correspond very well with the three clusters: type 1 corresponde 100% to the first cluster, and Type 3 corresponds fully with the second cluster. Type two has a few wines in other categories but 92% of that type are clustered into the third cluster.

Exercise 6:

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

Answer. Yes. See plot:


CLUSPLOT(wine2)
Component 2
Component 1
These two components explain 55.41 % of the point variability.