# 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 0:

# Install these packages if you don’t have them already:

# install.packages(c(“cluster”, “rattle.data”,“NbClust”))

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

data(wine, package="rattle.data")  
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:

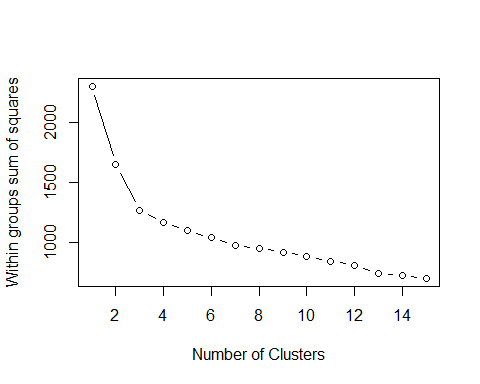
# Remove the first column from the data and scale it using the scale() function

df = scale(wine[-1])

# 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.

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?

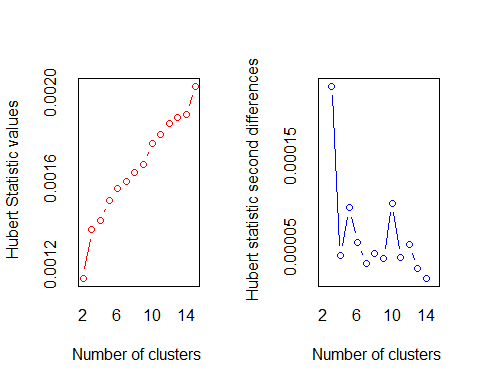
# \* Why does this method work? What’s the intuition behind it?

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

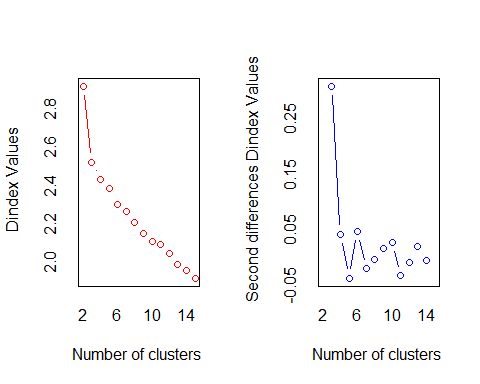
## This method suggests 3 clusters. The function for wssplot iterates the kmeans function across all columns within the df and extracts the $withinss value. Then, this figure is applied to each column and plotted.

# 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")

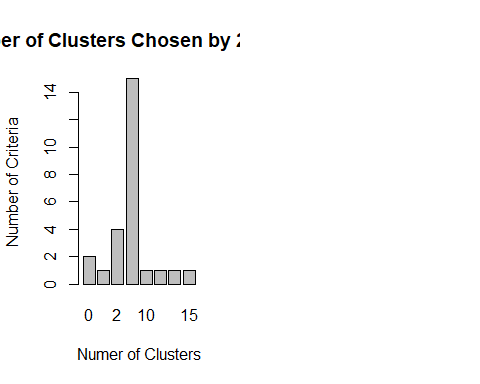


## \*\*\* : The Hubert index is a graphical method of determining the number of clusters.  
## In the plot of Hubert index, we seek a significant knee that corresponds to a   
## significant increase of the value of the measure i.e the significant peak in Hubert  
## index second differences plot.  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



## \*\*\* : 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 suggested 3 clusters according to majority rule from 15 indices.

# 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, nstart=25)

# Now we want to evaluate how well this clustering does.

# Exercise 5:

# \*Using the table() function, show how the clusters in fit.km$clusters # compares to the actual wine types in wine$Type.

# \*Would you consider this a good clustering?

wine\_val = table(wine$Type, fit.km$cluster)

## After reviewing the attached link, my numbers are in reverse order, but the actual figures are correct. After reviewing the table, 6 predictions were incorrect, mainly in the second cluster.

# Exercise 6:

# \* Visualize these clusters using function clusplot() from the cluster library

# \* Would you consider this a good clustering?

library(cluster)

## Warning: package 'cluster' was built under R version 3.4.4

clusplot(df, fit.km$cluster, main = "Cusplot")