Lab 27. K-means

Data Preprocessing

```
plants <- read.csv("plants.csv", sep=';')
head(plants)
```

```
##   plant.name  pdias longindex durflow height begflow mycor vegaer vegsout
## 1      Aceca  96.84 0.0000000       2      7       5     2      0       0
## 2      Aceps 110.72 0.0000000       3      8       4     2      0       0
## 3      Agrca   0.06 0.6666667       3      2       6     2      0       1
## 4      Agrst   0.08 0.4888889       2      2       7     1      2       0
## 5      Ajure   1.48 0.4761905       3      2       5     2      2       0
## 6      Allpe   2.33 0.5000000       3      5       4     0      0       0
##   autopoll insects wind lign piq ros semiros leafy suman winan monocarp
## 1        0       4    0    1   0   0       0     1     0     0        0
## 2        0       4    0    1   0   0       0     1     0     0        0
## 3        0       0    4    0   0   0       0     1     0     0        0
## 4        0       0    4    0   0   0       0     1     0     0        0
## 5        1       3    0    0   0   0       1     0     0     0        0
## 6        3       3    0    0   0   0       1     0     1     0        1
##   polycarp seasaes seashiv seasver everalw everparti elaio endozoo epizoo aquat
## 1        1       1       0       0       0          0     0       0      0     0
## 2        1       1       0       0       0          0     0       0      0     0
## 3        1       0       0       0       1          0     0       0      0     0
## 4        1       0       0       0       1          0     0       0      0     0
## 5        1       0       0       0       1          0     1       0      0     0
## 6        0       0       1       0       0          0     0       0      0     0
##   windgl unsp
## 1      1    0
## 2      1    0
## 3      0    1
## 4      0    1
## 5      0    0
## 6      0    1
```

1. Remove rows with at least 3 NAs
2. Replace NAs with column means
3. Scale columns with doubles

```
prepare_dataframe <- function(data) {
  data <- subset(data, select=-c(plant.name))

  cnt_na <- apply(data, 1, function(z) sum(is.na(z)))
  data <- data[cnt_na < 3,]

  mean_pdias <- mean(data[ ,'pdias'], na.rm = TRUE)
  mean_longindex <- mean(data[ ,'longindex'], na.rm = TRUE)
```

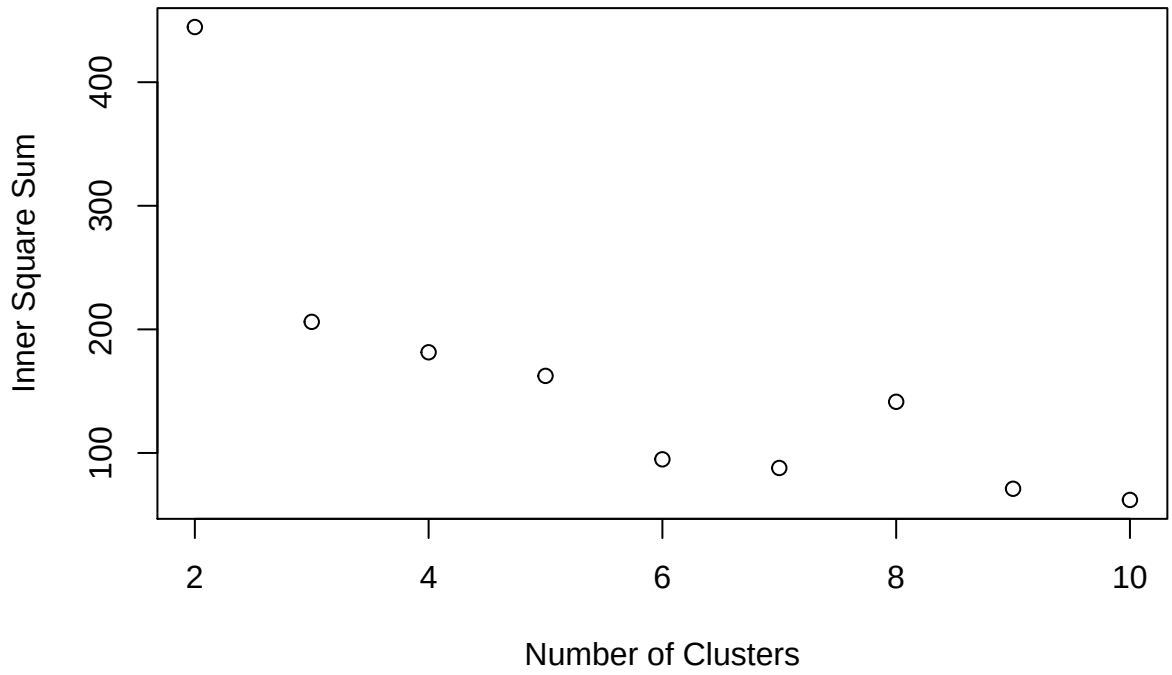
```
  data$pdias[is.na(data$pdias)] <- mean_pdias
  data$longindex[is.na(data$longindex)] <- mean_longindex

  data$pdias <- scale(data$pdias)
  data$longindex <- scale(data$longindex)

  return(data)
}

plants <- prepare_dataframe(plants)
```

```
plants <- subset(plants, select=c(pdias, longindex, insects, leafy))
```

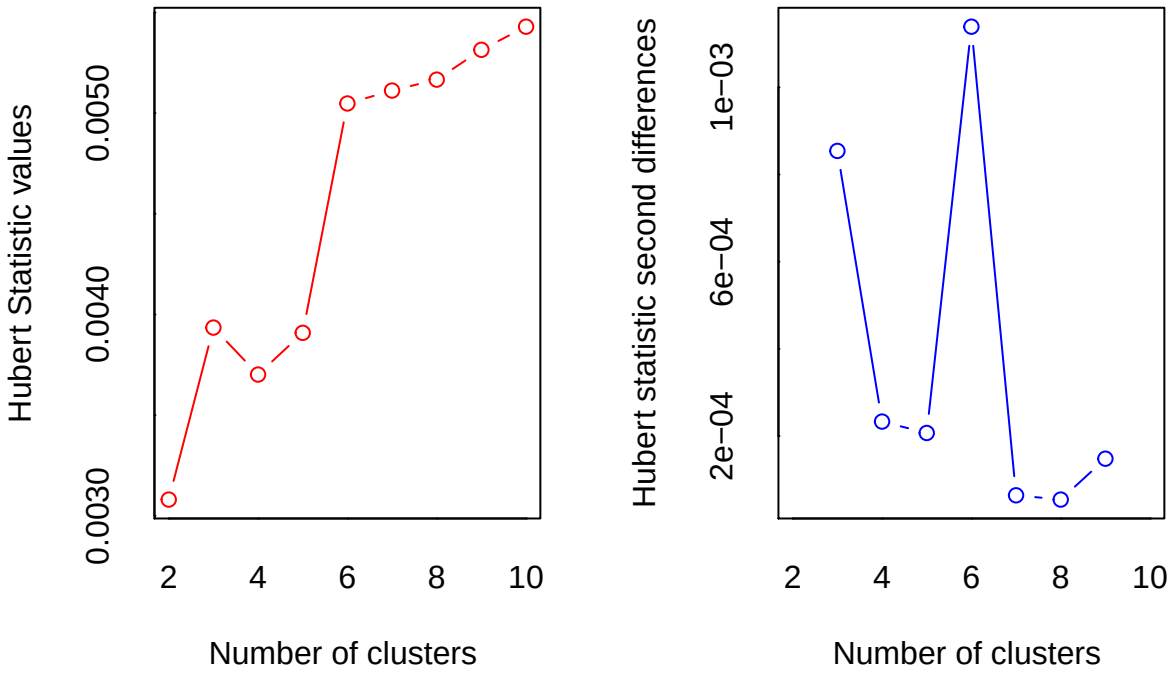
K-means

Find the last number of cluster that significantly decreases the error.

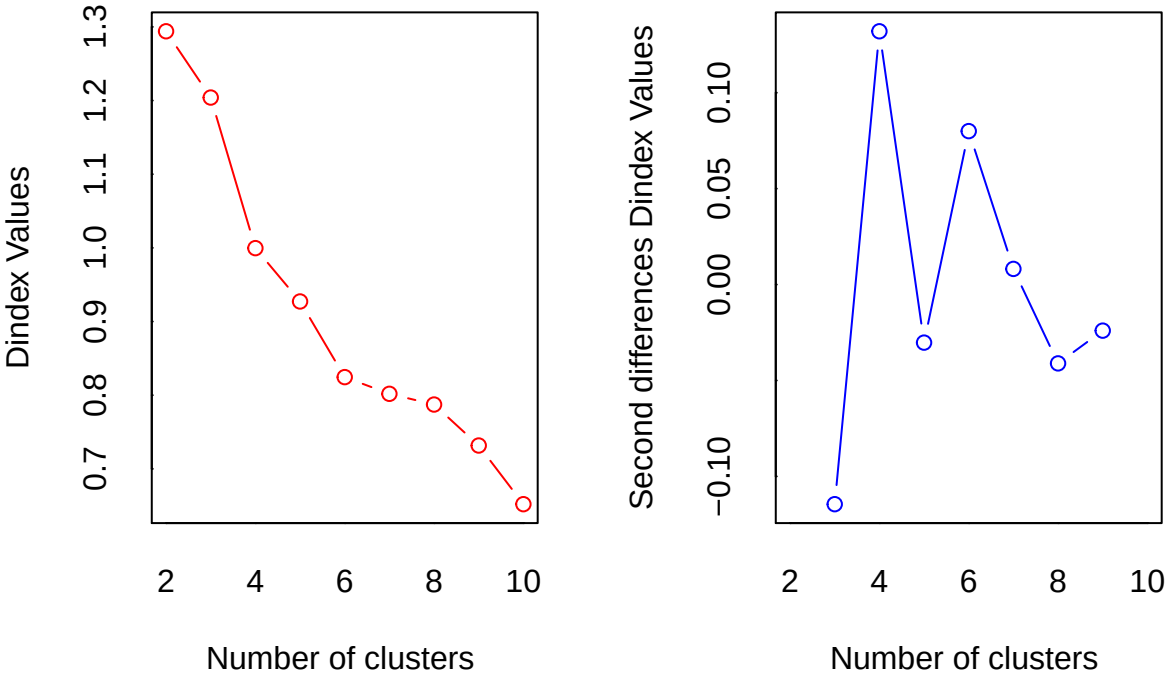
```
set.seed(1234)

cluster_num <- 2:10
inner_dists <- replicate(length(cluster_num), 0)
for (i in 1:length(cluster_num)) {
  model <- kmeans(plants, cluster_num[i])
  inner_dists[i] <- model[ 'tot.withinss' ]
}
plot(cluster_num, inner_dists, xlab="Number of Clusters", ylab="Inner Square Sum")
```




```
library(NbClust)
res <- NbClust(data = plants, distance = 'euclidean', min.nc = 2, max.nc = 10, 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:
## * 5 proposed 2 as the best number of clusters
## * 7 proposed 3 as the best number of clusters
## * 2 proposed 4 as the best number of clusters
## * 1 proposed 5 as the best number of clusters
## * 7 proposed 6 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *****************************************************************
```