Clustering (2020 Jul~Sep)

Hwang Seong-Yun

2022 9 14

SOM cluster

reference1 : https://data-make.tistory.com/91

reference2 : https://www.statmethods.net/advstats/cluster.html

```
water <- read.csv("C:/Users/HSY/Desktop/                /2020  7~9    .csv", sep=",", header=T)
water_name <- water[,1]
water <- water[,-1]
rownames(water) <- water_name
```

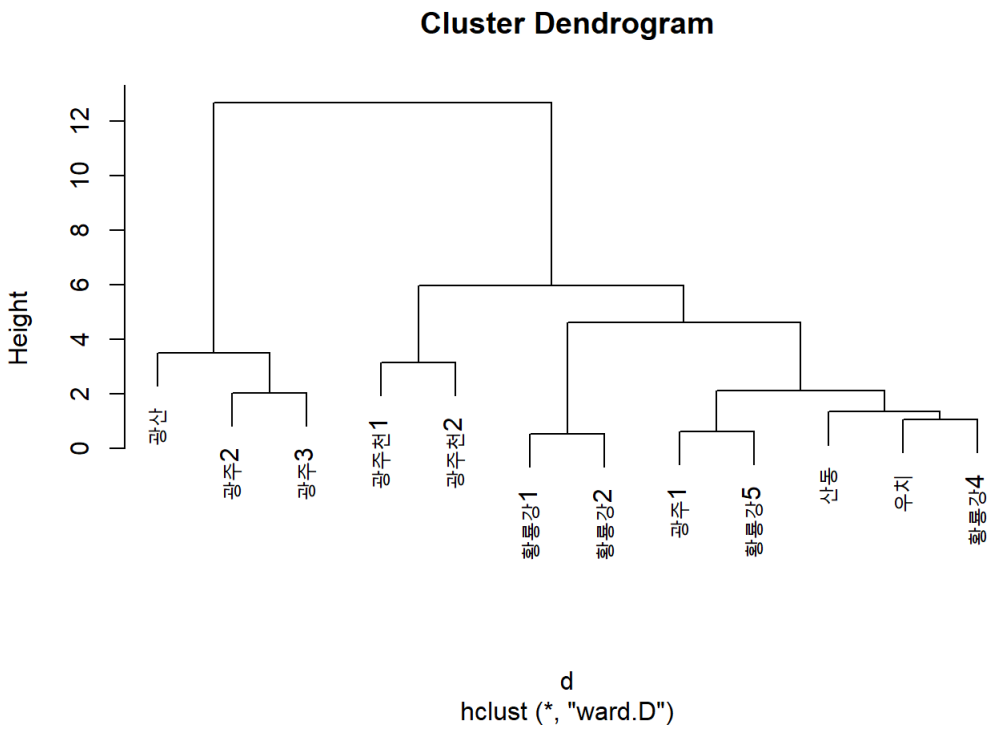
Distance matrix

```
water_scale <- scale(water)
d <- dist(water_scale, method="euclidean")
as.matrix(d)
```

```
##                      1     1     2     2     3
##       0.000000 1.245407 1.6191525 2.941653 3.079144 5.585021 4.328263
##       1.245407 0.000000 1.7676634 3.484646 3.176847 5.261175 4.076246
##   1   1.619153 1.767663 0.0000000 4.342995 3.512963 5.094870 4.081142
##   1 2.941653 3.484646 4.3429950 0.000000 3.158899 6.141986 4.797816
##   2 3.079144 3.176847 3.5129628 3.158899 0.000000 3.742468 2.700306
##   2   5.585021 5.261175 5.0948699 6.141986 3.742468 0.000000 2.043531
##   3   4.328263 4.076246 4.0811425 4.797816 2.700306 2.043531 0.000000
##   1 1.742728 2.633285 3.1131731 2.686601 4.014156 6.621235 5.463141
##   2 1.235760 2.119839 2.6154912 2.768595 3.736484 6.305887 5.143505
##   4 1.057289 1.303768 1.3683462 3.493088 3.532849 5.697739 4.473045
##   5 1.213004 1.501799 0.6294452 3.970434 3.493406 5.213621 4.189213
##       4.475317 4.659082 3.8798365 5.214110 3.231713 3.519801 2.750250
##            1      2     4     5
##       1.7427276 1.2357600 1.057289 1.2130039 4.475317
##       2.6332854 2.1198392 1.303768 1.5017995 4.659082
##   1   3.1131731 2.6154912 1.368346 0.6294452 3.879837
##   1 2.6866015 2.7685954 3.493088 3.9704336 5.214110
##   2 4.0141555 3.7364839 3.532849 3.4934056 3.231713
##   2   6.6212345 6.3058866 5.697739 5.2136208 3.519801
##   3   5.4631415 5.1435049 4.473045 4.1892134 2.750250
##   1 0.0000000 0.5462629 2.525125 2.5591158 5.747601
##   2 0.5462629 0.0000000 2.035674 2.0654661 5.416268
##   4 2.5251252 2.0356741 0.000000 1.1743290 4.321514
##   5 2.5591158 2.0654661 1.174329 0.0000000 4.196593
##       5.7476013 5.4162675 4.321514 4.1965933 0.000000
```

Apply Distance matrix model

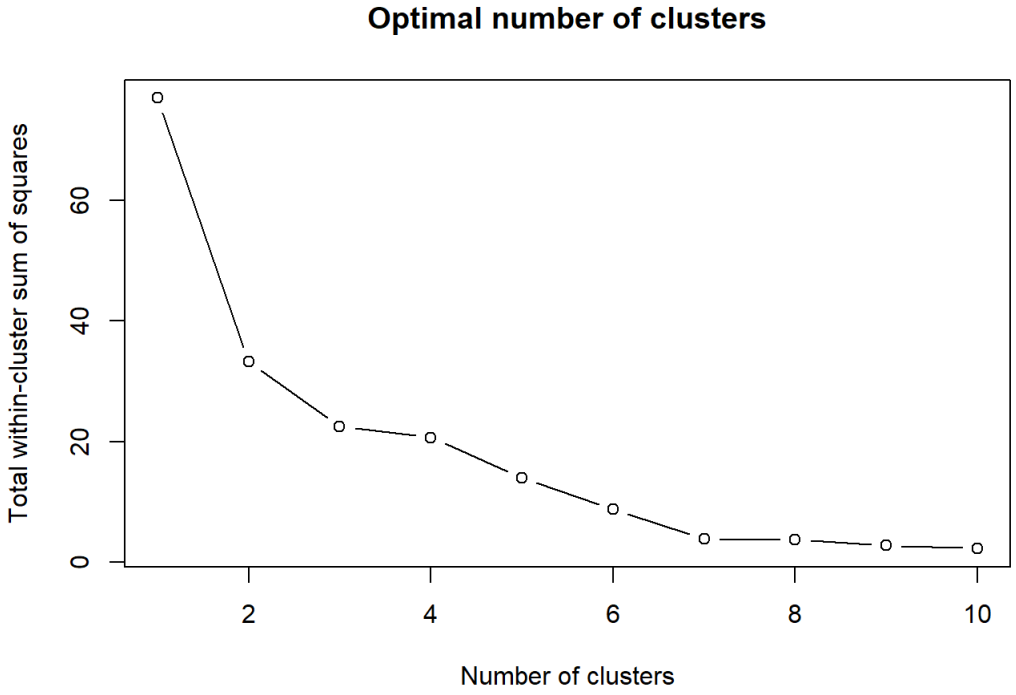
```
fit <- hclust(d, method="ward.D")
plot(fit)
```



Decide number of clusters

find the optimal number of clusters using Total within-cluster sum of squares

```
tot_withinss <- c()
for (i in 1:10){
  set.seed(1004) # for reproducibility
  kmeans_cluster <- kmeans(water_scale, centers = i, iter.max = 1000)
  tot_withinss[i] <- kmeans_cluster$tot.withinss}
plot(c(1:10), tot_withinss, type="b",
     main="Optimal number of clusters",
     xlab="Number of clusters",
     ylab="Total within-cluster sum of squares")
```



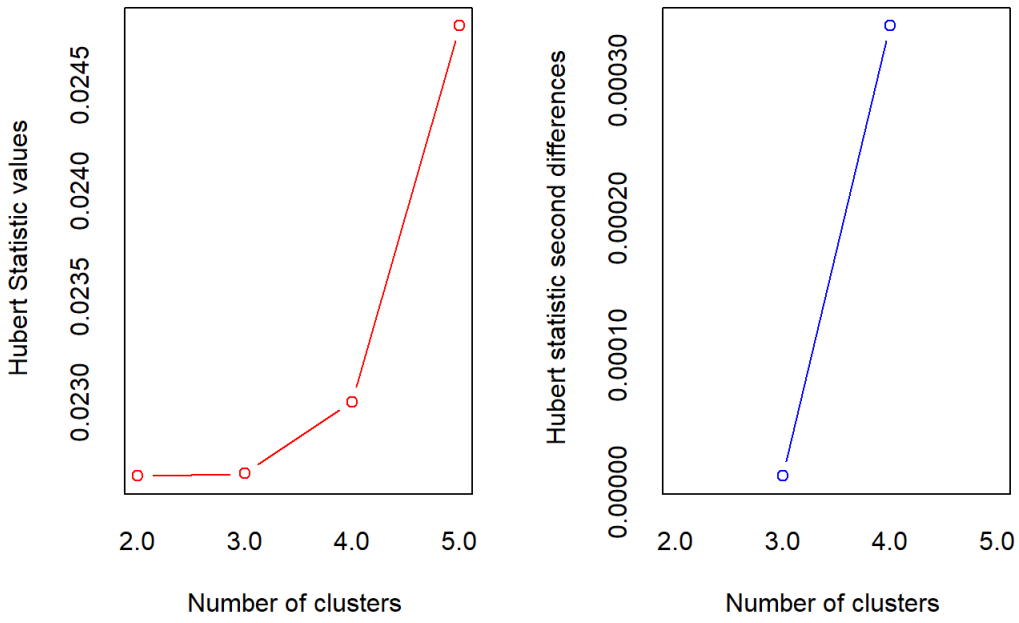
NbClust technique

```
library(NbClust)
```

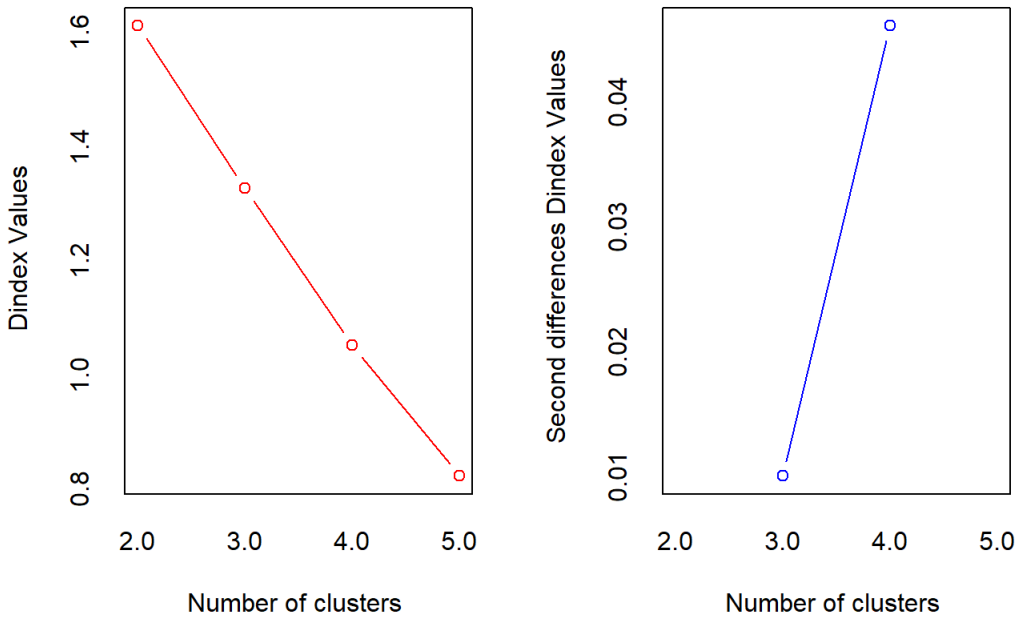
```
## Warning:    'NbClust'  R    4.1.3
```

```
nc <- NbClust(water_scale, distance="euclidean", method="ward.D",
              max.nc=5)
```

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

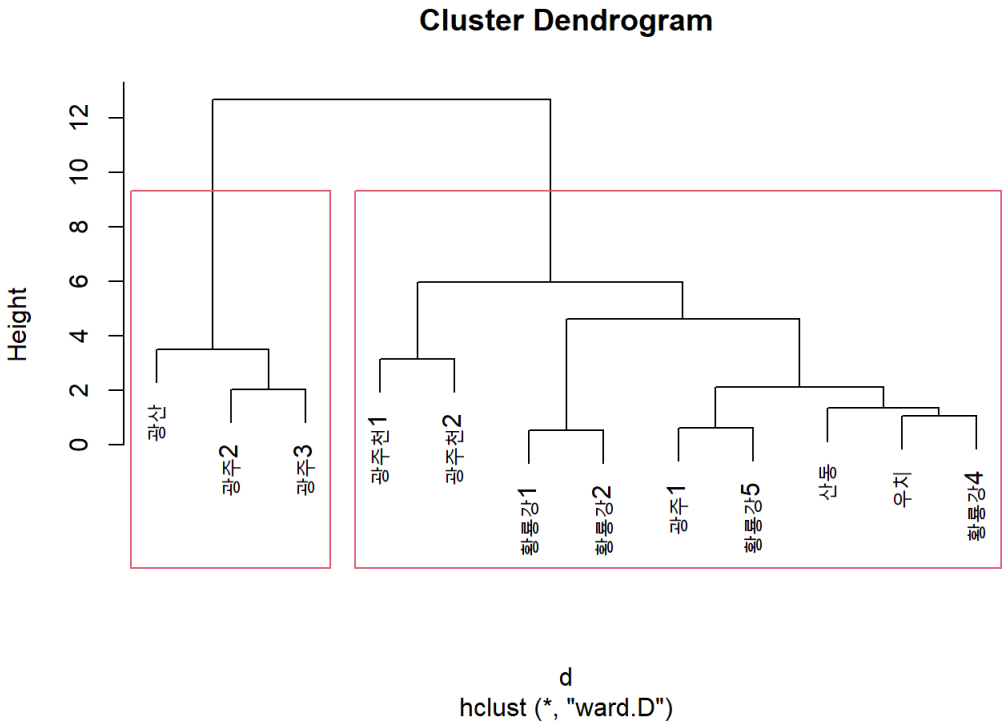



```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                In the plot of Hubert index, we seek a significant knee that corresponds to a
##                significant increase of the value of the measure i.e the significant peak in Hubert
##                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:
## * 9 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 5 proposed 5 as the best number of clusters
##
##                   ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

```
par(mfrow=c(1,1))
plot(fit)
rect.hclust(fit, k=2)
```



SOM cluster

```
library(SOMbrero)
```

```
## Warning:    'SOMbrero'  R    4.1.3
```

```
##               : igraph
```

```
## Warning:    'igraph'  R    4.1.2
```

```
##
##               : 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
##              : markdown
```

```
##
```

```
## ************************************************************
```

```
##
```

```
##       This is 'SOMbrero' package, v 1.4.1
```

```
##
```

```
## Citation details with citation('SOMbrero')
```

```
##
```

```
## Further information with help(SOMbrero)...
```

```
##
```

```
## Use sombreroGUI() to start the Graphical Interface.
```

```
##
```

```
## ************************************************************
```

```
library(kohonen)
```

```
## Warning:    'kohonen'  R    4.1.3
```

Normalization of data

```
water_scale <- data.frame(scale(water))
water_scale_matrix <- as.matrix(water_scale)
```

Training the SOM model

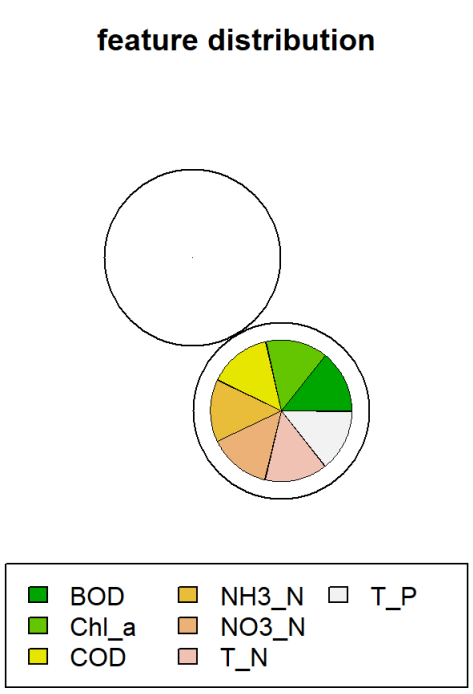
```
som_grid <- somgrid(xdim=1, ydim=2, topo="hexagonal")
som_model1 <- som(water_scale_matrix, grid=som_grid)
som_model2 <- trainSOM(x.data=water_scale, dimension=c(2,1),
                       nb.save=10, maxit=2000, scaling="none",
                       radius.type="letremy")
```

Visualization

```
table(som_model2$clustering)
```

```
##
## 1 2
## 4 8
```

```
plot(som_model1, main="feature distribution")
```




```
plot(som_model2, what="obs", type="names", print.title=T, scale=c(1,1))
```

```
## Warning in plot.somRes(som_model2, what = "obs", type = "names", print.title =
## T, : 'print.title' will be deprecated, please use 'show.names' instead
```