Clustering (2022 Jan~Mar)

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/              /2022  1~3    .csv", sep=",", header=T)
water_name <- water[,1]
water <- water[,-1]
rownames(water) <- water_name
```

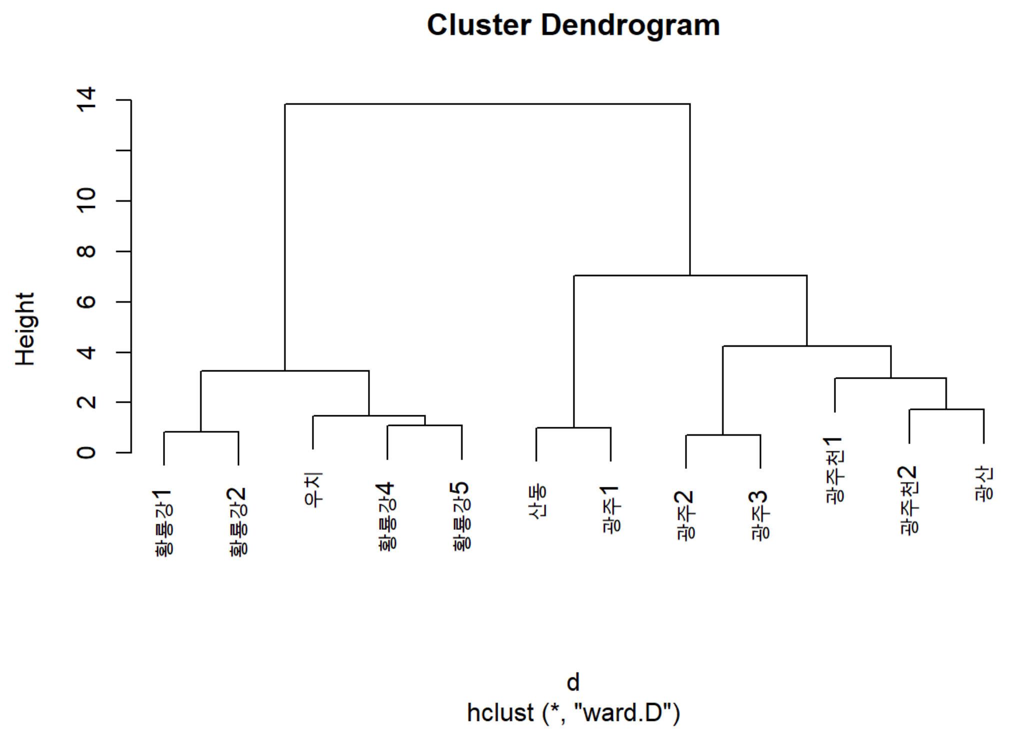
Distance matrix

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

```
##                          1     1     2       2       3
##       0.000000 3.3287189 2.4521894 3.676699 2.164418 4.1666223 3.9423763
##       3.328719 0.0000000 0.9987571 5.396733 2.851235 4.1469923 3.8417885
##   1    2.452189 0.9987571 0.0000000 4.776365 2.289749 3.8321497 3.5055820
##     1 3.676699 5.3967335 4.7763653 0.000000 2.965918 3.3728592 3.1665097
##     2 2.164418 2.8512354 2.2897489 2.965918 0.000000 3.2815589 2.9741302
##   2    4.166622 4.1469923 3.8321497 3.372859 3.281559 0.0000000 0.7211473
##   3    3.942376 3.8417885 3.5055820 3.166510 2.974130 0.7211473 0.0000000
##     1 2.927826 5.5626049 4.7240916 5.585957 5.045883 6.0782734 5.9617904
##     2 2.325973 5.2883872 4.3992623 4.902721 4.403877 5.6747837 5.5392744
##     4 1.588423 4.4198241 3.5184690 4.512582 3.626336 5.1382205 4.9417555
##     5 1.154780 3.7745976 2.8123322 4.026138 2.921601 4.2732392 4.0756814
##       2.769583 3.3536924 2.8145578 2.329468 1.712231 2.0110564 1.5346090
##              1       2     4     5
##       2.9278260 2.3259731 1.588423 1.154780 2.769583
##       5.5626049 5.2883872 4.419824 3.774598 3.353692
##   1    4.7240916 4.3992623 3.518469 2.812332 2.814558
##     1 5.5859570 4.9027211 4.512582 4.026138 2.329468
##     2 5.0458828 4.4038768 3.626336 2.921601 1.712231
##   2    6.0782734 5.6747837 5.138220 4.273239 2.011056
##   3    5.9617904 5.5392744 4.941756 4.075681 1.534609
##     1 0.0000000 0.8485064 1.553999 2.326272 5.139327
##     2 0.8485064 0.0000000 1.008206 1.812998 4.593280
##     4 1.5539987 1.0082061 0.000000 1.081911 3.913287
##     5 2.3262717 1.8129978 1.081911 0.000000 3.141883
##       5.1393267 4.5932801 3.913287 3.141883 0.000000
```

Apply Distance matrix model

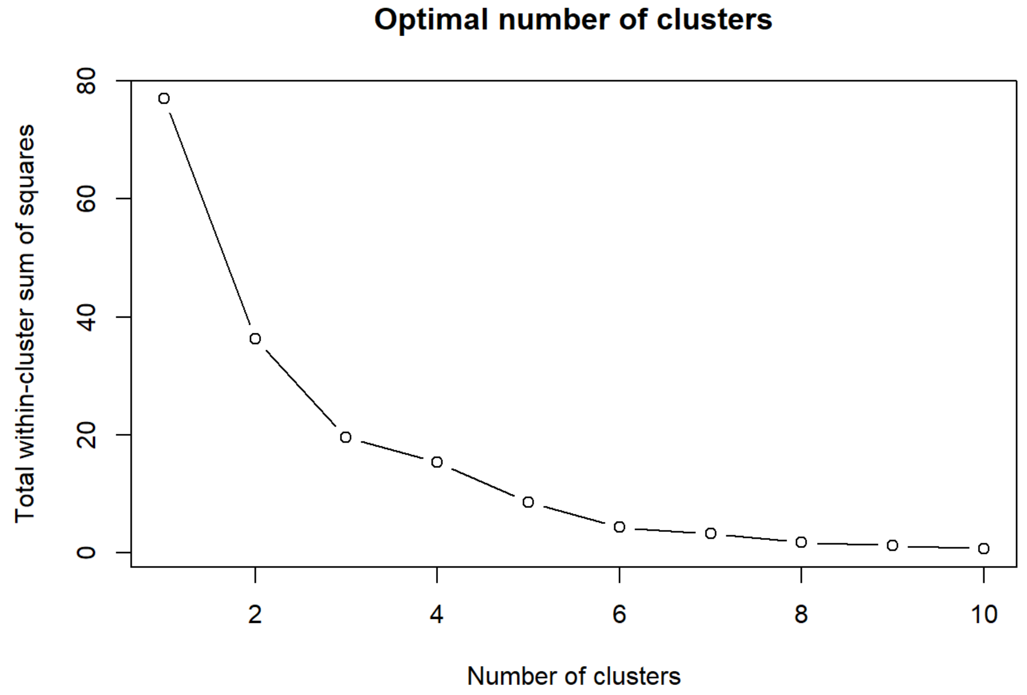
```
fit <- hclust(d, method="ward.D")
plot(fit)
```



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")
```

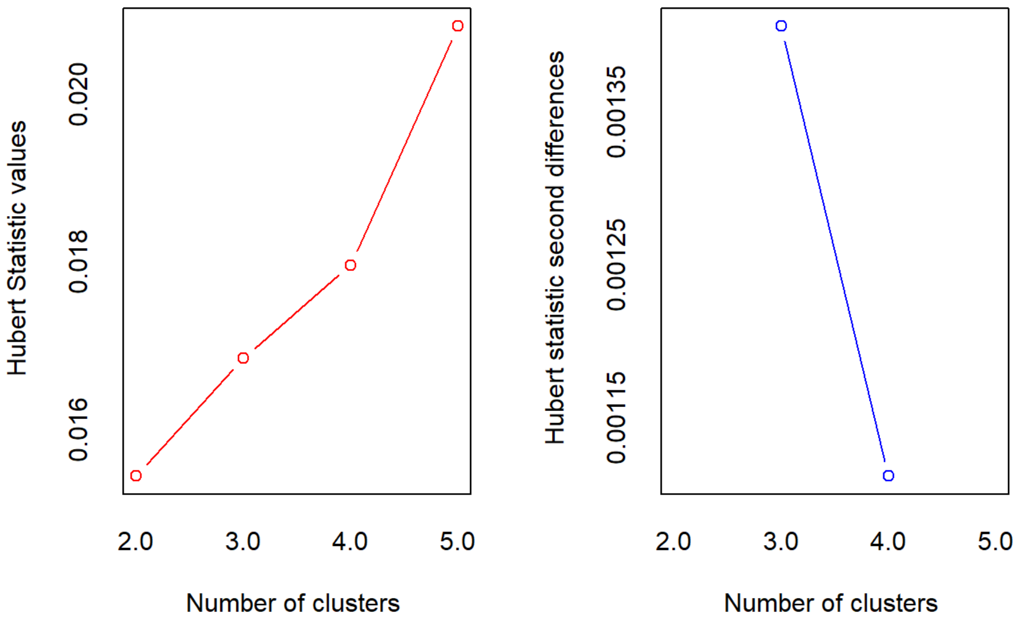


NbClust technique

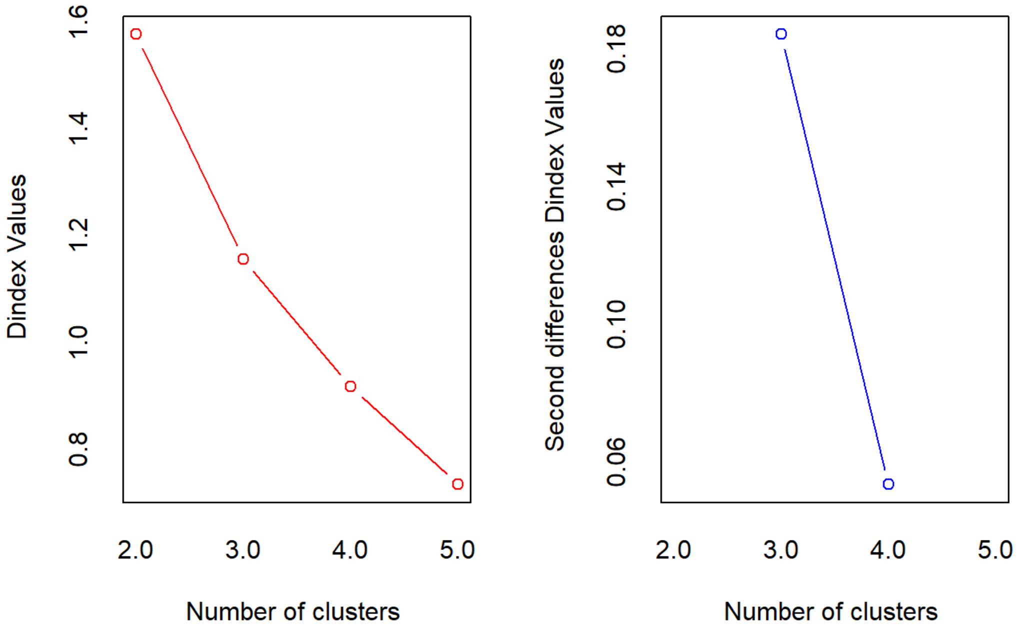
```
library(NbClust)
```

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

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

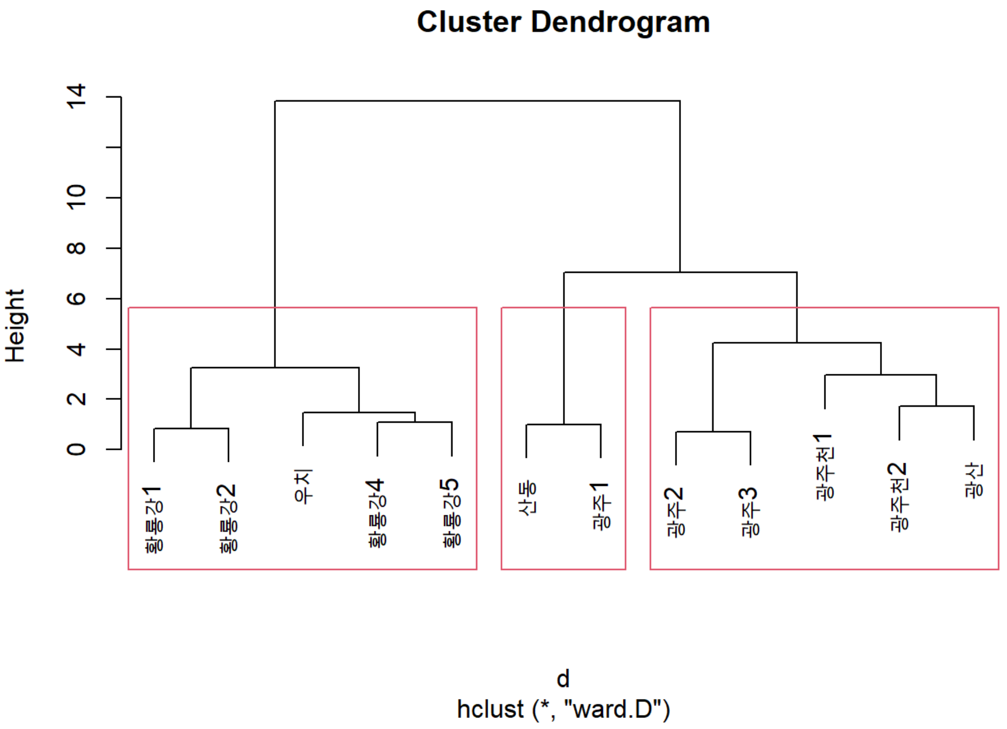


```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##       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

```
par(mfrow=c(1,1))
plot(fit)
rect.hclust(fit, k=3)
```



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

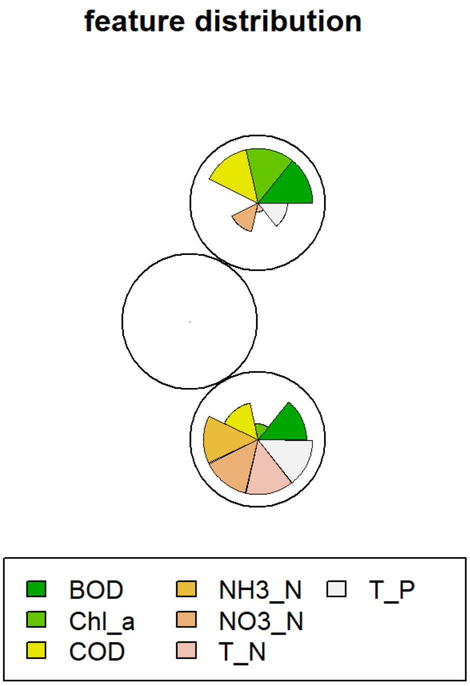
```
som_grid <- somgrid(xdim=1, ydim=3, topo="hexagonal")
som_model1 <- som(water_scale_matrix, grid=som_grid)
som_model2 <- trainSOM(x.data=water_scale, dimension=c(3,1),
                       nb.save=10, maxit=2000, scaling="none",
                       radius.type="letremy")
```

Visualization

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

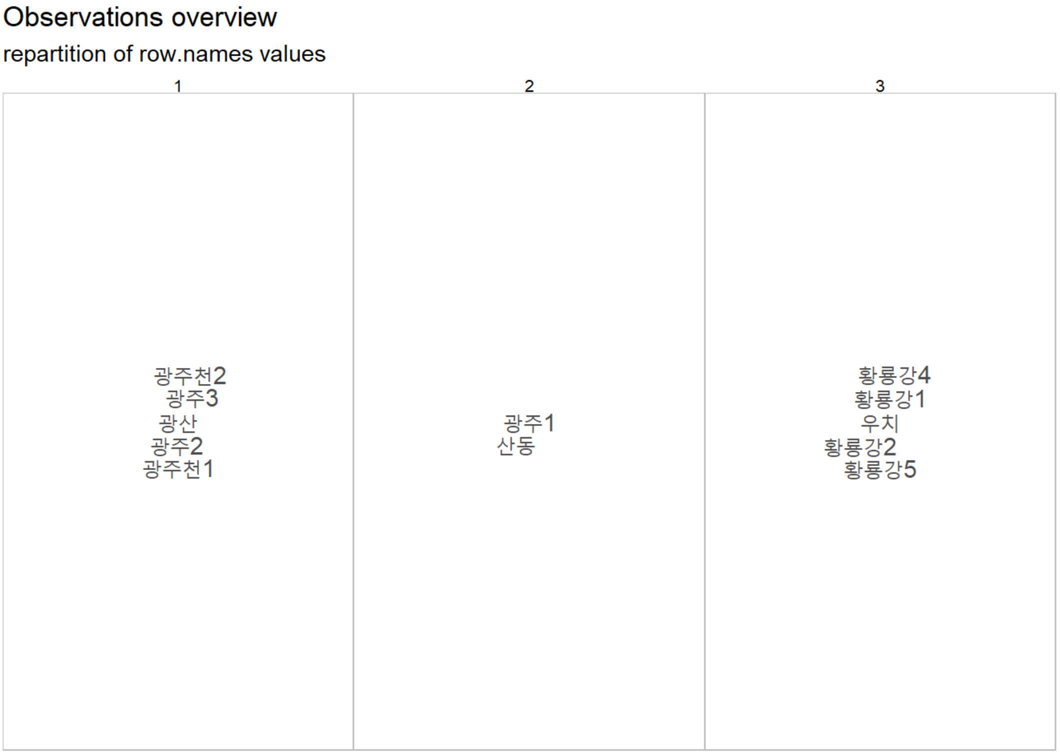
```
##
## 1 2 3
## 5 2 5
```

```
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
```


Observations overview
repartition of row.names values
광주천2
광주3
광산
광주2
광주천1
광주1
산동
황룡강4
황룡강1
우치
황룡강2
황룡강5