Clustering (2020 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/               /2020  1~3    .csv", sep=",", header=T)
water_name <- water[,1]
water <- water[,-1]
rownames(water) <- water_name
```

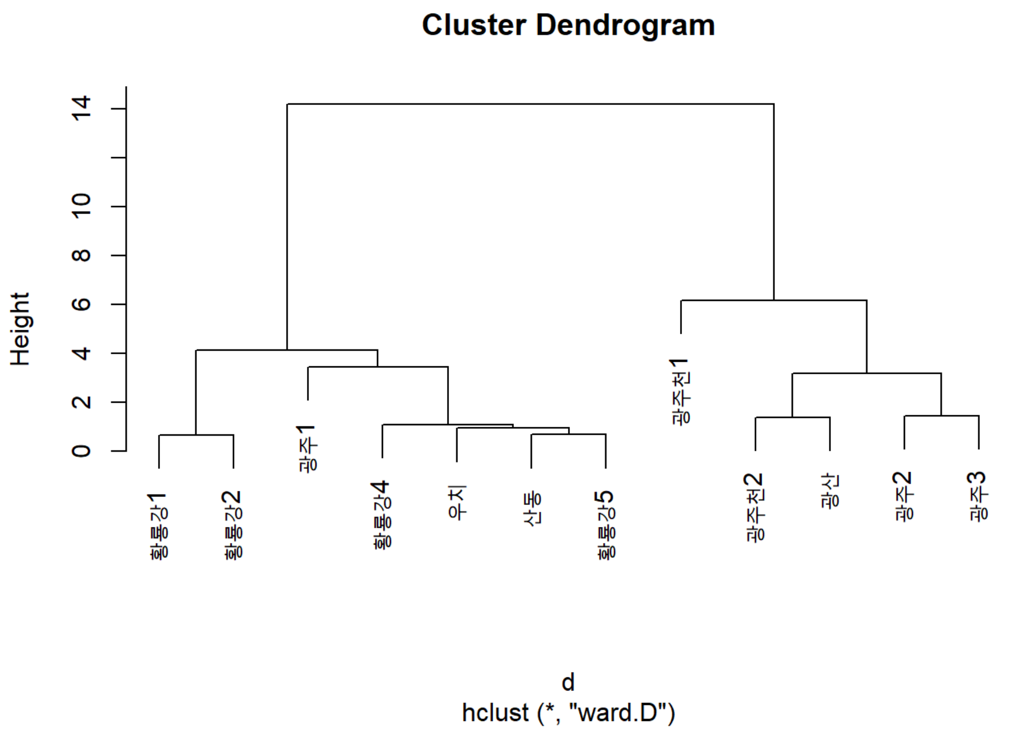
Distance matrix

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

```
##                          1     1     2     2     3
##       0.0000000 0.8842253 2.607582 4.118588 2.870907 4.779954 4.976605
##       0.8842253 0.0000000 2.496294 4.267416 2.544829 4.432041 4.787242
##   1    2.6075817 2.4962936 0.000000 5.303129 2.543686 4.381322 4.309680
##    1 4.1185885 4.2674161 5.303129 0.000000 3.904394 4.713814 5.251080
##    2 2.8709075 2.5448289 2.543686 3.904394 0.000000 2.782639 2.958605
##   2    4.7799538 4.4320409 4.381322 4.713814 2.782639 0.000000 1.453225
##   3    4.9766045 4.7872416 4.309680 5.251080 2.958605 1.453225 0.000000
##    1 1.9737094 2.3672227 4.070640 4.806865 4.708596 6.298948 6.643951
##    2 1.4459148 1.8066710 3.619840 4.655017 4.169473 5.877258 6.175887
##    4 1.1900748 1.0986154 2.577636 4.102314 3.111735 4.876312 5.310212
##    5 0.8755093 0.6865546 2.281624 4.485692 2.926041 4.748142 5.075161
##       3.6644875 3.3486963 2.936466 4.545220 1.392434 1.774625 1.701599
##             1      2      4      5
##       1.9737094 1.4459148 1.1900748 0.8755093 3.664488
##       2.3672227 1.8066710 1.0986154 0.6865546 3.348696
##   1    4.0706401 3.6198396 2.5776362 2.2816235 2.936466
##    1 4.8068648 4.6550167 4.1023145 4.4856920 4.545220
##    2 4.7085958 4.1694732 3.1117355 2.9260406 1.392434
##   2    6.2989481 5.8772578 4.8763121 4.7481419 1.774625
##   3    6.6439511 6.1758869 5.3102119 5.0751606 1.701599
##    1 0.0000000 0.6759965 1.8789352 2.0599895 5.422900
##    2 0.6759965 0.0000000 1.5281492 1.5846721 4.900505
##    4 1.8789352 1.5281492 0.0000000 0.7054777 3.900974
##    5 2.0599895 1.5846721 0.7054777 0.0000000 3.666358
##       5.4229001 4.9005053 3.9009740 3.6663582 0.000000
```

Apply Distance matrix model

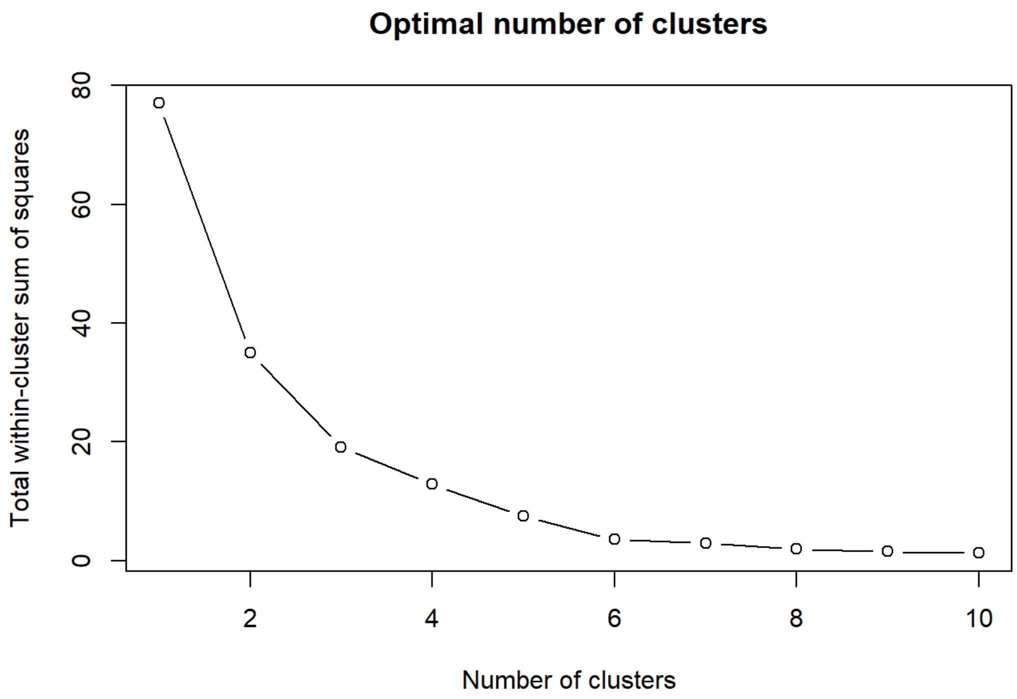
```
fit <- hclust(d, method="ward.D")
plot(fit)
```



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")
```

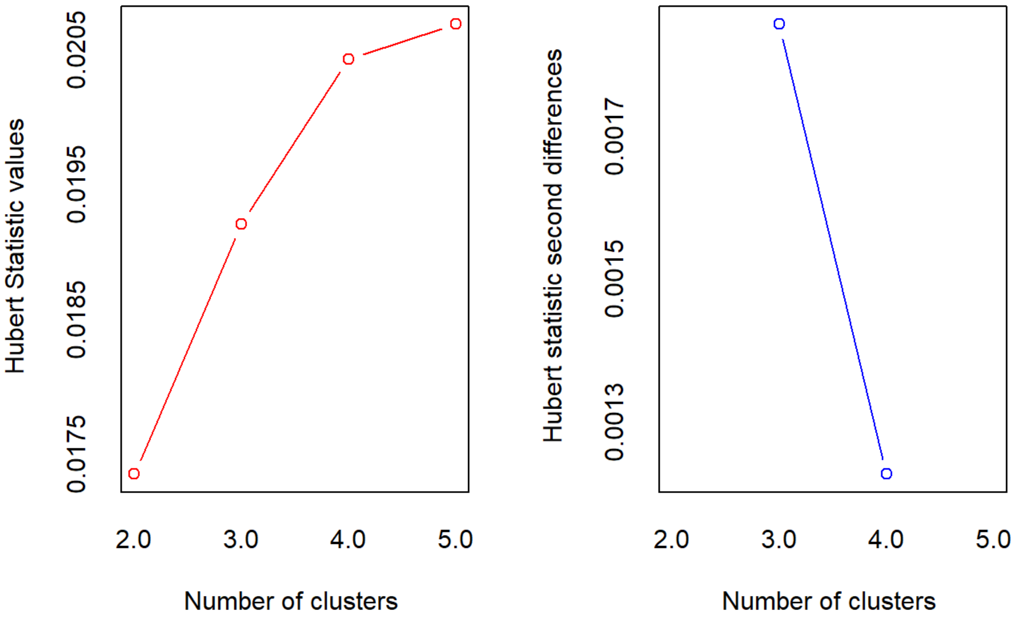


NbClust technique

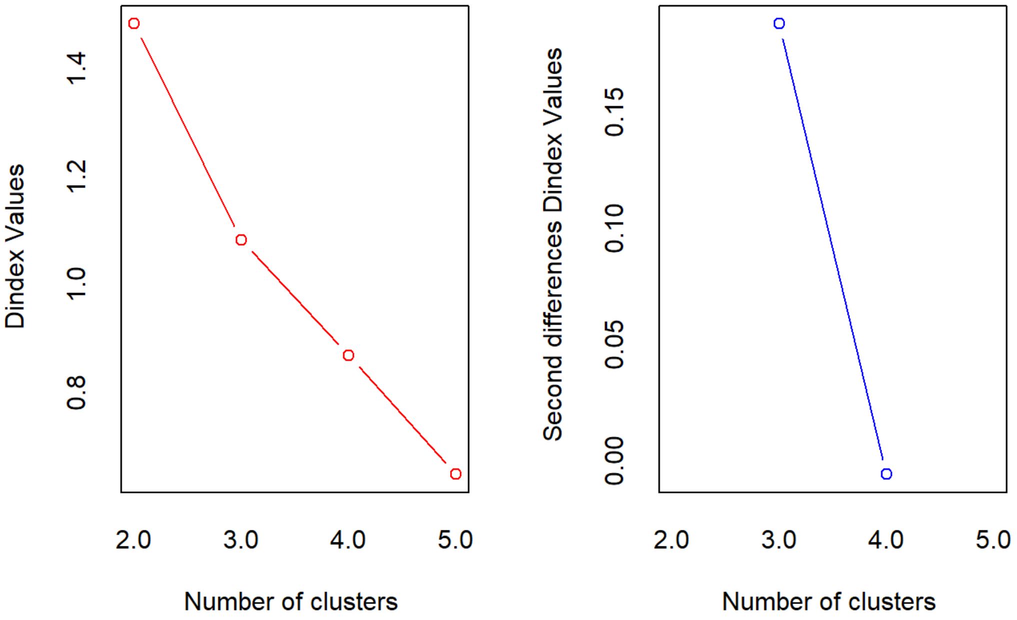
```
library(NbClust)
```

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

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

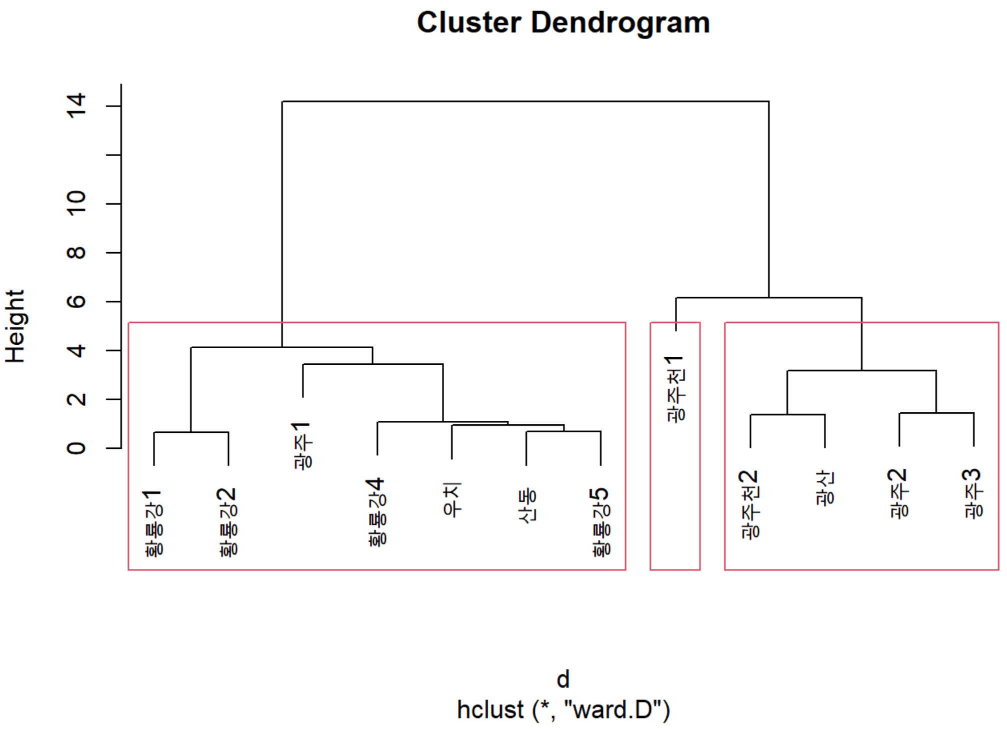


```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 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:
## * 5 proposed 2 as the best number of clusters
## * 11 proposed 3 as the best number of clusters
## * 7 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

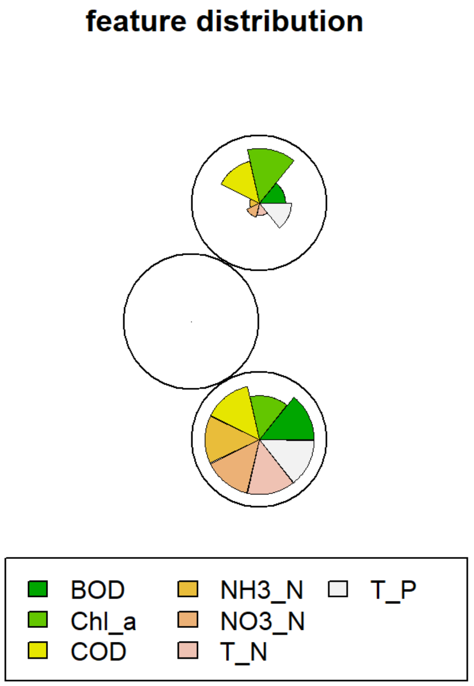
```
som_grid <- somgrid(xdim=1, ydim=3, topo="hexagonal")
som_model1 <- som(water_scale_matrix, grid=som_grid)
som_model2 <- trainSOM(x.data=water_scale, dimension=c(3,1),
                       nb.save=10, maxit=2000, scaling="none",
                       radius.type="letremy")
```

Visualization

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

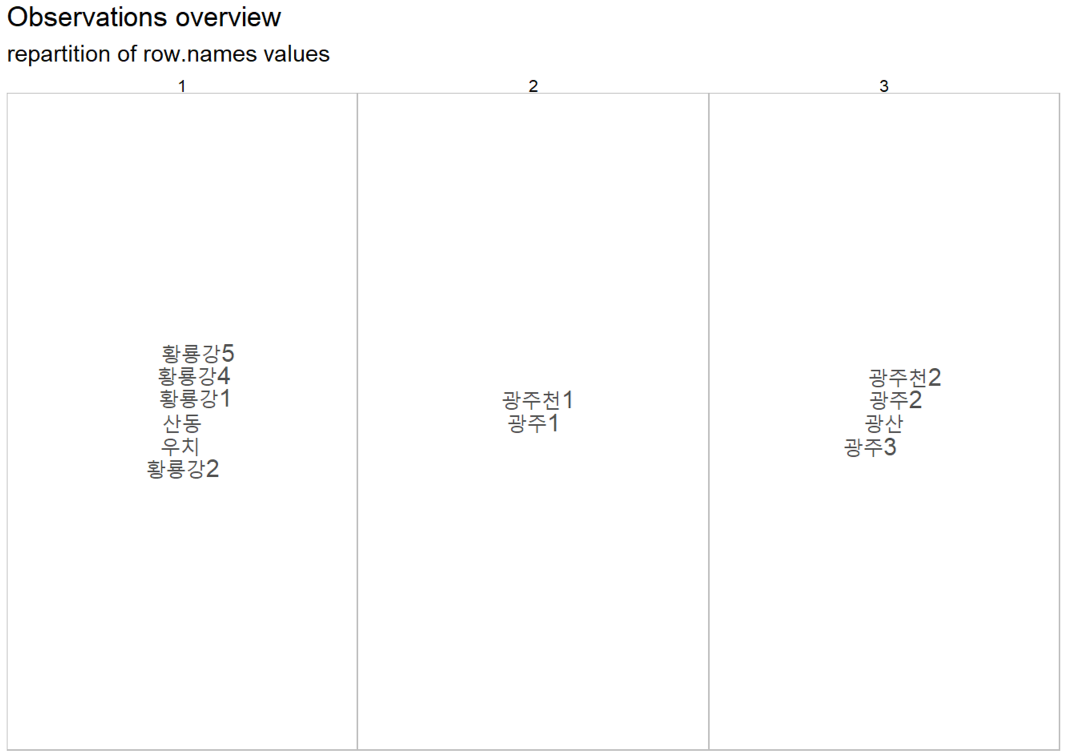
```
##
## 1 2 3
## 6 2 4
```

```
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
황룡강5
황룡강4
황룡강1
산동
우치
황룡강2
광주천1
광주1
광주천2
광주2
광산
광주3