Hierarchical Clustering

Kumar Rahul

23/05/2022

Hirarchial Clustering

More info at: https://uc-r.github.io/hc_clustering

```
#setwd("/Users/Rahul/Documents/Rahul Office/IIMB/Concepts/R/ML_using_R/R_code/R_clustering")
knitr::opts_knit$set(root.dir = "/Users/Rahul/Documents/Rahul Office/IIMB/Concepts/R/ML_using_R/R_code/
```

Preparing Data

Read data from a specified location

```
beer_df = read.csv('./data/Hclust_Beer.csv',header = TRUE,sep = ",",
                     na.strings = c(""," ", "NA"), row.names = c(2))

beer_df = beer_df[,-c(1)]
```

```
head(beer_df)
```

```
##             CAL SOD ALC COST
## Budweiser   144  15 4.7 0.43
## Schlitz     151  19 4.9 0.43
## Lowenbrau   157  15 4.9 0.48
## Kronenbourg 170   7 5.2 0.73
## Heineken    152  11 5.0 0.77
## Old Mil     145  23 4.6 0.28
```

Summary of the data

Summary of the data on which model is built and Standardizing the variables

```
str(beer_df)
```

```
## 'data.frame':    20 obs. of  4 variables:
##  $ CAL : int  144 151 157 170 152 145 175 149 99 113 ...
##  $ SOD : int  15 19 15 7 11 23 24 27 10 8 ...
##  $ ALC : num  4.7 4.9 4.9 5.2 5 4.6 5.5 4.7 4.3 3.7 ...
##  $ COST: num  0.43 0.43 0.48 0.73 0.77 0.28 0.4 0.42 0.43 0.44 ...
```

```
beer_df = na.omit(beer_df)
beer_scaled = scale(beer_df[,c(1:4)])
```

Hirerchial clustering - agnes package

Using euclidean as distance measure for hirerchial clustering. Note that if using hclust(), the dist matrix returned by dist() is used.

```r
#beer_dist = dist(beer_scaled, method = "euclidean") # distance matrix
```

With the agnes package we can also get the agglomerative coefficient, which measures the amount of clustering structure found (values closer to 1 suggest strong clustering structure)

```r
m = c( "average", "single", "complete", "ward")
names(m) = c( "average", "single", "complete", "ward")

# function to compute coefficient
ac = function(x) {
  agnes(beer_scaled, method = x)$ac
}

map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.7660820 0.5930611 0.8328273 0.8767939
```

On this data set, Ward's method identifies the strongest clustering structure of the four methods assessed. Thus applying ward method for further evaluation.

Dendrogram Plot

Plot using dendrogram plot to visulaize the clusters. In the dendrogram, each leaf corresponds to one observation. As we move up the tree, observations that are similar to each other are combined into branches, which are themselves fused at a higher height.

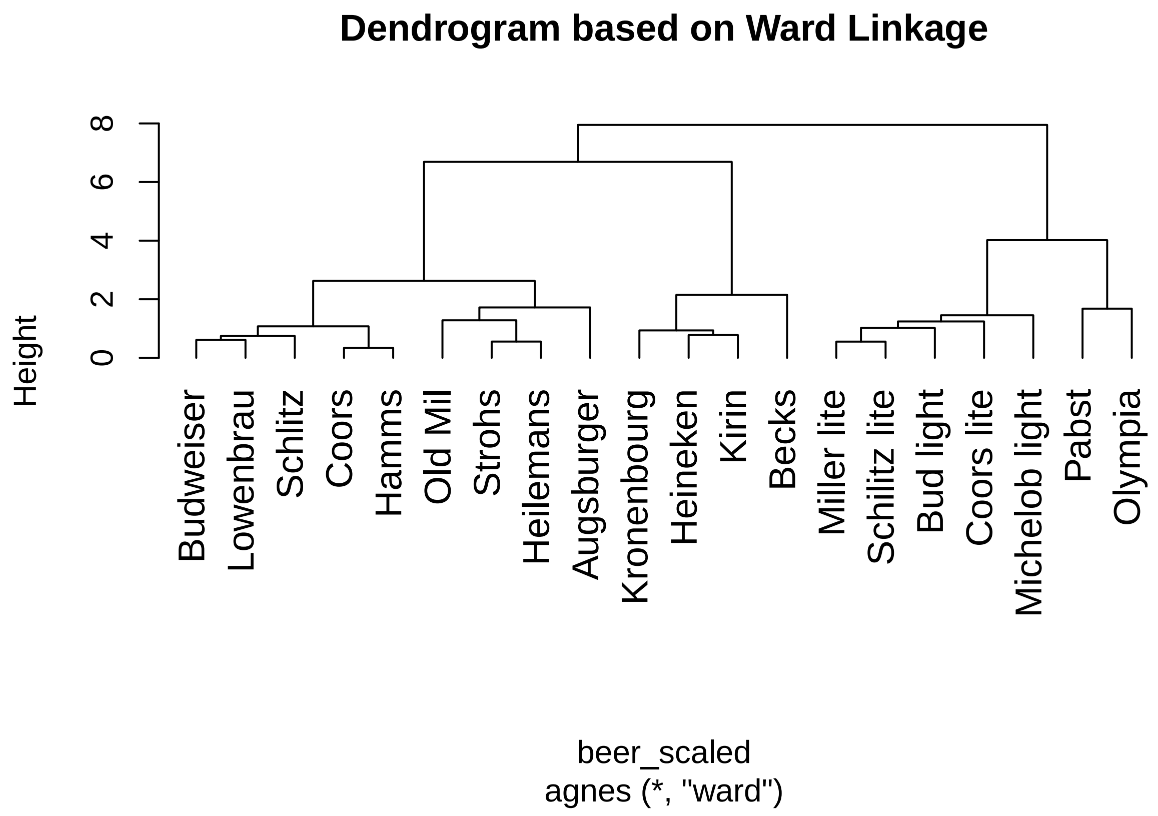
The height of the fusion, provided on the vertical axis, indicates the (dis)similarity between two observations. The higher the height of the fusion, the less similar the observations are. Note that, conclusions about the proximity of two observations can be drawn only based on the height where branches containing those two observations first are fused. We cannot use the proximity of two observations along the horizontal axis as a criteria of their similarity.

https://stats.stackexchange.com/questions/109949/what-algorithm-does-ward-d-in-hclust-implement-if-it-is-not-wards-criterion

Note that agnes(*, method=“ward”) corresponds to hclust(*, “ward.D2”). More at ?hclust.

```r
options(repr.plot.width=15, repr.plot.height=10)
```

```r
beer_hclust1 = agnes(beer_scaled, method = "ward")
pltree(beer_hclust1, cex = 1.2, hang = -1, main = "Dendrogram based on Ward Linkage")
```



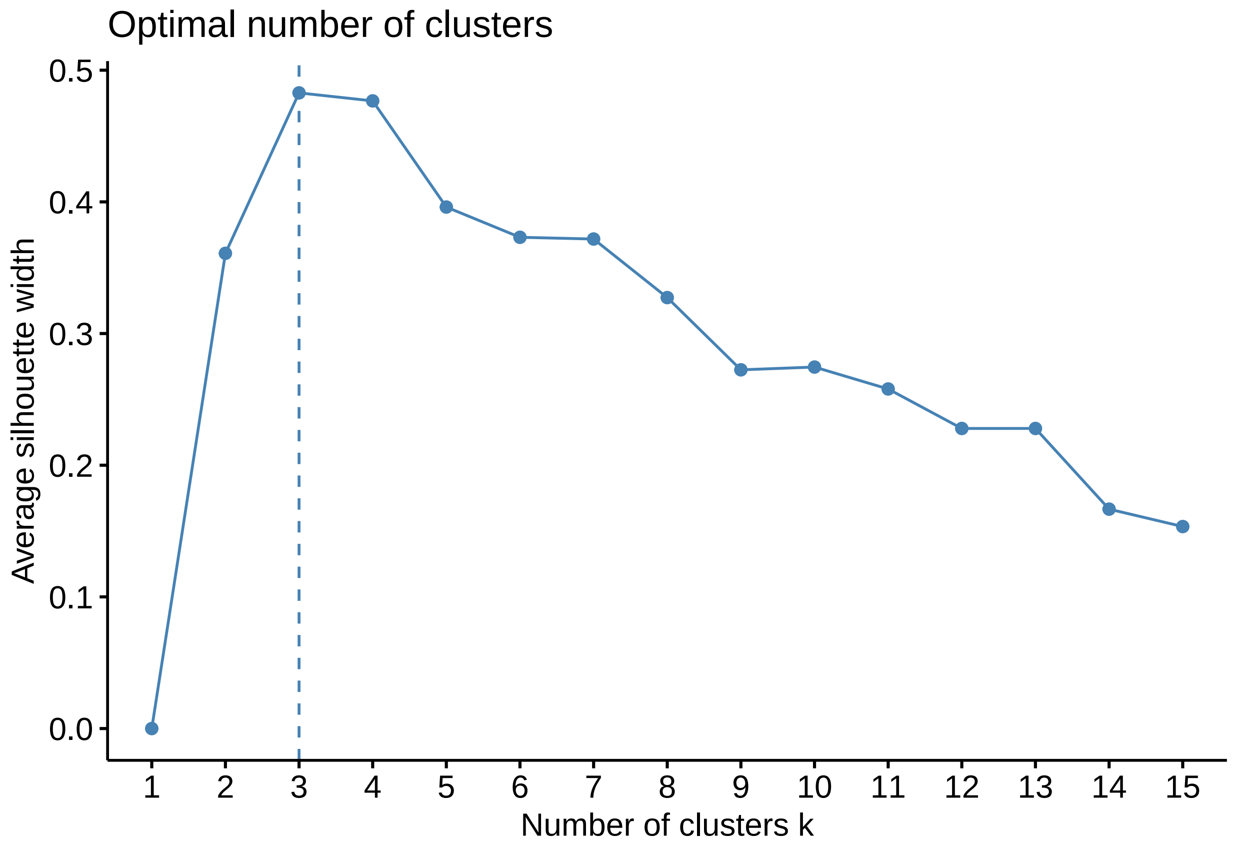
The height of the cut to the dendrogram controls the number of clusters obtained. It plays the same role as the k in k-means clustering. To identify the optimal k, we can use average silhoutee width or wss as metric.

Optimal cluster - Average Silhouette width

Using average silhouette width as the metric, the optimal number of clusters k is the one that maximizes the average silhouette over a range of possible values for k.

The results show that 2 clusters maximize the average silhouette values with 3 clusters coming in as second optimal number of clusters.

```
fviz_nbclust(beer_scaled, FUNcluster = hcut,hc_func = "agnes",hc_method = "ward.D2",hc_metric = "euclid
             method = "silhouette", k=15)
```

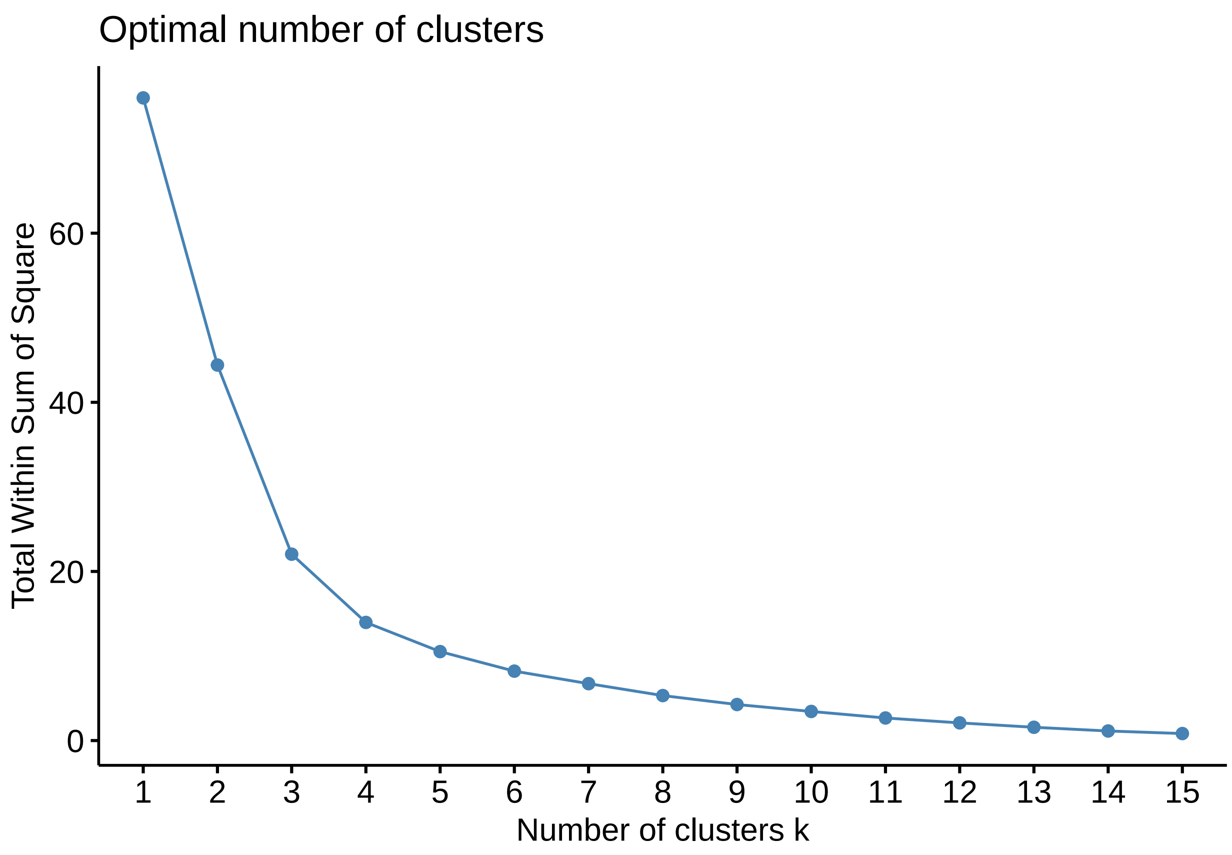


Optimal Cluster - Elbow method (wss)

The total within-cluster sum of square (wss) measures the compactness of the clustering and we want it to be as small as possible. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters.

Result shows k = 3 to be the optimal.

```
fviz_nbclust(beer_scaled, FUNcluster = hcut,hc_func = "agnes",hc_method =
             "ward.D2",hc_metric = "euclidean", method = "wss", k=15)
```



In order to identify sub-groups (i.e. clusters), we can cut the dendrogram with cutree. We will set k =3

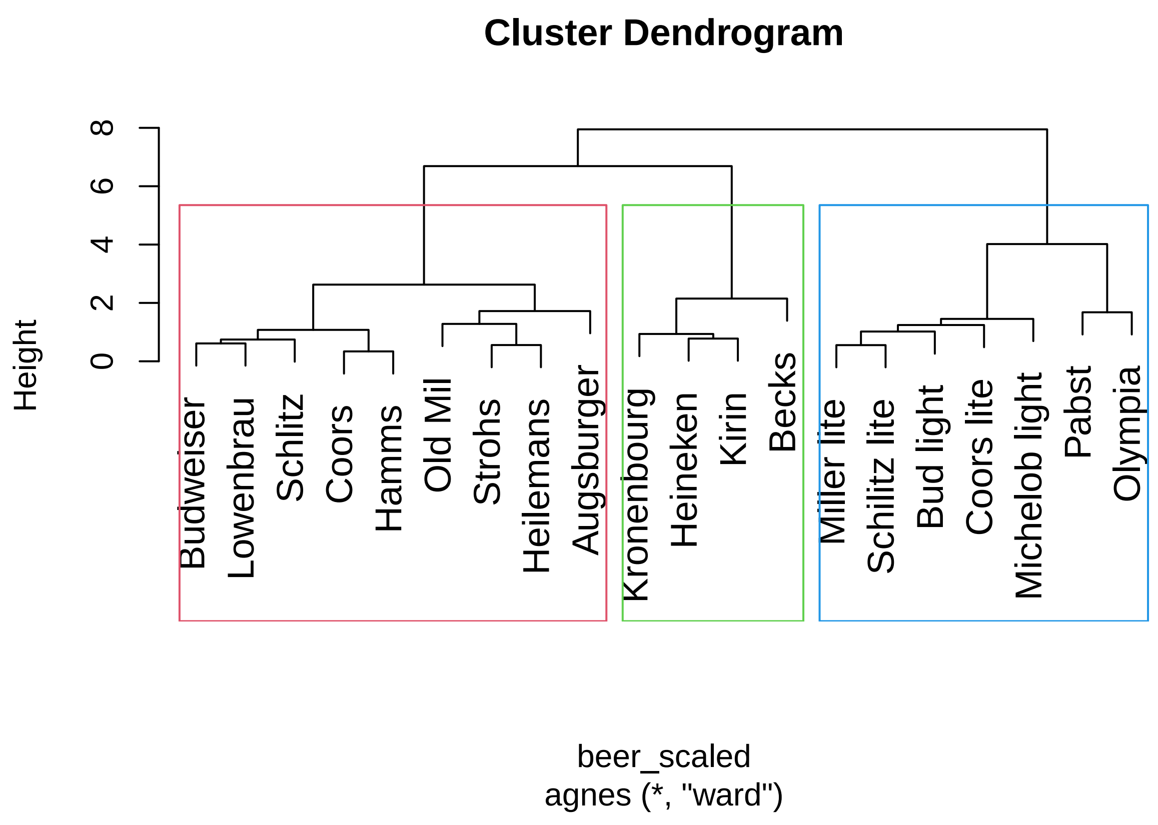
```
# Ward's method
# beer_hclust = hclust(beer_dist, method = "ward.D2" )
k = 3
beer_hclust2 = as.hclust(agnes(beer_scaled, method = "ward"))
#cutree((beer_hclust2), k = k)


# Cut tree into 4 groups
sub_grp = cutree(beer_hclust2, k = k)

# Number of members in each cluster
table(sub_grp)
```

```
## sub_grp
## 1 2 3
## 9 4 7
```

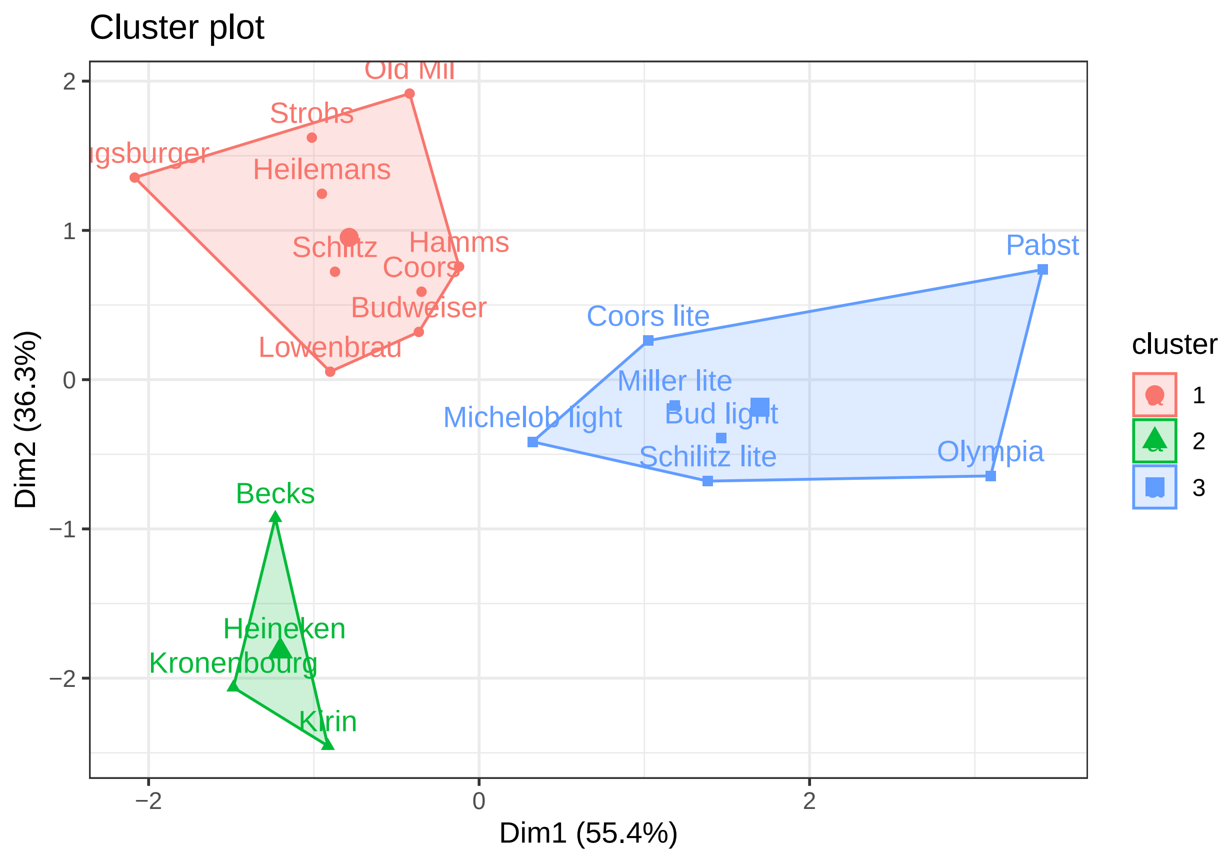
Visualize cluster

We can visualize the cluster as below:

```
plot(beer_hclust2, cex = 1.2)
rect.hclust(beer_hclust2, border = 2:4,cluster = sub_grp, k=k)
```



```
fviz_cluster(list(data = beer_scaled, cluster = sub_grp)) +
theme_bw()
```



The cluster can be tagged to the original dataframe as below:

```
beer_df %>%
mutate(cluster = sub_grp, name = row.names(beer_df)) %>%
head(10)
```

```
##    CAL SOD ALC COST cluster        name
## 1  144  15 4.7 0.43       1   Budweiser
## 2  151  19 4.9 0.43       1     Schlitz
## 3  157  15 4.9 0.48       1   Lowenbrau
## 4  170   7 5.2 0.73       2 Kronenbourg
## 5  152  11 5.0 0.77       2    Heineken
## 6  145  23 4.6 0.28       1     Old Mil
## 7  175  24 5.5 0.40       1  Augsburger
## 8  149  27 4.7 0.42       1      Strohs
## 9   99  10 4.3 0.43       3 Miller lite
## 10 113   8 3.7 0.44       3   Bud light
```

```
beer_df %>%
mutate(cluster = sub_grp, name = row.names(beer_df)) %>%
write.csv("beer_hirarchial_output.csv")
```