Exercise 3 solution

Isabelle Cretton

Oct. 1st, 2024

```
# Set global code chunk options
knitr::opts_chunk$set(warning = FALSE)
```

```
library(cluster)
library(stats)
par(mfrow = c(1, 1))
```

Problem 1 (Clustering)

1.A

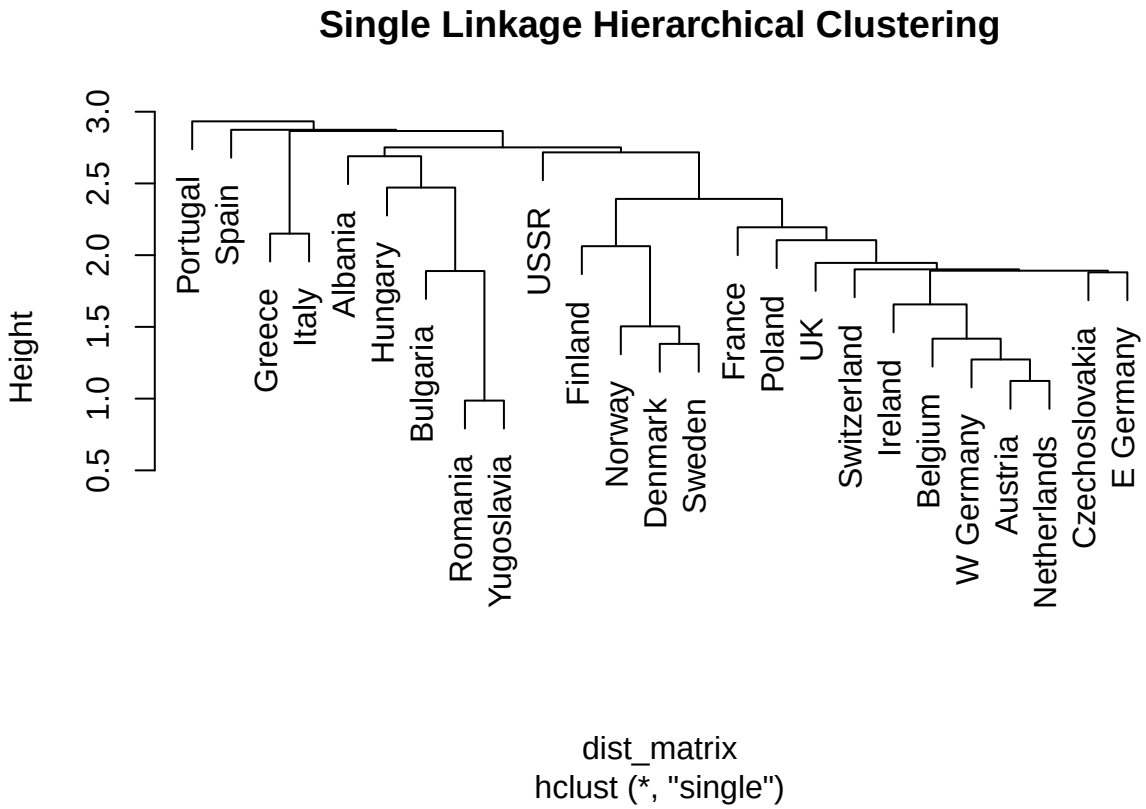
Prepare the data for clustering

```
# Load data
protein <- read.csv("~/Desktop/SEMESTER_5/STAT-MODELING/EXERCISES/StatModelEx/day3/data/protein.txt", s
row.names(protein) <- protein$Country
protein <- protein[, -1]  # Remove the country names column
protein <- scale(protein)  # Normalize the data

# Calculate the distance matrix
dist_matrix <- dist(protein)
```

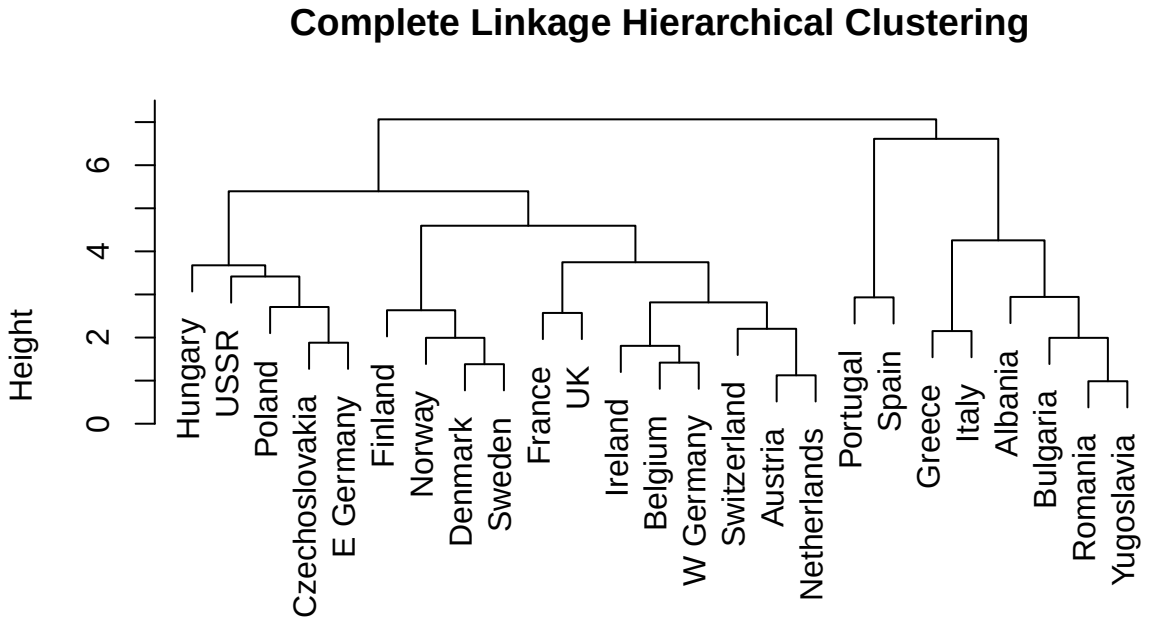
cluster analysis

```
# single linkage
single_link <- hclust(dist_matrix, method = "single")
plot(single_link, main = "Single Linkage Hierarchical Clustering")
```



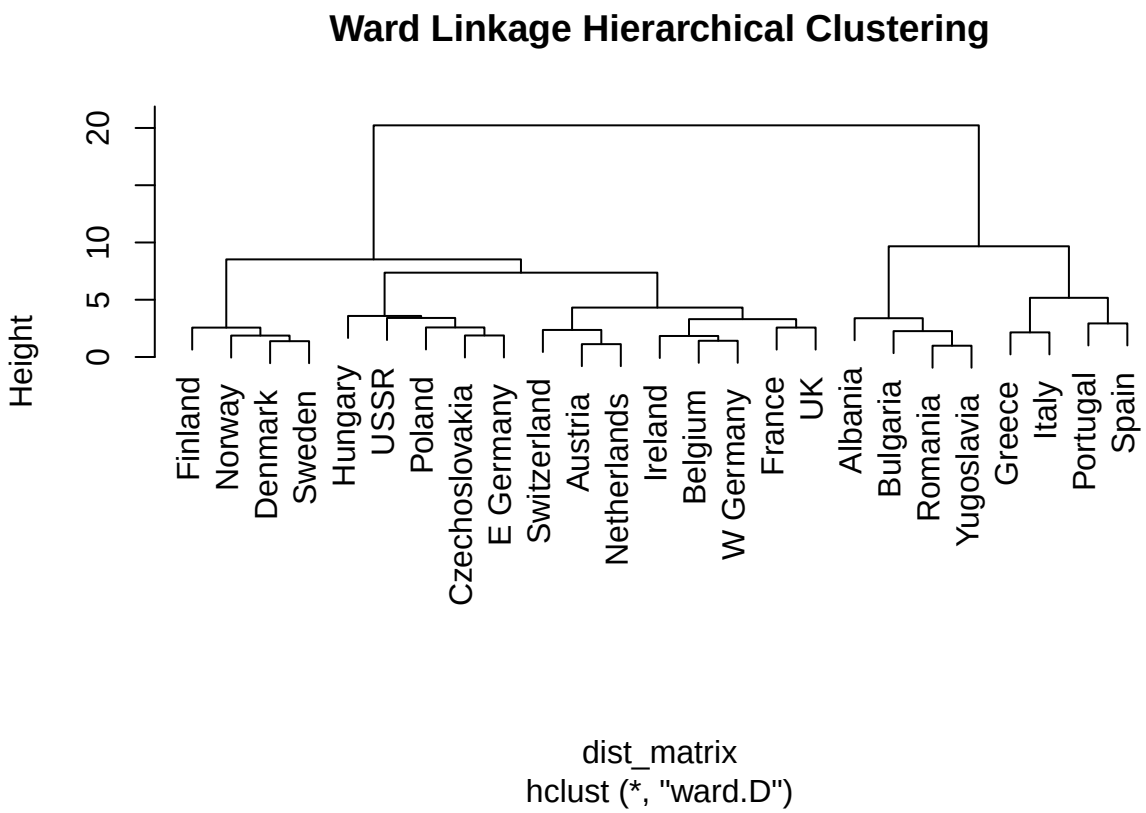
This method tends to create clusters by linking individual observations that are closest to each other, which can result in chains that are not indicative of actual clusters in the data. This method is sensitive to outliers as they can distort the shape of the dendrogram. For instance, countries like Romania and Yugoslavia are linked very early, suggesting minimal distance between their protein consumption patterns compared to other countries, but the larger structure suggests several longer chains that may not capture more meaningful groupings.

```
# complete linkage
comp_link <- hclust(dist_matrix, method = "complete")
plot(comp_link, main = "Complete Linkage Hierarchical Clustering")
```



Here, complete linkage hierarchicalclustering creates more balanced clusters compared to single linkage because it considers the maximum distance between observations in each cluster before combining them. This approach minimizes the influence of outliers and tends to produce more compact clusters. Clusters are more uniformly distributed, and countries that share more similarities in diet might group together more intuitively (eastern European countries like Czechoslovakia and East Germany are grouped)

```
# ward linkage
ward_link <- hclust(dist_matrix, method = "ward.D")
plot(ward_link, main = "Ward Linkage Hierarchical Clustering")
```



The dendrogram from the Ward Linkage Hierarchical Clustering presents a clear structure of how countries are grouped based on their dietary patterns. This clustering strategy, which minimizes within-cluster variance, has effectively revealed some distinct clusters that reflect regional dietary similarities and differences.

1.B

Yes, it is possible to cut the tree at a specific height to create clusters that can be interpreted meaningfully. By choosing an appropriate height to segment the dendrogram, we can form groups of countries with similar dietary protein patterns. Among the hierarchical clustering methods (single, complete, and Ward's linkage) Ward's linkage is recommended for deriving representative classifications. It minimizes within cluster variance, resulting in well-defined & compact clusters that are easier to interpret and distinguish.
Cutting the Dendrogram We would look for a significant vertical gap in the dendrogram produced by Ward's method, which indicates a natural division between clusters.

1.C

similarities: both PCA and hierarchical clustering help to reduce the complexity of the data. PCA does this by transforming the data into principal components that capture the most variance, while hierarchical clustering groups similar data points, simplifying the dataset's structure.

dissimilarities: the output of PCA is a set of principal components that each represent a combination of features, with components ranked by their ability to explain the variance in the data. Hierarchical clustering results in a dendrogram that illustrates how individual entries are grouped into clusters, providing a categorical classification rather than a continuous spectrum.

Problem 2 (my.kmeans)

```
my.kmean <- function(x, k, iter = 10) {
    set.seed(111)
    centroids <- sample(x, k, replace = FALSE)
    clusters <- numeric(length(x))

    for (i in 1:iter) {
        for (j in 1:length(x)) {
            distances <- abs(x[j] - centroids)
            clusters[j] <- which.min(distances)
        }
        for (c in 1:k) {
            centroids[c] <- mean(x[clusters == c])
        }
    }
    result <- data.frame(x = x, cluster = clusters)
    return(result)
}
x <- c(1,2,1,3,2,6,5,7,6,12)
k <- 3
result <- my.kmean(x, k)
result
```

```
##     x cluster
## 1   1       2
## 2   2       1
## 3   1       2
## 4   3       1
## 5   2       1
## 6   6       3
## 7   5       3
## 8   7       3
## 9   6       3
## 10 12       3
```