Exercise 3 solution

Tristan, Köning, Ali Movasati, Isabelle Cretton

Oct. 1st, 2024

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

```
library(cluster)
library(stats)
library(ggplot2)
library(ggrepel)
library(magrittr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tibble)
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:cluster':
##
##     votes.repub
```

```
library(fields)
```

```
## Loading required package: spam

## Spam version 2.10-0 (2023-10-23) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
par(mfrow = c(1, 1))
```

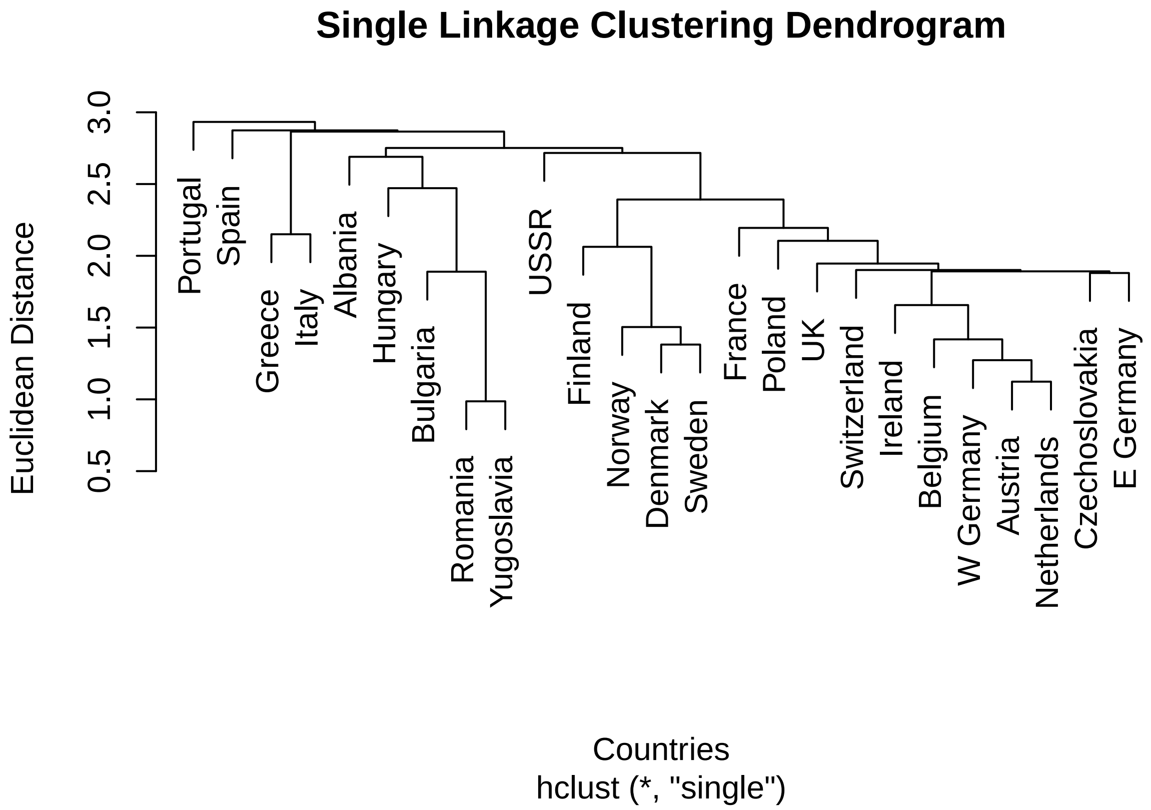
Problem 1 (Clustering)

1.A

```
# Setup data for clustering
protein <- read.csv("data/protein.txt", sep = "\t", header = TRUE)
row.names(protein) <- protein$Country
protein <- protein[, -1]
protein <- scale(protein)
```

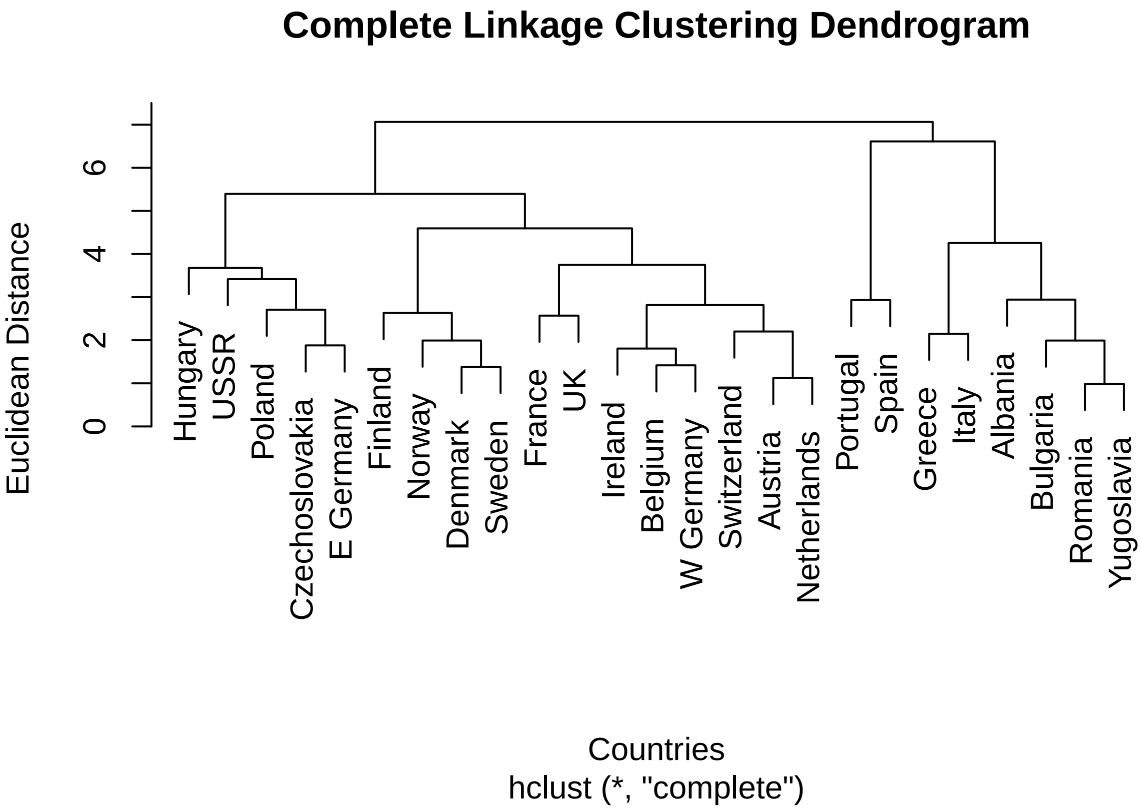
```
# Single Linkage Clustering
# Setup
single_linkage <- hclust(dist(protein), method = "single")

# Convert to dendrogram object
plot(single_linkage, main = "Single Linkage Clustering Dendrogram", xlab = "Countries", ylab = "Euclide
```



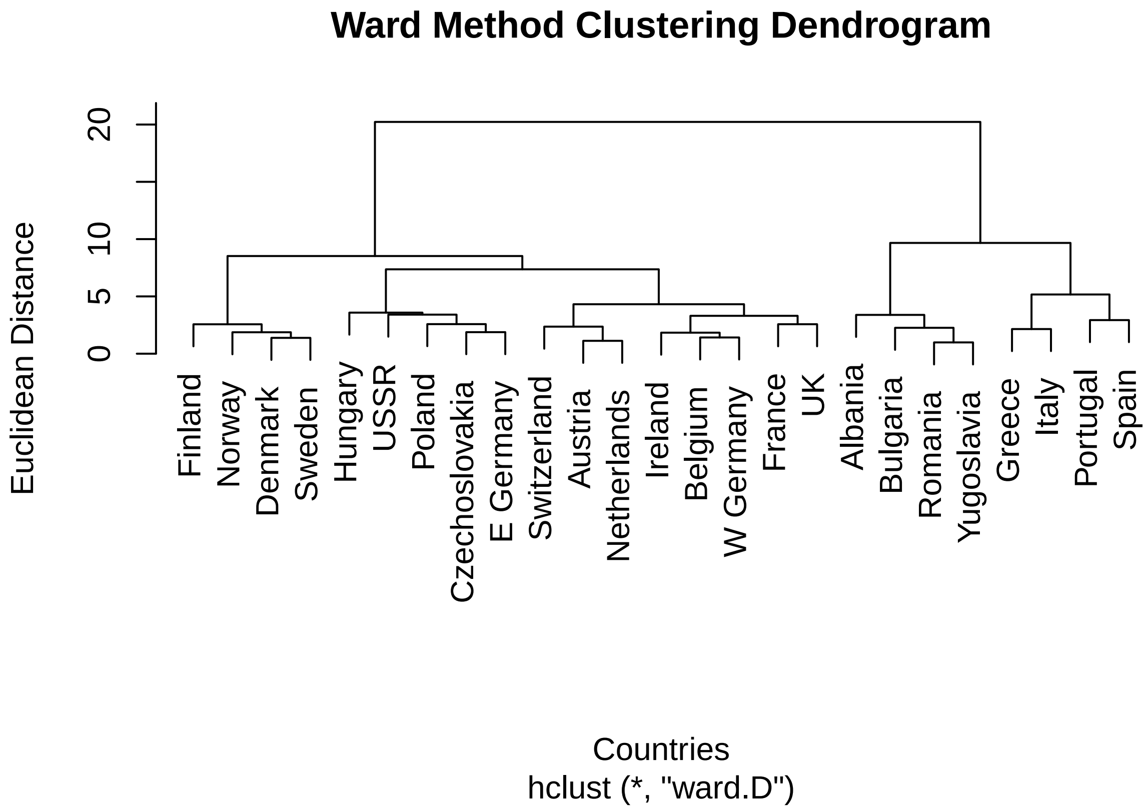
```
# Complete Linkage Clustering
# Setup
complete_linkage <- hclust(dist(protein), method = "complete")

plot(complete_linkage, main = "Complete Linkage Clustering Dendrogram", xlab = "Countries", ylab = "Eucl
```



```
# Ward Method Clustering
# Setup
ward_linkage <- hclust(dist(protein), method = "ward.D")

plot(ward_linkage, main = "Ward Method Clustering Dendrogram", xlab = "Countries", ylab = "Euclidean Di
```



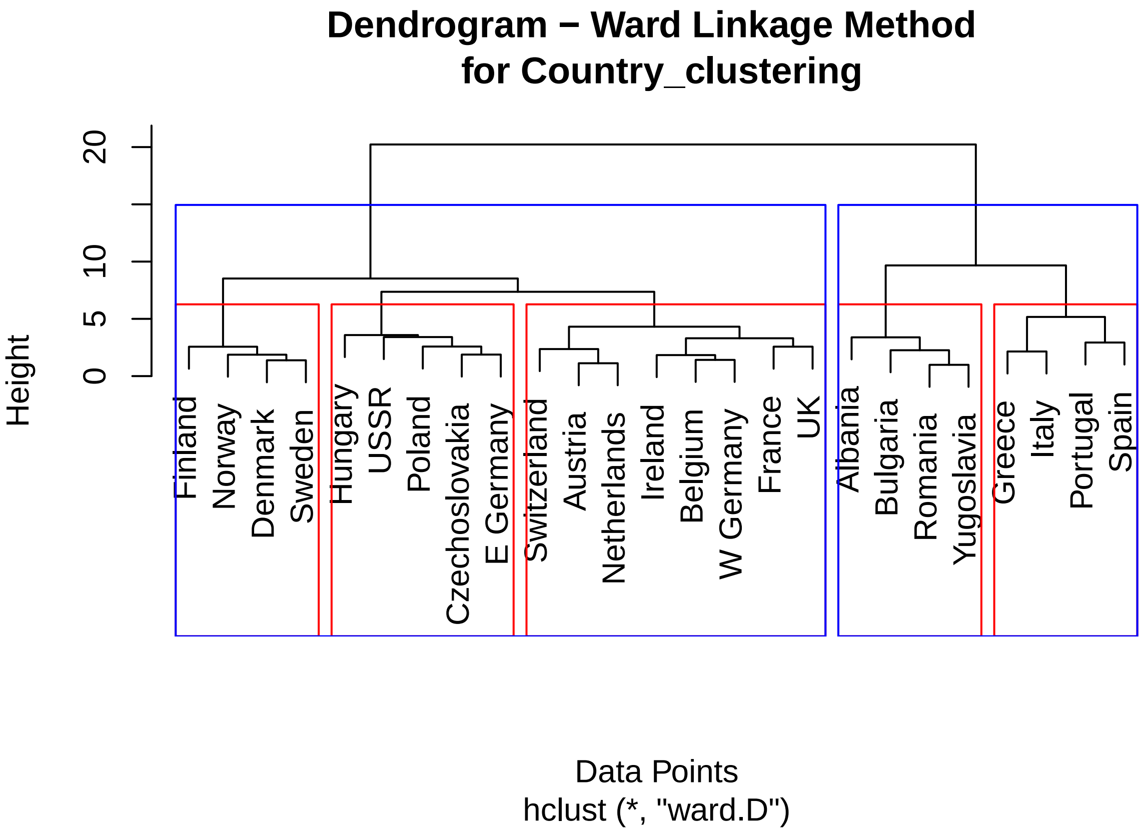
« **Comments** »

- We can observe that Ward method clustering shows the clearest structure in the dendrogram. It groups countries with small distances together early, and then forms larger clusters incrementally.
- The other two methods have a more erratic structure, with single linkage clustering showing the most erratic structure, where some countries dont belong to a cluster until the very end.

1.B

```
# perform clustering
for (i in c("Country_clustering")){
    if (i == "Country_clustering"){
        data <- protein
    } else if (i == "Protein_clustering"){
        data <- protein_t
    }
    distance_matrix <- dist(data, method = "euclidean")
    hc_single <- hclust(distance_matrix, method = "ward.D")
    plot(hc_single, main = paste0("Dendrogram - Ward Linkage Method \n for ", i), xlab = "Data Points",
    abline(h = 70, col = "red", lwd = 2, lty = 2)
    abline(h = 38, col = "blue", lwd = 2, lty = 2)
}
```

```
rect.hclust(ward_linkage, k = 5, border = "red")
rect.hclust(ward_linkage, k = 2, border = "blue")
```



« Comments »

- We can choose to look at either two or five groups, shown by the blue and red lines respectively. These can be representative of the economical status of the countries, with the two groups representing developed and developing countries, and the five groups being representative of different regions.

1.C

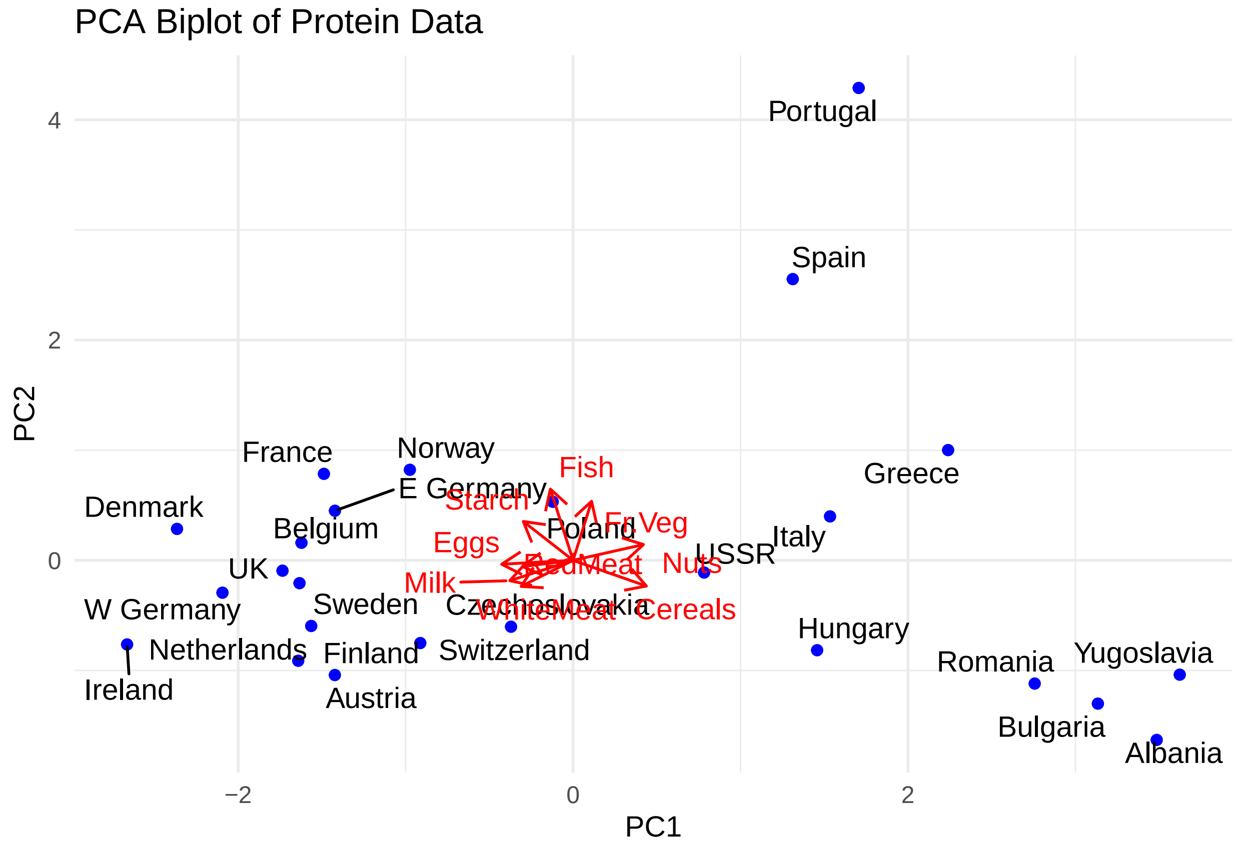
```
# Perform PCA on data
protein_pca <- scale(protein)

# Perform PCA
pca <- prcomp(protein_pca, scale = TRUE)

# Get loadings of PC
loadings <- pca$rotation
loadings_data <- data.frame(Variable = rownames(loadings), PC1 = loadings[, 1], PC2 = loadings[, 2])
```

```
# Perform Dimensionality reduction
scores <- pca$x
pca_data <- data.frame(Country = rownames(protein), PC1 = scores[, 1], PC2 = scores[, 2])


ggplot() +
  geom_point(data = pca_data, aes(x = PC1, y = PC2), color = "blue") +
  geom_text_repel(data = pca_data, aes(x = PC1, y = PC2, label = Country)) +
  geom_segment(data = loadings_data, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow = grid::arrow(length = unit(0.3, "cm")), color = "red") +
  geom_text_repel(data = loadings_data, aes(x = PC1, y = PC2, label = Variable), color = "red") +
  labs(title = "PCA Biplot of Protein Data", x = "PC1", y = "PC2") +
  theme_minimal()
```



« **Comments** » Looking at the plot of the Dimensionality-Reduction and the dendogram, we can observe that they build similar clusters, which are based on the geographic regions of europe, Although the PCA plot has more of a tendency to show Western vs. Eastern Europe in a socio-economic context, while the dendogram shows more of a geographic clustering.

Problem 2 (my.kmeans)

```r
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
```

« **Comments** » Note that when using different seeds that the final output may differ, as the initial centroids are randomly chosen.