Project - Multivariate Data Analysis

2024/25

Team Members

André Pires (64347)
Daniel Neves (64504)
Diana Santos (64478)
Matei Lupașcu (64471)
Vram Davtyan (64691)

Our work is about:

A set of data on temperature, humidity and evaporation is available, resulting from the observation of the following variables, in 46 days: TMAXDA - Maximum daily air temperature TMINDA – Minimum daily air temperature MTMEDA - Daily average air temperature TMAXDS - Maximum daily soil temperature TMINDS - Minimum daily soil temperature MTMEDS - Daily average soil temperature HRMAXD - Maximum daily relative humidity HRMIND - Minimum daily relative humidity MHMED - Average daily relative humidity FVENTOD – Wind speed.
EVAPOR – Evaporation

Question Number One:

1. Make a preliminary analysis of the data and discuss what you have learned from this analysis.

Before the Principal Component Analysis, we decided to perform a descriptive analysis of the data.

```r
# Run Data
data <- read.csv("data_8.csv", header = TRUE) # Modify the location accordingly
head (data)
```

```
##   tmaxda tminda mtmeda tmaxds tminds mtmeds hrmaxd hrmind mhmed fventod evapor
## 1     84     65    147     85     59    151     95     40   398     273     30
## 2     84     65    149     86     61    159     94     28   345     140     34
## 3     79     66    142     83     64    162     94     41   368     318     33
## 4     81     67    147     83     65    158     94     50   406     282     26
## 5     84     68    167     88     69    180     93     46   379     311     41
## 6     74     66    131     77     67    147     96     73   478     446      4
```

```r
# Type of data
str(data)
```

```
## 'data.frame':    50 obs. of  11 variables:
##  $ tmaxda : int  84 84 79 81 84 74 73 75 84 86 ...
##  $ tminda : int  65 65 66 67 68 66 66 67 68 72 ...
##  $ mtmeda : int  147 149 142 147 167 131 131 134 161 169 ...
##  $ tmaxds : int  85 86 83 83 88 77 78 84 89 91 ...
##  $ tminds : int  59 61 64 65 69 67 69 68 71 76 ...
##  $ mtmeds : int  151 159 162 158 180 147 159 159 195 206 ...
##  $ hrmaxd : int  95 94 94 94 93 96 96 95 95 93 ...
##  $ hrmind : int  40 28 41 50 46 73 72 70 63 56 ...
##  $ mhmed  : int  398 345 368 406 379 478 462 464 430 406 ...
##  $ fventod: int  273 140 318 282 311 446 294 313 455 604 ...
##  $ evapor : int  30 34 33 26 41 4 5 20 31 36 ...
```

```
# Dimension of data
dim(data)
```

```
## [1] 50 11
```

```
# Preliminary analysis of the data
summary(data)
```

```
##      tmaxda          tminda          mtmeda          tmaxds     
##  Min.   :58.00   Min.   :65.00   Min.   :131.0   Min.   :77.00  
##  1st Qu.:84.00   1st Qu.:68.25   1st Qu.:160.2   1st Qu.:87.25  
##  Median :88.00   Median :72.00   Median :175.5   Median :92.00  
##  Mean   :86.91   Mean   :71.26   Mean   :173.5   Mean   :90.74  
##  3rd Qu.:92.75   3rd Qu.:74.00   3rd Qu.:189.8   3rd Qu.:95.00  
##  Max.   :96.00   Max.   :76.00   Max.   :202.0   Max.   :97.00  
##  NA's   :4       NA's   :4       NA's   :4       NA's   :4      
##      tminds          mtmeds          hrmaxd          hrmind     
##  Min.   :59.00   Min.   :147.0   Min.   :93.00   Min.   :24.00  
##  1st Qu.:68.00   1st Qu.:170.8   1st Qu.:94.00   1st Qu.:43.00  
##  Median :70.00   Median :198.5   Median :95.00   Median :46.50  
##  Mean   :70.07   Mean   :190.7   Mean   :94.72   Mean   :48.74  
##  3rd Qu.:72.00   3rd Qu.:208.0   3rd Qu.:95.00   3rd Qu.:53.50  
##  Max.   :76.00   Max.   :215.0   Max.   :98.00   Max.   :73.00  
##  NA's   :4       NA's   :4       NA's   :4       NA's   :4      
##      mhmed          fventod          evapor     
##  Min.   :345.0   Min.   : 72.0   Min.   : 1.00  
##  1st Qu.:379.0   1st Qu.:174.0   1st Qu.:23.75  
##  Median :392.0   Median :235.0   Median :41.00  
##  Mean   :396.4   Mean   :277.7   Mean   :34.63  
##  3rd Qu.:405.8   3rd Qu.:384.0   3rd Qu.:45.00  
##  Max.   :478.0   Max.   :663.0   Max.   :54.00  
##  NA's   :4       NA's   :4       NA's   :4      
```

Since we have NA values, we decided to create a new variable $data_{clean}$ without these values included.

```
# Create a new variable with no NA values
suppressMessages(library(dplyr))
# We used suppressMessages on every library import for a cleaner output in the PDF Document
data_clean <- na.omit(data)
#data_selected_clean <- data_selected[1:(nrow(data_selected)-4),] ----> another way to remove Na values
```

```r
# Dimension of data
dim(data_clean)
```

```
## [1] 46 11
```

```r
# Preliminary analysis of the data
summary(data_clean)
```

```
##      tmaxda          tminda          mtmeda          tmaxds     
##  Min.   :58.00   Min.   :65.00   Min.   :131.0   Min.   :77.00  
##  1st Qu.:84.00   1st Qu.:68.25   1st Qu.:160.2   1st Qu.:87.25  
##  Median :88.00   Median :72.00   Median :175.5   Median :92.00  
##  Mean   :86.91   Mean   :71.26   Mean   :173.5   Mean   :90.74  
##  3rd Qu.:92.75   3rd Qu.:74.00   3rd Qu.:189.8   3rd Qu.:95.00  
##  Max.   :96.00   Max.   :76.00   Max.   :202.0   Max.   :97.00  
##      tminds          mtmeds          hrmaxd          hrmind     
##  Min.   :59.00   Min.   :147.0   Min.   :93.00   Min.   :24.00  
##  1st Qu.:68.00   1st Qu.:170.8   1st Qu.:94.00   1st Qu.:43.00  
##  Median :70.00   Median :198.5   Median :95.00   Median :46.50  
##  Mean   :70.07   Mean   :190.7   Mean   :94.72   Mean   :48.74  
##  3rd Qu.:72.00   3rd Qu.:208.0   3rd Qu.:95.00   3rd Qu.:53.50  
##  Max.   :76.00   Max.   :215.0   Max.   :98.00   Max.   :73.00  
##      mhmed          fventod          evapor     
##  Min.   :345.0   Min.   : 72.0   Min.   : 1.00  
##  1st Qu.:379.0   1st Qu.:174.0   1st Qu.:23.75  
##  Median :392.0   Median :235.0   Median :41.00  
##  Mean   :396.4   Mean   :277.7   Mean   :34.63  
##  3rd Qu.:405.8   3rd Qu.:384.0   3rd Qu.:45.00  
##  Max.   :478.0   Max.   :663.0   Max.   :54.00  
```

Let's start by checking if it is necessary to normalize the data.

```r
# Calculate the standard deviation
data_clean %>% summarise_if(is.numeric, sd)
```

```
##     tmaxda   tminda   mtmeda   tmaxds   tminds   mtmeds   hrmaxd  hrmind
## 1 7.461981 3.296023 20.06737 5.065942 3.666074 20.57512 1.204861 10.3052
##      mhmed  fventod  evapor
## 1 29.75952 149.0878 14.6308
```

The best approach is by using the correlation matrix, since the measures units are not all the same. And also, the mean and standard deviation are different.

Question Number Two:

2. Conduct a principal component analysis exploring the potentialities of this method. Include in your discussion topics like dimensionality reduction, interpretation of principal components.

Next we will determine the correlation matrix, as it is needed to calculate the eigenvalues and eigenvectors.

```
# Obtain Eigenvalues and Eigenvectors (based on the correlation matrix)

## 1st) Determine the correlation matrix
cor_data_clean <- cor(data_clean)


## 2nd) Obtain Eigenvalues and Eigenvectors
eigen_data_clean <- eigen(cor_data_clean)
eigen_data_clean$values
```

```
##  [1] 6.02758402 2.11220562 1.12660418 0.76325854 0.35561555 0.25937006
##  [7] 0.12318155 0.11081791 0.05994021 0.03945506 0.02196729
```

```
# We decided to only display the values, because only them are relevant for using the Kaiser's Criterio
```

Based on Kaiser's criterion we will retain the first three principal component, since the first three eigenvalues are greater than 1.

$$6.02758402, 2.11220562, 1.12660418 > 1$$

Now let's perform a Principal Component Analysis.

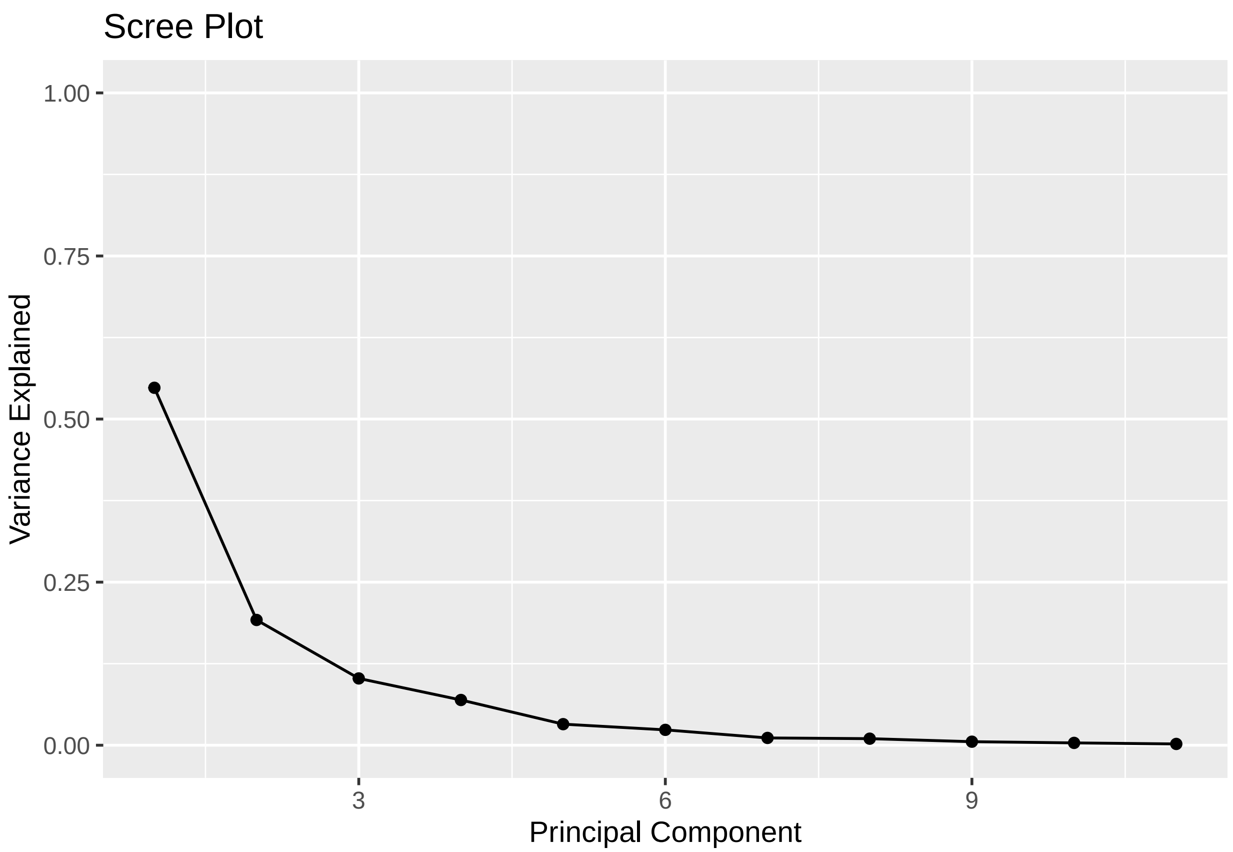
```
# Perform PCA
pca_data_clean <- princomp(data_clean,cor = TRUE)
# print(summary(pca_data_clean),loadings = TRUE)
# We decided to comment the execution of this because the output contain a very high number of rows
```

With three principal components we have a total of explain variance of 84.2%.

Let's do a scree-plot to confirm if we really need the first three principal components.

```
# Calculate total variance explained by each principal component

var_data_clean = pca_data_clean$sdev^2 / sum(pca_data_clean$sdev^2)

# Create scree plot - install ggplot2
suppressMessages(library(ggplot2))


suppressMessages(qplot(c(1:11),var_data_clean) + geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1))
```

```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Looking at the scree plot we realize that we must select the first 3 principal components, since from the fourth component onwards the curve starts to have a reduced slope.

Having reached the conclusion, with the methodologies presented previously, the ideal is to select 3 components, namely: Z_1, Z_2, Z_3

$Z_1 = -0,330Z_1 - 0,353Z_2 - 0,392Z_3 - 0,381Z_4 - 0,232Z_5 - 0,363Z_6 + 0,089Z_7 + 0,251Z_8 + 0,312Z_9 + 0,024Z_{10} - 0,336Z_{11}$ $Z_2 = -0,078Z_1 + 0,194Z_2 + 0,052Z_3 + 0,048Z_4 + 0,532Z_5 + 0,230Z_6 + 0,018Z_7 + 0,502Z_8 + 0,357Z_9 + 0,472Z_{10} - 0,113Z_{11}$ $Z_3 = -0,090Z_1 - 0,112Z_2 - 0,114Z_3 - 0,136Z_4 - 0,022Z_5 - 0,109Z_6 - 0,796Z_7 - 0,085Z_8 - 0,215Z_9 + 0,463Z_{10} + 0,185Z_{11}$

Identify the variables that contribute more for the explanation of each principal component retained.

```
# Make the correlation between the original values and the values obtained through PCA
cor(data_clean,pca_data_clean$scores)
```

```
##               Comp.1      Comp.2      Comp.3      Comp.4       Comp.5
## tmaxda    0.81006344  0.11398660  0.09589107  0.24779494  0.467756573
## tminda    0.86771785 -0.28158424  0.11886741  0.20404503 -0.015758888
## mtmeda    0.96163020 -0.07616727  0.12066408  0.12607485 -0.002234146
## tmaxds    0.93654077 -0.06956834  0.14440419  0.01151045 -0.098197824
## tminds    0.56863174 -0.77328045  0.02297285  0.06388193 -0.074058051
## mtmeds    0.89197021 -0.33412812  0.11571639 -0.12959498 -0.126546757
## hrmaxd   -0.21811935 -0.02635184  0.84449416 -0.47257701  0.107957198
## hrmind   -0.61550946 -0.72902778  0.09067574  0.13078519 -0.094182305
## mhmed    -0.76613882 -0.51914538  0.22827536  0.19901309  0.060260038
## fventod -0.05874778 -0.68627628 -0.49165965 -0.43258094  0.263938352
## evapor    0.82510901  0.16436105 -0.19676859 -0.39516129 -0.107725213
##                Comp.6      Comp.7      Comp.8       Comp.9      Comp.10
## tmaxda   0.20010039  0.01893305  0.01686027  0.017865818  0.011485410
## tminda  -0.28920356 -0.01023533  0.14767630 -0.033672981  0.004694435
## mtmeda  -0.13233266 -0.02242200  0.05711467  0.057587377 -0.010921571
## tmaxds  -0.04878672  0.22106963 -0.17789990 -0.008700848  0.066360130
## tminds   0.14951002 -0.15354990 -0.06155591 -0.132979245  0.030962363
## mtmeds   0.10678545 -0.04686181 -0.06521627  0.118647323 -0.114446769
## hrmaxd  -0.03829968 -0.03540743  0.01411686 -0.010106316  0.023505494
## hrmind   0.13892845  0.05490711  0.10797612  0.116895928  0.088389901
## mhmed    0.01049468  0.17098864  0.02282982 -0.074690872 -0.110901076
## fventod -0.15726188  0.02471895 -0.03452307  0.021866956  0.002704268
## evapor   0.19087515  0.11598960  0.17944758 -0.059251921 -0.007055807
##                Comp.11
## tmaxda   0.0166327756
## tminda   0.0711070454
## mtmeda  -0.1195569686
## tmaxds   0.0066406941
## tminds  -0.0154902683
## mtmeds   0.0389967546
## hrmaxd  -0.0009356466
## hrmind   0.0041546594
## mhmed   -0.0206406680
## fventod -0.0059148746
## evapor  -0.0075235914
```

```
# Apply rule for the 1st PC.   #Square Root of 6.03/11
sqrt(eigen_data_clean$values[1]/11)
```

```
## [1] 0.7402447
```

```
# Apply rule for the 2nd PC. #Square Root of 2,11/11
sqrt(eigen_data_clean$values[2]/11)
```

```
## [1] 0.4381994
```

```
# Apply rule for the 3st PC. #Square Root of 1,13/11
sqrt(eigen_data_clean$values[3]/11)
```

```
## [1] 0.320029
```

Important variables for each Principal Component

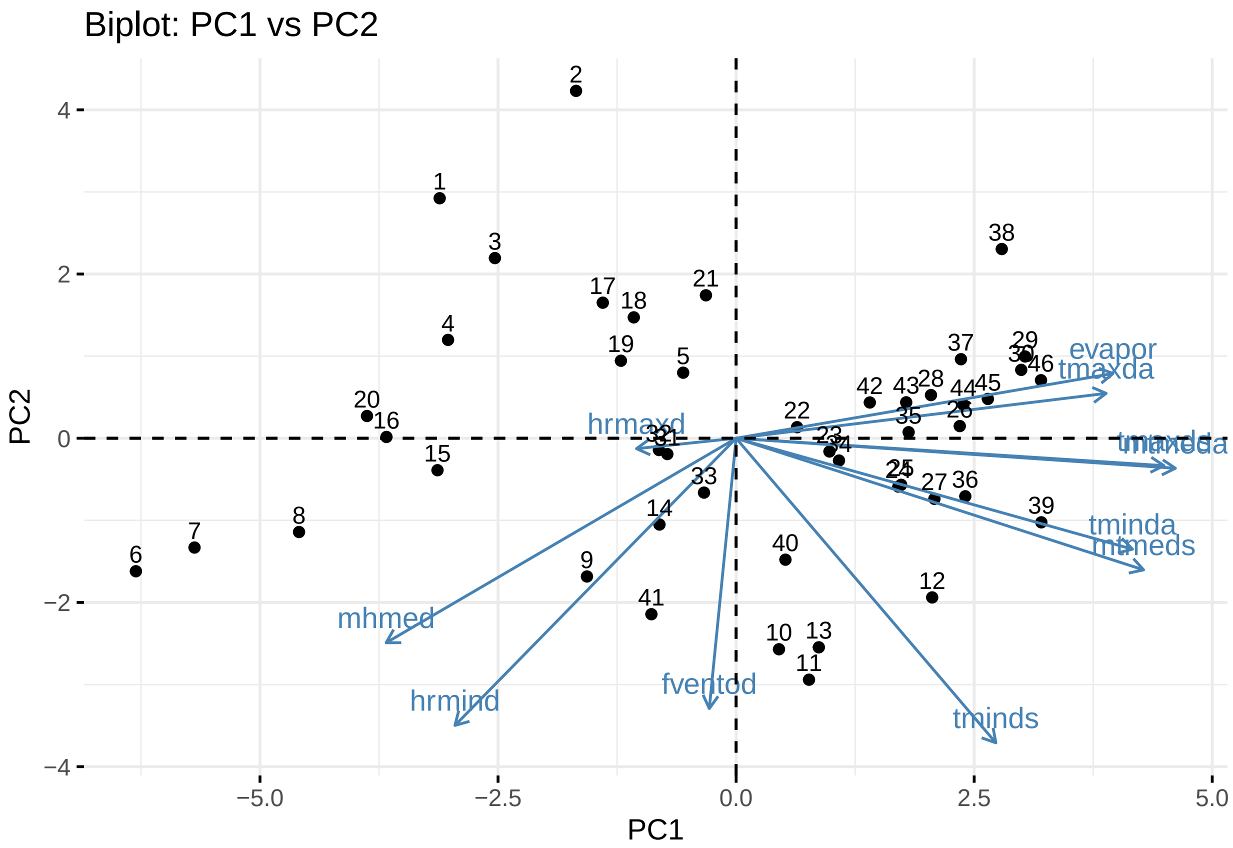
1st PC important variables: tmaxda, tminda, mtmeda, tmaxds,mtmeds, mhmed, evapor

2nd PC important variables: tminds, hrmind, fventod

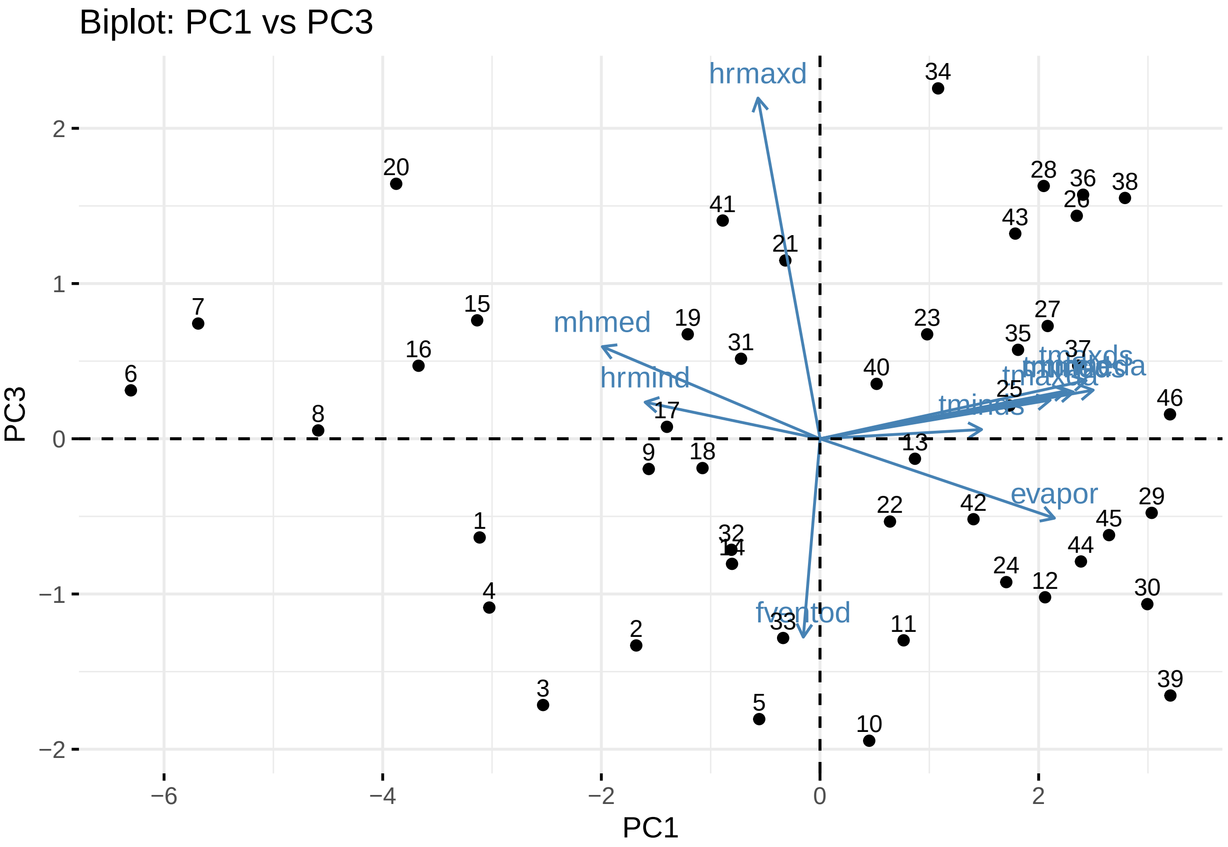
3rd PC important variables: hrmaxd

```r
suppressMessages(library(devtools))
#install_github("vqv/ggbiplot")
suppressMessages(require(ggbiplot))
#install.packages("patchwork")
suppressMessages(library(patchwork))
suppressMessages(library(ggbiplot))
```

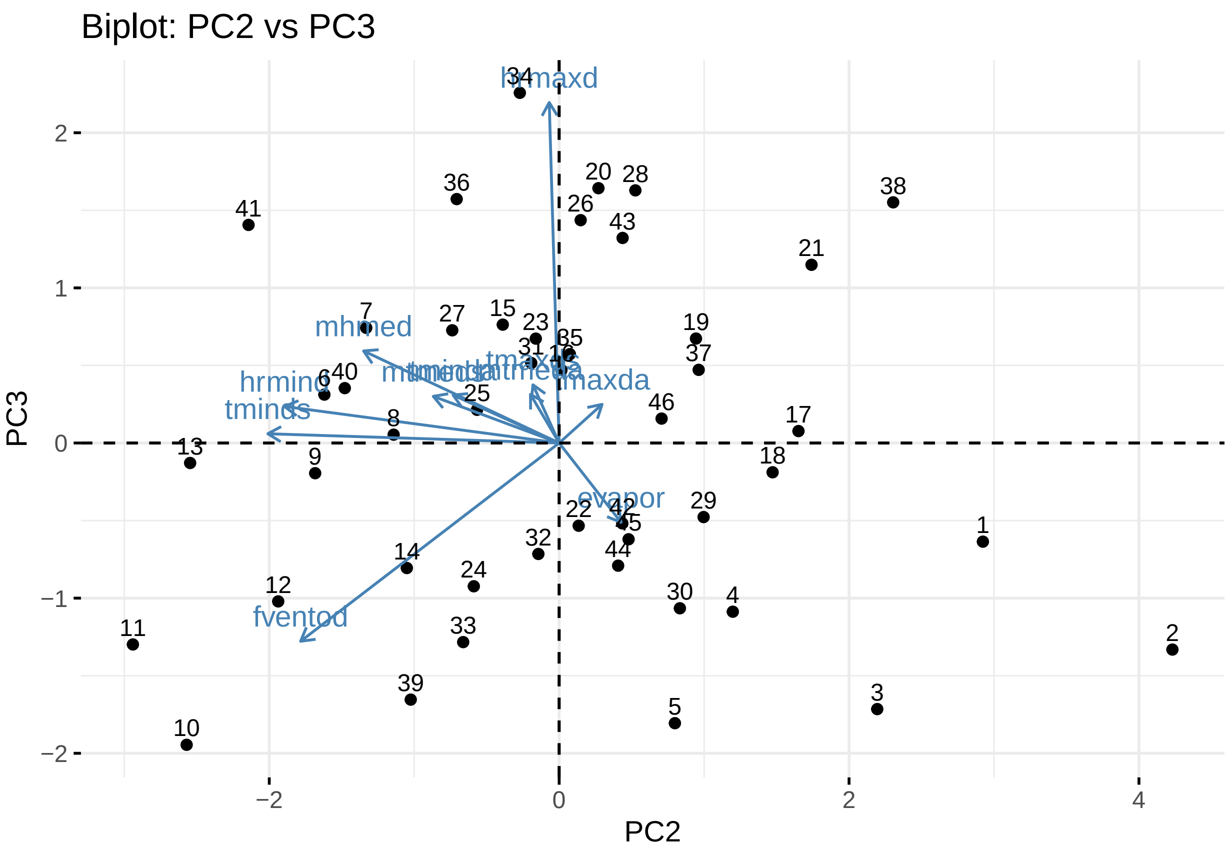
```r
# Representing the data based on the principal components

suppressMessages(library(factoextra))

# Biplot for PC1 vs PC2
fviz_pca_biplot(pca_data_clean, axes = c(1, 2), geom.ind = "point", label = "var") +
  xlab("PC1") +
  ylab("PC2") +
  ggtitle("Biplot: PC1 vs PC2") +
  geom_text(aes(label = rownames(pca_data_clean$scores)), vjust = -0.5, hjust = 0.5, size = 3, color =
```



```r
# Biplot for PC1 vs PC3
fviz_pca_biplot(pca_data_clean, axes = c(1, 3), geom.ind = "point", label = "var") +
  xlab("PC1") +
  ylab("PC3") +
  ggtitle("Biplot: PC1 vs PC3") +
  geom_text(aes(label = rownames(pca_data_clean$scores)), vjust = -0.5, hjust = 0.5, size = 3, color =
```



```
# Biplot for PC2 vs PC3
fviz_pca_biplot(pca_data_clean, axes = c(2, 3), geom.ind = "point", label = "var") +
  xlab("PC2") +
  ylab("PC3") +
  ggtitle("Biplot: PC2 vs PC3")+
  geom_text(aes(label = rownames(pca_data_clean$scores)), vjust = -0.5, hjust = 0.5, size = 3, color =
```



Conclusions

-> Variables with long vectors aligned with an axis dominate the corresponding principal component.

-> PCA was used to reduce the dimensionality of the data. Instead of working with all the original variables, the data was projected into a lower dimensional space (3 main components, represented by PC1, PC2 and PC3). The first principal component (PC1) explains 54.8% of the data variance, while PC2 explains 19.2% and PC3 explains 10.2%. Together, these three components explain a good part of the variance (84.2%), which suggests that most of the relevant information in the original data is preserved in these three axes.

-> In the PC1 vs PC2 plot, it appears that some variables (such as "*hrmind*" and "*mhmind*") are more correlated with PC1, while others (such as "*flventd*2") may be more correlated with PC2. Variables such as "*hrmaxd*" and "*mhmind*" appear to be strongly aligned, suggesting a positive correlation between them.

Question Number Three:

3. Conduct a cluster analysis exploring the hierarchical approach.Include in your discussion advantages and limitations of each methodology used.

```r
# Import the respective libraries:

suppressMessages(library(tidyverse)) #data manipulation
suppressMessages(library(cluster)) # clustering algorithm
suppressMessages(library(factoextra)) # clustering visualization
suppressMessages(library(dendextend)) # for comparing 2 dendograms
suppressMessages(library(laGP)) # squared euclidian distance
suppressMessages(library(metan)) # clustering algorithms
```

```
# Dissimilarity matrix: squared euclidian distance
dist_data_clean<- distance(data_clean)
# dist_data_clean # We decided to comment the execution of this because the output contain a very high
```

Let's perform the hierarchical Clustering

```
# Hierarchical clustering (Agglomerative) using single linkage
hc1<- agnes(dist_data_clean,method = 'single')
# hc1$merge # The hc1$merge object stores the matrix that describes the merging of clusters at each ste
# We decided to comment the execution of this because the output contain a very high number of rows
```

```
hc1$order # The command hc1$order returns the order of the observations (data points) as they appear in
```

```
##  [1]  1  4 30 42  3  5 34 25  7  8 15 20 16 17 23 35 36 44 38 26 43 45 22 27 37
## [26] 41  2 46 31 28 18 21 19 29  6  9 14 39 24 40 32 33 12 10 11 13
```

```
hc1$ac # The command hc1$ac returns the agglomerative coefficient, which measures the strength of the c
```

```
## [1] 0.906147
```

The Agglomerative coefficient is high!

The command plot(hc1) generates a dendrogram for the hierarchical clustering (hc1). A dendrogram is a tree-like diagram that shows how observations are grouped step-by-step during the clustering process. It visualizes the merging of clusters and their distances at each step.

To determine the number of clusters to consider from the dendrogram, we should look at where to "cut" the tree by choosing a height threshold. By cutting the dendrogram below this large gap (around the midpoint of the vertical axis), it appears reasonable to consider 4 clusters.

```
cutree(hc1,4)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  1  1  1  1  1  1  1  1  1  2  2  3  4  1  1  1  1  1  1  1  1  1  1  1  1  1
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
##  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

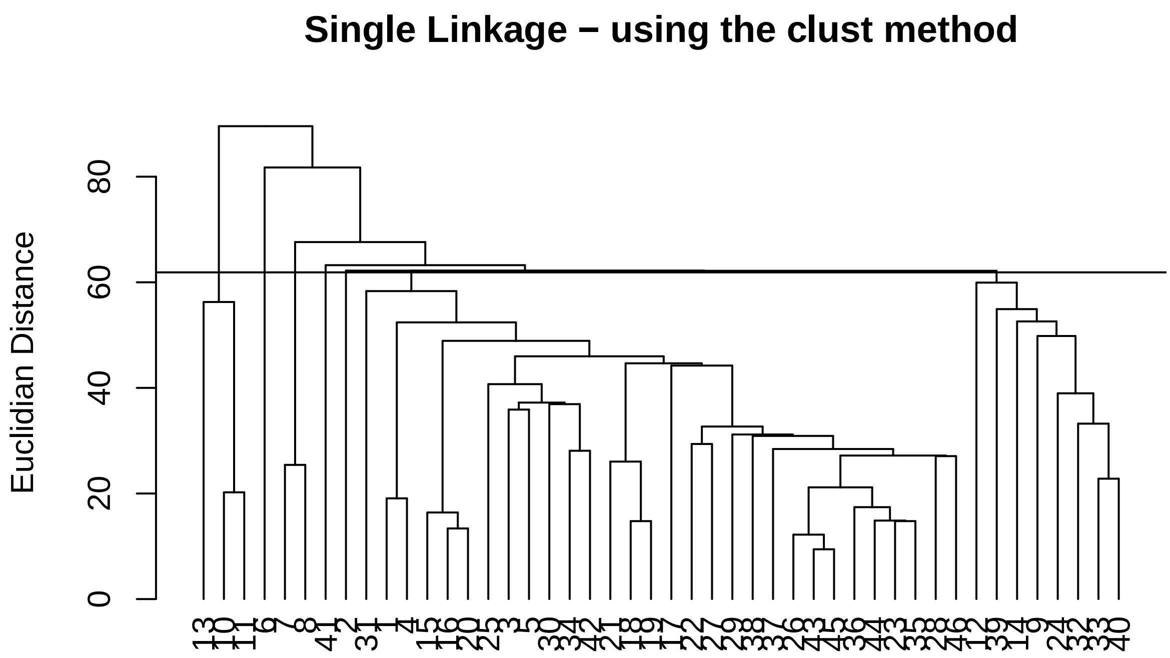
```
# cbind(row.names(data_clean),cutree(hc1,4)) # The code assigns each observation to one of 4 clusters f
# We decided to comment the execution of this because the output contain a very high number of rows
```

With another function to perform clustering –> cophenetic coefficient

```
clust_1<- clustering(data_clean,distmethod = 'euclidean',clustmethod = 'single')
clust_1$cophenetic
```

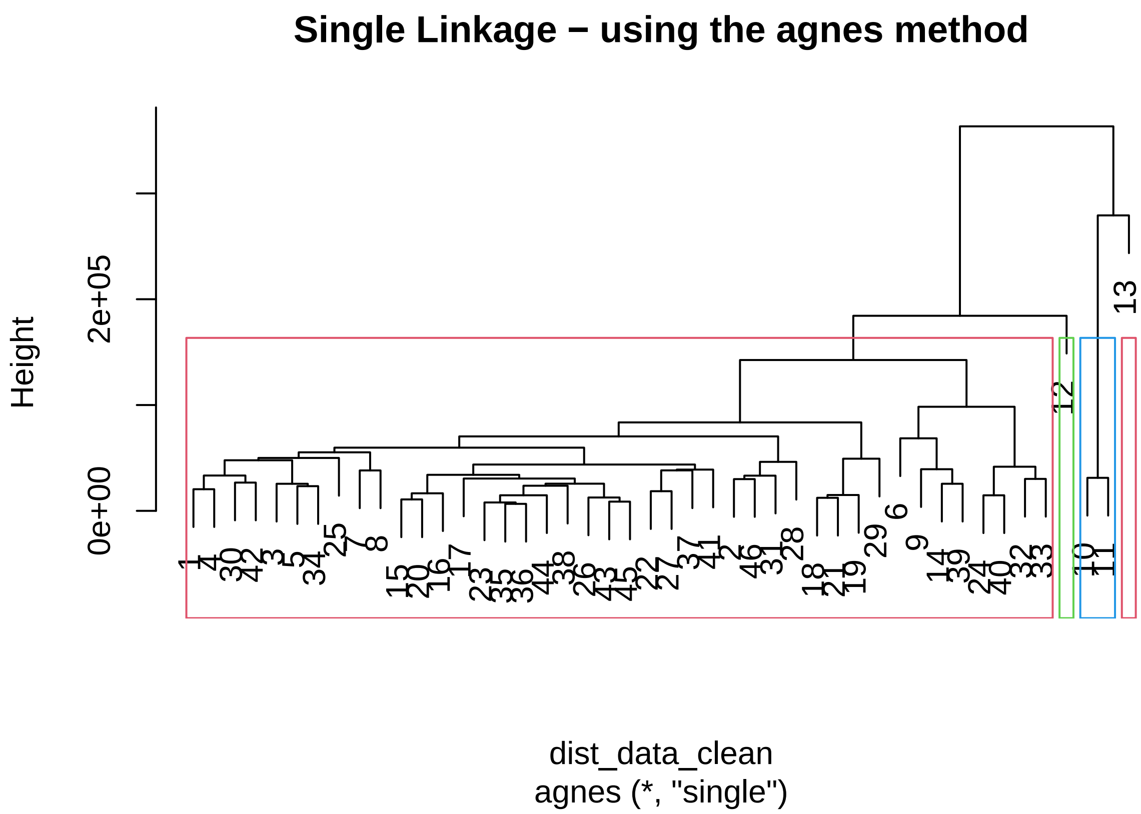
```
## [1] 0.7184685
```

```
plot(clust_1,horiz = F,ylab='Euclidian Distance',main='Single Linkage - using the clust method')
```



The command plot(hc1) generates a dendrogram for the hierarchical clustering (hc1). A dendrogram is a tree-like diagram that shows how observations are grouped step-by-step during the clustering process. It visualizes the merging of clusters and their distances at each step.

```
# Convert agnes object to hclust
hc1_hclust <- as.hclust(hc1)
plot(hc1_hclust, main = 'Single Linkage - using the agnes method')
rect.hclust(hc1_hclust, k = 4, border = 2:4)
```



plot(hc1): It plots the dendrogram of a hierarchical clustering object (hc1), visually representing how data points are grouped at different levels of similarity.

rect.hclust(hc1, k=5, border=2:4): This adds colored rectangles to the dendrogram, highlighting the 5 clusters (k=5) by drawing borders in colors specified by 2:4 (e.g., red, green, blue).

```
# Hierarchical clustering (Agglomerative) using complete linkage
hc2<- agnes(dist_data_clean,method = 'complete')
# hc2$merge
# We decided to comment the execution of this because the output contain a very high number of rows
```

```
hc2$order
```

```
##  [1]  1  4 30 42 22 27 41 37  3  5 34 25  7  8  2 46 28 31 18 21 19 29 15 20 16
## [26] 23 35 36 44 17 26 43 45 38  6  9 14 39 24 40 32 33 12 10 11 13
```

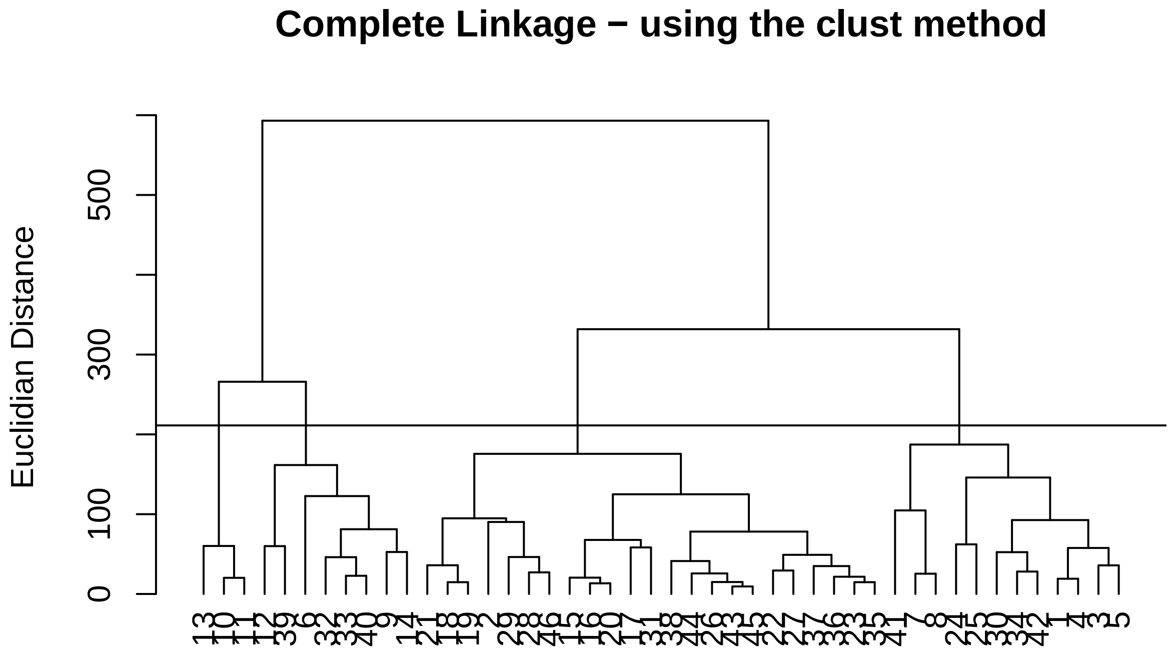
```
hc2$ac
```

```
## [1] 0.9686166
```

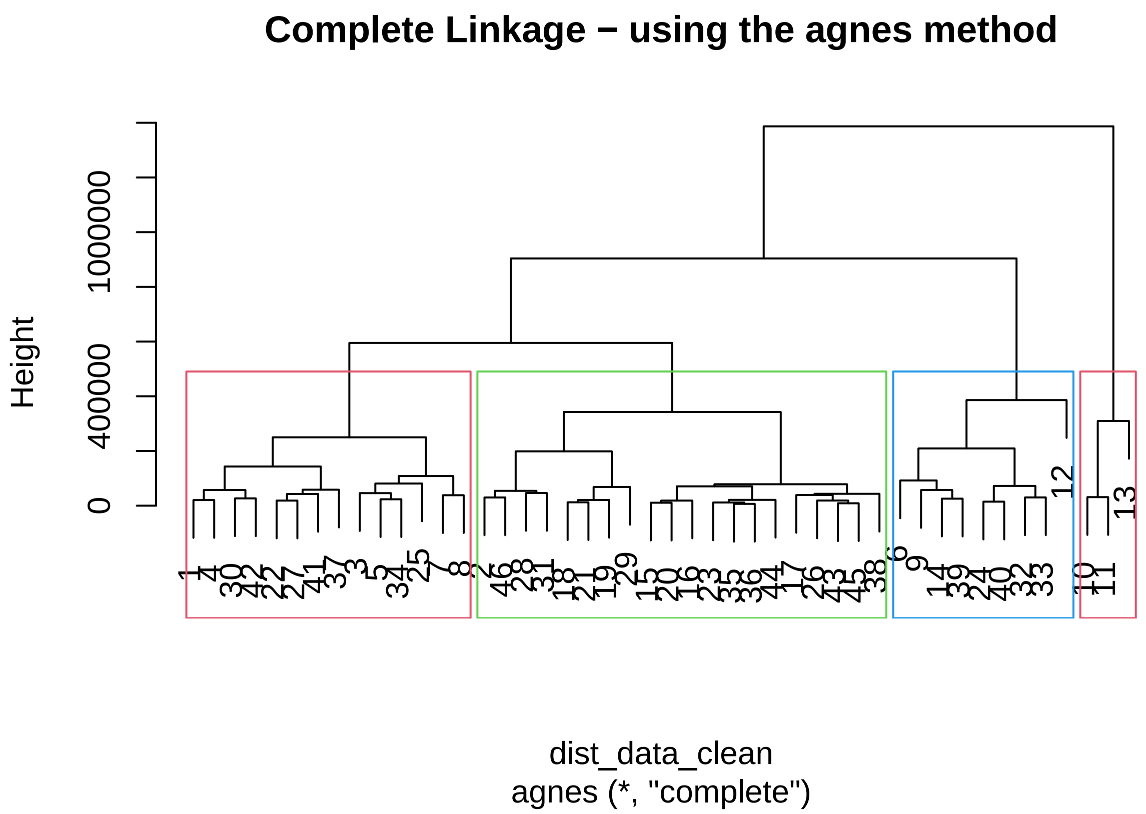
```
clust_2<- clustering(data_clean,distmethod = 'euclidean',clustmethod = 'complete')
clust_2$cophenetic
```

```
## [1] 0.7697676
```

```r
plot(clust_2,horiz = F,ylab='Euclidian Distance',main='Complete Linkage - using the clust method')
```



```r
# Convert agnes object to hclust
hc2_hclust <- as.hclust(hc2)
plot(hc2_hclust, main = 'Complete Linkage - using the agnes method')
rect.hclust(hc2_hclust, k = 4, border = 2:4)
```



```r
# Hierarchical clustering (Agglomerative) using Ward linkage
hc3<- agnes(dist_data_clean,method = 'ward')
# hc3$merge
# We decided to comment the execution of this because the output contain a very high number of rows
```

```r
hc3$order
```

```
##  [1]  1  4 30 42  3  5 34 25  7  8 15 20 16 17 38 26 43 45 23 35 36 44 22 27 41
## [26] 37  2 46 31 28 18 21 19 29  6  9 14 39 12 24 40 32 33 10 11 13
```

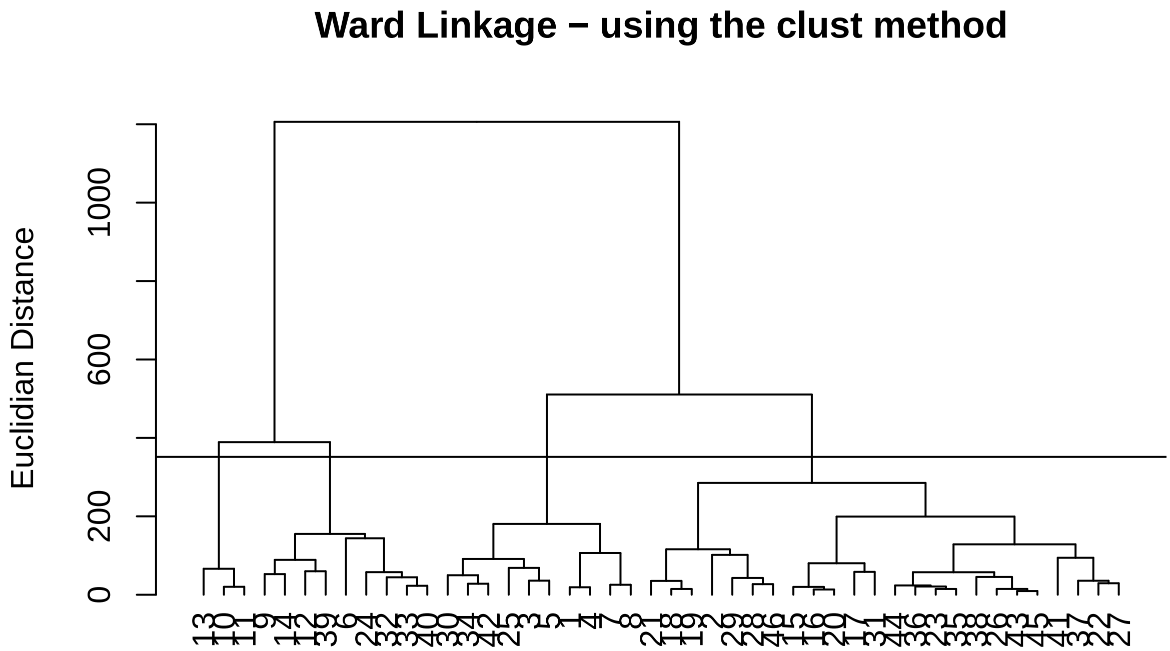
```r
hc3$ac
```

```
## [1] 0.9847143
```

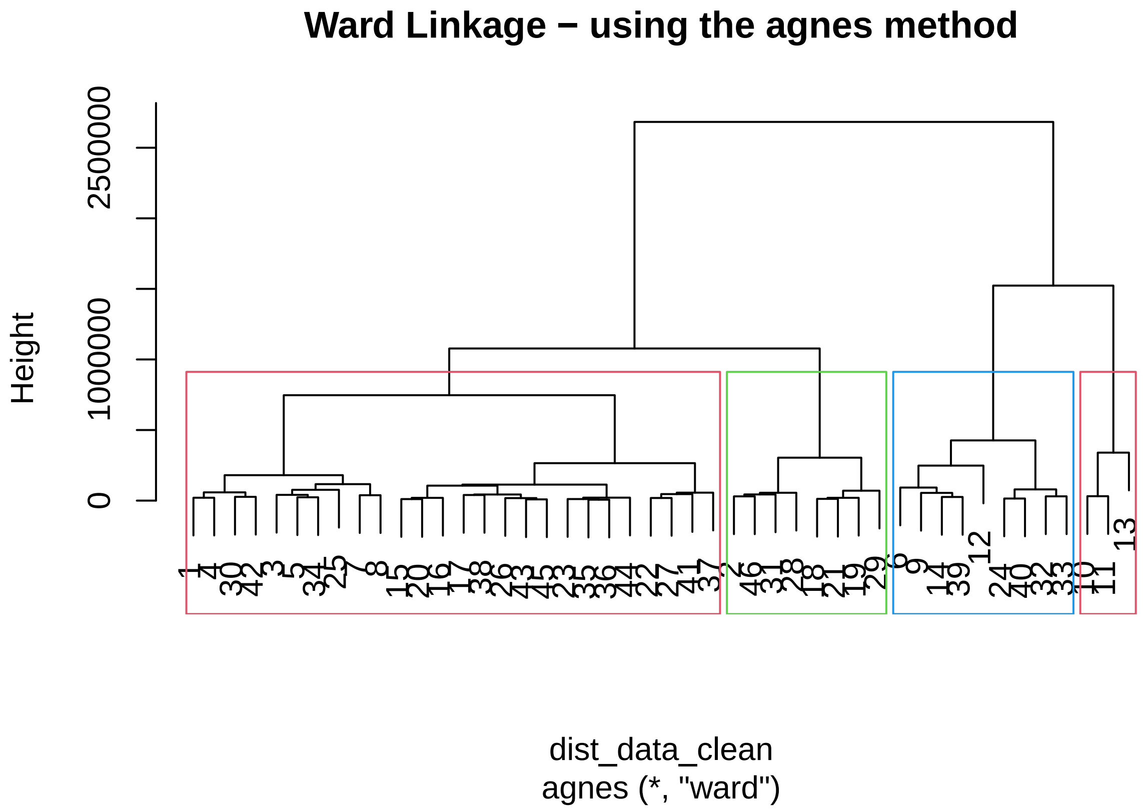
```r
clust_3<- clustering(data_clean,distmethod = 'euclidean',clustmethod = 'ward.D2')
clust_3$cophenetic
```

```
## [1] 0.7774111
```

```r
plot(clust_3,horiz = F,ylab='Euclidian Distance',main='Ward Linkage - using the clust method')
```

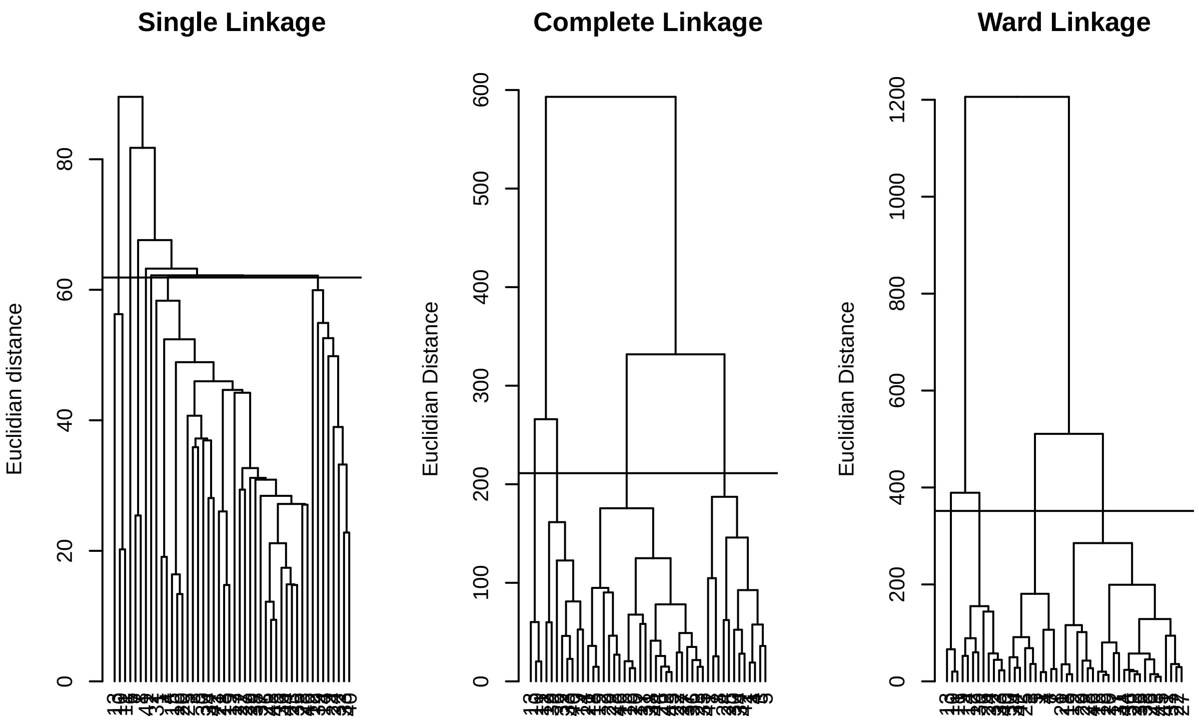


```
# Convert agnes object to hclust
hc3_hclust <- as.hclust(hc3)
plot(hc3_hclust, main = 'Ward Linkage - using the agnes method')
rect.hclust(hc3_hclust, k = 4, border = 2:4)
```



We found the Ward Linkage method to be the most suitable, as its cophenetic coefficent was the highest, therefore we considered the 4 clusters based on the visual output of the clustering with this method.

```
par(mfrow=c(1,3))
plot(clust_1,horiz = F,ylab='Euclidian distance',main= 'Single Linkage')

plot(clust_2,horiz = F,ylab='Euclidian Distance',main='Complete Linkage')

plot(clust_3,horiz = F,ylab='Euclidian Distance',main='Ward Linkage')
```



Question Number Four:

4. Compare the results obtained in (2) and (3).

PCA:

Reduces data to 3 principal components explaining 84.2% of the variance:
PC1 explains 54.8%: Dominated by temperature and evaporation-related variables.
PC2 explains 19.2%: Influenced by wind speed and minimum soil temperature.
PC3 explains 10.2%: Associated with maximum humidity.
Focuses on variance and helps identify key contributing variables for each component.

Hierarchical Clustering:

Groups data into 4 clusters based on Ward linkage, which had the highest agglomerative coefficient (0.86). Highlights natural groupings of days with similar weather patterns (e.g., clusters based on temperature or evaporation).

Comparison between the two methods:

Focus of analysis:

PCA focuses on identifying patterns in variables by reducing dimensionality and highlighting the major contributors to variability.

Cluster analysis groups observations (days) into meaningful clusters based on their overall similarity.

Outcome:

PCA reduces the dataset to three main components (PC1, PC2, PC3), summarizing the relationships between variables. Cluster analysis identifies four distinct groups, showing how the observations (days) differ based on their profiles.

Interpretation of results:

PCA results help understand which variables dominate and contribute to the dataset's variability. For example, temperature and humidity strongly influence the first component.
Cluster analysis provides actionable insights by grouping days with similar weather conditions, which may be used for targeted interventions or further analysis.

Use case:

PCA is ideal for dimensionality reduction and identifying variable relationships, often used as a preprocessing step for machine learning or further statistical analysis.
Cluster analysis is more practical for segmentation tasks, such as identifying patterns in weather conditions over different periods.

Question Number Five:

5. Interpretation of the results. Translate the statistical results in current language, accessible to the researcher who, hypothetically, have proposed the project.

Key variables:

PCA showed that temperature-related variables (tmaxda, tminda, etc.) explain most of the variation in the dataset.
Humidity (hrmaxd) and wind speed (fventod) also play significant roles but contribute differently to different patterns.

How the data is being grouped:

Using clustering, we identified 4 distinct groups of observations. These clusters may represent patterns such as days with similar weather conditions or evaporation rates.

The importance:

Instead of analyzing all variables, researchers can focus on the three principal components or representative clusters to simplify their study.
For example, patterns in temperature and evaporation could inform predictions about agricultural outcomes or water usage.

The usage of the information:

Use PCA for predictive models to reduce complexity.
Use clustering to segment data into meaningful categories for targeted analysis (e.g., comparing high-evaporation days vs. low-evaporation days).
In essence, these techniques simplify the complex dataset, highlighting the most important variables and grouping observations for actionable insights.