Descriptive Analysis of the Data

Alberto Fallocco, Ana Araújo, Chris Anaya, Leonor Ferreira

2023-12-28

```
# Run data
data8 <- read.csv("data_8.csv", header = TRUE)
head(data8)
```

```
##    T1 T2 T3  T4 T6  T5
## 1 121 22 74 223 54 254
## 2 108 30 80 175 40 300
## 3 122 49 87 266 41 223
## 4  77 37 66 178 80 209
## 5 140 35 71 175 38 261
## 6 108 37 57 241 59 245
```

```
str(data8)
```

```
## 'data.frame':    40 obs. of  6 variables:
##  $ T1: int  121 108 122 77 140 108 124 130 149 129 ...
##  $ T2: int  22 30 49 37 35 37 39 34 55 38 ...
##  $ T3: int  74 80 87 66 71 57 52 89 91 72 ...
##  $ T4: int  223 175 266 178 175 241 194 200 198 162 ...
##  $ T6: int  54 40 41 80 38 59 72 85 50 47 ...
##  $ T5: int  254 300 223 209 261 245 242 242 277 268 ...
```

```
dim(data8)
```

```
## [1] 40  6
```

```
summary(data8)
```

```
##        T1              T2              T3              T4
##  Min.   : 47.0   Min.   :17.00   Min.   : 52.00   Min.   :152.0
##  1st Qu.:116.5   1st Qu.:30.75   1st Qu.: 75.50   1st Qu.:186.0
##  Median :129.5   Median :36.50   Median : 82.50   Median :223.0
##  Mean   :126.9   Mean   :34.90   Mean   : 81.80   Mean   :214.7
##  3rd Qu.:142.0   3rd Qu.:39.00   3rd Qu.: 88.25   3rd Qu.:240.2
##  Max.   :164.0   Max.   :55.00   Max.   :105.00   Max.   :291.0
##        T6              T5
##  Min.   :27.00   Min.   :209.0
##  1st Qu.:37.50   1st Qu.:242.0
##  Median :47.50   Median :262.5
##  Mean   :48.95   Mean   :265.2
##  3rd Qu.:58.00   3rd Qu.:293.2
##  Max.   :88.00   Max.   :324.0
```

```
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
```

```
data8 %>% summarise_if(is.numeric, sd)
```

```
##         T1       T2       T3       T4       T6       T5
## 1 22.98249 8.301375 12.04947 36.96526 16.25904 32.69733
```

Principal Component Analysis

Alberto Fallocco, Ana Araújo, Chris Anaya, Leonor Ferreira

2022-12-28

PCA, Step 0 - Recalling data, libraries

```
# Run data
data8 <- read.csv("data_8.csv", header = TRUE)
head(data8)
```

```
##    T1 T2 T3  T4 T6  T5
## 1 121 22 74 223 54 254
## 2 108 30 80 175 40 300
## 3 122 49 87 266 41 223
## 4  77 37 66 178 80 209
## 5 140 35 71 175 38 261
## 6 108 37 57 241 59 245
```

```
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(ggplot2)
library(AMR)
library(devtools)
```

```
## Loading required package: usethis
```

```
require(ggbiplot)
```

```
## Loading required package: ggbiplot

## Loading required package: plyr
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## Loading required package: scales

## Loading required package: grid
```

```
data8 %>% summarise_if(is.numeric, sd)
```

```
##         T1       T2       T3       T4       T6       T5
## 1 22.98249 8.301375 12.04947 36.96526 16.25904 32.69733
```

PCA, Step 1 - Principal Components Analysis

Step 1.1. - Data standardization, Eigenvalues and Eigenfactors

```
# Obtain Eigenvalues and Eigenvectors (based on the Correlation Matrix)

## 1) Determine the Correlation Matrix
cor_data8 <- cor(data8)
cor_data8
```

```
##             T1          T2          T3         T4          T6          T5
## T1  1.00000000  0.18863907  0.10103566  0.1229213 -0.05360504  0.38744776
## T2  0.18863907  1.00000000 -0.12094150 -0.2664102 -0.24453244 -0.06735388
## T3  0.10103566 -0.12094150  1.00000000  0.2893548 -0.15750071  0.07458298
## T4  0.12292126 -0.26641020  0.28935484  1.0000000 -0.18898014  0.32696061
## T6 -0.05360504 -0.24453244 -0.15750071 -0.1889801  1.00000000 -0.15021611
## T5  0.38744776 -0.06735388  0.07458298  0.3269606 -0.15021611  1.00000000
```

```
## 2) Obtain Eigenvalues and Eigenvectors
eigen_data8 <- eigen(cor_data8)
eigen_data8
```

```
## eigen() decomposition
## $values
## [1] 1.7751277 1.3544159 1.0726505 0.8147958 0.5306128 0.4523973
##
```

```
## $vectors
##             [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## [1,] -0.40239072  0.3964661  0.4617841 -0.3928149 -0.2103062  0.5187674
## [2,]  0.09715877  0.7472294 -0.1752970 -0.1315611 -0.2801896 -0.5528697
## [3,] -0.38541311 -0.2181560 -0.4329575 -0.7177525  0.2585104 -0.1855163
## [4,] -0.54333623 -0.3144601 -0.1065065  0.2453920 -0.7066663 -0.1869825
## [5,]  0.31188931 -0.3559400  0.6268314 -0.3992852 -0.2012981 -0.4279773
## [6,] -0.53629229  0.1062657  0.4053555  0.3058981  0.5201339 -0.4155385
```

Step 1.2. - PCA

```
# Perform PCA
pca_data8 <- princomp(data8,cor=TRUE)
print(summary(pca_data8,loadings=TRUE))
```

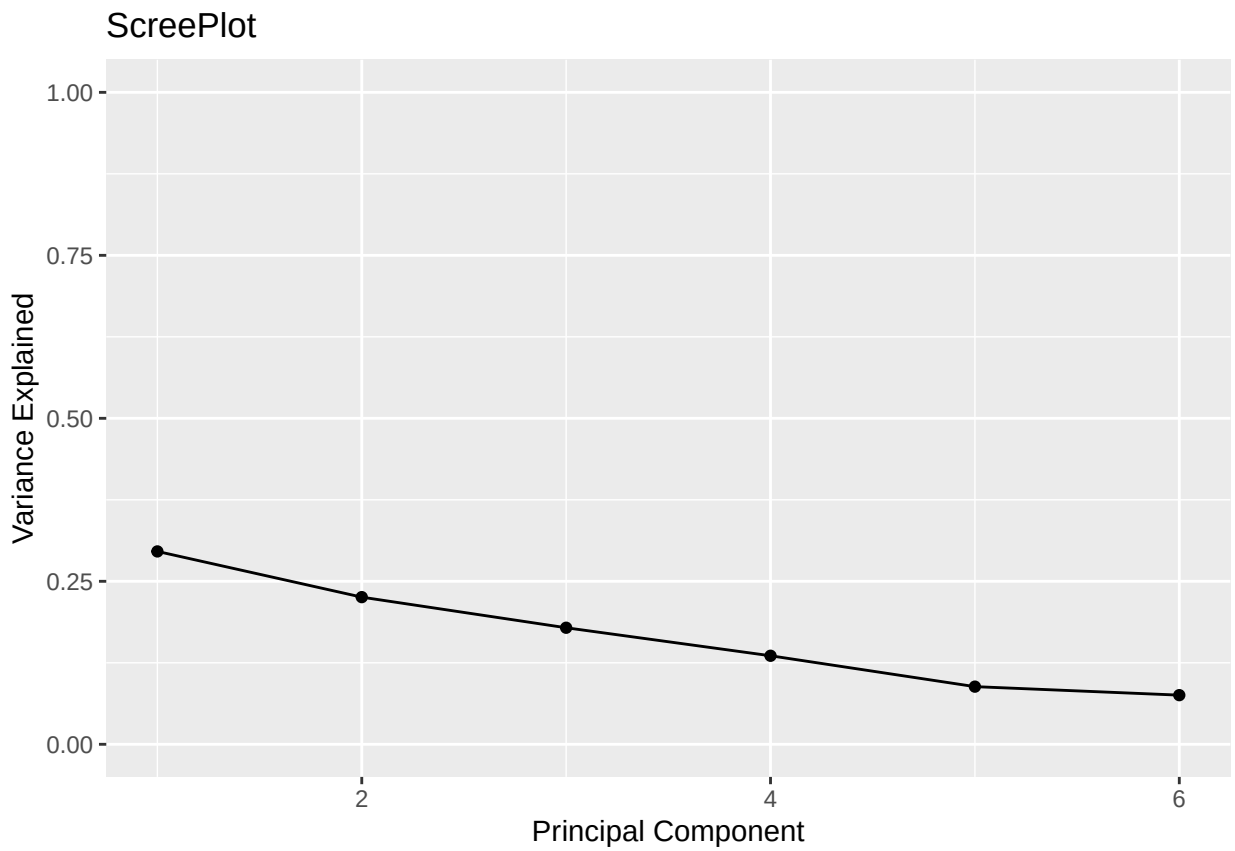
```
## Importance of components:
##                            Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     1.3323392 1.1637937 1.0356884 0.9026604 0.72843175
## Proportion of Variance 0.2958546 0.2257360 0.1787751 0.1357993 0.08843547
## Cumulative Proportion  0.2958546 0.5215906 0.7003657 0.8361650 0.92460045
##                            Comp.6
## Standard deviation     0.67260486
## Proportion of Variance 0.07539955
## Cumulative Proportion  1.00000000
##
## Loadings:
##    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
## T1  0.402  0.396  0.462  0.393  0.210  0.519
## T2         0.747 -0.175  0.132  0.280 -0.553
## T3  0.385 -0.218 -0.433  0.718 -0.259 -0.186
## T4  0.543 -0.314 -0.107 -0.245  0.707 -0.187
## T6 -0.312 -0.356  0.627  0.399  0.201 -0.428
## T5  0.536  0.106  0.405 -0.306 -0.520 -0.416
```

```
# Calculate total variance explained by each principal component
var_explained_data8 <- pca_data8$sdev^2 / sum(pca_data8$sdev^2)
var_explained_data8
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## 0.29585461 0.22573598 0.17877509 0.13579930 0.08843547 0.07539955
```

```
# Create ScreePlot
qplot(c(1:6),var_explained_data8) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("ScreePlot") +
  ylim(0,1)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



Step 1.3. - Interpretation of the PCs

```
component_matrix <- cor(data8,pca_data8$scores)
component_matrix
```

```
##        Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6
## T1  0.5361209  0.4614047  0.4782644  0.3545784  0.1531937  0.3489255
## T2 -0.1294484  0.8696209 -0.1815530  0.1187550  0.2040990 -0.3718628
## T3  0.5135010 -0.2538886 -0.4484090  0.6478867 -0.1883072 -0.1247792
## T4  0.7239081 -0.3659667 -0.1103075 -0.2215057  0.5147582 -0.1257653
## T6 -0.4155423 -0.4142408  0.6492020  0.3604190  0.1466319 -0.2878596
## T5  0.7145232  0.1236713  0.4198220 -0.2761221 -0.3788820 -0.2794932
```

More important variables for the explanations of the 1st and 2nd PC:

- **1st PC**: T1, T3, T4, T5, T6
- **2nd PC**: T2

```
# 1st PC
## T1
a_11_square <- (component_matrix[1,1]/sqrt(eigen_data8$values[1]))^2
a_11_square
```

```
## [1] 0.1619183
```

```
# 1st PC
## T3
a_31_square <- (component_matrix[3,1]/sqrt(eigen_data8$values[1]))^2
a_31_square
```

```
## [1] 0.1485433
```

```
# 1st PC
## T4
a_41_square <- (component_matrix[4,1]/sqrt(eigen_data8$values[1]))^2
a_41_square
```

```
## [1] 0.2952143
```

```
# 1st PC
## T6
a_61_square <- (component_matrix[5,1]/sqrt(eigen_data8$values[1]))^2
a_61_square
```

```
## [1] 0.09727494
```

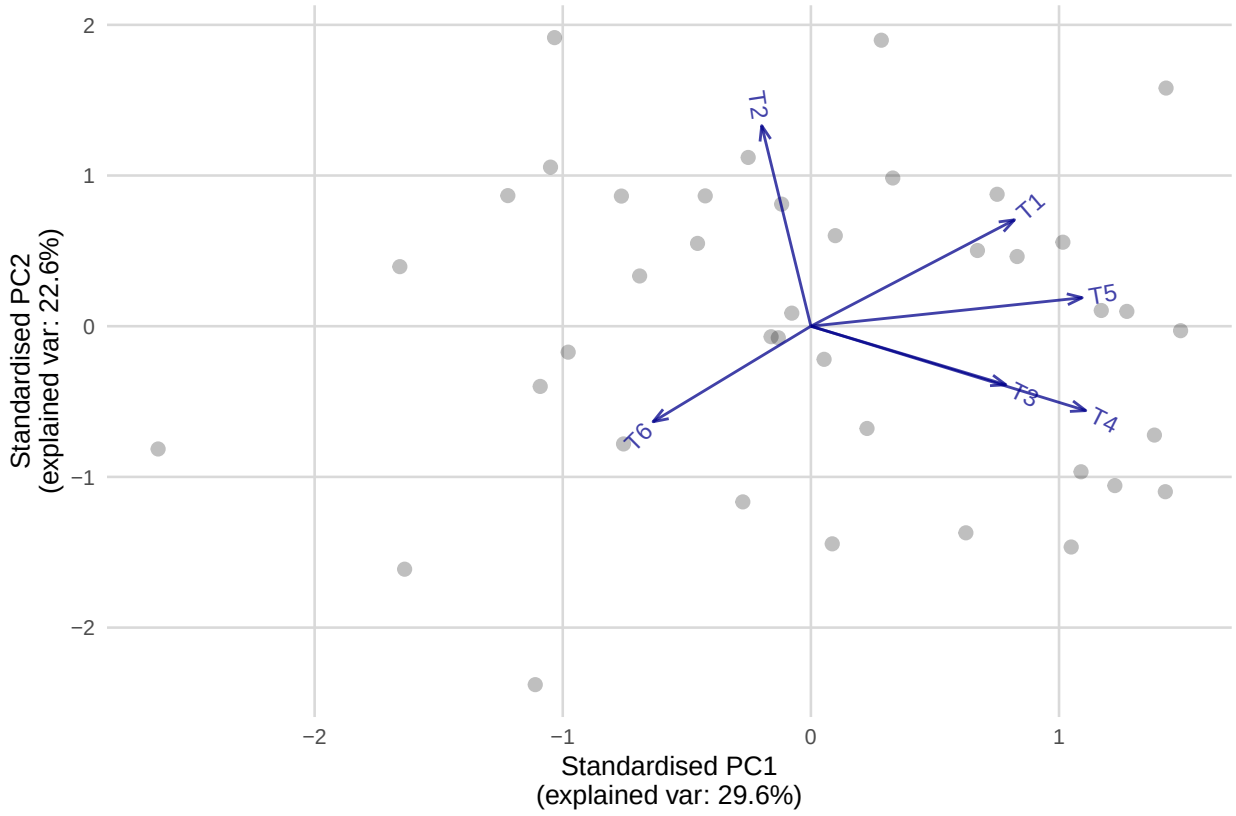
```
# 1st PC
## T5
a_51_square <- (component_matrix[6,1]/sqrt(eigen_data8$values[1]))^2
a_51_square
```

```
## [1] 0.2876094
```

```
# 2nd PC
## T2
a_22_square <- (component_matrix[2,2]/sqrt(eigen_data8$values[1]))^2
a_22_square
```

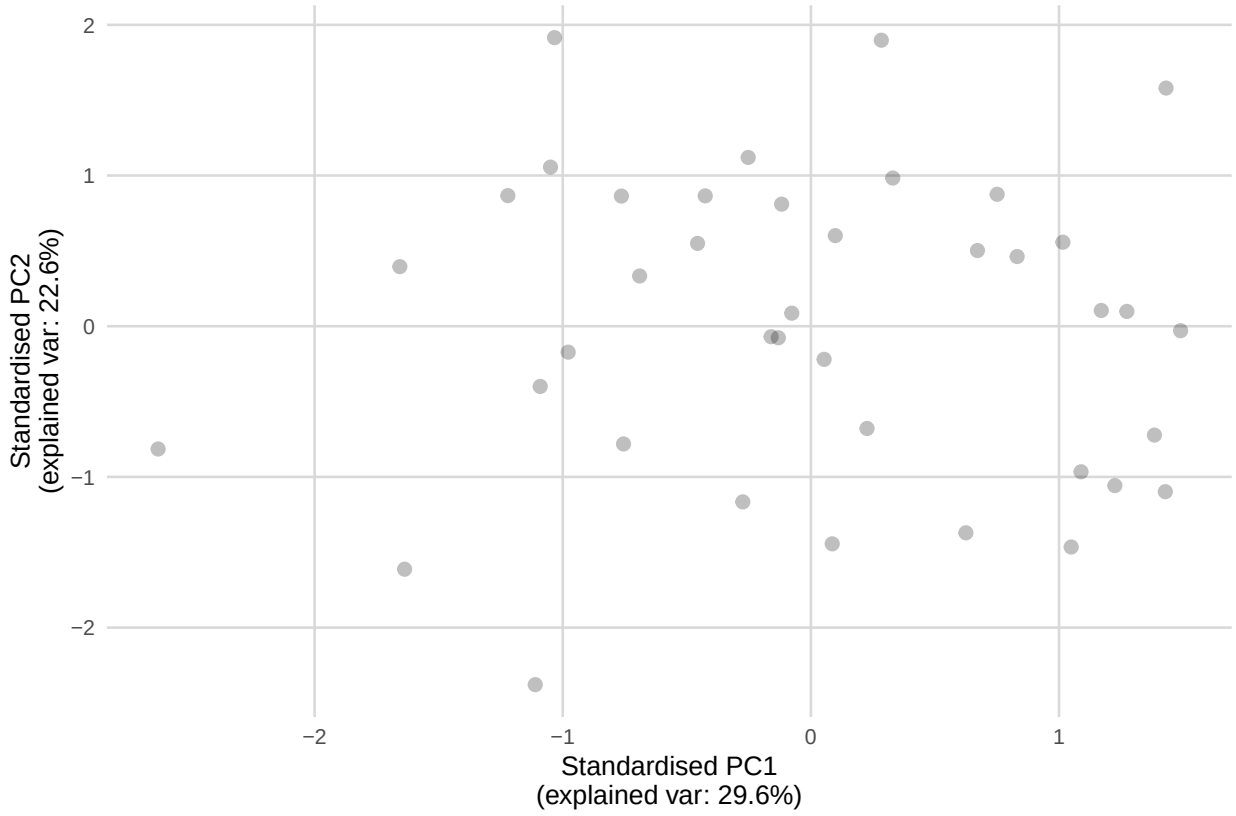
```
## [1] 0.4260203
```

```
library(AMR)
ggplot_pca(pca_data8)
```



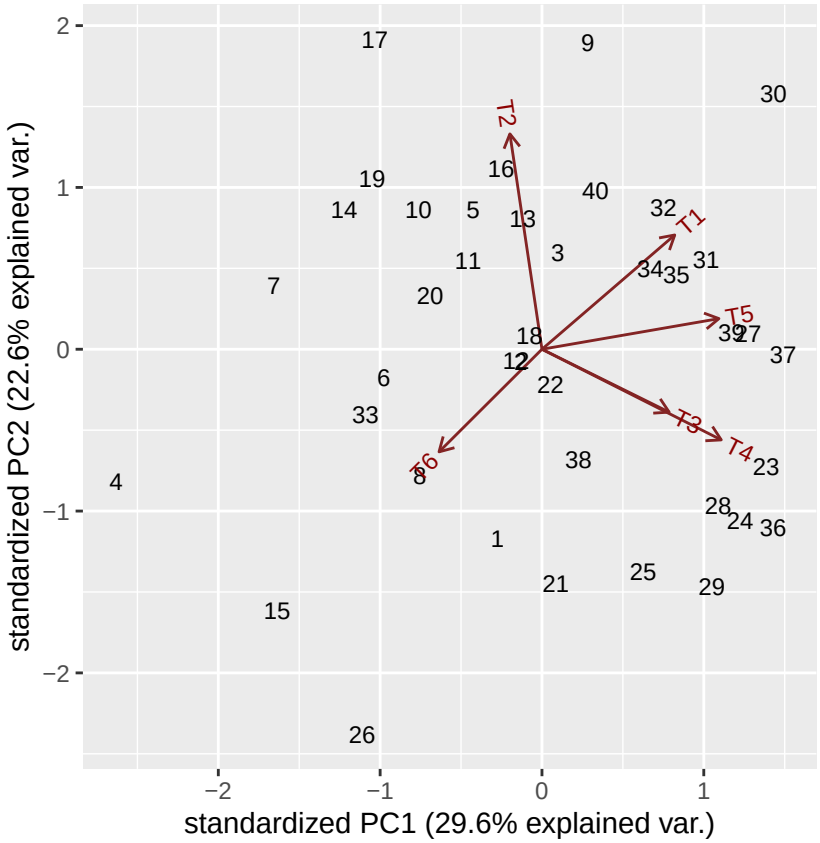
Total explained variance: 52.2%

```
# Scores
ggplot_pca(pca_data8,arrows = FALSE)
```



Total explained variance: 52.2%

```
# Biplot
ggbiplot(pca_data8, labels = 1:40)
```




```
data8
```

```
##      T1 T2  T3  T4 T6  T5
## 1   121 22  74 223 54 254
## 2   108 30  80 175 40 300
## 3   122 49  87 266 41 223
## 4    77 37  66 178 80 209
## 5   140 35  71 175 38 261
## 6   108 37  57 241 59 245
## 7   124 39  52 194 72 242
## 8   130 34  89 200 85 242
## 9   149 55  91 198 50 277
## 10  129 38  72 162 47 268
## 11  154 37  87 170 60 244
## 12  145 33  88 208 51 228
## 13  112 40  60 232 29 279
## 14  120 39  73 159 39 233
## 15  118 21  83 152 88 233
## 16  141 42  80 195 36 241
## 17  135 49  73 152 42 249
## 18  151 37  76 223 74 268
## 19   97 46  83 164 31 243
## 20  109 42  82 188 57 267
## 21  132 17  77 232 50 249
## 22  123 32  79 192 64 315
## 23  129 31  96 250 55 319
```

```
## 24 131 23  67 291 48 310
## 25 110 24  96 239 42 268
## 26  47 22  87 231 40 217
## 27 125 32  87 227 30 324
## 28 129 29 102 234 58 300
## 29 130 26 104 256 58 270
## 30 147 47  82 240 30 322
## 31 159 37  80 227 58 317
## 32 135 41  83 216 39 306
## 33 100 35  83 183 57 242
## 34 149 37  94 227 30 240
## 35 149 38  78 258 42 271
## 36 153 27  89 283 66 291
## 37 136 31  83 257 31 311
## 38  97 36 100 252 30 225
## 39 141 37 105 250 27 243
## 40 164 32  76 187 30 264
```

```
summary(data8)
```

```
##        T1              T2              T3              T4
##  Min.   : 47.0   Min.   :17.00   Min.   : 52.00   Min.   :152.0
##  1st Qu.:116.5   1st Qu.:30.75   1st Qu.: 75.50   1st Qu.:186.0
##  Median :129.5   Median :36.50   Median : 82.50   Median :223.0
##  Mean   :126.9   Mean   :34.90   Mean   : 81.80   Mean   :214.7
##  3rd Qu.:142.0   3rd Qu.:39.00   3rd Qu.: 88.25   3rd Qu.:240.2
##  Max.   :164.0   Max.   :55.00   Max.   :105.00   Max.   :291.0
##        T6              T5
##  Min.   :27.00   Min.   :209.0
##  1st Qu.:37.50   1st Qu.:242.0
##  Median :47.50   Median :262.5
##  Mean   :48.95   Mean   :265.2
##  3rd Qu.:58.00   3rd Qu.:293.2
##  Max.   :88.00   Max.   :324.0
```

Cluster Analysis

Alberto Fallocco, Ana Araújo, Chris Anaya, Leonor Ferreira

2022-12-28

CA, Step 0 - Recalling data, libraries

```
# Run data
data8 <- read.csv("data_8.csv", header = TRUE)
head(data8)
```

```
##    T1 T2 T3  T4 T6  T5
## 1 121 22 74 223 54 254
## 2 108 30 80 175 40 300
## 3 122 49 87 266 41 223
## 4  77 37 66 178 80 209
## 5 140 35 71 175 38 261
## 6 108 37 57 241 59 245
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.16.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
```

```
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ----------------------
##
##
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(laGP)
```

CA, Step 1 - Cluster Analysis

Step 1.1 - Scaling data, calculating distances

```
##scaling the dataframe
data8_s <- scale(data8)
```

```
## Dissimilarity matrix: euclidean distance
dist_data8 <- dist(data8_s,method = "euclidean")
dist_data8
```

```
##           1        2        3        4        5        6        7
## 2  2.4296901
## 3  3.8256756 4.1900923
## 4  3.6475085 4.2053636 4.5348717
## 5  2.4286187 2.0729566 3.5589050 4.1174136
## 6  2.4459964 3.4259952 3.2856022 2.8569850 2.9055467
## 7  3.0854649 3.7873705 4.2026956 2.6529637 2.8597180 1.7260447
## 8  2.8195248 3.6014487 3.7765977 3.2495533 3.4039094 3.4489930 3.2443195
## 9  4.5044529 3.7875558 2.9035609 5.1810388 3.1439009 4.2969165 4.2851416
## 10 2.6373083 1.8631340 3.6529566 3.5978516 0.9178705 2.8294917 2.5617936
## 11 2.9624592 3.0853364 3.5427289 4.1192453 2.0764883 3.7281061 3.3420343
## 12 2.2421668 3.0441419 2.7544148 4.0617379 2.1332437 3.2790746 3.5040747
## 13 3.0352939 2.7357541 3.2725714 4.3879399 2.3806842 2.1835568 3.1716796
## 14 2.9092424 2.4850442 3.3609631 3.3253467 1.3933674 2.9432180 2.8564655
## 15 3.0106463 3.8377279 5.4293442 3.1880174 3.9153153 4.2039265 3.7045624
## 16 2.9609345 2.7857775 2.4122171 4.2379472 1.3974380 3.1036490 3.3138673
## 17 3.9004902 3.1291034 3.4375734 4.0310736 1.8712054 3.4839665 3.0755288
## 18 2.5870639 3.3672952 3.4470122 3.9912712 2.6634123 2.7513711 2.5823131
## 19 3.8299559 2.7276044 3.1289766 3.7723117 2.6125105 3.6563554 3.9659256
## 20 2.7579849 2.0867535 2.9042899 3.0472056 2.2106047 2.6819668 2.8666058
## 21 0.8927675 2.9654218 4.1859449 4.4045394 2.8509384 3.1671132 3.7924541
## 22 2.4885513 1.7583350 4.3083448 4.1417775 2.5719042 3.2491932 3.3111134
## 23 3.0198161 2.8138927 3.8602119 5.4144561 3.6224695 4.1321979 4.8212317
## 24 2.6447629 3.6138279 4.5210950 5.5795726 3.8502944 3.2823189 4.3297895
## 25 2.1298106 2.5027398 3.5115988 4.7068395 3.2967285 3.8114096 4.7422676
```

```
## 26 3.6878365 4.1291190 4.7076548 4.0217329 4.9710389 4.3221070 5.4144250
## 27 3.0690730 1.9592345 3.9144746 5.5987777 2.8728349 4.0337322 4.7863619
## 28 2.8914225 2.8206615 3.8516523 5.1812963 3.5852058 4.3104500 4.8760815
## 29 2.7715080 3.4618964 3.6084111 5.1752631 3.9058529 4.3173817 5.0430492
## 30 4.1839588 3.3202199 3.3960765 6.0584500 3.1353311 4.1669160 4.7261662
## 31 3.1672440 3.0183069 3.9357401 5.3458448 2.7793253 3.6829225 3.8179634
## 32 3.0957464 2.1132133 3.1059989 4.9807509 2.1655532 3.4233432 3.9001157
## 33 2.2770274 2.1982988 3.1966167 2.4673971 2.4023054 2.7056657 2.9801484
## 34 3.1429645 3.3117917 2.3766385 5.2275546 2.5472283 3.9949609 4.5677627
## 35 2.6496929 3.1594563 2.4289046 4.9560501 2.4118677 2.8585788 3.6114375
## 36 2.8828755 3.9612286 3.9712875 5.5793530 3.9779206 3.9740758 4.6140732
## 37 2.8364265 2.6270199 3.5866134 5.6995166 2.9493837 3.7269445 4.6407935
## 38 3.4887873 3.6722169 2.3247666 4.7368455 3.8859865 4.0770892 5.1753916
## 39 3.7477382 3.8564634 2.5114802 5.8078879 3.5915551 4.6762385 5.4688543
## 40 2.8635868 2.7930854 3.8683578 5.2656547 1.3225712 3.7999188 3.8560123
##            8         9        10        11        12        13        14
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9  3.5906865
## 10 3.0620187 2.9147328
## 11 2.0676753 2.6133806 1.9930810
## 12 2.2473159 3.0721970 2.3908798 1.4124590
## 13 4.5629922 3.8721712 2.5537774 4.0106749 3.5857863
## 14 3.4096383 3.3045869 1.2526939 2.3380591 2.3623309 2.7490179
## 15 2.1837825 5.2779495 3.5824585 3.0975255 3.3375122 5.3647871 3.8100150
## 16 3.2889738 2.3167133 1.6976929 1.9143968 1.6671445 2.6349879 1.5314610
## 17 3.7150853 2.3767649 1.5284963 2.3683603 2.9099459 3.0831412 1.4786626
## 18 1.8991227 2.9951458 2.5537884 2.0460546 2.2258601 3.5504927 3.2736419
## 19 4.0471276 3.1667605 2.2927478 3.2627182 3.1822920 3.0336834 1.6598946
## 20 2.3993313 2.5281464 1.5975481 2.2659962 2.3934440 2.8152916 1.9595741
## 21 3.2591730 4.9427150 3.2475085 3.2595012 2.3887231 3.5964122 3.4645783
## 22 2.7457424 3.4736754 2.1785571 2.7843903 3.0651351 3.2651555 3.2290162
## 23 3.3540675 3.6071155 3.6083982 3.5075962 3.1750042 3.8651814 4.3211278
## 24 4.5448110 5.1771951 4.1565551 4.6834400 4.0279709 3.1664842 4.7744589
## 25 3.3594133 4.3063949 3.4537615 3.4531466 2.5380468 3.6656497 3.6001967
## 26 4.9099575 6.3366052 4.9088853 5.4643245 4.5720286 4.6664219 4.4350650
## 27 4.2893623 3.6155011 3.0365891 3.7045207 3.3661997 2.8610926 3.6802708
## 28 2.8772959 3.6173941 3.5473267 3.0996243 2.7560386 4.2484510 4.1325327
## 29 2.8747860 4.0989143 4.0090238 3.3010484 2.5341065 4.5244303 4.3350349
## 30 4.6837044 2.4884386 3.2788336 3.7943686 3.7124217 2.8537553 3.9222021
## 31 3.2913846 2.8422602 2.8204124 2.7860680 2.9837111 3.4085246 3.8242101
## 32 3.6096018 2.2666081 2.1814957 2.7996601 2.7518112 2.4304408 2.9214724
## 33 2.2679445 3.5052438 1.9721871 2.4187602 2.2002160 3.2065926 1.8442195
## 34 3.6008862 2.8587863 3.0039827 2.4863647 1.6054613 3.4826989 2.9023958
## 35 3.4629317 2.8754253 2.8022878 2.8626116 2.2242481 2.4648173 3.2150139
## 36 3.2195550 4.2277363 4.1727701 3.6093925 3.0545964 4.3178857 4.8032689
## 37 4.2788098 3.7567951 3.3075463 3.7636911 3.1776520 2.7091957 3.8811661
## 38 4.0798599 4.1342528 4.0183450 3.9990815 2.9288023 3.8343479 3.5845556
## 39 4.0839511 3.3517086 3.9436904 3.3698118 2.4358272 4.1393893 3.8267969
```

```
## 40 3.9280912 3.3781321 2.1254182 2.3185403 2.2119979 3.0834142 2.4934677
##           15        16        17        18        19        20        21
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16 4.3708741
## 17 4.5668796 1.6327562
## 18 3.4191333 2.7166624 3.2460206
## 19 4.7321765 2.1819904 2.0371464 4.1424669
## 20 3.4963693 2.0742684 2.1567103 2.4374286 2.0072730
## 21 3.3521209 3.3298554 4.4623344 3.0117437 4.4220188 3.4854531
## 22 3.3988963 3.1883660 3.4337261 2.2487395 3.7116397 2.0582249 3.0712622
## 23 4.5727482 3.6141033 4.5458011 2.9198773 4.4002029 3.0365425 3.2035177
## 24 5.2800680 4.2818613 5.2767538 3.4278968 5.3594712 4.1838291 2.6937221
## 25 4.0131241 3.2433145 4.4466985 3.5285741 3.6967818 2.9679964 2.1747167
## 26 4.8156900 4.9414920 5.6683423 5.6060906 4.1743975 4.2472349 4.0083669
## 27 5.1523080 3.0987886 3.9567197 3.5699331 3.6720506 3.0029466 3.2923064
## 28 4.0209872 3.5012160 4.4706941 2.9181770 4.2428024 2.9215798 3.0148341
## 29 4.0340200 3.6403009 4.8619973 3.1240502 4.5318136 3.3585008 2.6981681
## 30 6.1000077 2.8659164 3.4763627 3.4629160 3.8500510 3.2653604 4.4947289
## 31 4.5913259 2.9931662 3.5954999 1.8593679 4.3930951 2.9289440 3.4397761
## 32 4.8390189 2.1102854 2.7727354 2.6598386 3.0049972 2.1267996 3.5081909
## 33 2.8056876 2.3937608 2.7299076 2.8664100 2.1436329 1.2141559 2.9793529
## 34 4.8223727 1.6493682 3.1992382 3.2105463 3.1693790 2.9893812 3.1542430
## 35 4.8747491 2.0648344 3.3128089 2.1972955 3.7252836 2.7970768 2.8535858
## 36 4.5415814 3.9377292 5.0781717 2.4483009 5.3456624 3.8363743 2.7927917
## 37 5.3022543 3.0587788 4.1849636 3.2947959 4.0986970 3.3158950 2.9224800
## 38 5.1179937 3.1138541 4.3138210 4.3784483 3.0685122 3.2298271 3.6804668
## 39 5.4081138 2.6816847 4.1381062 3.9314107 3.6942454 3.5639011 3.6918141
## 40 4.5409298 1.8001812 2.7385357 2.9949297 3.5332469 3.1926327 2.9010494
##           22        23        24        25        26        27        28
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
```

```
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23 2.2037301
## 24 3.2330867 2.8670604
## 25 2.9585204 2.1342905 3.2259683
## 26 5.0102288 5.0309208 5.2052664 3.2596055
## 27 2.4069732 1.8377840 2.8991567 2.3443440 4.9059581
## 28 2.3409088 0.9300820 3.4375498 1.7996319 4.7609599 2.2914851
## 29 3.1540039 1.7639858 3.5118189 1.5619776 4.4244357 2.9780670 1.1645035
## 30 3.2438174 2.8502702 3.6954785 3.8589000 6.2388977 2.1168215 3.3933417
## 31 1.9646176 2.1010598 2.9856606 3.4757460 6.1597199 2.4290694 2.5044704
## 32 2.1023220 2.1577126 3.3065134 2.8831941 5.2523646 1.4740874 2.5060970
## 33 2.5439052 3.4401202 4.3404836 2.6244263 3.3533043 3.4478775 3.1072215
## 34 3.7044627 3.1458154 4.1575278 2.5967216 4.9177788 2.8966571 2.9147266
## 35 2.9391650 2.5575537 2.6584987 2.8684430 5.2197827 2.4589050 2.8419139
## 36 3.0609899 1.9100190 2.4682367 2.8388730 5.5964992 3.1765911 2.0946454
## 37 2.7696701 1.8791918 2.1631937 2.3620256 5.0346657 1.0836521 2.4279978
## 38 4.3741578 3.6115914 4.6084868 2.2175710 3.0825553 3.5362778 3.3378793
## 39 4.2569398 3.1178850 4.4806603 2.5207022 4.8745267 3.1043588 2.8504172
## 40 3.1730991 3.6314005 3.8620420 3.4268030 5.6623647 2.8724929 3.5917026
##           29        30        31        32        33        34        35
## 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
## 27
```

```
## 28
## 29
## 30 3.9952899
## 31 3.1617947 2.2054574
## 32 3.1778078 1.3301521 1.7181333
## 33 3.2486554 4.1713642 3.6595595 2.9487979
## 34 2.7479153 2.9771596 3.1702809 2.4302259 3.1008580
## 35 2.8984970 2.1231716 1.9703486 1.7733219 3.2561911 1.9810620
## 36 1.9374794 3.6557610 2.2905421 3.1557618 4.0454048 3.3591584 2.3727494
## 37 2.7926760 2.0686393 2.2440756 1.7171768 3.7018520 2.6558041 1.7771328
## 38 2.9145395 4.1984073 4.6246144 3.5325382 2.9213470 2.4596007 3.3258394
## 39 2.5169969 3.3327974 3.7499996 2.9558480 3.6420850 1.1765045 2.6053737
## 40 3.8202674 3.0434094 2.6989521 2.3826334 3.3830258 2.1748601 2.2925215
##           36        37        38        39
## 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
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37 2.5553927
## 38 4.1983514 3.4889601
## 39 3.4917848 2.8852650 2.0474472
## 40 3.7529695 2.7372001 4.1488344 3.2415901
```

Step 1.2 - Single Linkage

```
# Hierarchical clustering, Single Linkage method
hc1 <- agnes(dist_data8, method = "single")
hc1$merge
```

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

```
round(hc1$height, 3)
```

```
##  [1] 0.893 2.130 1.758 1.863 0.918 1.253 1.323 1.397 1.479 1.598 1.214 1.649
## [13] 1.412 1.605 1.177 1.660 1.959 1.899 1.859 0.930 1.165 1.562 1.838 1.084
## [25] 1.474 1.330 1.718 1.773 1.910 2.047 2.163 2.184 2.267 2.325 2.381 1.726
## [37] 2.184 2.467 3.083
```

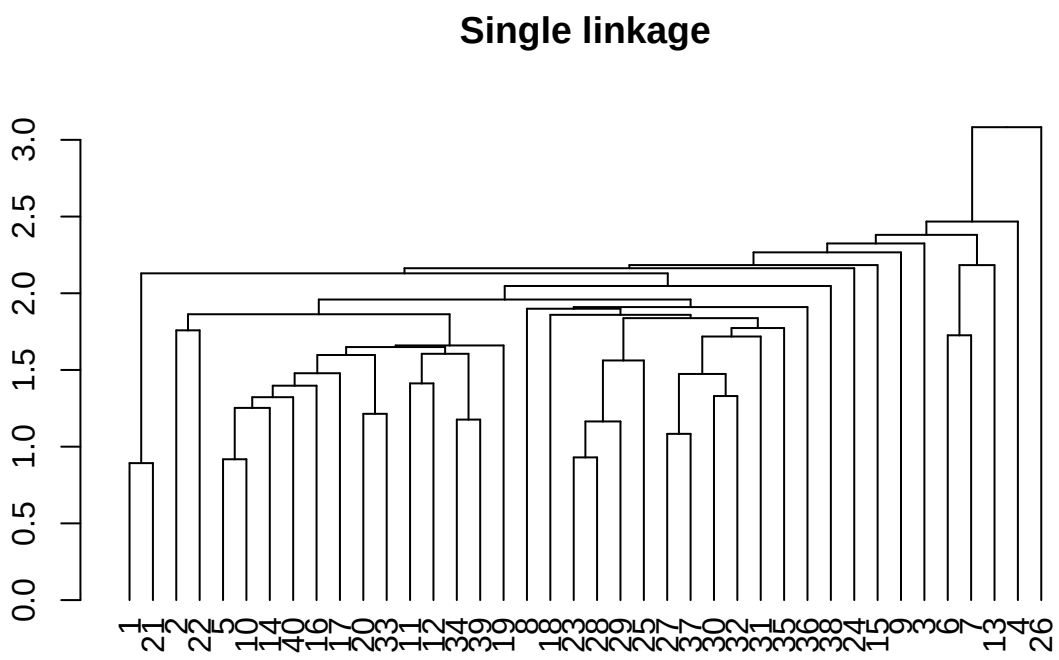
```
hc1$order
```

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

```
hc1$ac
```

```
## [1] 0.4922957
```

```
dend1 <- as.dendrogram(hc1)
plot(dend1, main = "Single linkage")
```



Step 1.3 - Average linkage

```
# Hierarchical clustering, Average Linkage method
hc2 <- agnes(dist_data8, method = "average")
hc2$merge
```

```
##       [,1] [,2]
##  [1,]   -1  -21
##  [2,]   -5  -10
##  [3,]  -23  -28
##  [4,]  -27  -37
##  [5,]  -34  -39
```

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

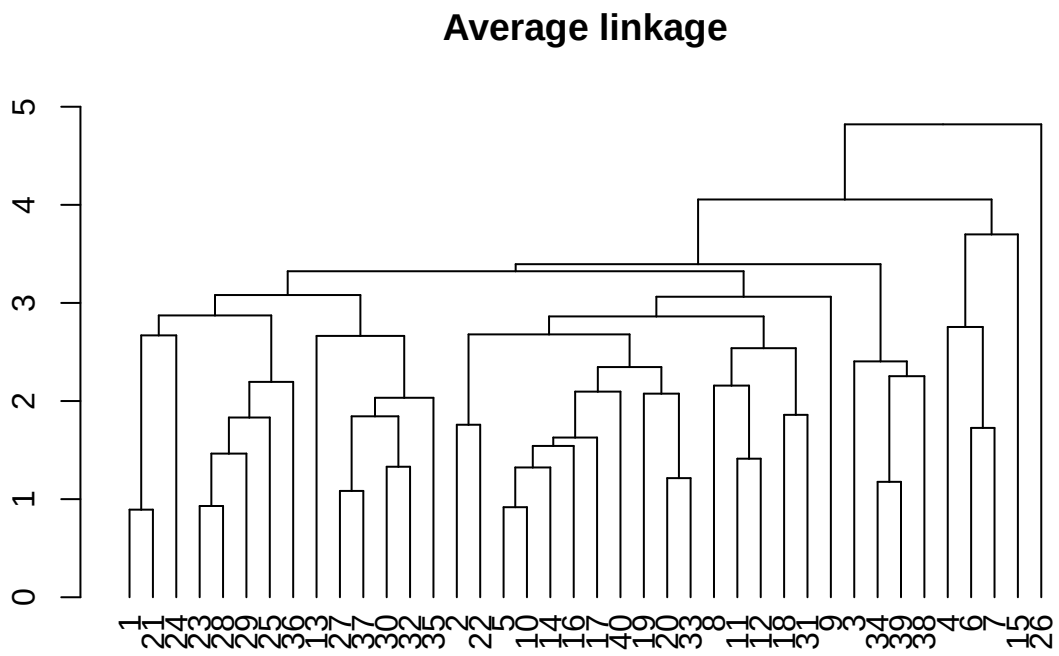
```
hc2$order
```

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

```
hc2$ac
```

```
## [1] 0.6303391
```

```
dend2 <- as.dendrogram(hc2)
plot(dend2, main = "Average linkage")
```



Step 1.4 - Ward

```
# Hierarchical clustering, Ward method
hc3 <- agnes(dist_data8, method = "ward")
hc3$merge
```

```
##       [,1] [,2]
##  [1,]   -1  -21
##  [2,]   -5  -10
##  [3,]  -23  -28
##  [4,]  -27  -37
##  [5,]  -34  -39
##  [6,]  -20  -33
##  [7,]  -30  -32
##  [8,]  -11  -12
##  [9,]    2  -14
## [10,]  -25  -29
## [11,]  -16  -17
## [12,]   -6   -7
## [13,]   -2  -22
## [14,]    9   11
## [15,]  -18  -31
## [16,]    3   10
## [17,]    7  -35
## [18,]   -8  -15
## [19,]  -19    6
```

```
## [20,]   -3  -38
## [21,]    4   17
## [22,]  -24  -36
## [23,]    8  -40
## [24,]   20    5
## [25,]   12  -13
## [26,]   14   23
## [27,]   -9   15
## [28,]    1   22
## [29,]   13   19
## [30,]   -4   18
## [31,]   27   21
## [32,]   28   16
## [33,]   29   26
## [34,]   33   25
## [35,]   30  -26
## [36,]   24   31
## [37,]   32   36
## [38,]   34   35
## [39,]   37   38
```

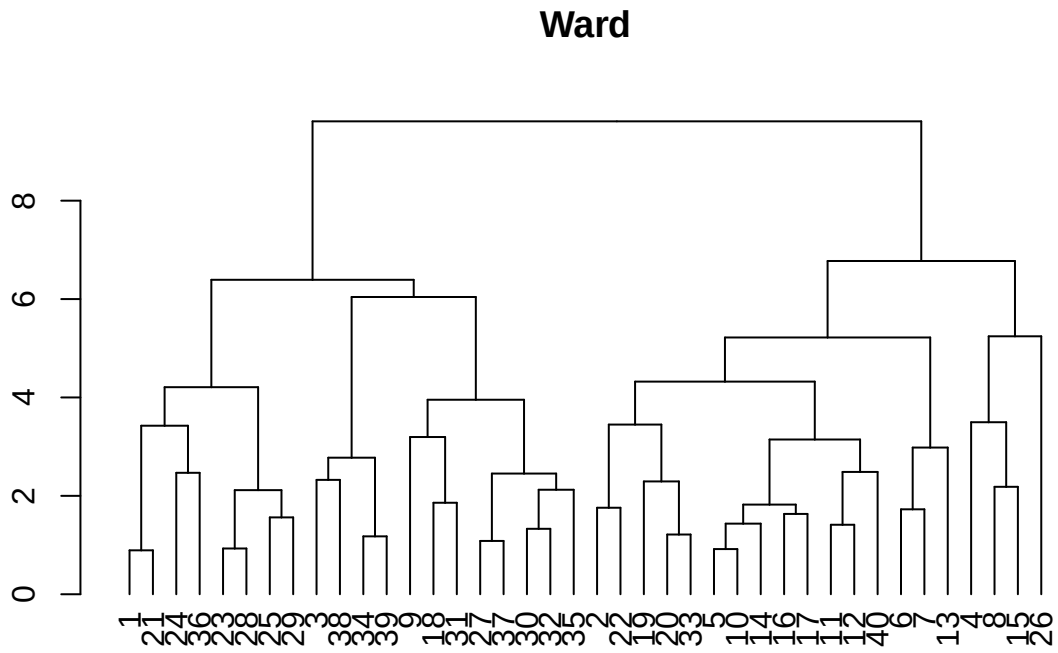
```
hc3$order
```

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

```
hc3$ac
```

```
## [1] 0.8122188
```

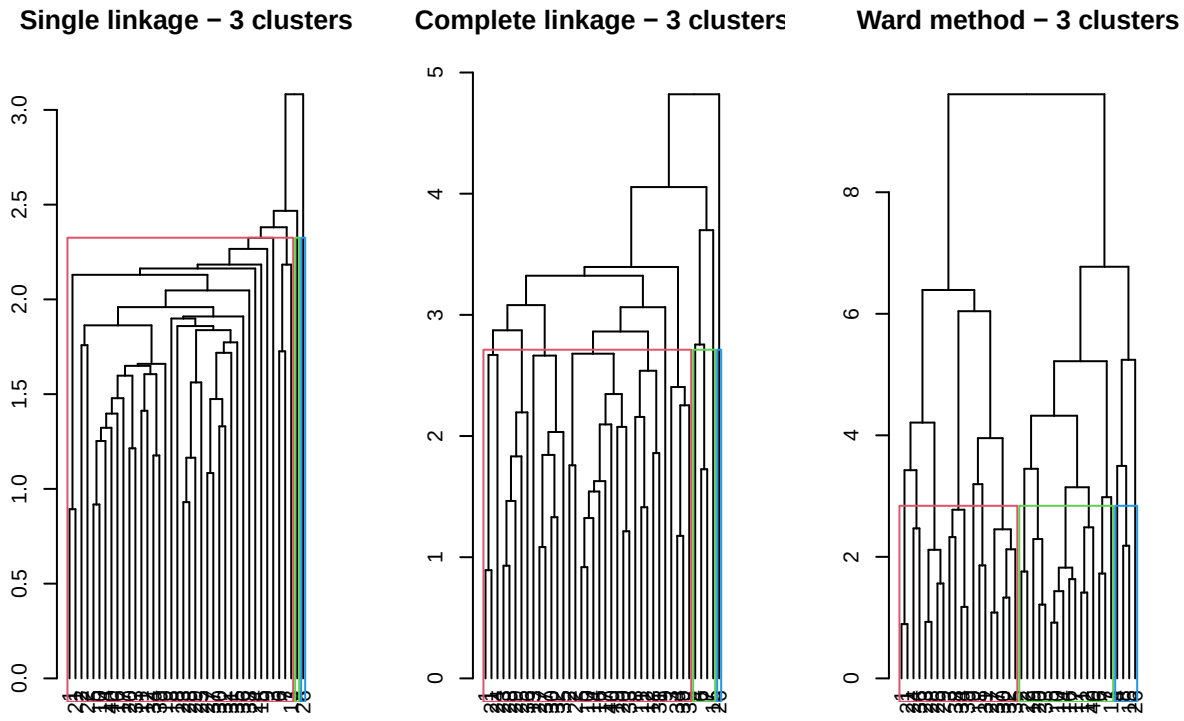
```
dend3 <- as.dendrogram(hc3)
plot(dend3, main = "Ward")
```



CA, Step 2 - Comparing results

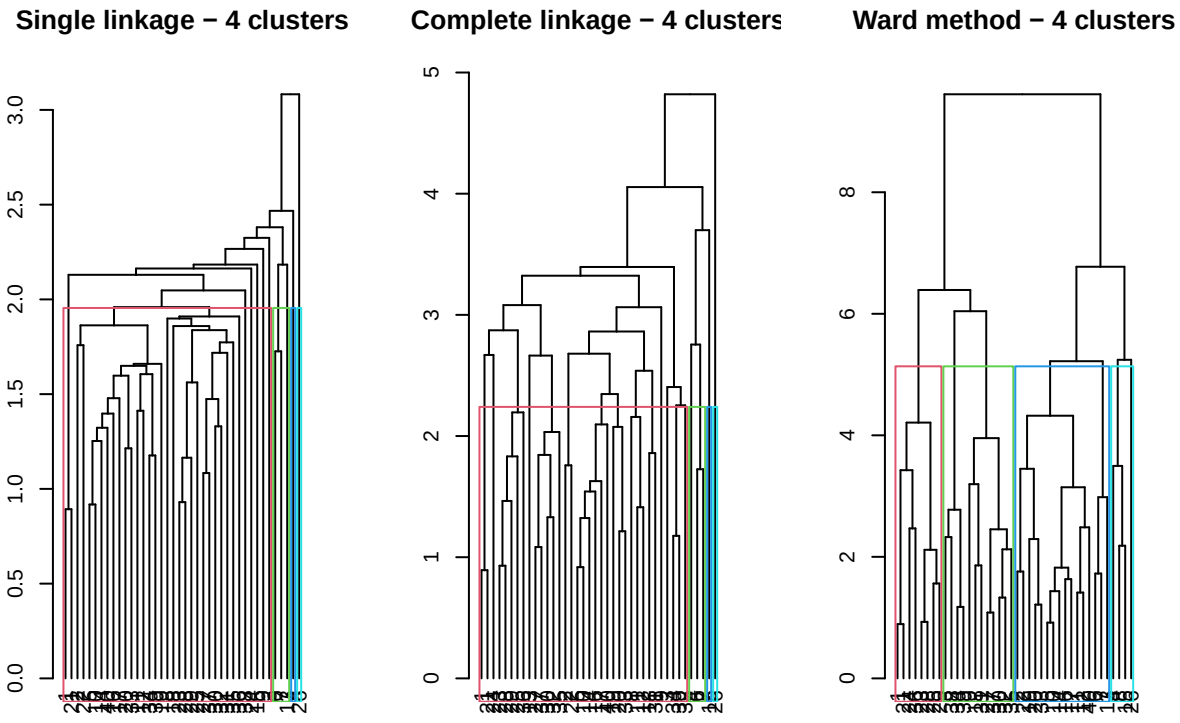
3 clusters

```
par(mfrow=c(1,3))
plot(dend1,main = "Single linkage - 3 clusters")
rect.hclust(hc1, k = 3, border = 2:4)
plot(dend2,main = "Complete linkage - 3 clusters")
rect.hclust(hc2, k = 3, border = 2:4)
plot(dend3,main = "Ward method - 3 clusters")
rect.hclust(hc3, k = 3, border = 2:4)
```



4 clusters

```
par(mfrow=c(1,3))
plot(dend1, main = "Single linkage - 4 clusters")
rect.hclust(hc1, k = 4, border = 2:5)
plot(dend2, main = "Complete linkage - 4 clusters")
rect.hclust(hc2, k = 4, border = 2:5)
plot(dend3, main = "Ward method - 4 clusters")
rect.hclust(hc3, k = 4, border = 2:5)
```


Single linkage − 4 clusters
Complete linkage − 4 clusters
Ward method − 4 clusters
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
1
2
3
4
5
0
2
4
6
8