CLUSTERING ALGORITHMS on STUDENT DATA

Bogdan Tanasa

1. INTRODUCTION

2. DATA EXPLORATION

3. DATA SELECTION

4. DATA FILTERING

5. K-MEANS CLUSTERING

6. HIERARCHICAL CLUSTERING

7. HIERARCHICAL CLUSTERING (AGNES and DIANA)

8. HIERARCHICAL CLUSTERING (THE HEATMAPS)

9. HIERARCHICAL CLUSTERING (CLUSTER TENDENCY)

10. CONCLUSIONS

1. INTRODUCTION

We are using the data from **UCI** : !(https://archive.ics.uci.edu/ml/datasets/Student+Performance)

We are reading a file about **STUDENTS**, and we aim to predict whether they have passed or not the exams **(PASS/no_PASS)**;

The attributes in the **INPUT FILE** are the following:

- 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)
- 2 sex - student's sex (binary: "F" - female or "M" - male)
- 3 age - student's age (numeric: from 15 to 22)
- 4 address - student's home address type (binary: "U" - urban or "R" - rural)
- 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)
- 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)
- 7 Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 8 Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
- 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other")
- 11 reason - reason to choose this school (nominal: close to "home", school "reputation", "course" preference or "other")
- 12 guardian - student's guardian (nominal: "mother", "father" or "other")
- 13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
- 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
- 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)
- 16 schoolsup - extra educational support (binary: yes or no)
- 17 famsup - family educational support (binary: yes or no)
- 18 paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- 19 activities - extra-curricular activities (binary: yes or no)
- 20 nursery - attended nursery school (binary: yes or no)

- 21 higher - wants to take higher education (binary: yes or no)
- 22 internet - Internet access at home (binary: yes or no)
- 23 romantic - with a romantic relationship (binary: yes or no)
- 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
- 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)
- 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)
- 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
- 29 health - current health status (numeric: from 1 - very bad to 5 - very good)
- 30 absences - number of school absences (numeric: from 0 to 93)

NOTES

DATA EXPLORATION and **DATA SELECTION** and **DATA FILTERING** have been presented also in the previous documents, and here, we have not fully included all the figures in those sections.

2. DATA EXPLORATION

```r
options(warn=-1)
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(reshape2))
suppressPackageStartupMessages(library(readxl))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(tidyr))
suppressPackageStartupMessages(library(purrr))
suppressPackageStartupMessages(library(ggpubr))
suppressPackageStartupMessages(library(broom))
suppressPackageStartupMessages(library(tibble))
suppressPackageStartupMessages(library(class))
suppressPackageStartupMessages(library(gmodels))
suppressPackageStartupMessages(library(caret))
suppressPackageStartupMessages(library(e1071))
suppressPackageStartupMessages(library(ISLR))
suppressPackageStartupMessages(library(pROC))
suppressPackageStartupMessages(library(lattice))
suppressPackageStartupMessages(library(kknn))
suppressPackageStartupMessages(library(multiROC))
suppressPackageStartupMessages(library(MLeval))
suppressPackageStartupMessages(library(AppliedPredictiveModeling))
suppressPackageStartupMessages(library(corrplot))
suppressPackageStartupMessages(library(Hmisc))
suppressPackageStartupMessages(library(rattle))
suppressPackageStartupMessages(library(Hmisc))
suppressPackageStartupMessages(library(broom)) # to add : AUGMENT
suppressPackageStartupMessages(library(rattle))
suppressPackageStartupMessages(library(quantmod))
suppressPackageStartupMessages(library(nnet))
suppressPackageStartupMessages(library(NeuralNetTools))
suppressPackageStartupMessages(library(neuralnet))
suppressPackageStartupMessages(library(klaR))
suppressPackageStartupMessages(library(kernlab))
suppressPackageStartupMessages(library(gridExtra))
suppressPackageStartupMessages(library(cluster))
suppressPackageStartupMessages(library(factoextra))
suppressPackageStartupMessages(library(magrittr))
suppressPackageStartupMessages(library(fpc))
suppressPackageStartupMessages(library(gplots))
suppressPackageStartupMessages(library(pheatmap))
# suppressPackageStartupMessages(library(d3heatmap))
suppressPackageStartupMessages(library(clValid))
suppressPackageStartupMessages(library(clustertend))
suppressPackageStartupMessages(library(factoextra))

######################################################
######################################################

FILE1="student.mat.txt"

######################################################
######################################################
# FILE2="student.por.txt"
```

```r
# FILE3="student.mat.and.por.txt"
########################################################
########################################################

# using the data for CLUSTERING

########################################################
########################################################

student <- read.delim(FILE1, sep="\t", header=T, stringsAsFactors=F)

########################################################
########################################################

summary(student)
```

```
##     school              sex                 age         address         
##  Length:395         Length:395         Min.   :15.0   Length:395        
##  Class :character   Class :character   1st Qu.:16.0   Class :character  
##  Mode  :character   Mode  :character   Median :17.0   Mode  :character  
##                                        Mean   :16.7                     
##                                        3rd Qu.:18.0                     
##                                        Max.   :22.0                     
##    famsize            Pstatus               Medu            Fedu      
##  Length:395         Length:395         Min.   :0.000   Min.   :0.000  
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.:2.000  
##  Mode  :character   Mode  :character   Median :3.000   Median :2.000  
##                                        Mean   :2.749   Mean   :2.522  
##                                        3rd Qu.:4.000   3rd Qu.:3.000  
##                                        Max.   :4.000   Max.   :4.000  
##      Mjob               Fjob              reason            guardian        
##  Length:395         Length:395         Length:395         Length:395        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##    traveltime      studytime        failures        schoolsup        
##  Min.   :1.000   Min.   :1.000   Min.   :0.0000   Length:395        
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   Class :character  
##  Median :1.000   Median :2.000   Median :0.0000   Mode  :character  
##  Mean   :1.448   Mean   :2.035   Mean   :0.3342                     
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000                     
##  Max.   :4.000   Max.   :4.000   Max.   :3.0000                     
##     famsup              paid            activities          nursery         
##  Length:395         Length:395         Length:395         Length:395        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##     higher            internet           romantic             famrel     
##  Length:395         Length:395         Length:395         Min.   :1.000  
##  Class :character   Class :character   Class :character   1st Qu.:4.000  
```

```
##  Mode  :character   Mode  :character   Mode  :character   Median :4.000
##                                                            Mean   :3.944
##                                                            3rd Qu.:5.000
##                                                            Max.   :5.000
##     freetime         goout            Dalc            Walc
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :3.000   Median :1.000   Median :2.000
##  Mean   :3.235   Mean   :3.109   Mean   :1.481   Mean   :2.291
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##      health         absences            G1              G2
##  Min.   :1.000   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00
##  1st Qu.:3.000   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00
##  Median :4.000   Median : 4.000   Median :11.00   Median :11.00
##  Mean   :3.554   Mean   : 5.709   Mean   :10.91   Mean   :10.71
##  3rd Qu.:5.000   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00
##  Max.   :5.000   Max.   :75.000   Max.   :19.00   Max.   :19.00
##        G3
##  Min.   : 0.00
##  1st Qu.: 8.00
##  Median :11.00
##  Mean   :10.42
##  3rd Qu.:14.00
##  Max.   :20.00
```

```r
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : chr  "GP" "GP" "GP" "GP" ...
##  $ sex       : chr  "F" "F" "F" "F" ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : chr  "U" "U" "U" "U" ...
##  $ famsize   : chr  "GT3" "GT3" "LE3" "GT3" ...
##  $ Pstatus   : chr  "A" "T" "T" "T" ...
##  $ Medu      : int  4 1 1 4 3 4 2 4 3 3 ...
##  $ Fedu      : int  4 1 1 2 3 3 2 4 2 4 ...
##  $ Mjob      : chr  "at_home" "at_home" "at_home" "health" ...
##  $ Fjob      : chr  "teacher" "other" "other" "services" ...
##  $ reason    : chr  "course" "course" "other" "home" ...
##  $ guardian  : chr  "mother" "father" "mother" "mother" ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : chr  "yes" "no" "yes" "no" ...
##  $ famsup    : chr  "no" "yes" "no" "yes" ...
##  $ paid      : chr  "no" "no" "yes" "yes" ...
##  $ activities: chr  "no" "no" "no" "yes" ...
##  $ nursery   : chr  "yes" "no" "yes" "yes" ...
##  $ higher    : chr  "yes" "yes" "yes" "yes" ...
##  $ internet  : chr  "no" "yes" "yes" "yes" ...
##  $ romantic  : chr  "no" "no" "no" "yes" ...
##  $ famrel    : int  4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : int  3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : int  4 3 2 2 2 2 4 4 2 1 ...
```

```
##  $ Dalc      : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : int  1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : int  3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```r
class(student)
```

```
## [1] "data.frame"
```

Here we are starting to display the data for visual exploration.

```r
################################################################################################
################################################################################################
# 1 school - student's school (binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira)

# unique(student$school)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=school, fill=school))

# ggsave("display.1.school.png")
# student$school = as.character(student$school)
student$school = as.factor(student$school)

################################################################################################
################################################################################################
# 2 sex - student's sex (binary: "F" - female or "M" - male)

# unique(student$sex)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=sex , fill=sex))

# ggsave("display.2.sex.png")
student$sex = as.factor(student$sex)

################################################################################################
################################################################################################
# 3 age - student's age (numeric: from 15 to 22)

# unique(student$age)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=age , fill=age))

# ggplot(data=student, aes(x=age)) +
#       geom_histogram(aes(y=..density..), colour="black", fill="white")+
#       geom_density(alpha=.2, fill="#FF6666")

# ggsave("display.3.age.png")
# AGE is already on the numerical scale !!
student$age = as.integer(student$age)
```

```r
#################################################################################################
#################################################################################################
# 4 address - student's home address type (binary: "U" - urban or "R" - rural)

# unique(student$address) ## [1] "U" "R"

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=address, fill=address))

# ggsave("display.4.address.png")
student$address = as.factor(student$address)

#################################################################################################
#################################################################################################
# 5 famsize - family size (binary: "LE3" - less or equal to 3 or "GT3" - greater than 3)

# unique(student$famsize)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=famsize, fill=famsize))

# ggsave("display.5.famsize.png")
student$famsize = as.factor(student$famsize)

#################################################################################################
#################################################################################################
# 6 Pstatus - parent's cohabitation status (binary: "T" - living together or "A" - apart)

# unique(student$Pstatus)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Pstatus, fill=Pstatus))

# ggsave("display.6.Pstatus.png")
student$Pstatus = as.factor(student$Pstatus)

#################################################################################################
#################################################################################################
# 7 Medu - mother's education (numeric: 0 - none,  1 - primary education (4th grade),
# 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

# unique(student$Medu)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Medu, fill=Medu))

# ggsave("display.7.Medu.png")
# we may wanna use the numerical values in various regression models
# student$Medu = as.integer(student$Medu)
student$Medu = as.factor(student$Medu)

#################################################################################################
#################################################################################################
```

```r
# 8 Fedu - father's education (numeric: 0 - none,  1 - primary education (4th grade),
# 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)

# unique(student$Fedu)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Fedu, fill=Fedu))

# ggsave("display.8.Fedu.png")
# we may wanna use the numerical values in various regression models
# student$Fedu = as.integer(student$Fedu)
student$Fedu = as.factor(student$Fedu)

##################################################################################################
##################################################################################################
# 9 Mjob - mother's job (nominal: "teacher", "health" care related, civil "services"
# (e.g. administrative or police), "at_home" or "other")

# unique(student$Mjob)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Mjob, fill=Mjob))

# ggsave("display.9.Mjob.png")
student$Mjob = as.factor(student$Mjob)

##################################################################################################
##################################################################################################
# 10 Fjob - father's job (nominal: "teacher", "health" care related, civil "services"
# (e.g. administrative or police), "at_home" or "other")

# unique(student$Fjob)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Fjob, fill=Fjob))

# ggsave("display.10.Fjob.png")
student$Fjob = as.factor(student$Fjob)

##################################################################################################
##################################################################################################
# 11 reason - reason to choose this school
# (nominal: close to "home", school "reputation", "course" preference or "other")

# unique(student$reason)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=reason, fill=reason))

# ggsave("display.11.reason.png")
student$reason = as.factor(student$reason)

##################################################################################################
```

```r
###############################################################################################
# 12 guardian - student's guardian (nominal: "mother", "father" or "other")

# unique(student$guardian)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=guardian, fill=guardian))

# ggsave("display.12.guardian.png")
student$guardian = as.factor(student$guardian)

###############################################################################################
###############################################################################################
# 13 traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min.,
# 3 - 30 min. to 1 hour, or 4 - >1 hour)

# unique(student$traveltime)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=traveltime, fill=traveltime))

# ggsave("display.13.traveltime.png")
# student$traveltime = as.factor(student$traveltime)
# we may wanna use the NUMERICAL VALUES :
student$traveltime = as.integer(student$traveltime)

###############################################################################################
###############################################################################################
# 14 studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours,
# 3 - 5 to 10 hours, or 4 - >10 hours)

# unique(student$studytime)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=studytime, fill=studytime))

# ggsave("display.14.studytime.png")
# student$studytime = as.factor(student$studytime)
# we may wanna use the NUMERICAL VALUES :
student$studytime = as.integer(student$studytime)

###############################################################################################
###############################################################################################
# 15 failures - number of past class failures (numeric: n if 1<=n<3, else 4)

# unique(student$failures)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=failures, fill=failures))

# ggsave("display.15.failures.png")
# we may wanna use the NUMERICAL VALUES :
student$failures = as.integer(student$failures)
```

```
################################################################################################
################################################################################################
# 16 schoolsup - extra educational support (binary: yes or no)

# unique(student$schoolsup)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=schoolsup, fill=schoolsup))

# ggsave("display.16.schoolsup.png")
student$schoolsup = as.factor(student$schoolsup)

################################################################################################
################################################################################################
# 17 famsup - family educational support (binary: yes or no)

# unique(student$famsup)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=famsup, fill=famsup))

# ggsave("display.17.famsup.png")
student$famsup = as.factor(student$famsup)

################################################################################################
################################################################################################
# 18 paid - extra paid classes within the course subject (Math or Portuguese)
# (binary: yes or no)

# unique(student$paid)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=paid, fill=paid))

# ggsave("display.18.paid.png")
student$paid = as.factor(student$paid)

################################################################################################
################################################################################################
# 19 activities - extra-curricular activities (binary: yes or no)

# unique(student$activities)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=activities, fill=activities))

# ggsave("display.19.activities.png")
student$activities = as.factor(student$activities)

################################################################################################
################################################################################################
# 20 nursery - attended nursery school (binary: yes or no)
```

```r
# unique(student$nursery)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=nursery, fill=nursery))

# ggsave("display.20.nursery.png")
student$nursery = as.factor(student$nursery)

#################################################################################################
#################################################################################################
# 21 higher - wants to take higher education (binary: yes or no)

# unique(student$higher)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=higher, fill=higher))

# ggsave("display.21.higher.png")
student$higher = as.factor(student$higher)

#################################################################################################
#################################################################################################
# 22 internet - Internet access at home (binary: yes or no)

# unique(student$internet)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=internet, fill=internet))

# ggsave("display.22.internet.png")
student$internet = as.factor(student$internet)

#################################################################################################
#################################################################################################
# 23 romantic - with a romantic relationship (binary: yes or no)

# unique(student$romantic)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=romantic, fill=romantic))

# ggsave("display.23.romantic.png")
student$romantic = as.factor(student$romantic)

#################################################################################################
#################################################################################################
# 24 famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)

# unique(student$famrel)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=famrel, fill=famrel))
```

```
# ggsave("display.24.famrel.png")
# i believe that we can keep these as numerical : or factor ?
student$famrel = as.factor(student$famrel)

################################################################################################
################################################################################################
# 25 freetime - free time after school (numeric: from 1 - very low to 5 - very high)

# unique(student$freetime)

# ggplot(data = student) +
#       geom_bar(mapping = aes(x=freetime, fill=freetime))

# ggsave("display.25.freetime.png")
# i believe that we can keep these as numerical :
student$freetime = as.factor(student$freetime)

################################################################################################
################################################################################################
# 26 goout - going out with friends (numeric: from 1 - very low to 5 - very high)

# unique(student$goout)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=goout, fill=goout))

# ggsave("display.26.goout.png")
# i believe that we can keep these as numerical :
student$goout = as.factor(student$goout)

################################################################################################
################################################################################################
# 27 Dalc - workday alcohol consumption (numeric: from 1 - very low to 5 - very high)

# unique(student$Dalc)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Dalc, fill=Dalc))

# ggsave("display.27.Dalc.png")
# i believe that we can keep these as numerical :
student$Dalc = as.factor(student$Dalc)

################################################################################################
################################################################################################
# 28 Walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)

# unique(student$Walc)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=Walc, fill=Walc))

# ggsave("display.28.Walc.png")
```

```r
# i believe that we can keep these as numerical :
student$Walc = as.factor(student$Walc)

#################################################################################################
#################################################################################################
# 29 health - current health status (numeric: from 1 - very bad to 5 - very good)

# unique(student$health)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=health, fill=health))

# ggsave("display.29.health.png")
# i believe that we can keep these as numerical :
student$health = as.factor(student$health)

#################################################################################################
#################################################################################################
# 30 absences - number of school absences (numeric: from 0 to 93)

# unique(student$absences)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=absences, fill=absences))

# ggplot(data=student, aes(x=absences)) +
#      geom_histogram(aes(y=..density..), colour="black", fill="white")+
#      geom_density(alpha=.2, fill="#FF6666")

# ggsave("display.30.absences.png")
# i believe that we can keep these as numerical :
student$absences = as.integer(student$absences)

#################################################################################################
#################################################################################################
# $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...

# unique(student$G1)

# ggplot(data = student) +
#         geom_bar(mapping = aes(x=G1, fill=G1))

# ggplot(data=student, aes(x=G1)) +
#       geom_histogram(aes(y=..density..), colour="black", fill="white")+
#       geom_density(alpha=.2, fill="#FF6666")

# ggsave("display.0.G1.png")
# i believe that we can keep these as numerical, although we may not need it :
student$G1 = as.integer(student$G1)

#################################################################################################
#################################################################################################
# $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
```

```r
# unique(student$G2)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=G2, fill=G2))

# ggplot(data=student, aes(x=G2)) +
#       geom_histogram(aes(y=..density..), colour="black", fill="white")+
#       geom_density(alpha=.2, fill="#FF6666")

# ggsave("display.0.G2.png")
# i believe that we can keep these as numerical, although we may not need it :
student$G2 = as.integer(student$G2)

###############################################################################################
###############################################################################################
# $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...

# unique(student$G3)

# ggplot(data = student) +
#        geom_bar(mapping = aes(x=G3, fill=G3))

# ggplot(data=student, aes(x=G3)) +
#       geom_histogram(aes(y=..density..), colour="black", fill="white")+
#       geom_density(alpha=.2, fill="#FF6666")

# ggsave("display.0.G3.png")
# i believe that we can covert it into RANGES of VALUES :
student$G3 = as.integer(student$G3)

###############################################################################################
summary(student)
```

```
##  school   sex          age       address famsize   Pstatus Medu    Fedu
##  GP:349   F:208   Min.   :15.0   R: 88   GT3:281   A: 41   0:  3   0:  2
##  MS: 46   M:187   1st Qu.:16.0   U:307   LE3:114   T:354   1: 59   1: 82
##                   Median :17.0                             2:103   2:115
##                   Mean   :16.7                             3: 99   3:100
##                   3rd Qu.:18.0                             4:131   4: 96
##                   Max.   :22.0
##       Mjob          Fjob              reason      guardian     traveltime
##  at_home : 59   at_home : 20   course     :145   father: 90   Min.   :1.000
##  health  : 34   health  : 18   home       :109   mother:273   1st Qu.:1.000
##  other   :141   other   :217   other      : 36   other : 32   Median :1.000
##  services:103   services:111   reputation:105                 Mean   :1.448
##  teacher : 58   teacher : 29                                  3rd Qu.:2.000
##                                                               Max.   :4.000
##    studytime        failures      schoolsup famsup     paid     activities
##  Min.   :1.000   Min.   :0.0000   no :344   no :153   no :214   no :194
##  1st Qu.:1.000   1st Qu.:0.0000   yes: 51   yes:242   yes:181   yes:201
##  Median :2.000   Median :0.0000
##  Mean   :2.035   Mean   :0.3342
##  3rd Qu.:2.000   3rd Qu.:0.0000
```

```
##  Max.   :4.000   Max.   :3.0000
##  nursery   higher    internet  romantic  famrel  freetime goout   Dalc
##  no : 81   no : 20   no : 66   no :263   1:  8   1: 19    1: 23   1:276
##  yes:314   yes:375   yes:329   yes:132   2: 18   2: 64    2:103   2: 75
##                                          3: 68   3:157    3:130   3: 26
##                                          4:195   4:115    4: 86   4:  9
##                                          5:106   5: 40    5: 53   5:  9
##
##  Walc    health     absences            G1              G2
##  1:151   1: 47   Min.   : 0.000   Min.   : 3.00   Min.   : 0.00
##  2: 85   2: 45   1st Qu.: 0.000   1st Qu.: 8.00   1st Qu.: 9.00
##  3: 80   3: 91   Median : 4.000   Median :11.00   Median :11.00
##  4: 51   4: 66   Mean   : 5.709   Mean   :10.91   Mean   :10.71
##  5: 28   5:146   3rd Qu.: 8.000   3rd Qu.:13.00   3rd Qu.:13.00
##                  Max.   :75.000   Max.   :19.00   Max.   :19.00
##        G3
##  Min.   : 0.00
##  1st Qu.: 8.00
##  Median :11.00
##  Mean   :10.42
##  3rd Qu.:14.00
##  Max.   :20.00
```

```r
str(student)
```

```
## 'data.frame':    395 obs. of  33 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 5 4 5 3 5 4 4 ...
##  $ Fedu      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 3 4 4 3 5 3 5 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : Factor w/ 5 levels "1","2","3","4",..: 4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : Factor w/ 5 levels "1","2","3","4",..: 4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 5 5 5 3 1 1 5 ...
```

```
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G1        : int  5 5 7 15 6 15 12 6 16 14 ...
##  $ G2        : int  6 5 8 14 10 15 12 5 18 15 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```r
class(student)
```

```
## [1] "data.frame"
```

```r
###############################################################################################
# knitr::kable(summary(student, format = "html"))
###############################################################################################
```

3. DATA SELECTION

```r
## the OUTPUT VARIABLES is G3
## we may remove G1 and G2
## and some other features

student1 <- subset(student, select = -c(G1, G2))

student2 <- subset(student1,
                   select = -c(school, sex, address, famsize, Pstatus,
                   Mjob, Fjob, reason, guardian, schoolsup, famsup,
                   paid, activities, nursery,
                   higher, internet, romantic))

### shall we decide to keep ALL the FEATURES (ATTRIBUTES)
student2 = student1

str(student2)
```

```
## 'data.frame':    395 obs. of  31 variables:
##  $ school    : Factor w/ 2 levels "GP","MS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex       : Factor w/ 2 levels "F","M": 1 1 1 1 1 2 2 1 2 2 ...
##  $ age       : int  18 17 15 15 16 16 16 17 15 15 ...
##  $ address   : Factor w/ 2 levels "R","U": 2 2 2 2 2 2 2 2 2 2 ...
##  $ famsize   : Factor w/ 2 levels "GT3","LE3": 1 1 2 1 1 2 2 1 2 1 ...
##  $ Pstatus   : Factor w/ 2 levels "A","T": 1 2 2 2 2 2 2 1 1 2 ...
##  $ Medu      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 5 4 5 3 5 4 4 ...
##  $ Fedu      : Factor w/ 5 levels "0","1","2","3",..: 5 2 2 3 4 4 3 5 3 5 ...
##  $ Mjob      : Factor w/ 5 levels "at_home","health",..: 1 1 1 2 3 4 3 3 4 3 ...
##  $ Fjob      : Factor w/ 5 levels "at_home","health",..: 5 3 3 4 3 3 3 5 3 3 ...
##  $ reason    : Factor w/ 4 levels "course","home",..: 1 1 3 2 2 4 2 2 2 2 ...
##  $ guardian  : Factor w/ 3 levels "father","mother",..: 2 1 2 2 1 2 2 2 2 2 ...
##  $ traveltime: int  2 1 1 1 1 1 1 2 1 1 ...
##  $ studytime : int  2 2 2 3 2 2 2 2 2 2 ...
##  $ failures  : int  0 0 3 0 0 0 0 0 0 0 ...
##  $ schoolsup : Factor w/ 2 levels "no","yes": 2 1 2 1 1 1 1 2 1 1 ...
##  $ famsup    : Factor w/ 2 levels "no","yes": 1 2 1 2 2 2 1 2 2 2 ...
##  $ paid      : Factor w/ 2 levels "no","yes": 1 1 2 2 2 2 1 1 2 2 ...
##  $ activities: Factor w/ 2 levels "no","yes": 1 1 1 2 1 2 1 1 1 2 ...
##  $ nursery   : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...
##  $ higher    : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ internet  : Factor w/ 2 levels "no","yes": 1 2 2 2 1 2 2 1 2 2 ...
##  $ romantic  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ famrel    : Factor w/ 5 levels "1","2","3","4",..: 4 5 4 3 4 5 4 4 4 5 ...
##  $ freetime  : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 2 3 4 4 1 2 5 ...
##  $ goout     : Factor w/ 5 levels "1","2","3","4",..: 4 3 2 2 2 2 4 4 2 1 ...
##  $ Dalc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 1 ...
##  $ Walc      : Factor w/ 5 levels "1","2","3","4",..: 1 1 3 1 2 2 1 1 1 1 ...
##  $ health    : Factor w/ 5 levels "1","2","3","4",..: 3 3 3 5 5 5 3 1 1 5 ...
##  $ absences  : int  6 4 10 2 4 10 0 6 0 0 ...
##  $ G3        : int  6 6 10 15 10 15 11 6 19 15 ...
```

```r
### depending on the algorithm that we may choose to use
student2$G3 = as.factor(student2$G3)
### student2$G3 = as.integer(student2$G3)
```

```r
table(student2$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

```r
### for simplicity, to work with a copy of STUDENT2, let's call it STUDENT3

student3 = subset(student2,
                  select= c(age, traveltime, studytime, failures, absences, G3))

### shall we decide to keep ALL the FEATURES (ATTRIBUTES)
### student3 = student2

table(student3$G3)
```

```
##
##  0  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 38  1  7 15  9 32 28 56 47 31 31 27 33 16  6 12  5  1
```

4. DATA FILTERING

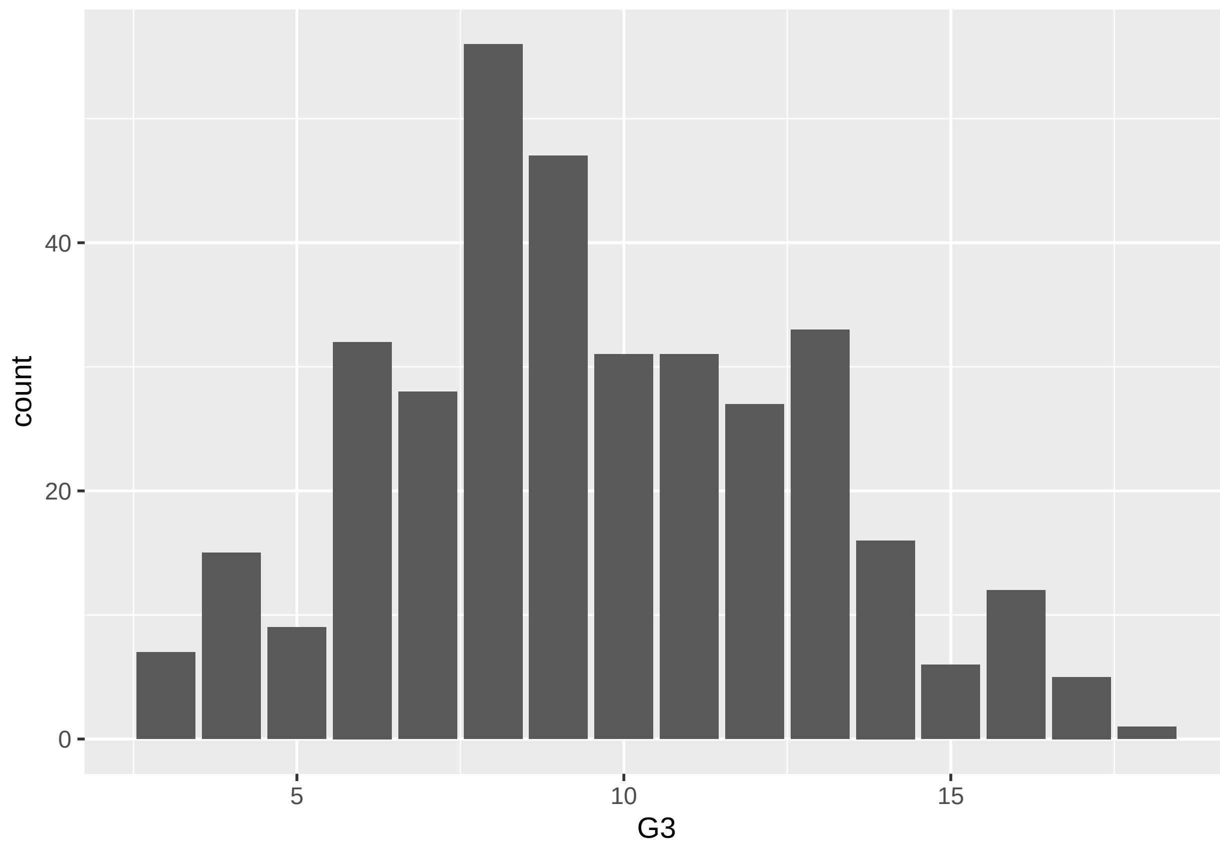
```r
## in order to KEEP the RECORDS where the GRADE 3 is > 2 :

dim(student3)
```

```
## [1] 395   6
```

```r
student3$G3 = as.integer(student3$G3)

student4 = student3[student3$G3 > 2, ]

dim(student4)
```

```
## [1] 356   6
```

```r
ggplot(data = student4) +
       geom_bar(mapping = aes(x=G3, fill=G3))
```




```r
ggsave("display.0.G3.after.filtering.grade3.frequency.png")
```

```
## Saving 6.5 x 4.5 in image
```

```r
student3 = student4

## FOR CLUSTERING, we may not use the R code below ;

## TRANSFORMING G3 into RANGES of PASS and NO-PASS :

## student3$G3 = as.integer(student3$G3)
```

```r
## student3$RESULT[student3$G3 <= 10] = "NO_PASS"
## student3$RESULT[student3$G3 >=10 ] = "PASS"

## student3 <- subset(student3, select = -c(G3))

## student3$RESULT = as.factor(student3$RESULT)

## DISPLAYING THE FEATURES (ATTRIBUTES) in THE CURRENT DATASET :

colnames(student3)
```

```
## [1] "age"        "traveltime" "studytime"  "failures"   "absences"
## [6] "G3"
```

```r
summary(student3)
```

```
##       age          traveltime      studytime        failures
##  Min.   :15.00   Min.   :1.000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:16.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000
##  Median :17.00   Median :1.000   Median :2.000   Median :0.0000
##  Mean   :16.65   Mean   :1.433   Mean   :2.042   Mean   :0.2669
##  3rd Qu.:18.00   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:0.0000
##  Max.   :22.00   Max.   :4.000   Max.   :4.000   Max.   :3.0000
##     absences            G3
##  Min.   : 0.000   Min.   : 3.000
##  1st Qu.: 2.000   1st Qu.: 7.000
##  Median : 4.000   Median : 9.000
##  Mean   : 6.272   Mean   : 9.545
##  3rd Qu.: 8.000   3rd Qu.:12.000
##  Max.   :75.000   Max.   :18.000
```

5. DATA SCALING

```r
student3 <- na.omit(student3)
dim(student3)
```

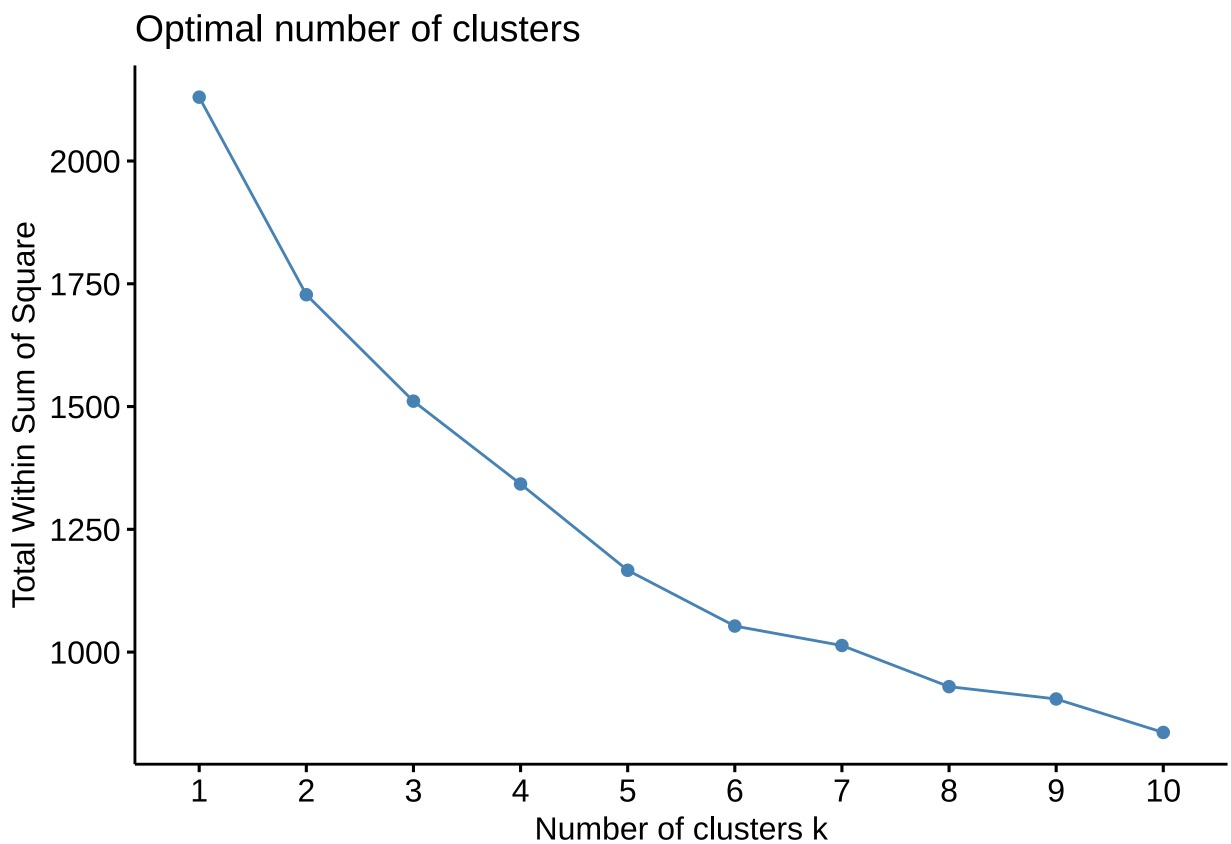
```
## [1] 356   6
```

```r
## in order to SCALE the DATA

student3_scaled = scale(student3)
summary(student3_scaled)
```

```
##       age            traveltime        studytime           failures
##  Min.   :-1.3028   Min.   :-0.6300   Min.   :-1.25097   Min.   :-0.4004
##  1st Qu.:-0.5154   1st Qu.:-0.6300   1st Qu.:-1.25097   1st Qu.:-0.4004
##  Median : 0.2721   Median :-0.6300   Median :-0.05058   Median :-0.4004
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 1.0595   3rd Qu.: 0.8263   3rd Qu.:-0.05058   3rd Qu.:-0.4004
##  Max.   : 4.2093   Max.   : 3.7390   Max.   : 2.35020   Max.   : 4.1014
##     absences             G3
##  Min.   :-0.7690   Min.   :-2.0405
##  1st Qu.:-0.5238   1st Qu.:-0.7934
##  Median :-0.2786   Median :-0.1699
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2118   3rd Qu.: 0.7654
##  Max.   : 8.4259   Max.   : 2.6360
```

5. K-MEANS CLUSTERING

```r
## Looking at the OPTIMAL NUMBER of CLUSTERS by WSS method
## i.e. "within cluster sums of squares"

optimalclusters_WSS <- fviz_nbclust(student3_scaled, kmeans, method="wss")
print(optimalclusters_WSS)
```

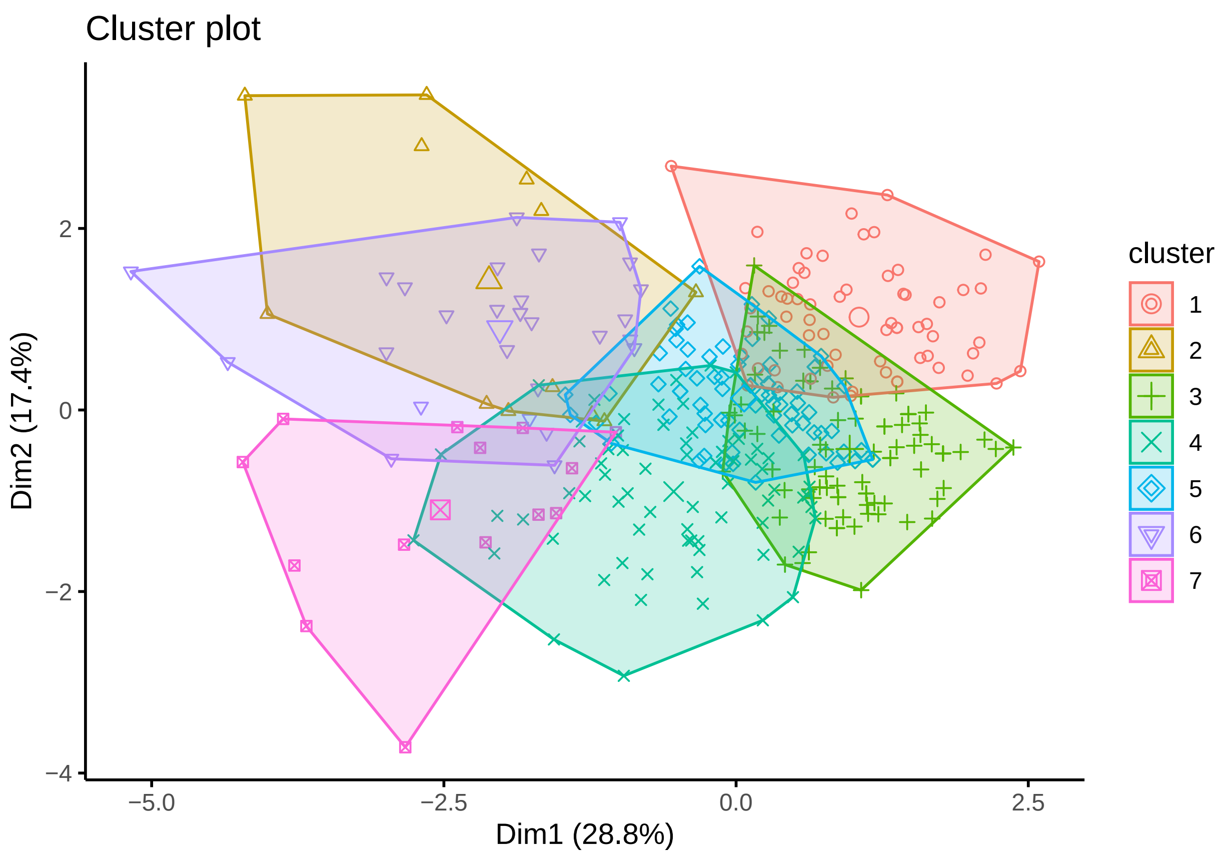



```r
## Running the K-MEANS clustering algorithm by using 7 CLUSTERS
kmeans.df <- kmeans(student3_scaled, 7, nstart=25)
print(kmeans.df)
```

```
## K-means clustering with 7 clusters of sizes 64, 12, 86, 73, 79, 28, 14
##
## Cluster means:
##           age traveltime  studytime   failures    absences         G3
## 1 -0.04783343 -0.2431478  1.5812008 -0.3535504 -0.28434990  0.5461968
## 2  0.40331227  0.2195404 -0.2506430  0.2248110  3.73651831 -0.4556840
## 3 -0.53369731 -0.4775788 -0.5949407 -0.3306486 -0.35273277  0.9212931
## 4  0.23970875  1.3649871 -0.3301209 -0.2154374 -0.17615723 -0.3236451
## 5 -0.18644781 -0.6115509 -0.1417470 -0.3814494  0.01004804 -0.7026650
## 6  1.67823830 -0.3699265 -0.1791913  1.6361019  0.54894103 -0.5262030
## 7 -0.40289095  0.9303681 -0.4792886  3.4582749  0.02789474 -1.0606631
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   4   5   7   1   5   3   5   4   3   3   5   4   3   4   1   3   1   4   7   5
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
```

```
##   3   3   3   4   5   7   3   3   5   5   3   3   3   3   3   4   1   1   1   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   2   3   3   3   4   5   5   1   3   5   4   3   4   3   3   5   3   3   5   3
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   5   4   5   5   5   4   1   1   4   1   1   1   7   3   2   5   1   1   7   5
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   1   5   3   5   7   5   1   4   5   5   3   5   4   1   1   3   5   3   5
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   5   1   3   2   3   1   1   1   4   1   3   1   3   3   5   3   3   3   4   3
## 121 122 123 124 125 126 127 128 130 133 134 139 140 142 143 144 146 148 150 152
##   3   1   4   3   5   3   5   6   3   5   5   3   3   7   1   3   5   5   7   3
## 153 155 156 157 158 159 160 162 164 165 166 167 168 170 172 173 175 176 177 178
##   7   3   5   3   7   4   3   7   5   7   4   5   3   3   3   5   4   4   4   5
## 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   5   5   5   3   3   2   5   5   3   3   5   5   3   5   4   5   3   3   3   4
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 218 219
##   2   5   3   5   5   5   1   2   7   3   4   4   1   5   3   6   5   4   6   4
## 220 221 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 241
##   1   4   3   4   1   6   3   5   4   1   3   4   5   3   5   1   3   5   4   4
## 242 244 246 247 248 249 250 251 252 253 254 255 256 257 258 259 261 262 263 264
##   4   3   4   4   6   6   3   4   4   6   4   3   4   1   6   3   3   5   1   5
## 266 267 268 269 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286
##   4   5   4   5   6   1   4   3   4   4   2   2   6   4   2   6   1   4   5   5
## 287 288 289 290 291 292 293 294 295 296 298 299 300 301 302 303 304 305 306 307
##   1   1   1   3   5   1   6   1   1   5   4   1   3   5   4   1   1   6   6   3
## 308 309 310 312 313 314 315 316 318 319 320 321 322 323 324 325 326 327 328 329
##   2   6   6   4   6   6   6   2   5   1   5   2   5   1   1   1   1   3   4   5
## 330 331 332 336 337 339 340 341 343 345 346 347 348 349 350 351 352 353 354 355
##   1   1   1   1   6   1   5   6   3   1   1   1   1   1   6   7   4   6   4   4
## 356 357 358 359 360 361 362 363 364 365 366 367 369 370 371 372 373 374 375 376
##   5   4   4   4   1   4   4   4   3   4   4   1   4   4   6   4   1   5   1   4
## 377 378 379 380 381 382 383 385 386 387 389 391 392 393 394 395
##   6   5   3   5   3   4   4   6   4   4   5   6   3   6   4   5
##
## Within cluster sum of squares by cluster:
## [1] 176.81415  87.44409 167.04971 225.08240 138.98671 106.62924  68.82676
##  (between_SS / total_SS =  54.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
## Visualization :
optimalclusters_WSS_fviz = fviz_cluster(kmeans.df,
                                        data = student3_scaled,
                                        geom = c("point"),
                                        ggtheme=theme_classic())

gridExtra::grid.arrange(optimalclusters_WSS_fviz)
```




```r
## Numerical SUMMARY of the CLUSTERS
clusters_aggregate <- aggregate(student3, by=list(cluster=kmeans.df$cluster), mean)
print(clusters_aggregate)
```

```
##   cluster      age traveltime studytime   failures  absences        G3
## 1       1 16.59375   1.265625  3.359375 0.03125000  3.953125 11.296875
## 2       2 17.16667   1.583333  1.833333 0.41666667 36.750000  8.083333
## 3       3 15.97674   1.104651  1.546512 0.04651163  3.395349 12.500000
## 4       4 16.95890   2.369863  1.767123 0.12328767  4.835616  8.506849
## 5       5 16.41772   1.012658  1.924051 0.01265823  6.354430  7.291139
## 6       6 18.78571   1.178571  1.892857 1.35714286 10.750000  7.857143
## 7       7 16.14286   2.071429  1.642857 2.57142857  6.500000  6.142857
```

```r
student3 %>%
  mutate(Cluster = kmeans.df$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 7 x 7
##   Cluster   age traveltime studytime failures absences    G3
##     <int> <dbl>      <dbl>     <dbl>    <dbl>    <dbl> <dbl>
## 1       1  16.6       1.27      3.36   0.0312     3.95 11.3
## 2       2  17.2       1.58      1.83   0.417     36.8   8.08
## 3       3  16.0       1.10      1.55   0.0465     3.40 12.5
## 4       4  17.0       2.37      1.77   0.123      4.84  8.51
## 5       5  16.4       1.01      1.92   0.0127     6.35  7.29
## 6       6  18.8       1.18      1.89   1.36      10.8   7.86
```

```
## 7       7  16.1       2.07      1.64   2.57       6.5   6.14
```

```r
### Just in case that we will need the information on CLUSTERS in the BIG DATA FRAME

clusterbind_student3 <- cbind(student3, kmeans.df$cluster)
head(clusterbind_student3)
```

```
##   age traveltime studytime failures absences G3 kmeans.df$cluster
## 1  18          2         2        0        6  4                 4
## 2  17          1         2        0        4  4                 5
## 3  15          1         2        3       10  8                 7
## 4  15          1         3        0        2 13                 1
## 5  16          1         2        0        4  8                 5
## 6  16          1         2        0       10 13                 3
```

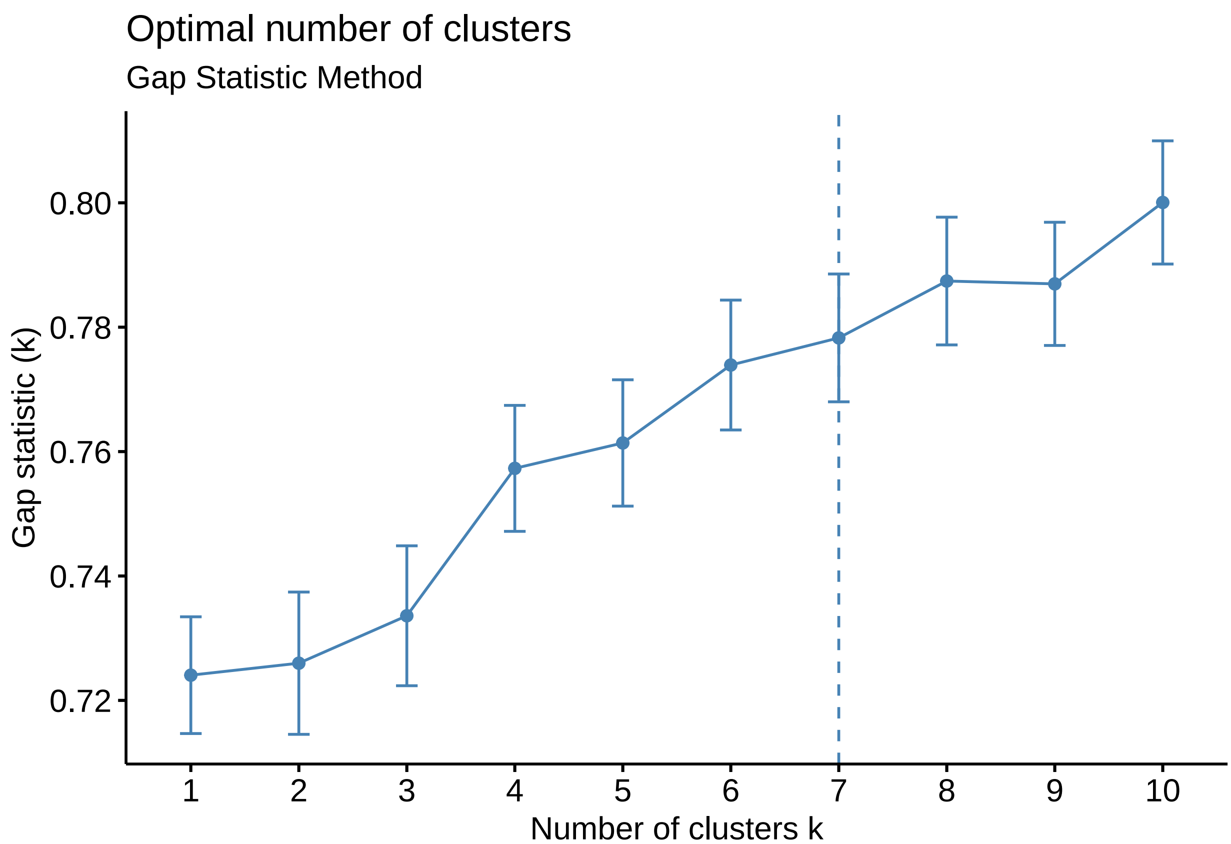
```r
tail(clusterbind_student3)
```

```
##     age traveltime studytime failures absences G3 kmeans.df$cluster
## 389  18          1         2        0        0  6                 5
## 391  20          1         2        2       11  7                 6
## 392  17          2         1        0        3 14                 3
## 393  21          1         1        3        3  5                 6
## 394  18          3         1        0        0  8                 4
## 395  19          1         1        0        5  7                 5
```

```r
## Looking at the OPTIMAL NUMBER of CLUSTERS by GAP method

optimalclusters_GAP <- fviz_nbclust(student3_scaled, kmeans, method="gap_stat",
                                    nstart=25, nboot=50) +
                                    labs(subtitle = "Gap Statistic Method")
```

```
## Clustering k = 1,2,..., K.max (= 10): .. done
## Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
## .................................................. 50
```

```r
print(optimalclusters_GAP)
```



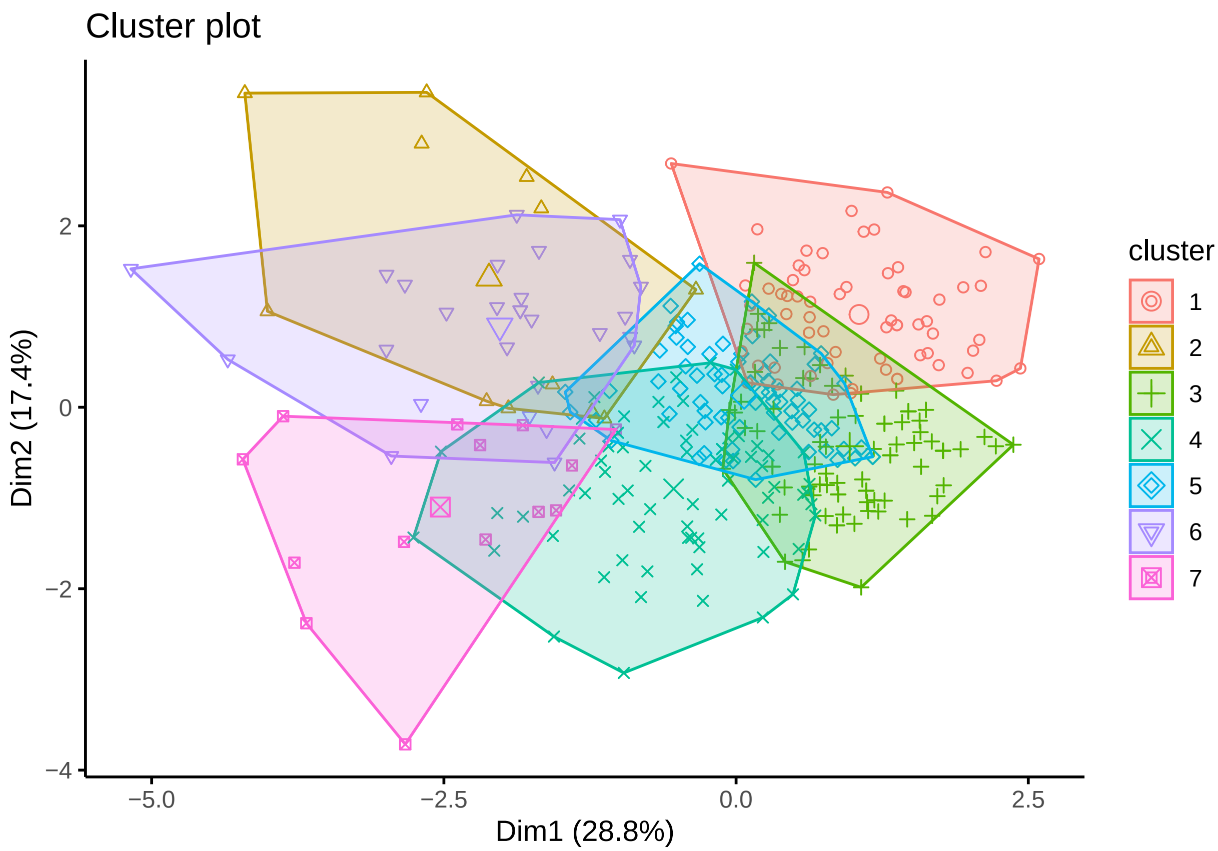

```r
## Running the K-MEANS clustering algorithm by using 7 CLUSTERS
kmeans.df <- kmeans(student3_scaled, 7, nstart=25)
print(kmeans.df)
```

```
## K-means clustering with 7 clusters of sizes 64, 12, 86, 73, 79, 28, 14
##
## Cluster means:
##           age traveltime  studytime   failures    absences         G3
## 1 -0.04783343 -0.2431478  1.5812008 -0.3535504 -0.28434990  0.5461968
## 2  0.40331227  0.2195404 -0.2506430  0.2248110  3.73651831 -0.4556840
## 3 -0.53369731 -0.4775788 -0.5949407 -0.3306486 -0.35273277  0.9212931
## 4  0.23970875  1.3649871 -0.3301209 -0.2154374 -0.17615723 -0.3236451
## 5 -0.18644781 -0.6115509 -0.1417470 -0.3814494  0.01004804 -0.7026650
## 6  1.67823830 -0.3699265 -0.1791913  1.6361019  0.54894103 -0.5262030
## 7 -0.40289095  0.9303681 -0.4792886  3.4582749  0.02789474 -1.0606631
```

```
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   4   5   7   1   5   3   5   4   3   3   5   4   3   4   1   3   1   4   7   5
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   3   3   3   4   5   7   3   3   5   5   3   3   3   3   3   4   1   1   1   3
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   2   3   3   3   4   5   5   1   3   5   4   3   4   3   3   5   3   3   5   3
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   5   4   5   5   5   4   1   1   4   1   1   1   7   3   2   5   1   1   7   5
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   3   1   5   3   5   7   5   1   4   5   5   3   5   4   1   1   3   5   3   5
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   5   1   3   2   3   1   1   1   4   1   3   1   3   3   5   3   3   3   4   3
## 121 122 123 124 125 126 127 128 130 133 134 139 140 142 143 144 146 148 150 152
##   3   1   4   3   5   3   5   6   3   5   5   3   3   7   1   3   5   5   7   3
## 153 155 156 157 158 159 160 162 164 165 166 167 168 170 172 173 175 176 177 178
##   7   3   5   3   7   4   3   7   5   7   4   5   3   3   3   5   4   4   4   5
## 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   5   5   5   3   3   2   5   5   3   3   5   5   3   5   4   5   3   3   3   4
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 218 219
##   2   5   3   5   5   5   1   2   7   3   4   4   1   5   3   6   5   4   6   4
## 220 221 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 241
##   1   4   3   4   1   6   3   5   4   1   3   4   5   3   5   1   3   5   4   4
## 242 244 246 247 248 249 250 251 252 253 254 255 256 257 258 259 261 262 263 264
##   4   3   4   4   6   6   3   4   4   6   4   3   4   1   6   3   3   5   1   5
## 266 267 268 269 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286
##   4   5   4   5   6   1   4   3   4   4   2   2   6   4   2   6   1   4   5   5
## 287 288 289 290 291 292 293 294 295 296 298 299 300 301 302 303 304 305 306 307
##   1   1   1   3   5   1   6   1   1   5   4   1   3   5   4   1   1   6   6   3
## 308 309 310 312 313 314 315 316 318 319 320 321 322 323 324 325 326 327 328 329
##   2   6   6   4   6   6   6   2   5   1   5   2   5   1   1   1   1   3   4   5
## 330 331 332 336 337 339 340 341 343 345 346 347 348 349 350 351 352 353 354 355
##   1   1   1   1   6   1   5   6   3   1   1   1   1   1   6   7   4   6   4   4
## 356 357 358 359 360 361 362 363 364 365 366 367 369 370 371 372 373 374 375 376
##   5   4   4   4   1   4   4   4   3   4   4   1   4   4   6   4   1   5   1   4
## 377 378 379 380 381 382 383 385 386 387 389 391 392 393 394 395
##   6   5   3   5   3   4   4   6   4   4   5   6   3   6   4   5
##
## Within cluster sum of squares by cluster:
## [1] 176.81415  87.44409 167.04971 225.08240 138.98671 106.62924  68.82676
##  (between_SS / total_SS =  54.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
## Visualization :
optimalclusters_GAP_fviz = fviz_cluster(kmeans.df,
                                        data = student3_scaled,
                                        geom = c("point"),
                                        ggtheme=theme_classic())
```

```r
gridExtra::grid.arrange(optimalclusters_GAP_fviz)
```




```r
## Numerical SUMMARY of the CLUSTERS
clusters_aggregate <- aggregate(student3, by=list(cluster=kmeans.df$cluster), mean)
print(clusters_aggregate)
```

```
##   cluster      age traveltime studytime   failures  absences        G3
## 1       1 16.59375   1.265625  3.359375 0.03125000  3.953125 11.296875
## 2       2 17.16667   1.583333  1.833333 0.41666667 36.750000  8.083333
## 3       3 15.97674   1.104651  1.546512 0.04651163  3.395349 12.500000
## 4       4 16.95890   2.369863  1.767123 0.12328767  4.835616  8.506849
## 5       5 16.41772   1.012658  1.924051 0.01265823  6.354430  7.291139
## 6       6 18.78571   1.178571  1.892857 1.35714286 10.750000  7.857143
## 7       7 16.14286   2.071429  1.642857 2.57142857  6.500000  6.142857
```

```r
student3 %>%
  mutate(Cluster = kmeans.df$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 7 x 7
##   Cluster   age traveltime studytime failures absences    G3
##     <int> <dbl>      <dbl>     <dbl>    <dbl>    <dbl> <dbl>
## 1       1  16.6       1.27      3.36   0.0312     3.95 11.3
## 2       2  17.2       1.58      1.83   0.417     36.8   8.08
## 3       3  16.0       1.10      1.55   0.0465     3.40 12.5
## 4       4  17.0       2.37      1.77   0.123      4.84  8.51
```

```
## 5       5  16.4       1.01      1.92   0.0127      6.35  7.29
## 6       6  18.8       1.18      1.89   1.36       10.8   7.86
## 7       7  16.1       2.07      1.64   2.57        6.5   6.14
```

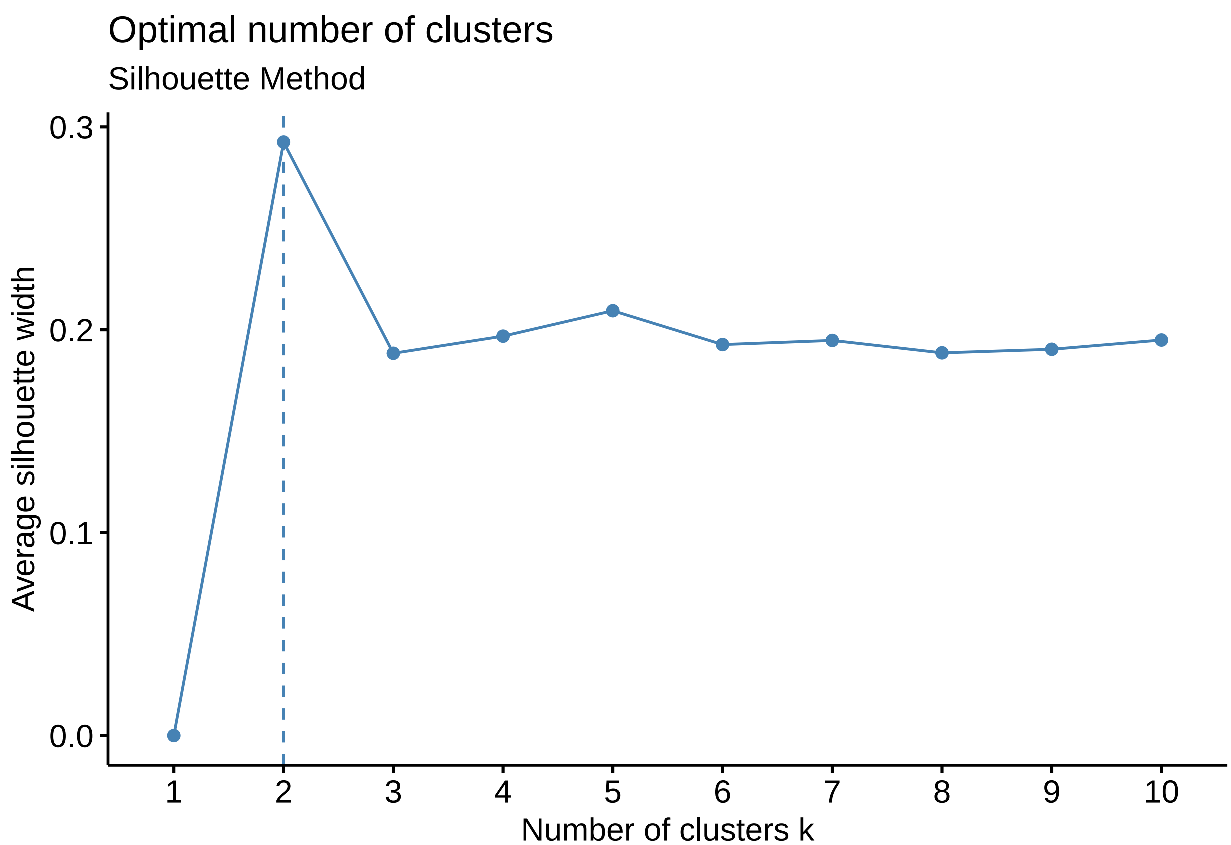
```
### just in case that we will need the information on CLUSTERS in the BIG DATA FRAME

clusterbind_student3 <- cbind(student3, kmeans.df$cluster)
head(clusterbind_student3)
```

```
##   age traveltime studytime failures absences G3 kmeans.df$cluster
## 1  18          2         2        0        6  4                 4
## 2  17          1         2        0        4  4                 5
## 3  15          1         2        3       10  8                 7
## 4  15          1         3        0        2 13                 1
## 5  16          1         2        0        4  8                 5
## 6  16          1         2        0       10 13                 3
```

```
tail(clusterbind_student3)
```

```
##     age traveltime studytime failures absences G3 kmeans.df$cluster
## 389  18          1         2        0        0  6                 5
## 391  20          1         2        2       11  7                 6
## 392  17          2         1        0        3 14                 3
## 393  21          1         1        3        3  5                 6
## 394  18          3         1        0        0  8                 4
## 395  19          1         1        0        5  7                 5
```

```r
## Using K-MEANS
## Looking at the OPTIMAL NUMBER of CLUSTERS by SILHOUETTE METHOD

optimalclusters_SILHOUETTE <- fviz_nbclust(student3_scaled, kmeans, nstart=25,
                                          method="silhouette", nboot=50) +
                                          labs(subtitle = "Silhouette Method")
print(optimalclusters_SILHOUETTE)
```

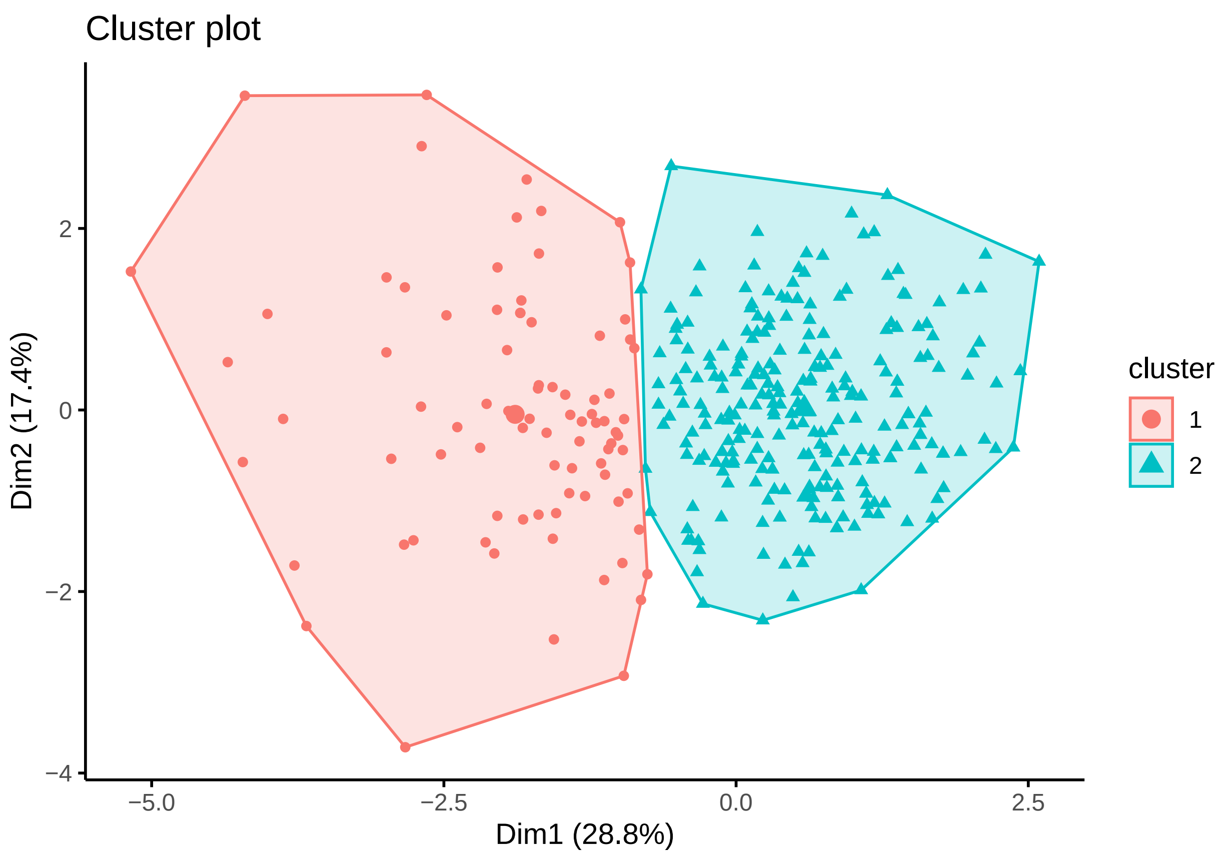



```r
## Running the K-MEANS clustering algorithm by using 2 CLUSTERS
kmeans.df <- kmeans(student3_scaled, 2, nstart=25)
print(kmeans.df)
```

```
## K-means clustering with 2 clusters of sizes 84, 272
##
## Cluster means:
##          age traveltime  studytime   failures   absences         G3
## 1  0.7220438  0.6009601 -0.4507079  1.1716264  0.7999773 -0.7488947
## 2 -0.2229841 -0.1855906  0.1391892 -0.3618258 -0.2470518  0.2312763
##
## Clustering vector:
##    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##    1   2   1   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   1   2
##   21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##    2   2   2   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2
##   41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##    1   2   2   2   1   2   2   2   2   2   2   2   1   2   2   2   2   2   2   2
```

```
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   1   2   2   2   2   2   2   2   2   2   2   1   2   1   2   2   2   1   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   2   2   1   2   2   1   2   2   2   2   2   2   2   2   2   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   1   2   2   1   2   2   2   2   2   2   2   2   2   2   2   2   2   2   1   2
## 121 122 123 124 125 126 127 128 130 133 134 139 140 142 143 144 146 148 150 152
##   2   2   2   2   2   2   2   1   2   2   2   2   2   1   2   2   2   2   1   2
## 153 155 156 157 158 159 160 162 164 165 166 167 168 170 172 173 175 176 177 178
##   1   2   2   2   1   2   2   1   2   1   1   2   2   2   2   2   2   2   2   2
## 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   2   2   2   2   2   1   2   2   2   2   2   2   2   2   1   2   2   2   2   1
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 218 219
##   1   2   2   2   2   1   2   1   1   2   2   2   2   2   2   1   2   2   1   1
## 220 221 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 241
##   2   2   2   2   2   1   2   2   1   2   2   2   2   2   1   2   2   2   2   2
## 242 244 246 247 248 249 250 251 252 253 254 255 256 257 258 259 261 262 263 264
##   2   2   2   2   1   1   2   1   2   1   2   2   1   2   2   2   2   2   2   2
## 266 267 268 269 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286
##   2   2   2   2   1   2   2   2   2   2   1   1   1   1   1   1   2   2   2   2
## 287 288 289 290 291 292 293 294 295 296 298 299 300 301 302 303 304 305 306 307
##   2   2   2   2   2   2   1   2   2   2   1   2   2   2   2   2   2   1   1   2
## 308 309 310 312 313 314 315 316 318 319 320 321 322 323 324 325 326 327 328 329
##   1   1   1   1   1   1   1   1   2   2   2   2   2   2   2   2   2   2   1   2
## 330 331 332 336 337 339 340 341 343 345 346 347 348 349 350 351 352 353 354 355
##   2   2   2   2   1   2   2   1   2   2   2   2   2   2   1   1   2   1   1   2
## 356 357 358 359 360 361 362 363 364 365 366 367 369 370 371 372 373 374 375 376
##   2   2   2   1   2   2   1   2   2   2   2   2   2   1   1   1   2   1   2   1
## 377 378 379 380 381 382 383 385 386 387 389 391 392 393 394 395
##   1   2   2   2   2   1   2   1   2   1   2   1   2   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 737.0046 990.7037
##  (between_SS / total_SS =  18.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
## Visualization :
optimalclusters_SILHOUETTE_fviz = fviz_cluster(kmeans.df,
                                         data = student3_scaled,
                                         geom = c("point"),
                                         ggtheme=theme_classic())

gridExtra::grid.arrange(optimalclusters_SILHOUETTE_fviz)
```




```r
## Numerical SUMMARY of the CLUSTERS
clusters_aggregate <- aggregate(student3, by=list(cluster=kmeans.df$cluster), mean)
print(clusters_aggregate)
```

```
##   cluster      age traveltime studytime   failures  absences        G3
## 1       1 17.57143   1.845238  1.666667 1.04761905 12.797619  7.142857
## 2       2 16.37132   1.305147  2.158088 0.02573529  4.257353 10.286765
```

```r
student3 %>%
  mutate(Cluster = kmeans.df$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 2 x 7
##   Cluster   age traveltime studytime failures absences    G3
##     <int> <dbl>      <dbl>     <dbl>    <dbl>    <dbl> <dbl>
## 1       1  17.6       1.85      1.67   1.05       12.8  7.14
## 2       2  16.4       1.31      2.16   0.0257      4.26 10.3
```

```r
### just in case that we will need the information on CLUSTERS in the BIG DATA FRAME

clusterbind_student3 <- cbind(student3, kmeans.df$cluster)
head(clusterbind_student3)
```

```
##   age traveltime studytime failures absences G3 kmeans.df$cluster
## 1  18          2         2        0        6  4                 1
## 2  17          1         2        0        4  4                 2
## 3  15          1         2        3       10  8                 1
```

```
## 4  15          1         3        0        2 13                  2
## 5  16          1         2        0        4  8                  2
## 6  16          1         2        0       10 13                  2
```

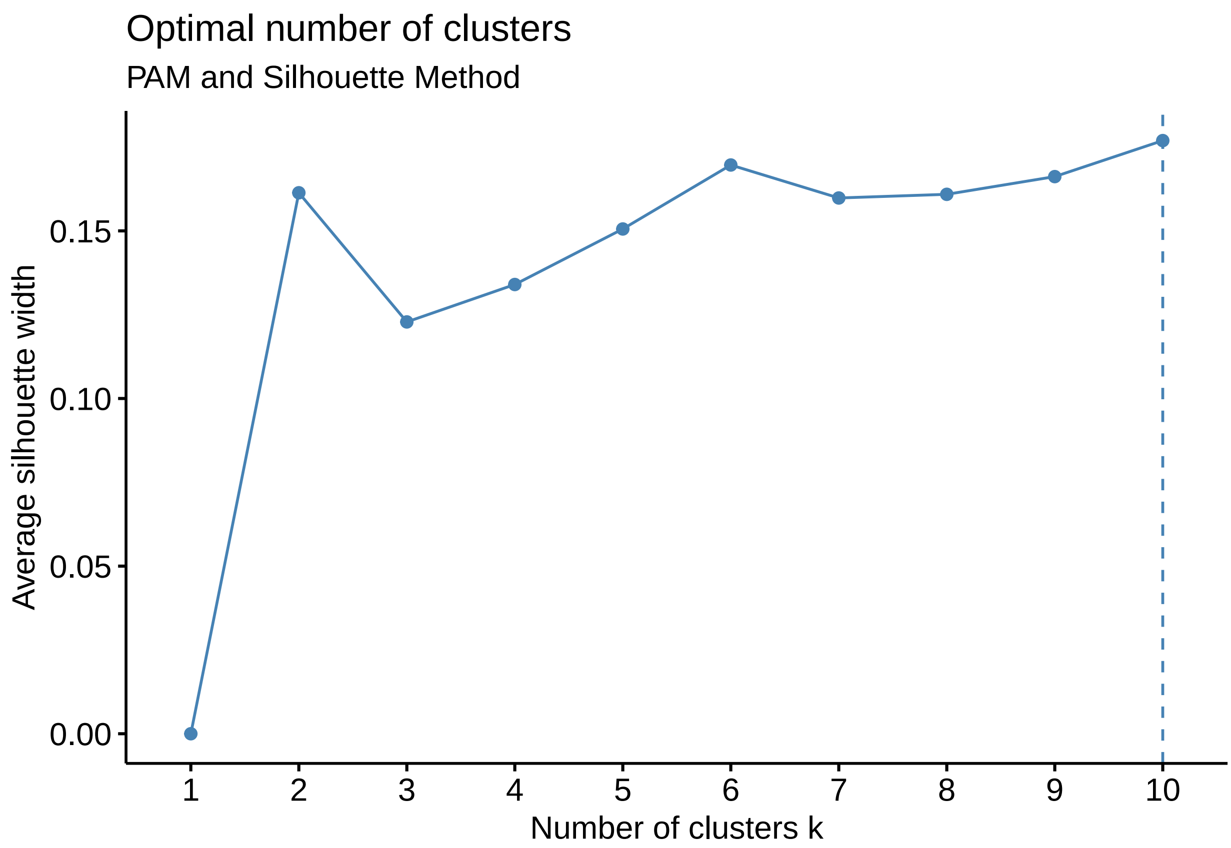
```r
tail(clusterbind_student3)
```

```
##     age traveltime studytime failures absences G3 kmeans.df$cluster
## 389  18          1         2        0        0  6                 2
## 391  20          1         2        2       11  7                 1
## 392  17          2         1        0        3 14                 2
## 393  21          1         1        3        3  5                 1
## 394  18          3         1        0        0  8                 1
## 395  19          1         1        0        5  7                 1
```

```r
## Comparing the summaries above to the dataset grouped by GRADE G3
## (age, traveltime, studytime, failures, absences, G3)

## tapply(student3$age, student3$G3, summary)
## tapply(student3$traveltime, student3$G3, summary)
## tapply(student3$studytime, student3$G3, summary)
## tapply(student3$failures, student3$G3, summary)
## tapply(student3$absences, student3$G3, summary)
## tapply(student3$G3, student3$G3, summary)
```

5.4. PAM-based CLUSTERING (after SILHOUETTE method)

```r
## Using PAM Partitioning Around Medoids
## Looking at the OPTIMAL NUMBER of CLUSTERS by SILHOUETTE

optimalclusters_PAM <- fviz_nbclust(student3_scaled,
                                    pam,
                                    method="silhouette") +
                                    labs(subtitle = "PAM and Silhouette Method")
print(optimalclusters_PAM)
```




```r
#  Using "PAM" function for Partitioning (clustering) of the data into 'k' clusters
# "around medoids"", a more robust version of K-means.
```

```r
pam_clusters <- pam(student3_scaled, 2)

print(pam_clusters)
```

```
## Medoids:
##      ID        age traveltime   studytime   failures    absences         G3
## 276 250  0.2720699  0.8263445 -0.05057819 -0.4004445 -0.03340482  0.1418721
## 29   29 -0.5153844 -0.6299854 -0.05057819 -0.4004445 -0.27860307 -0.1698963
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   2   2   2   2   2   2   1   2   2   2   1   2   2   2   2   2   1   2   2
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   2   2   2   1   2   2   2   2   2   2   2   1   2   2   2   2   2   1   2   2
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
```

```
##   1   2   2   2   1   2   2   2   2   2   1   2   1   2   2   2   2   2   2   2
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   2   1   2   2   2   1   2   2   2   1   1   2   2   2   1   2   1   2   1   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   2   2   2   1   2   2   2   2   1   2   2   2   2   1   2   2   1   2   2   2
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
##   2   2   2   1   2   2   2   2   1   2   2   2   2   2   2   2   1   1   1   2
## 121 122 123 124 125 126 127 128 130 133 134 139 140 142 143 144 146 148 150 152
##   2   2   1   2   2   2   2   1   2   2   2   2   1   1   2   2   2   2   1   2
## 153 155 156 157 158 159 160 162 164 165 166 167 168 170 172 173 175 176 177 178
##   1   2   2   2   1   1   2   2   2   1   1   2   2   2   1   2   1   1   1   2
## 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   2   2   2   2   2   1   2   2   2   2   2   2   2   2   1   2   1   2   2   1
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 218 219
##   1   2   2   2   2   2   1   2   2   2   1   1   1   2   2   2   2   1   2   1
## 220 221 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 241
##   2   1   2   1   2   2   2   2   1   1   2   1   2   2   1   1   2   2   1   1
## 242 244 246 247 248 249 250 251 252 253 254 255 256 257 258 259 261 262 263 264
##   1   2   1   1   1   2   2   1   1   2   1   2   1   2   1   1   1   2   2   2
## 266 267 268 269 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286
##   1   2   1   2   1   1   1   2   1   1   1   1   2   1   1   2   1   1   2   2
## 287 288 289 290 291 292 293 294 295 296 298 299 300 301 302 303 304 305 306 307
##   1   2   1   1   1   2   1   1   1   2   1   2   1   1   1   1   2   1   1   1
## 308 309 310 312 313 314 315 316 318 319 320 321 322 323 324 325 326 327 328 329
##   1   1   1   1   1   1   1   1   2   2   2   2   2   2   2   1   2   2   1   2
## 330 331 332 336 337 339 340 341 343 345 346 347 348 349 350 351 352 353 354 355
##   1   2   2   2   1   1   2   1   1   2   1   1   2   2   1   1   1   2   1   1
## 356 357 358 359 360 361 362 363 364 365 366 367 369 370 371 372 373 374 375 376
##   2   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1   2   2   1   1
## 377 378 379 380 381 382 383 385 386 387 389 391 392 393 394 395
##   1   2   1   2   1   1   1   1   1   1   2   1   1   1   1   1
## Objective function:
##    build     swap
## 2.084444 2.062337
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```r
str(pam_clusters)
```

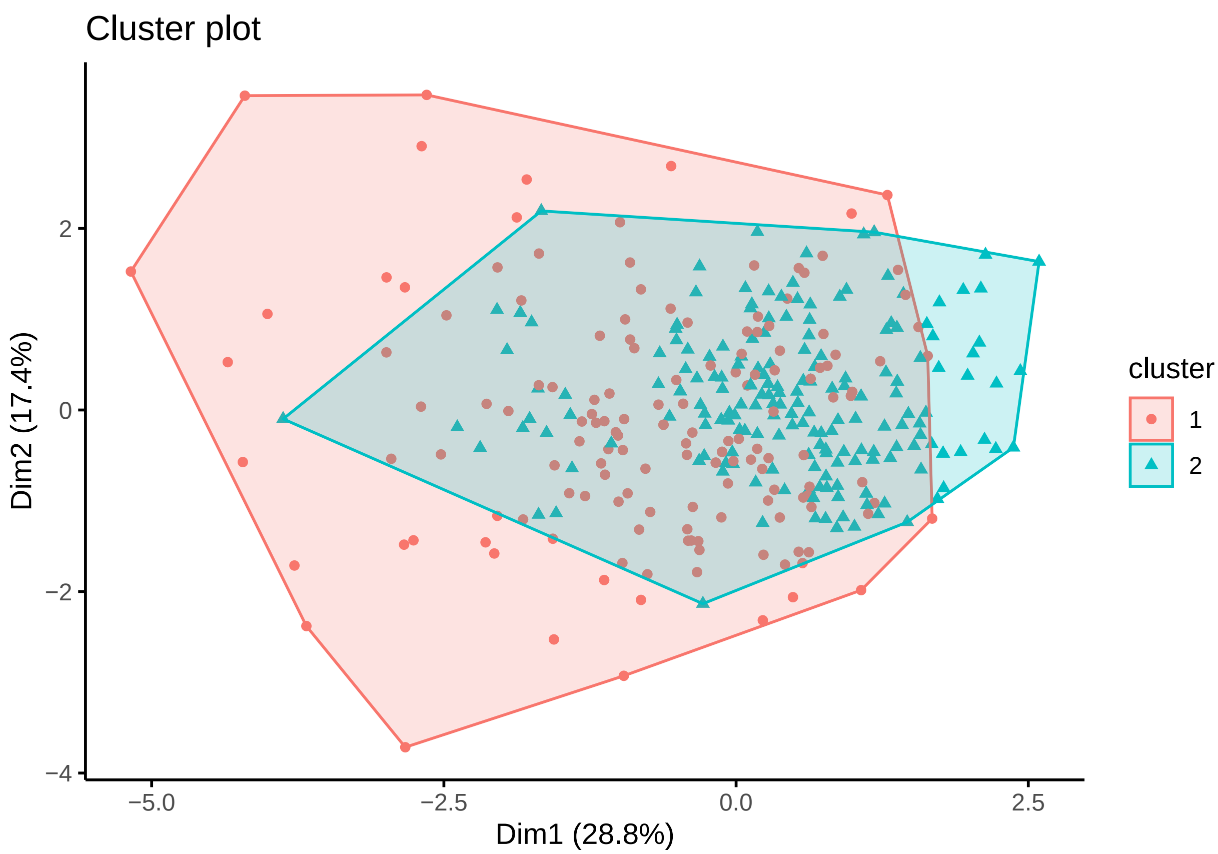
```
## List of 10
##  $ medoids   : num [1:2, 1:6] 0.2721 -0.5154 0.8263 -0.63 -0.0506 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "276" "29"
##   .. ..$ : chr [1:6] "age" "traveltime" "studytime" "failures" ...
##  $ id.med    : int [1:2] 250 29
##  $ clustering: Named int [1:356] 1 2 2 2 2 2 2 1 2 2 ...
##   ..- attr(*, "names")= chr [1:356] "1" "2" "3" "4" ...
##  $ objective : Named num [1:2] 2.08 2.06
##   ..- attr(*, "names")= chr [1:2] "build" "swap"
##  $ isolation : Factor w/ 3 levels "no","L","L*": 1 1
##   ..- attr(*, "names")= chr [1:2] "1" "2"
##  $ clusinfo  : num [1:2, 1:5] 151 205 8.55 5.29 2.45 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "size" "max_diss" "av_diss" "diameter" ...
##  $ silinfo   :List of 3
##   ..$ widths         : num [1:356, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:356] "370" "229" "312" "354" ...
##   .. .. ..$ : chr [1:3] "cluster" "neighbor" "sil_width"
##   ..$ clus.avg.widths: num [1:2] -0.00856 0.28653
##   ..$ avg.width      : num 0.161
##  $ diss      : NULL
##  $ call      : language pam(x = student3_scaled, k = 2)
##  $ data      : num [1:356, 1:6] 1.06 0.272 -1.303 -1.303 -0.515 ...
##   ..- attr(*, "scaled:center")= Named num [1:6] 16.654 1.433 2.042 0.267 6.272 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "scaled:scale")= Named num [1:6] 1.27 0.687 0.833 0.666 8.157 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:356] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:6] "age" "traveltime" "studytime" "failures" ...
##  - attr(*, "class")= chr [1:2] "pam" "partition"
```

```r
print(pam_clusters$medoids)
```

```
##            age traveltime   studytime   failures    absences         G3
## 276  0.2720699  0.8263445 -0.05057819 -0.4004445 -0.03340482  0.1418721
## 29  -0.5153844 -0.6299854 -0.05057819 -0.4004445 -0.27860307 -0.1698963
```

```r
## Visualization :
optimalclusters_SILHOUETTE_fviz = fviz_cluster(pam_clusters,
                                               data = student3_scaled,
                                               geom = c("point"),
                                               ggtheme=theme_classic())

gridExtra::grid.arrange(optimalclusters_SILHOUETTE_fviz)
```




```
## Numerical SUMMARY of the CLUSTERS
pamclusters_aggregate <- aggregate(student3, by=list(cluster=pam_clusters$cluster), mean)
print(pamclusters_aggregate)
```

```
##   cluster      age traveltime studytime  failures absences       G3
## 1       1 17.34437   1.980132  2.013245 0.4105960 8.039735 9.536424
## 2       2 16.14634   1.029268  2.063415 0.1609756 4.970732 9.551220
```

```
student3 %>%
  mutate(Cluster = pam_clusters$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 2 x 7
##   Cluster   age traveltime studytime failures absences    G3
##     <int> <dbl>      <dbl>     <dbl>    <dbl>    <dbl> <dbl>
## 1       1  17.3       1.98      2.01    0.411     8.04  9.54
## 2       2  16.1       1.03      2.06    0.161     4.97  9.55
```

```
### just in case that we will need the information on CLUSTERS in the BIG DATA FRAME

clusterbind_student3 <- cbind(student3, pam_clusters$cluster)
head(clusterbind_student3)
```

```
##   age traveltime studytime failures absences G3 pam_clusters$cluster
## 1  18          2         2        0        6  4                    1
## 2  17          1         2        0        4  4                    2
## 3  15          1         2        3       10  8                    2
```

```
## 4  15          1         3        0        2 13                    2
## 5  16          1         2        0        4  8                    2
## 6  16          1         2        0       10 13                    2
```

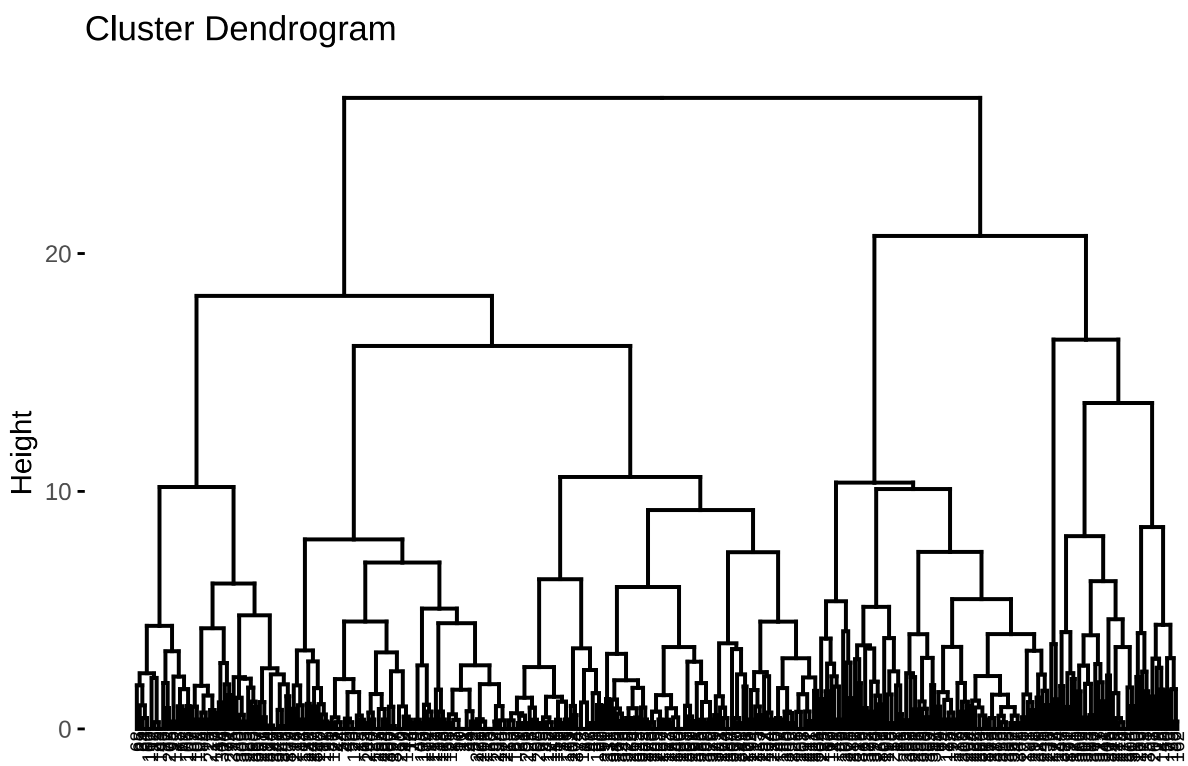
```r
tail(clusterbind_student3)
```

```
##     age traveltime studytime failures absences G3 pam_clusters$cluster
## 389  18          1         2        0        0  6                    2
## 391  20          1         2        2       11  7                    1
## 392  17          2         1        0        3 14                    1
## 393  21          1         1        3        3  5                    1
## 394  18          3         1        0        0  8                    1
## 395  19          1         1        0        5  7                    1
```

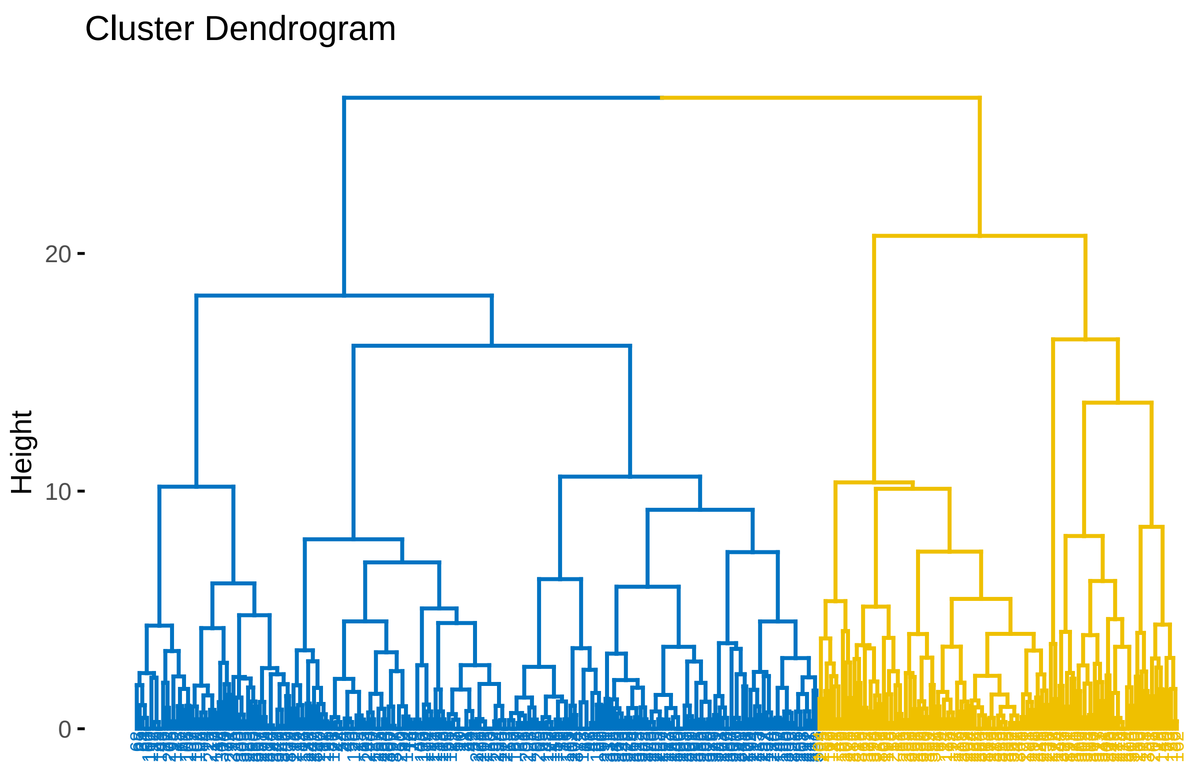
6. HIERARCHICAL CLUSTERING

6.1 HIERARCHICAL CLUSTERING (using EUCLIDEAN distance)

```r
student3_dist = dist(student3_scaled, method="euclidean")
# as.matrix(student3_dist)[1:2,1:2]

agg_tree_ward = hclust(d = student3_dist, method="ward.D2")
#print(agg_tree_ward)

# Visualizing the Dendogram
fviz_dend(agg_tree_ward, cex=.5)
```




```r
# Cutting the tree to create 2 clusters and visualizing it:
agg_tree_warddend <- fviz_dend(agg_tree_ward, cex=.5, k=2, palette = "jco")
agg_tree_warddend
```

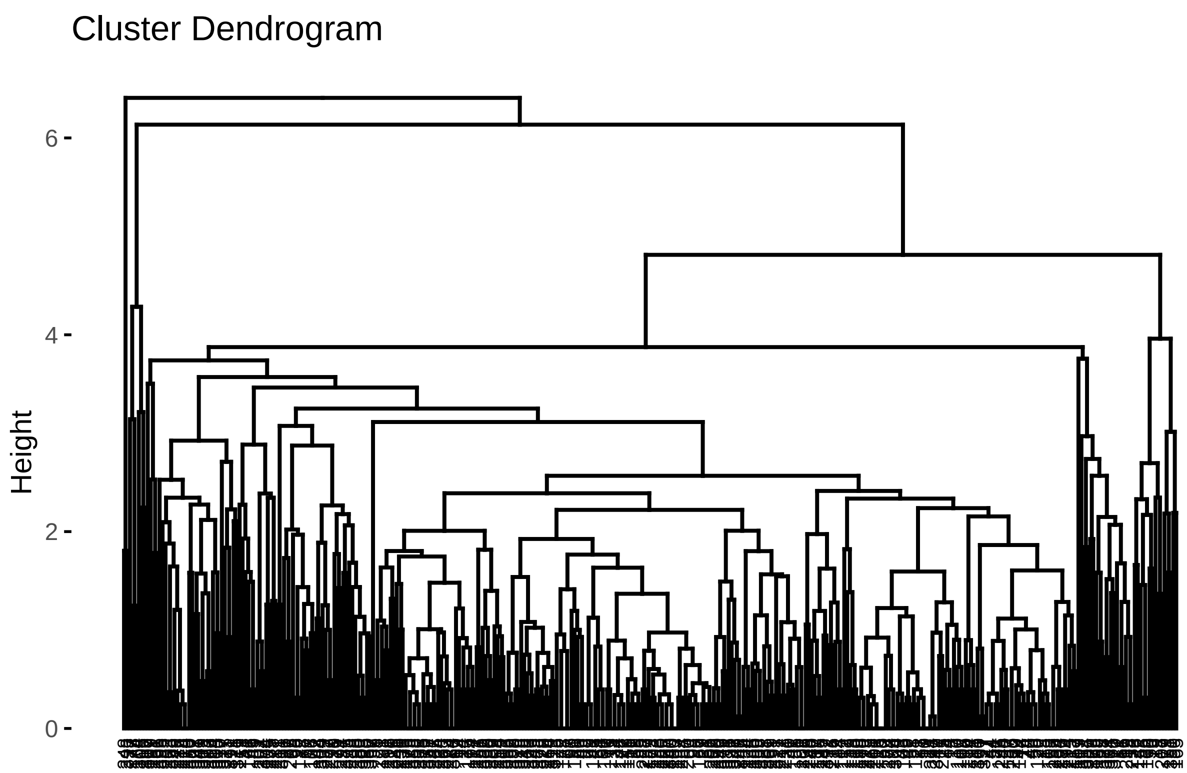



```
# To access the partition accuracy of the cluster tree (created by hclust()) there should be a strong
# correlation between the original distance matrix and the object linkage distance defined as cophenetic
# distances.
# Calculating Cophenetic Distances

agg_cophenetic <- cophenetic(agg_tree_ward)

# head(agg_cophenetic)
# tail(agg_cophenetic)

# Calculating the correlation between Cophenetic Distances and Original Distances for :
cor(student3_dist, agg_cophenetic)
```

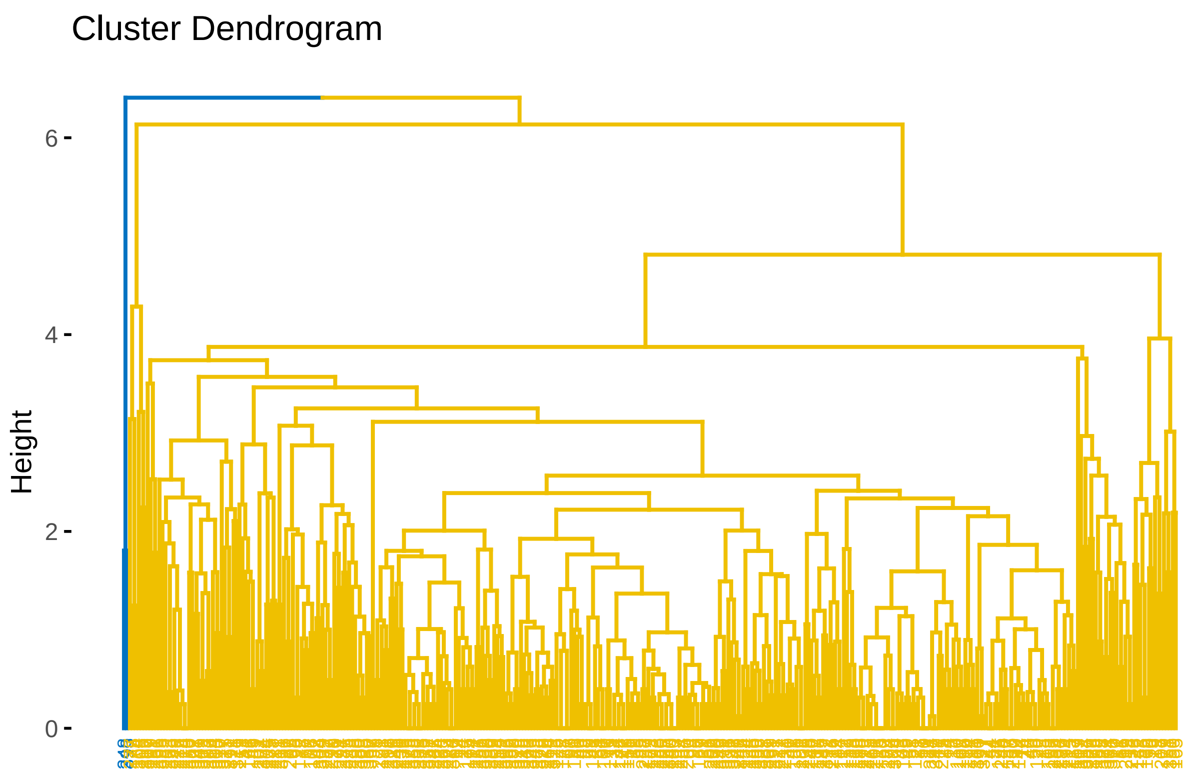
```
## [1] 0.4707307
```

6.2 HIERARCHICAL CLUSTERING (using MINKOWSKI distance)

```
student3_dist = dist(student3_scaled, method="minkowski")
# as.matrix(student3_dist)[1:2,1:2]

agg_tree_ward = hclust(d = student3_dist, method="average")
#print(agg_tree_ward)

# Visualizing the Dendogram
fviz_dend(agg_tree_ward, cex=.5)
```




```
# Cutting the tree to create 2 clusters and visualizing it:
agg_tree_warddend <- fviz_dend(agg_tree_ward, cex=.5, k=2, palette = "jco")
agg_tree_warddend
```

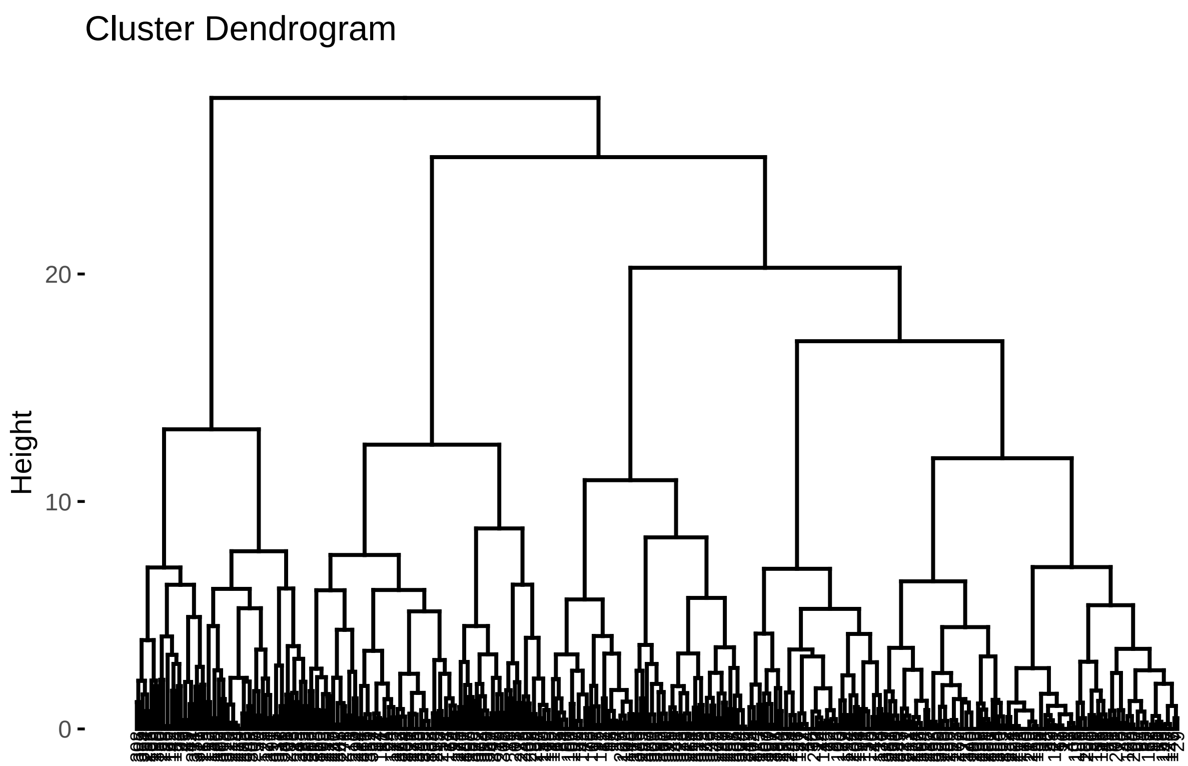



```
# To access the partition accuracy of the cluster tree (created by hclust()) there should be a strong
# correlation between the original distance matrix and the object linkage distance defined as copheneti
# distances.
# Calculating Cophenetic Distances

agg_cophenetic <- cophenetic(agg_tree_ward)

# head(agg_cophenetic)
# tail(agg_cophenetic)

# Calculating the correlation between Cophenetic Distances and Original Distances for :
cor(student3_dist, agg_cophenetic)
```

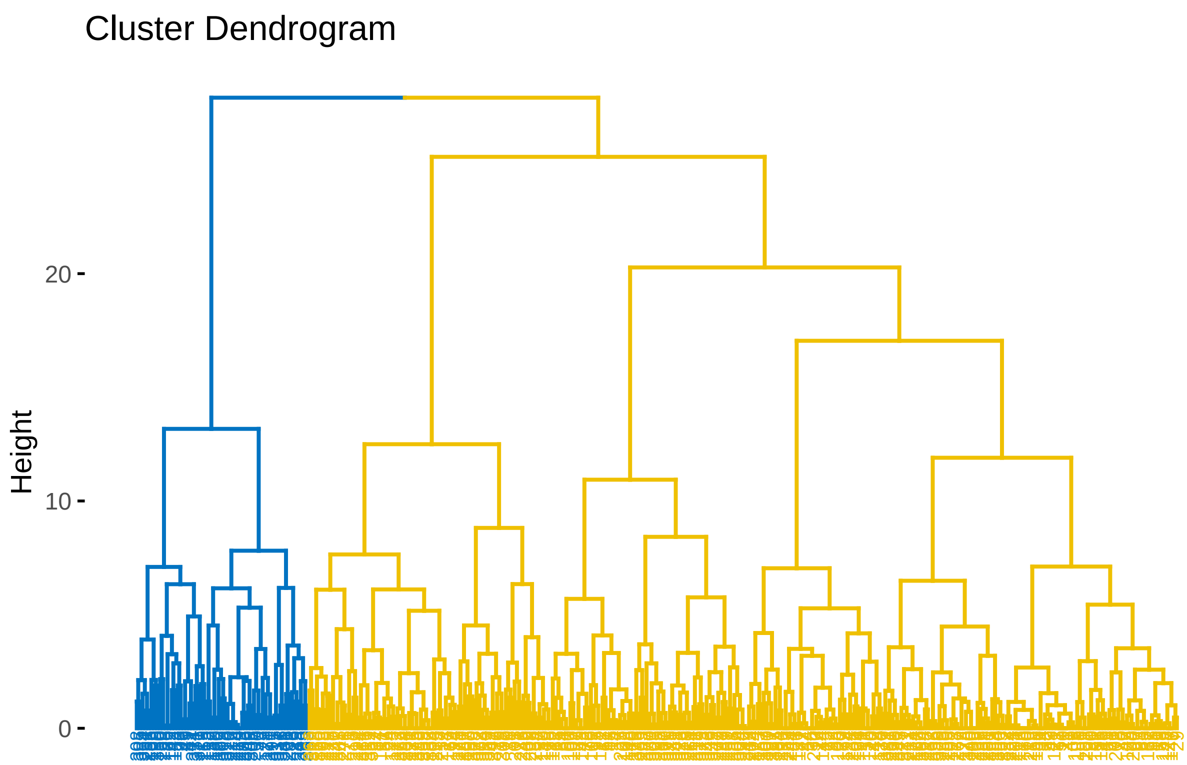
```
## [1] 0.783801
```

6.3 HIERARCHICAL CLUSTERING (using CANBERRA distance)

```r
student3_dist = dist(student3_scaled, method="canberra")
# as.matrix(student3_dist)[1:2,1:2]

agg_tree_ward = hclust(d = student3_dist, method="ward.D2")
#print(agg_tree_ward)

# Visualizing the Dendogram
fviz_dend(agg_tree_ward, cex=.5)
```




```r
# Cutting the tree to create 2 clusters and visualizing it:
agg_tree_warddend <- fviz_dend(agg_tree_ward, cex=.5, k=2, palette = "jco")
agg_tree_warddend
```

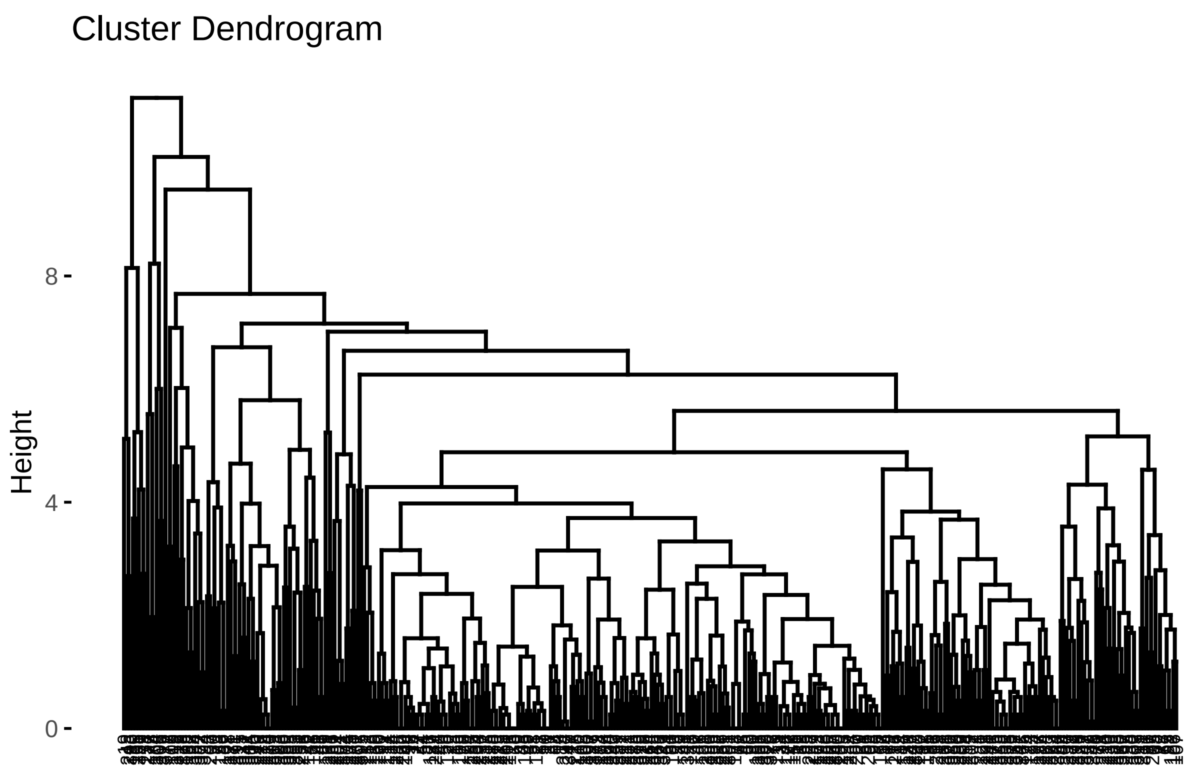



```
# To access the partition accuracy of the cluster tree (created by hclust()) there should be a strong
# correlation between the original distance matrix and the object linkage distance defined as copheneti
# distances.
# Calculating Cophenetic Distances

agg_cophenetic <- cophenetic(agg_tree_ward)

# head(agg_cophenetic)
# tail(agg_cophenetic)

# Calculating the correlation between Cophenetic Distances and Original Distances for :
cor(student3_dist, agg_cophenetic)
```

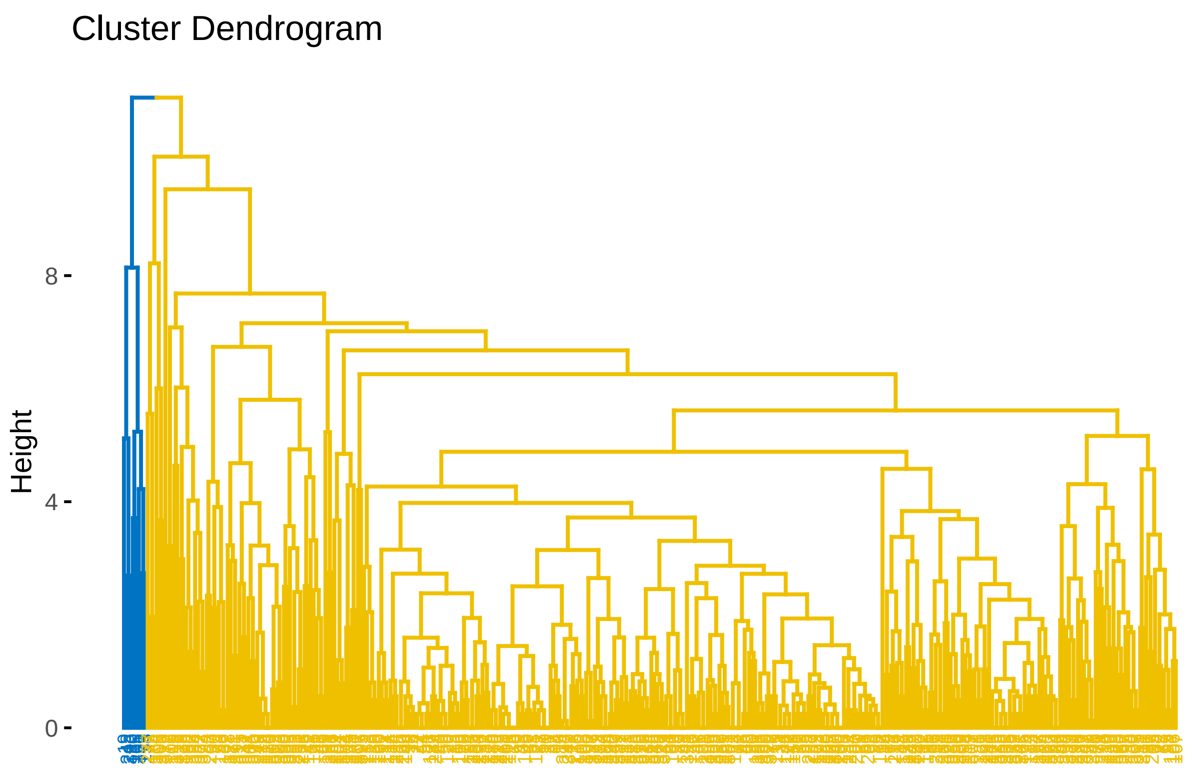
```
## [1] 0.6233455
```

6.4 HIERARCHICAL CLUSTERING (using MANHATTAN distance)

```r
student3_dist = dist(student3_scaled, method="manhattan")
# as.matrix(student3_dist)[1:2,1:2]

agg_tree_ward = hclust(d = student3_dist, method="average")
#print(agg_tree_ward)

# Visualizing the Dendogram
fviz_dend(agg_tree_ward, cex=.5)
```




```r
# Cutting the tree to create 2 clusters and visualizing it:
agg_tree_warddend <- fviz_dend(agg_tree_ward, cex=.5, k=2, palette = "jco")
agg_tree_warddend
```




```
# To access the partition accuracy of the cluster tree (created by hclust()) there should be a strong
# correlation between the original distance matrix and the object linkage distance defined as copheneti
# distances.
# Calculating Cophenetic Distances

agg_cophenetic <- cophenetic(agg_tree_ward)

# head(agg_cophenetic)
# tail(agg_cophenetic)

# Calculating the correlation between Cophenetic Distances and Original Distances for :
cor(student3_dist, agg_cophenetic)
```

```
## [1] 0.7838249
```

7. HIERARCHICAL CLUSTERING (by AGNES and DIANA)

7.1 HIERARCHICAL CLUSTERING (by AGNES)

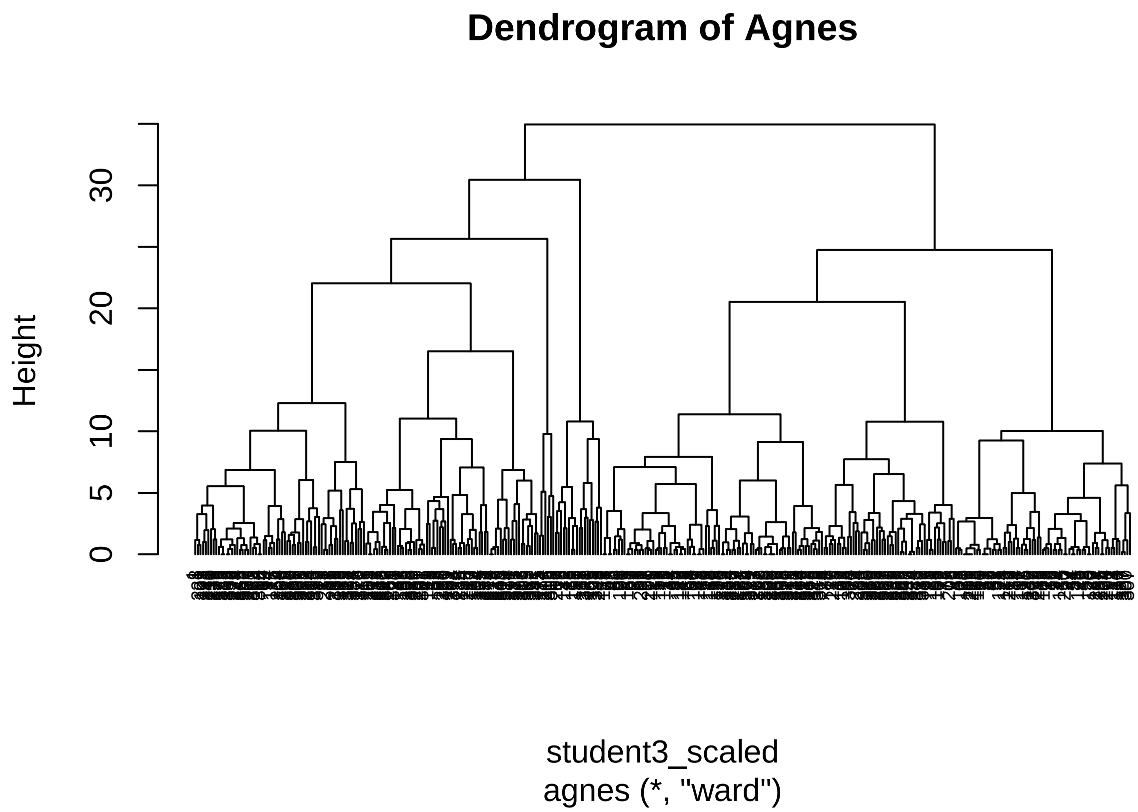
```r
### AGGLOMERATIVE

agnes_cluster <- agnes(x=student3_scaled, stand=TRUE, metric = "euclidean", method="ward")

str(agnes_cluster)
```

```
## List of 9
##  $ order    : int [1:356] 1 8 202 175 271 220 192 210 216 158 ...
##  $ height   : num [1:355] 1.171 0.759 3.265 1.014 1.957 ...
##  $ ac       : num 0.972
##  $ merge    : int [1:355, 1:2] -266 -258 -249 -212 -196 -185 -178 -173 -172 -162 ...
##  $ diss     : NULL
##  $ call     : language agnes(x = student3_scaled, metric = "euclidean", stand = TRUE, method = "ward
##  $ method   : chr "ward"
##  $ order.lab: chr [1:356] "1" "8" "221" "193" ...
##  $ data     : num [1:356, 1:6] 1.266 0.325 -1.556 -1.556 -0.616 ...
##   ..- attr(*, "scaled:center")= Named num [1:6] 1.35e-15 -1.10e-16 2.50e-16 4.93e-17 6.19e-17 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "scaled:scale")= Named num [1:6] 0.837 0.832 0.698 0.661 0.646 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:356] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:6] "age" "traveltime" "studytime" "failures" ...
##  - attr(*, "class")= chr [1:2] "agnes" "twins"
```

```r
agnes_cluster$ac
```

```
## [1] 0.9718767
```

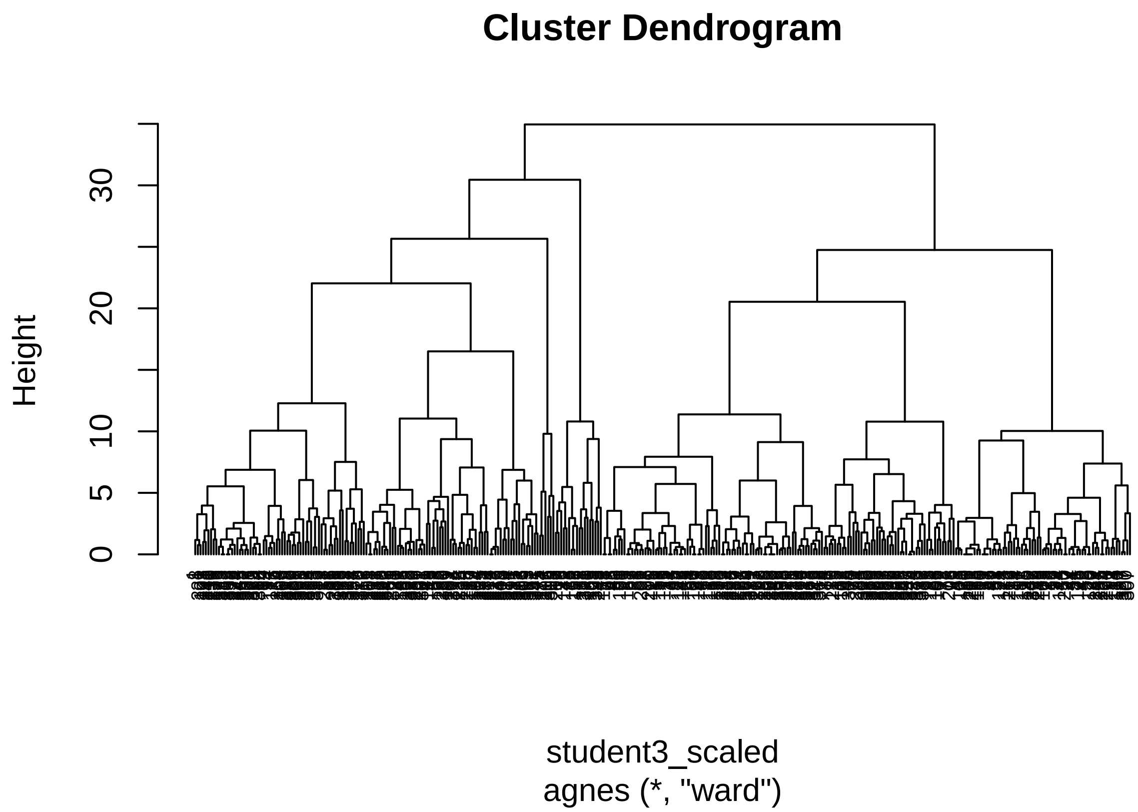
```r
agnes_tree <- pltree(agnes_cluster, cex = 0.6, hang = -1, main = "Dendrogram of Agnes")
```



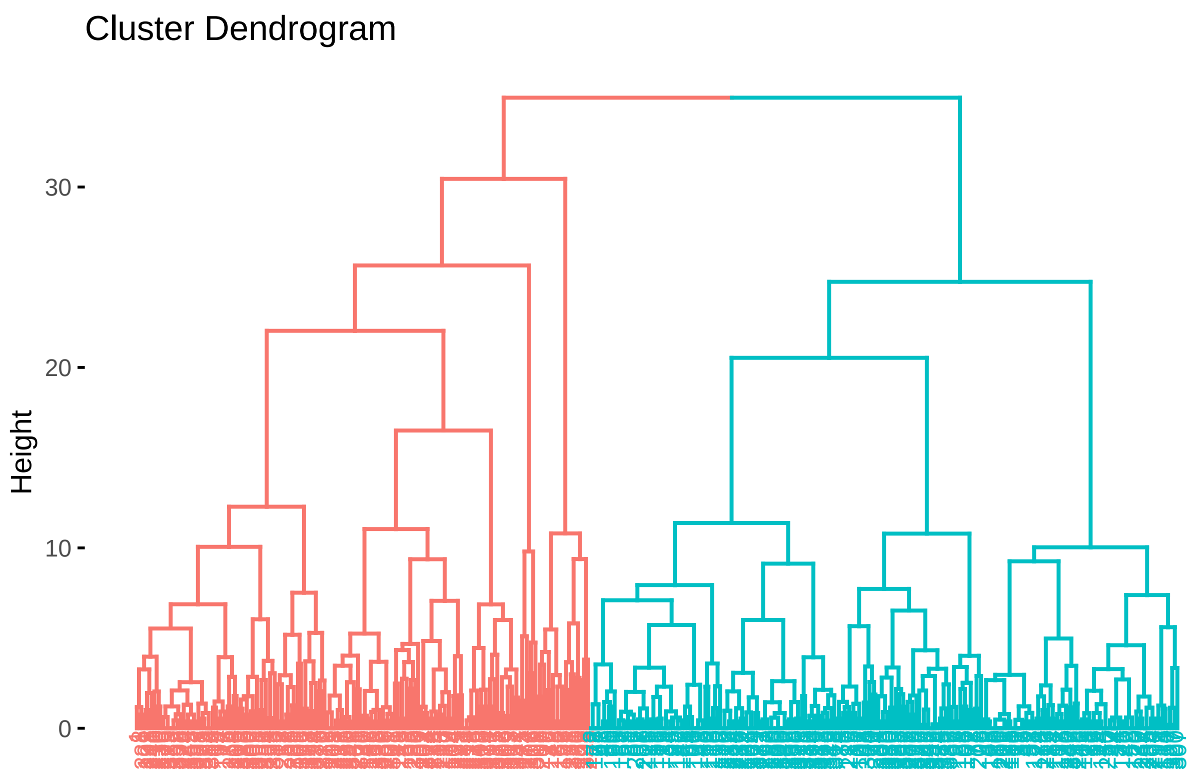

```r
print(agnes_tree)
```

```
## NULL
```

```r
plot(as.hclust(agnes_cluster), cex = 0.6, hang = -1)
```




```
fviz_dend(agnes_cluster, cex=.6, k=2)
```


Cluster Dendrogram
Height
30
20
10
0

7.2 HIERARCHICAL CLUSTERING (by DIANA)

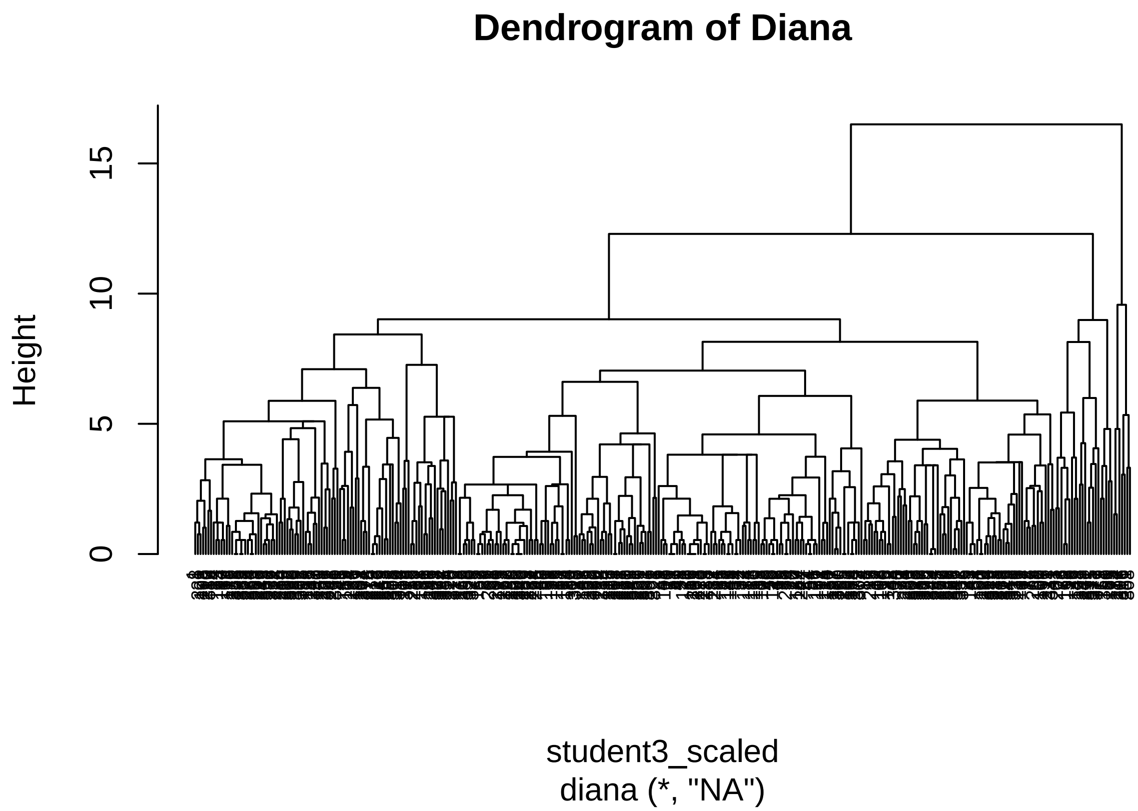
```r
### Divisive
diana_cluster <- diana(x=student3_scaled, stand=TRUE, metric = "euclidean")

str(diana_cluster)
```

```
## List of 8
##  $ order    : int [1:356] 1 8 202 175 271 192 349 14 24 123 ...
##  $ height   : num [1:355] 1.209 0.759 2.045 1.014 2.833 ...
##  $ dc       : num 0.935
##  $ merge    : int [1:355, 1:2] -91 -39 -266 -178 -142 -44 -35 -55 -31 -127 ...
##  $ diss     : NULL
##  $ call     : language diana(x = student3_scaled, metric = "euclidean", stand = TRUE)
##  $ order.lab: chr [1:356] "1" "8" "221" "193" ...
##  $ data     : num [1:356, 1:6] 1.266 0.325 -1.556 -1.556 -0.616 ...
##   ..- attr(*, "scaled:center")= Named num [1:6] 1.35e-15 -1.10e-16 2.50e-16 4.93e-17 6.19e-17 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "scaled:scale")= Named num [1:6] 0.837 0.832 0.698 0.661 0.646 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "traveltime" "studytime" "failures" ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:356] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:6] "age" "traveltime" "studytime" "failures" ...
##  - attr(*, "class")= chr [1:2] "diana" "twins"
```

```r
diana_cluster$dc
```

```
## [1] 0.9352352
```

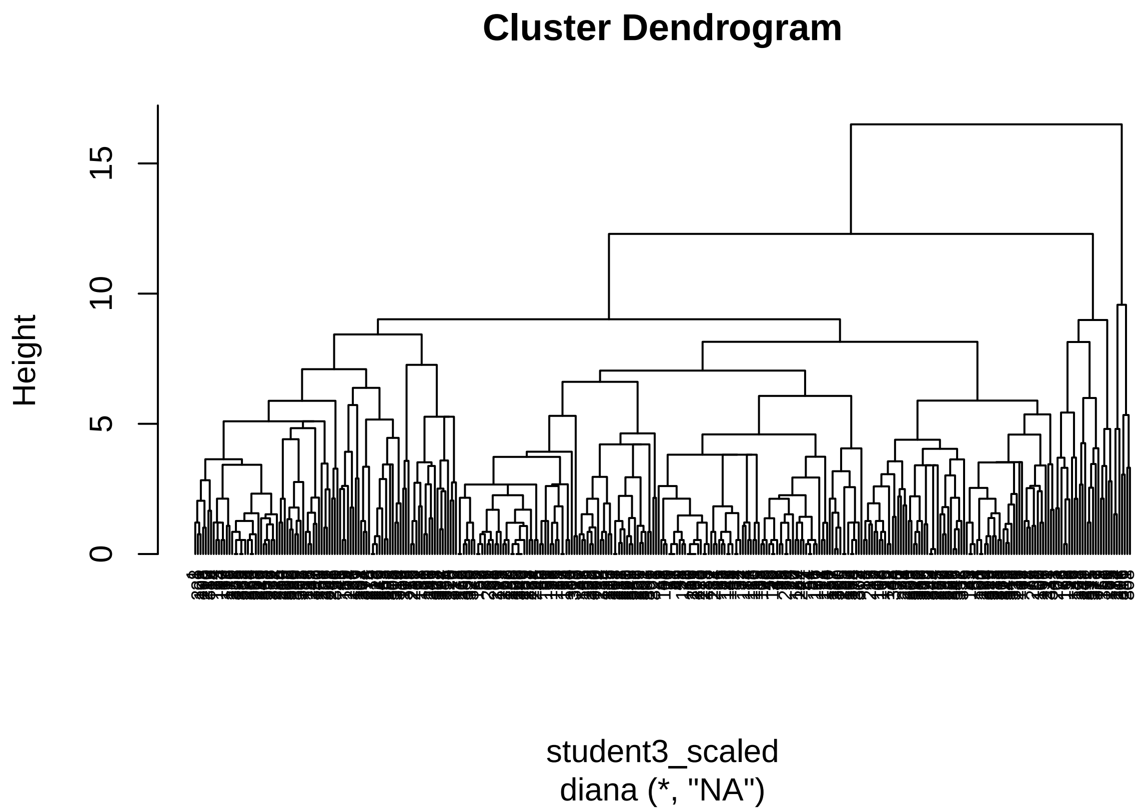
```r
diana_tree <- pltree(diana_cluster, cex = 0.6, hang = -1, main = "Dendrogram of Diana")
```



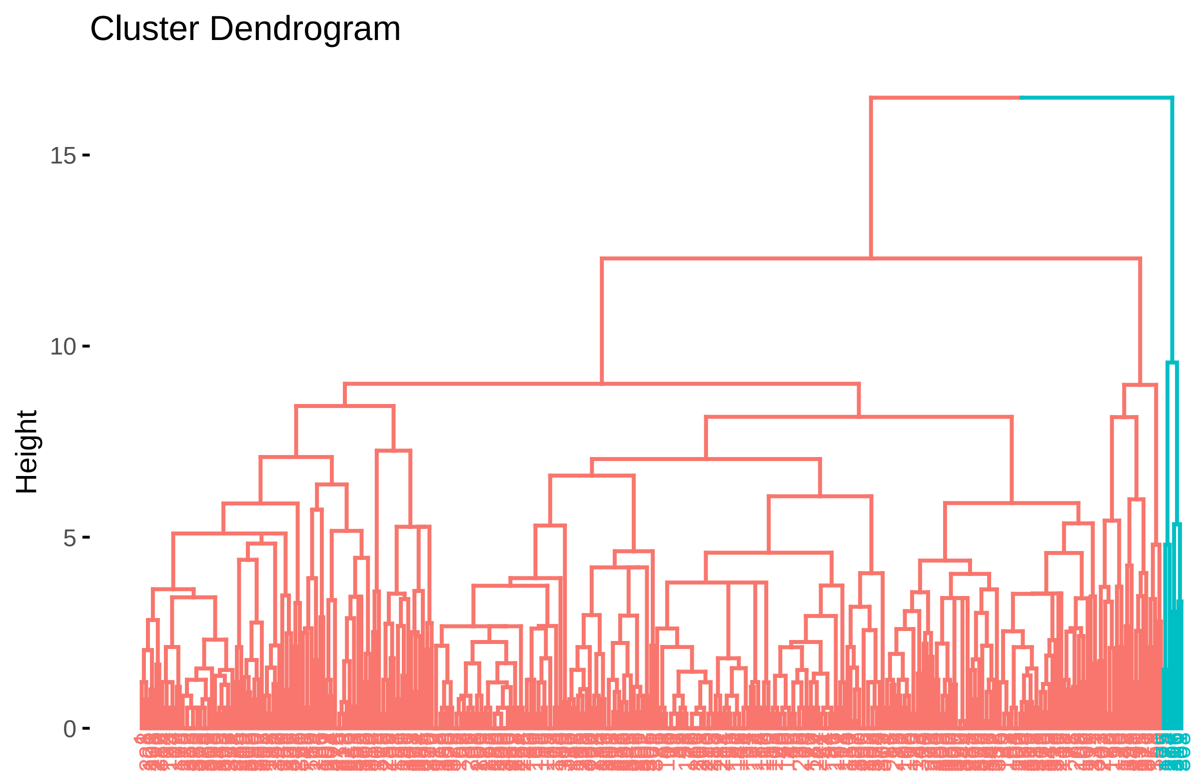

```r
print(diana_tree)
```

```
## NULL
```

```r
plot(as.hclust(diana_cluster), cex = 0.6, hang = -1)
```



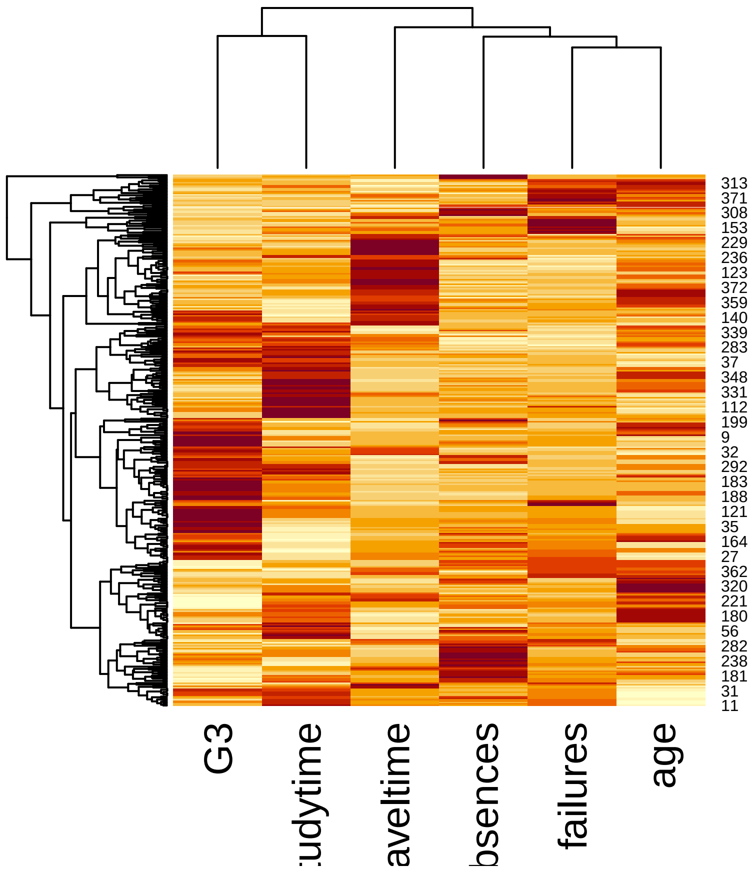

```
fviz_dend(diana_cluster, cex=.6, k=2)
```



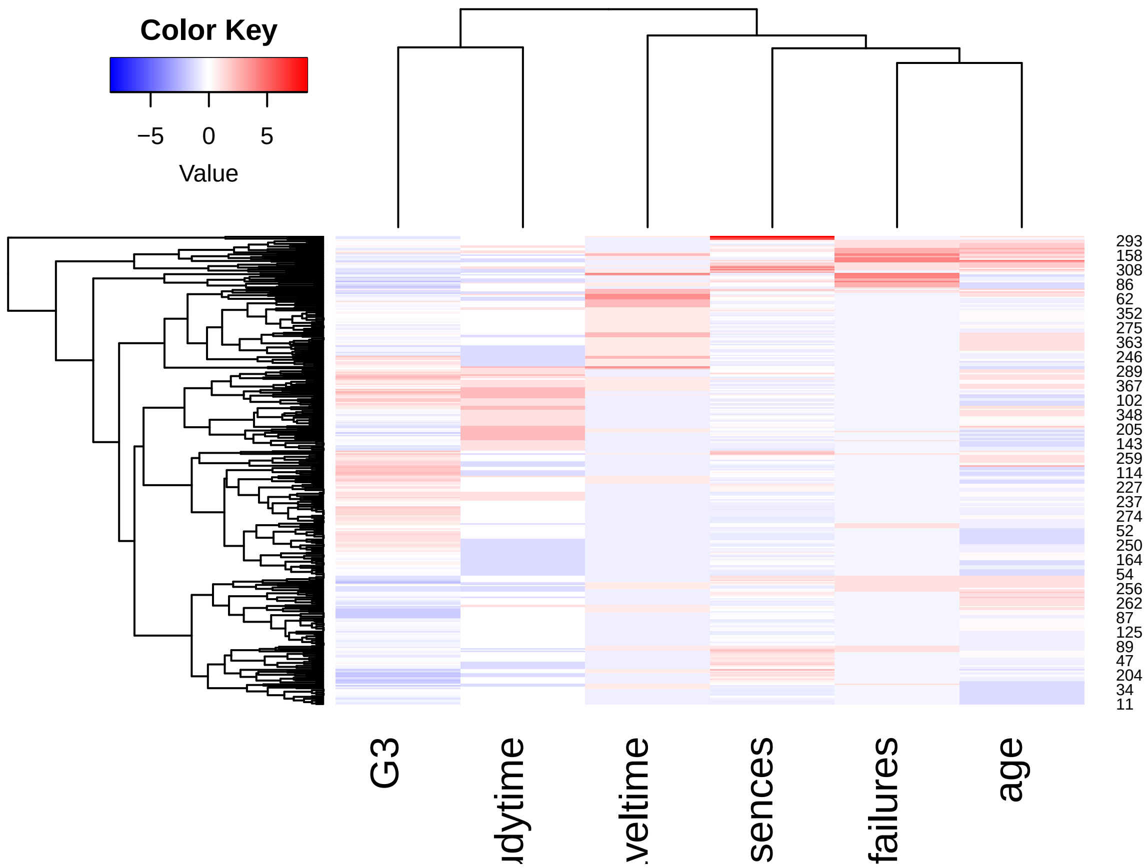
Cluster Dendrogram
Height
15
10
5
0

8. HIERARCHICAL CLUSTERING (THE HEATMAPS)

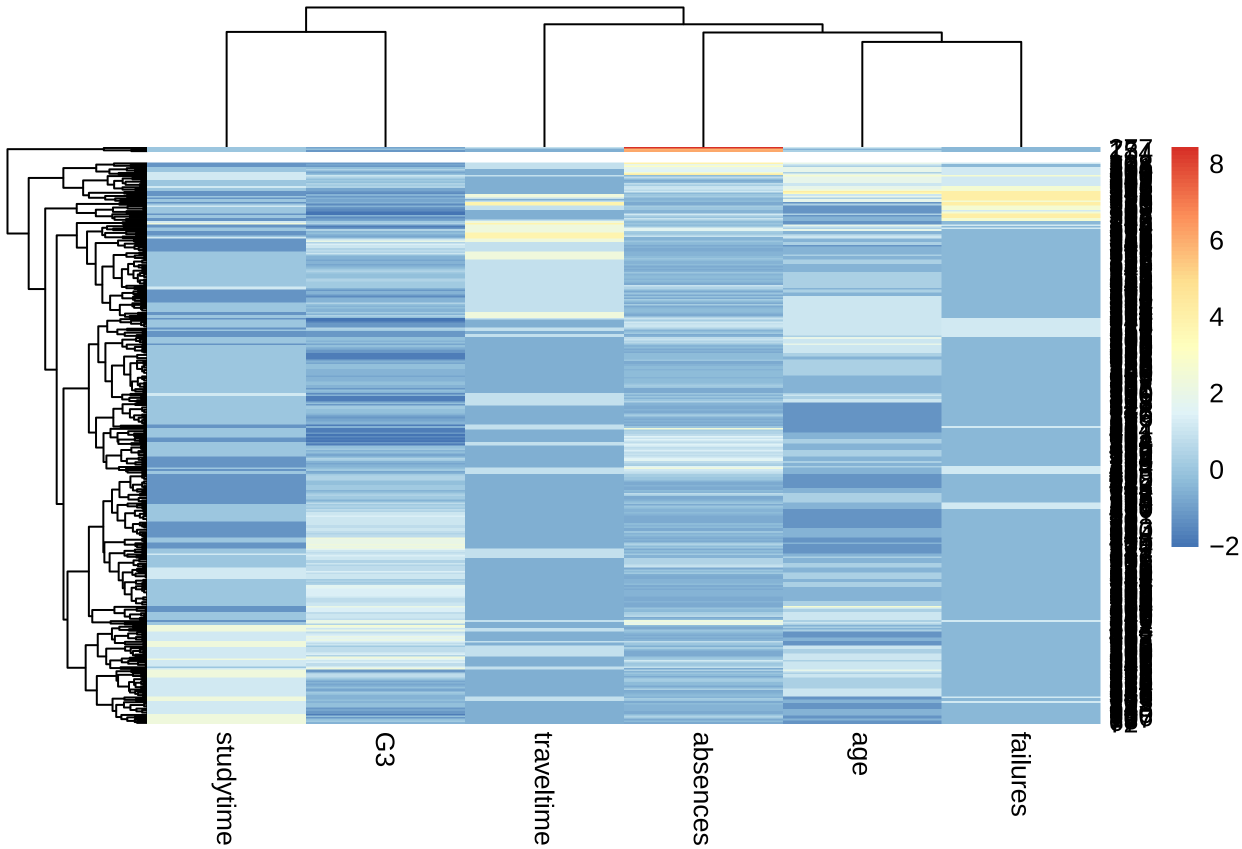
```r
# using HEATMAP : the high values are in red and low in yellow.

heatmap(student3_scaled)
```




```r
heatmap.2(student3_scaled,
          scale="none",
          col=bluered(100),
          trace = "none", density.info = "none")
```



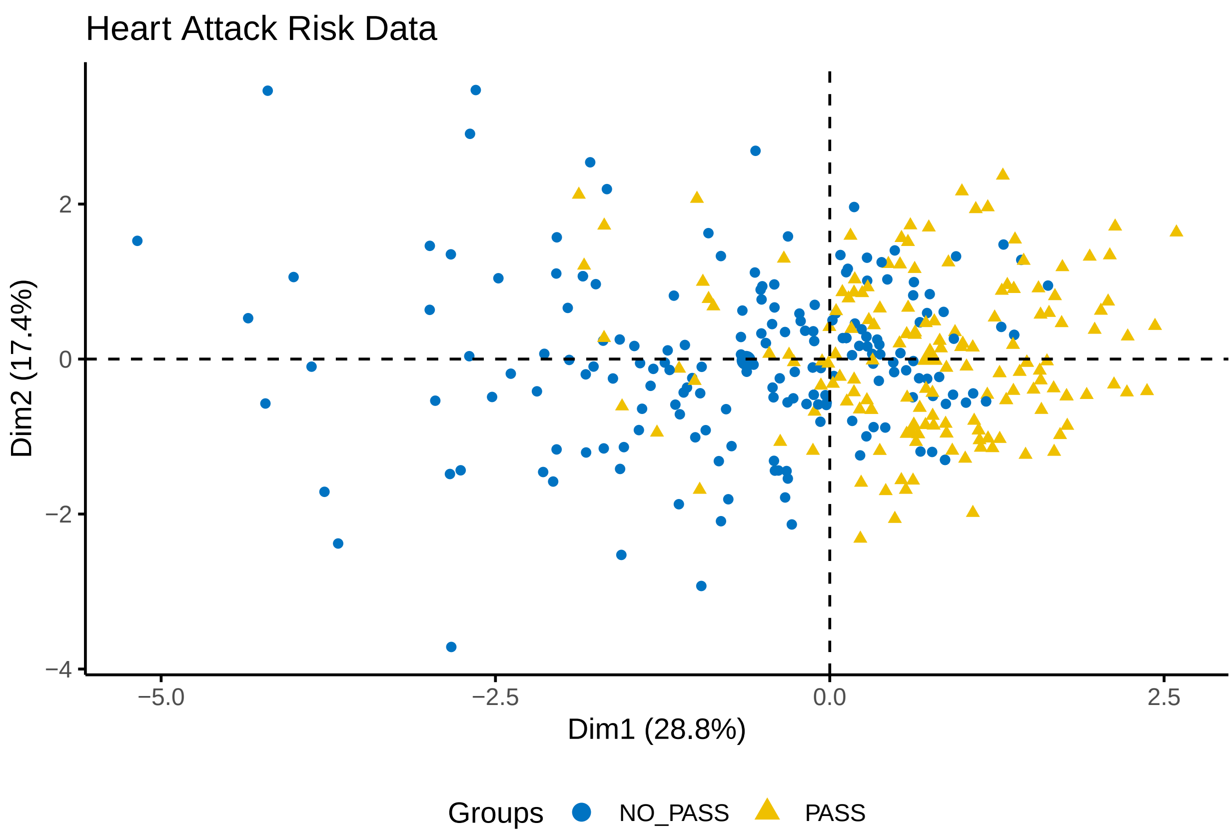

```
# using PHEATMAP

pheatmap(student3_scaled, cutree_rows = 2)
```




```r
# using D3HEATMAP

# d3heatmap(scale(student3), k_row=4, k_col=2)
```

9. HIERARCHICAL CLUSTERING (CLUSTER TENDENCY)

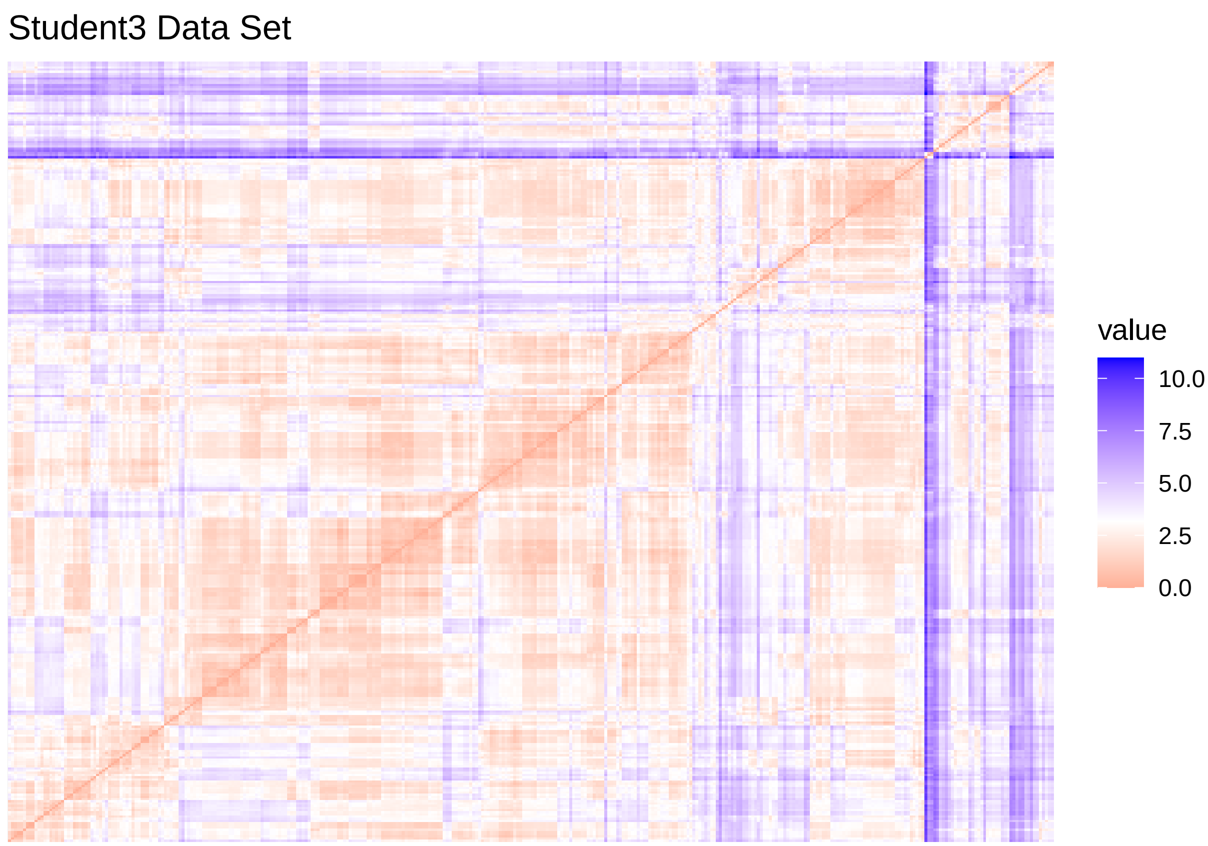
```
## TRANSFORMING G3 into RANGES of PASS and NO-PASS :

student3$G3 = as.integer(student3$G3)

student3$RESULT[student3$G3 <= 10]  = "NO_PASS"
student3$RESULT[student3$G3 >=10 ]  = "PASS"

student3 <- subset(student3, select = -c(G3))

student3$RESULT = as.factor(student3$RESULT)

## displaying the PCA analysis :

fviz_pca_ind(prcomp(student3_scaled),
             title="Heart Attack Risk Data",
             habillage = student3$RESULT,
             palette = "jco",
             geom = "point", ggtheme=theme_classic(), legend="bottom" )
```




```
# Calculating Hopkins Statistics to check if the data does exhibit inherent patterns :

hopkins(student3_scaled, n=nrow(student3_scaled) - 1)
```

```
## $H
## [1] 0.1844118
```

```r
# Visualizing the Dissimilarity Matrix
# where RED depicts high similarity and BLUE low similarity

fviz_dist(dist(student3_scaled), show_labels = FALSE) +
         labs(title = "Student3 Data Set")
```




```r
# using : validation="internal"
cluster_method <- c("hierarchical", "kmeans", "pam", "diana", "agnes")

check <- clValid(student3_scaled,
                 nClust=2:6,
                 clMethods=cluster_method, validation="internal")

summary(check)
```

```
##
## Clustering Methods:
##  hierarchical kmeans pam diana agnes
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                   2       3       4       5       6
##
## hierarchical Connectivity    4.3579 13.6944 25.1456 27.5996 31.3520
```

```
##              Dunn            0.1895   0.2597   0.2033   0.2033   0.2033
##              Silhouette      0.4931   0.4669   0.3837   0.3526   0.3072
## kmeans       Connectivity   52.6861 103.7401 109.0377 156.4282  76.1706
##              Dunn            0.0447   0.0324   0.0450   0.0376   0.1357
##              Silhouette      0.3051   0.1945   0.2004   0.1680   0.2219
## pam          Connectivity   90.1119 112.4353 127.4925 169.9107 178.9607
##              Dunn            0.0305   0.0284   0.0300   0.0300   0.0300
##              Silhouette      0.1614   0.1228   0.1340   0.1506   0.1697
## diana        Connectivity   11.6155  32.8206  39.6988  65.1560  66.2250
##              Dunn            0.1392   0.1768   0.1823   0.1363   0.1400
##              Silhouette      0.4674   0.4036   0.3717   0.1944   0.1964
## agnes        Connectivity    4.3579  13.6944  25.1456  27.5996  31.3520
##              Dunn            0.1895   0.2597   0.2033   0.2033   0.2033
##              Silhouette      0.4931   0.4669   0.3837   0.3526   0.3072
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 4.3579 hierarchical 2
## Dunn         0.2597 hierarchical 3
## Silhouette   0.4931 hierarchical 2
```

```r
# using : validation="stability"
cluster_method <- c("hierarchical", "kmeans", "pam", "diana", "agnes")

check_stability <- clValid(student3_scaled,
                           nClust=2:6,
                           clMethods=cluster_method, validation="stability")

optimalScores(check_stability)
```

```
##          Score Method Clusters
## APN 0.01297107  agnes        2
## AD  2.53949039    pam        6
## ADM 0.11002497  agnes        2
## FOM 0.96950356    pam        5
```

10. CONCLUSIONS

Here above we have compared the algorithms that perform the CLUSTERING, particularly K-MEANS, PAM (PARTITIONING AROUND MEDOIDS) and HC (HIERARCHICAL CLUSTERING).

We could draw several conclusions from our study:

1. Referring to the optimal number of clusters to be used for the K-MEANS algorithm,
   WSS and GAP methods suggest to call 7 CLUSTERS by K-MEANS, while SILHOUETTE method suggests to use 2 CLUSTERS.

2. We have employed both PAM and K-MEANS on 2 clusters, although after visual examination, PAM does not seem to have worked too well (by visualizing the data on a dimensionality reduction plot), in contrast to K-MEANS that has achieved a better separation.

3. Referring to the COPHENETIC DISTANCES and the set of CLUSTERING METHODS, we obtain the following values for the COPHENETIC DISTANCES (euclidean : 0.47, minkowski : 0.73, canberra : 0.62, manhattan : 0.78), suggesting that the MANHATTAN DISTANCE may provide more accurate results (followed by MINKOWSKI DISTANCE).

As we have read in some text books, "It can be argued that a dendrogram is an appropriate summary of some data if the correlation between the original distances and the cophenetic distances is high. Otherwise, it should simply be viewed as the description of the output of the clustering algorithm."

4. The DENDROGRAMS and the CLUSTERING data generated by the measures "WARD.D2/euclidean" and "WARD.D2/canberra" look similar, while the results of the pipelines "Minkowski/average" and "Manhattan/average" look very similar too.

5. The results of AGNES algorithm look more like "WARD.D2/euclidean" and "WARD.D2/canberra", while the results of DIANA algorithm look more like "Minkowski/average" and "Manhattan/average".

6. We have also displayed the HEATMAPS using the functions "heatmap.2", "pheatmap" or "d3.heatmap" functions.
7. Referring to Hopkins statistics value of the data, it is 0.1850252, suggesting that the data is uniformly distributed (according to the interpretation that we can read in Wikipedia "a value close to 1 tends to indicate the data is highly clustered, random data will tend to result in values around 0.5, and uniformly distributed data will tend to result in values close to 0").

Therefore, the lower the number of the clusters is, the better the modelling approach is.

8. At the end, we will have compared all these approaches ("hierarchical", "kmeans", "pam", "diana", "agnes") and we'll have evaluated the performance by using the function *clValid()*. (https://cran.r-project.org/web/packages/clValid/vignettes/clValid.pdf)

9. According to the documentation of clValid on "Internal Validation", we recall that "the connectivity should be minimized, while both the Dunn Index and the Silhouette Width should be maximized."

We have obtained optimal scores for the “Hierarchical Clustering” approach, using 2 or 3 clusters.

10. Shall we consider the “Stability Score” (and the associated measures APN, AD, ADM, and FOM), we recall that “these measures should be minimized in each case”.

In our case, we have obtained optimal scores on PAM (6 clusters) and AGNES (2 clusters).