

Notebook 3

Group 10 (Umar, Cory, Caroline, Benji)

10/9/2022

Source: <https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset>

This is a dataset based off of 70,000 records of patient data (Heart Related). Columns (13): ID, Age, Height(cm), Weight(kg), Gender, Systolic Blood Pressure (AP_HIGH), Diastolic Blood Pressure (AP_LOW), Cholesterol, Glucose, Smoking, Alcohol Intake, Physical Activity, Presence or Absence of cardiovascular disease.

The .csv file needed to be edited a bit in Microsoft Excel before using it in R. I just performed a split column delimiter function around semicolons, to divide the singular column that existed into 13. Each row had 13 variables in 1 column separated by semicolons, the function I ran split it up into 13 columns, making a 70,000 x 13 table.

<https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings>

Visit the website above to better understand Systolic and Diastolic Blood Pressure

Cleaning Data

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(flexclust)
```

```
## Warning: package 'flexclust' was built under R version 4.1.3
```

```
## Loading required package: grid
```

```
## Loading required package: lattice
```

```
## Loading required package: modeltools
```

```
## Loading required package: stats4
```

```
library(mclust)
```

```
## Warning: package 'mclust' was built under R version 4.1.3
```

```
## Package 'mclust' version 5.4.10
```

```
## Type 'citation("mclust")' for citing this R package in publications.
```

```

# Read in .csv file
heart <- read.csv("cardio_train.csv")

# Clean out any rows that have an unrealistic blood pressure (AP_HIGH & AP_LOW)
# They looked to be input errors by the person who made the data set
s <- subset(heart, AP_HIGH > 50)

s1 <- subset(s, AP_HIGH < 200)
s2 <- subset(s1, AP_LOW > 25)
s3 <- subset(s2, AP_LOW < 200)

# Removing little people(4'10") and Giants(7'3"), values are in cm
s4 <- subset(s3, HEIGHT > 147)

s5 <- subset(s4, HEIGHT < 220)

# Removing anyone below 90 lbs and above 375 lbs, the values are in kg
s6 <- subset(s5, WEIGHT > 40)

h1 <- subset(s6, WEIGHT < 180)

# AGE is in days so to get years i just divide by 365
h1$AGE <- (h1$AGE / 365)

# Removing people under 40
h <- subset(h1, AGE > 39)

# Checking for any NA values
# There is none
colSums(is.na(h))

```

```

##          ID          AGE          GENDER          HEIGHT
##          0           0           0           0
##      WEIGHT      AP_HIGH      AP_LOW      CHOLESTEROL
##          0           0           0           0
##      GLUCOSE      SMOKE      ALCOHOL PHYSICAL_ACTIVITY
##          0           0           0           0
##  CARDIO_DISEASE
##          0

```

```

# Everything that should be factored is factored
h$GENDER <- factor(h$GENDER)
h$CHOLESTEROL <- factor(h$CHOLESTEROL)
h$GLUCOSE <- factor(h$GLUCOSE)
h$SMOKE <- factor(h$SMOKE)
h$ALCOHOL <- factor(h$ALCOHOL)
h$PHYSICAL_ACTIVITY <- factor(h$PHYSICAL_ACTIVITY)
h$CARDIO_DISEASE <- factor(h$CARDIO_DISEASE)

```

```
# There is now 67,685 rows
```

```
str(h)
```

```
## 'data.frame':    67685 obs. of  13 variables:
## $ ID           : int  0 1 2 3 4 8 9 12 13 14 ...
## $ AGE          : num  50.4 55.4 51.7 48.3 47.9 ...
## $ GENDER       : Factor w/ 2 levels "1","2": 2 1 1 2 1 1 1 2 1 1 ...
## $ HEIGHT       : int  168 156 165 169 156 151 157 178 158 164 ...
## $ WEIGHT       : num  62 85 64 82 56 67 93 95 71 68 ...
## $ AP_HIGH      : int  110 140 130 150 100 120 130 130 110 110 ...
## $ AP_LOW       : int  80 90 70 100 60 80 80 90 70 60 ...
## $ CHOLESTEROL  : Factor w/ 3 levels "1","2","3": 1 3 3 1 1 2 3 3 1 1 ...
## $ GLUCOSE      : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 2 1 3 1 1 ...
## $ SMOKE        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ ALCOHOL      : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ PHYSICAL_ACTIVITY: Factor w/ 2 levels "0","1": 2 2 1 2 1 1 2 2 2 1 ...
## $ CARDIO_DISEASE : Factor w/ 2 levels "0","1": 1 2 2 2 1 1 1 2 1 1 ...
```

Kmean Clusters

Determining how many numbers of clusters to use for the data

```
# Function for finding optimal number of clusters
```

```
hplot <- function(data, nc = 15, seed = 2354)
{
  hss <- (nrow(h) - 1) * sum(apply(h, 2, var))
  for (i in 2:nc)
  {
    set.seed(seed)
    hss[i] <- sum(kmeans(h, centers = i)$withinss)
  }
  plot(1:nc, hss, type = "b", xlab = "Number of Clusters",
       ylab = "Within groups sum of squares")
}
```

```
# Show graph
```

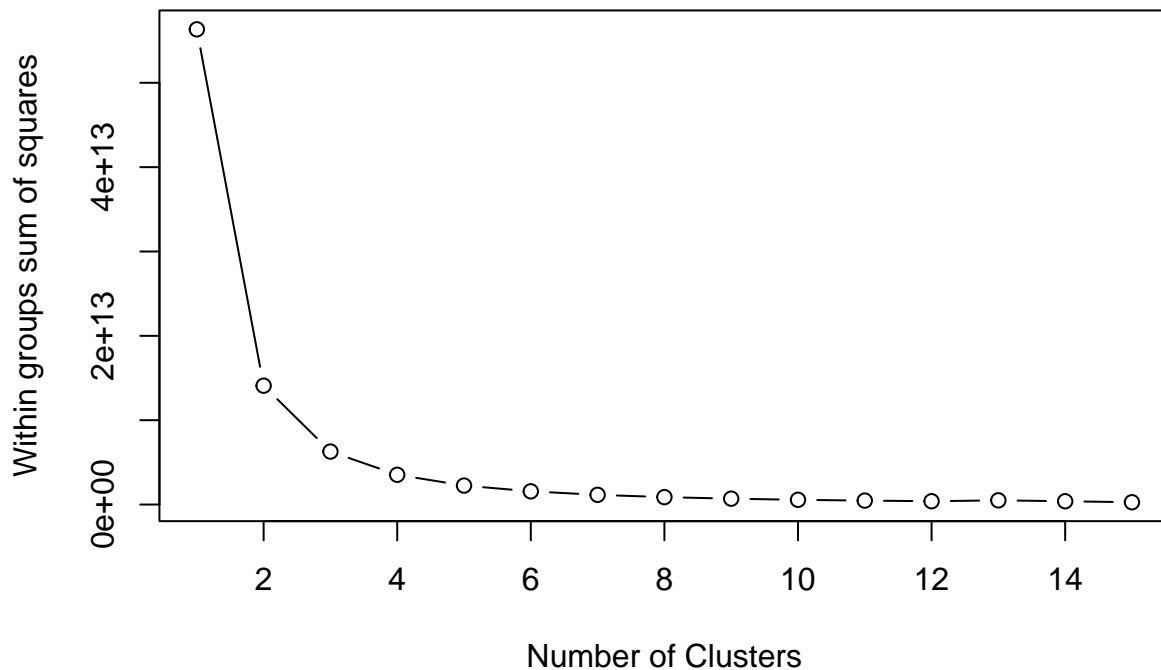
```
hplot(h)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 3384250)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 3384250)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 3384250)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 3384250)
```



```
cat("Looking at the graph, three seems to be the optimal number of clusters.\n
That's where the 'elbow' is.")
```

```
## Looking at the graph, three seems to be the optimal number of clusters.
##
## That's where the 'elbow' is.
```

Making the kmean clusters based off of 3 clusters

```
# Set unique and repeatable random variable

set.seed(2354)

# Clustering off of Systolic and Diastolic blood pressure

heartCluster <- kmeans(h[, 6:7], 3, nstart = 20)

# Data is too large to show the summary for

cat("K-means clustering with 3 clusters of sizes 12542, 18119, 37024

Cluster means:
  AP_HIGH  AP_LOW
1 106.5830 70.24629
2 148.3140 90.72096
```

```
3 122.4537 80.46778
```

```
Clustering vector:
```

```
[ reached getOption('max.print') -- omitted 66685 entries ]
```

```
Within cluster sum of squares by cluster:
```

```
[1] 1330107 3536136 1932918  
(between_SS / total_SS = 71.9 %)
```

```
Available components:
```

```
[1] 'cluster'      'centers'      'totss'        'withinss'  
[5] 'tot.withinss' 'betweenss'    'size'         'iter'  
[9] 'ifault'      ")
```

```
## K-means clustering with 3 clusters of sizes 12542, 18119, 37024
```

```
##
```

```
## Cluster means:
```

```
##      AP_HIGH  AP_LOW
```

```
## 1 106.5830 70.24629
```

```
## 2 148.3140 90.72096
```

```
## 3 122.4537 80.46778
```

```
##
```

```
## Clustering vector:
```

```
##
```

```
## [ reached getOption('max.print') -- omitted 66685 entries ]
```

```
##
```

```
## Within cluster sum of squares by cluster:
```

```
## [1] 1330107 3536136 1932918
```

```
## (between_SS / total_SS = 71.9 %)
```

```
##
```

```
## Available components:
```

```
##
```

```
## [1] 'cluster'      'centers'      'totss'        'withinss'
```

```
## [5] 'tot.withinss' 'betweenss'    'size'         'iter'
```

```
## [9] 'ifault'
```

Looking at clusters through the lens of having heart disease

```
# Comparing the cluster and the presence of heart disease
```

```
# A '1' on the x-axis means they have heart disease
```

```
table(heartCluster$cluster, h$CARDIO_DISEASE)
```

```
##
```

```
##      0      1
```

```
## 1 9621 2921
```

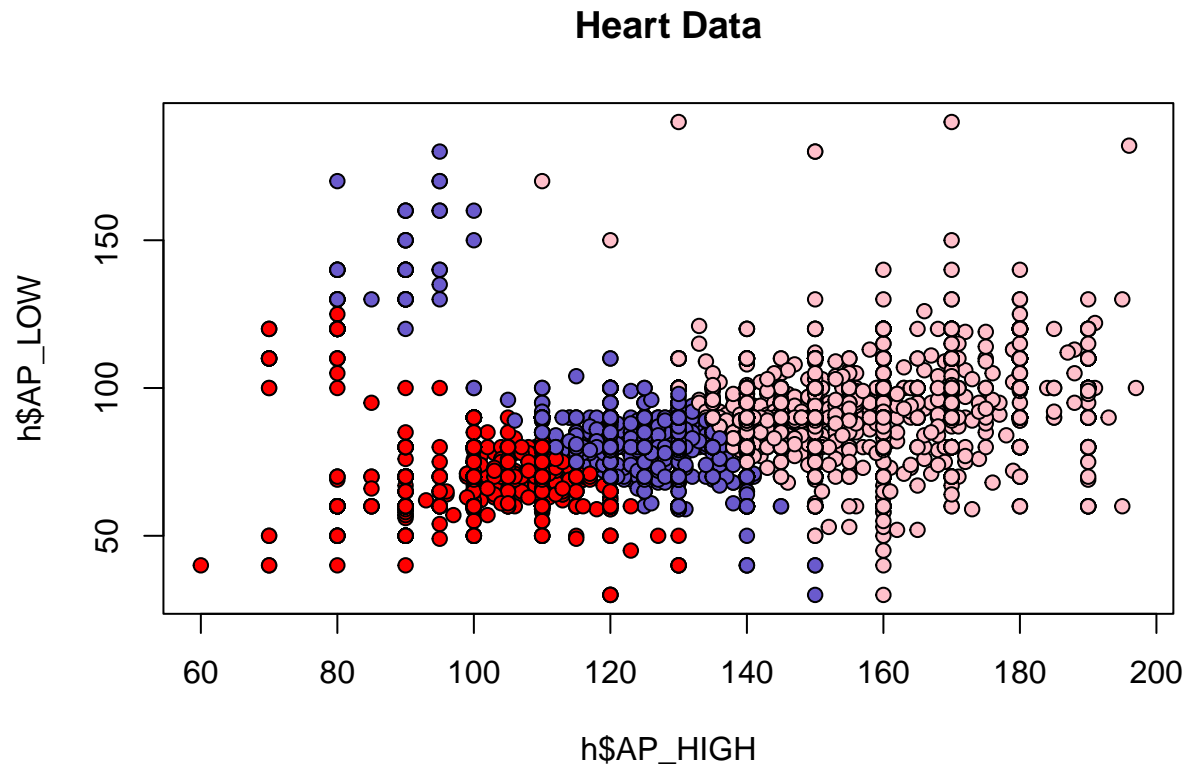
```
## 2 2976 15143
```

```
## 3 21662 15362
```

Displaying clusters on scatter plot

```
# Plotting the clusters on a scatter plot
```

```
plot(h$AP_HIGH, h$AP_LOW, pch = 21, bg = c("red", "pink", "slateblue")
      [unclass(heartCluster$cluster)], main = "Heart Data")
```



Hierarchical Clustering

```
# Removing any categorical data and ID's
```

```
hrt <- subset(h, select = -c(ID, GENDER, CHOLESTEROL, GLUCOSE, SMOKE, ALCOHOL,
                             PHYSICAL_ACTIVITY, CARDIO_DISEASE))
```

```
# Limiting data points to 50 for the sake of the visibility of the table
```

```
hrt <- hrt[1:50,]
```

```
# Displaying new table
```

```
head(hrt)
```

```
##      AGE HEIGHT WEIGHT AP_HIGH AP_LOW
## 1 50.39178   168    62    110     80
## 2 55.41918   156    85    140     90
```

```
## 3 51.66301    165    64    130    70
## 4 48.28219    169    82    150   100
## 5 47.87397    156    56    100    60
## 6 60.03836    151    67    120    80
```

```
# Scaling the data
```

```
hrt.scaled <- scale(hrt)
```

```
head(hrt.scaled)
```

```
##      AGE      HEIGHT      WEIGHT      AP_HIGH      AP_LOW
## 1 -0.2993365  0.5634783 -0.7089041 -1.1066532  0.04066093
## 2  0.4338026 -1.0619399  0.8588646  1.0352563  1.05718406
## 3 -0.1139542  0.1571238 -0.5725764  0.3212864 -0.97586221
## 4 -0.6069752  0.6989298  0.6543730  1.7492261  2.07370719
## 5 -0.6665053 -1.0619399 -1.1178873 -1.8206231 -1.99238534
## 6  1.1074115 -1.7391974 -0.3680848 -0.3926834  0.04066093
```

Displaying Hierarchical Graph (Dendrogram)

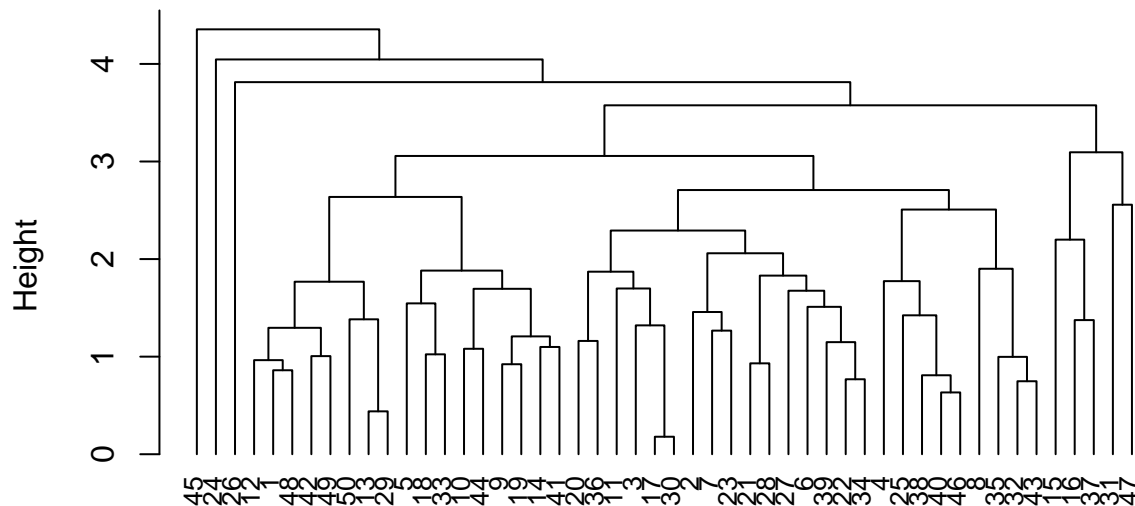
```
# Finding the distances between each data point
```

```
di <- dist(hrt.scaled)
```

```
fit.average <- hclust(di, method = "average")
```

```
plot(fit.average, hang = -1, cex = .8, main = "Hierarchical Clustering")
```

Hierarchical Clustering



di
hclust (*, "average")

Cutting data

```
# Learning more about the data through cutting

# SIDENOTE: Hierarchical Clustering isn't the easiest to understand, but is good
# for learning more about your data

for (c in 3:11)
{
  cluster_cut <- cutree(fit.average, c)
  table_cut <- table(cluster_cut, hrt$AP_LOW)
  print(table_cut)
  ri <- randIndex(table_cut)
  print(paste("cut=", c, "Rand Index = ", ri))
}

##
## cluster_cut 60 70 80 85 90 100
##           1  3 11 18  2 12  2
##           2  1  0  0  0  0  0
##           3  0  0  1  0  0  0
## [1] "cut= 3 Rand Index = 0.00629370629370629"
##
## cluster_cut 60 70 80 85 90 100
##           1  3 11 18  2 12  1
##           2  1  0  0  0  0  0
```



```

##          3  0  0  0  0  0  1
##          4  0  0  1  0  0  0
## [1] "cut= 4 Rand Index =  0.0311563810665069"
##
## cluster_cut 60 70 80 85 90 100
##          1  3 11 15  2 10  1
##          2  0  0  3  0  2  0
##          3  1  0  0  0  0  0
##          4  0  0  0  0  0  1
##          5  0  0  1  0  0  0
## [1] "cut= 5 Rand Index = -0.000822481151473578"
##
## cluster_cut 60 70 80 85 90 100
##          1  3 11 15  2 10  1
##          2  0  0  2  0  1  0
##          3  1  0  0  0  0  0
##          4  0  0  0  0  0  1
##          5  0  0  1  0  1  0
##          6  0  0  1  0  0  0
## [1] "cut= 6 Rand Index = -0.00495526496902962"
##
## cluster_cut 60 70 80 85 90 100
##          1  3  7  7  0  0  0
##          2  0  4  8  2 10  1
##          3  0  0  2  0  1  0
##          4  1  0  0  0  0  0
##          5  0  0  0  0  0  1
##          6  0  0  1  0  1  0
##          7  0  0  1  0  0  0
## [1] "cut= 7 Rand Index =  0.0695719844357977"
##
## cluster_cut 60 70 80 85 90 100
##          1  3  7  7  0  0  0
##          2  0  4  8  2  2  0
##          3  0  0  0  0  8  1
##          4  0  0  2  0  1  0
##          5  1  0  0  0  0  0
##          6  0  0  0  0  0  1
##          7  0  0  1  0  1  0
##          8  0  0  1  0  0  0
## [1] "cut= 8 Rand Index =  0.166334841628959"
##
## cluster_cut 60 70 80 85 90 100
##          1  0  1  7  0  0  0
##          2  0  4  8  2  2  0
##          3  0  0  0  0  8  1
##          4  3  6  0  0  0  0
##          5  0  0  2  0  1  0
##          6  1  0  0  0  0  0
##          7  0  0  0  0  0  1
##          8  0  0  1  0  1  0
##          9  0  0  1  0  0  0
## [1] "cut= 9 Rand Index =  0.237241379310345"
##

```

```
## cluster_cut 60 70 80 85 90 100
##      1  0  1  7  0  0  0
##      2  0  4  8  2  2  0
##      3  0  0  0  0  8  1
##      4  3  6  0  0  0  0
##      5  0  0  2  0  1  0
##      6  1  0  0  0  0  0
##      7  0  0  0  0  0  1
##      8  0  0  1  0  0  0
##      9  0  0  1  0  0  0
##     10  0  0  0  0  1  0
## [1] "cut= 10 Rand Index =  0.238717632552404"
##
## cluster_cut 60 70 80 85 90 100
##      1  0  1  7  0  0  0
##      2  0  4  8  2  2  0
##      3  0  0  0  0  4  1
##      4  3  6  0  0  0  0
##      5  0  0  0  0  4  0
##      6  0  0  2  0  1  0
##      7  1  0  0  0  0  0
##      8  0  0  0  0  0  1
##      9  0  0  1  0  0  0
##     10  0  0  1  0  0  0
##     11  0  0  0  0  1  0
## [1] "cut= 11 Rand Index =  0.189734513274336"
```

Model Based

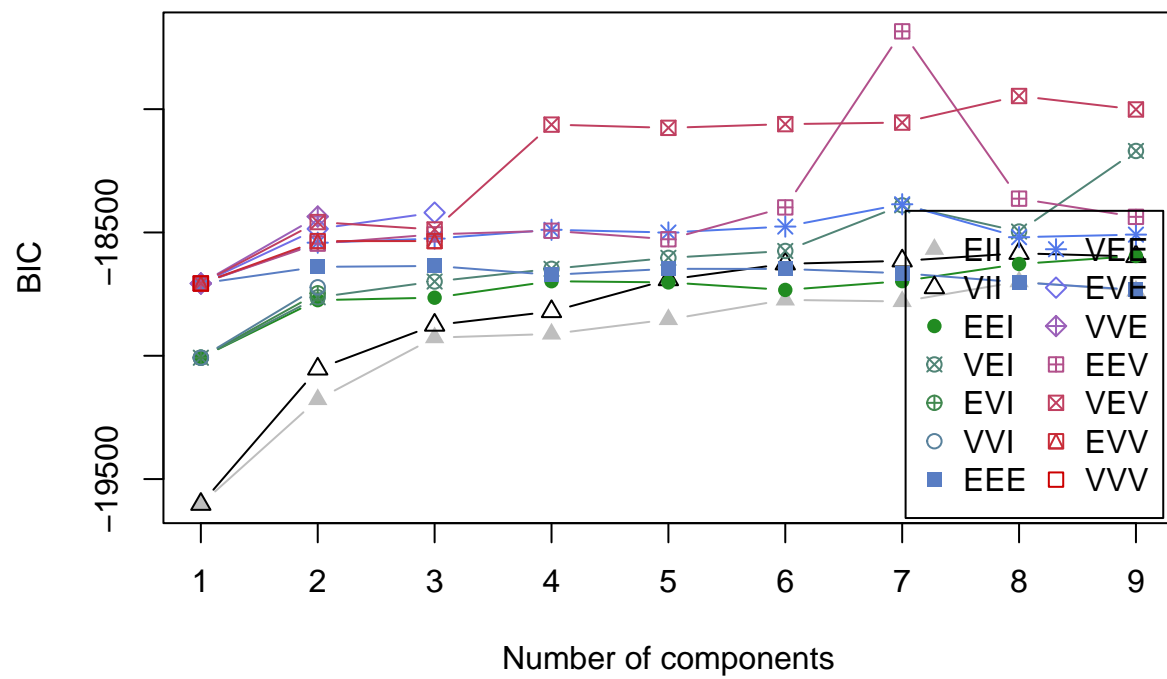
```
# Shrink data set to 500 for the sake of the algorithms

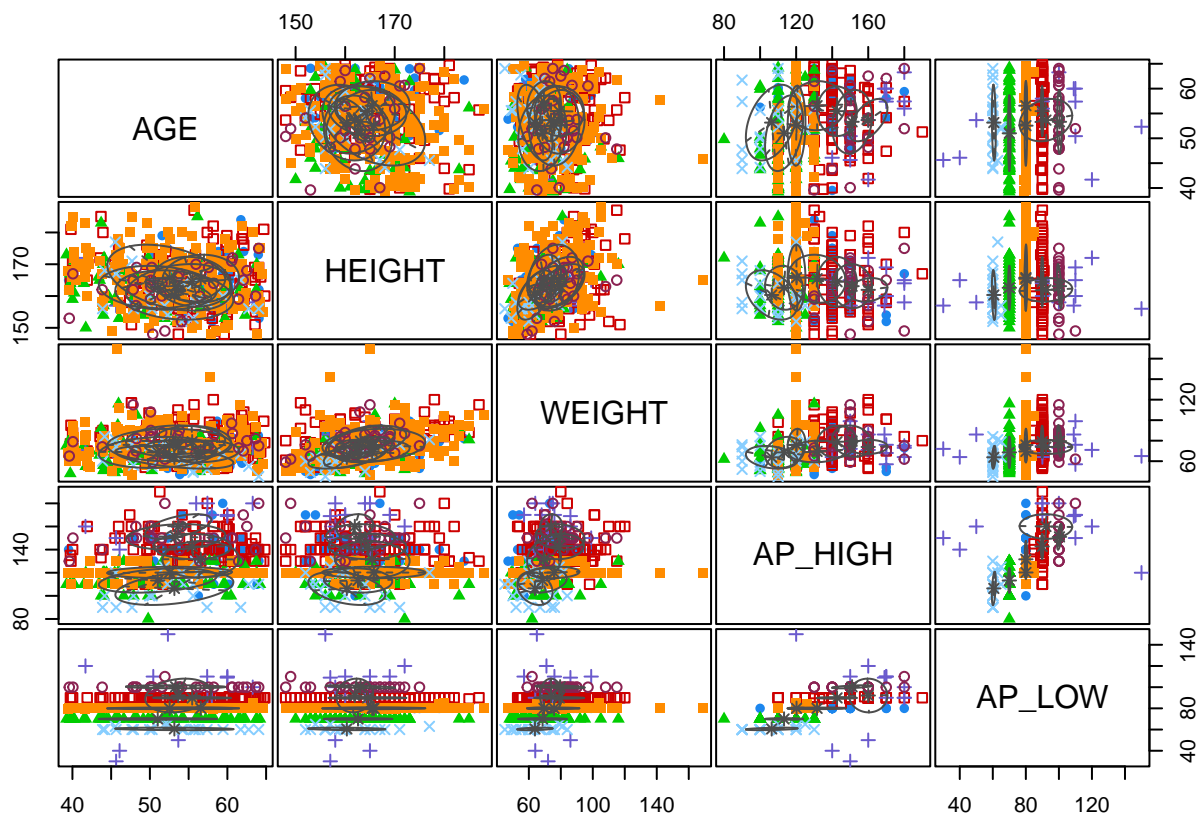
hhrt <- subset(h[1:500,], select = -c(ID, GENDER, CHOLESTEROL, GLUCOSE, SMOKE, ALCOHOL,
                                     PHYSICAL_ACTIVITY, CARDIO_DISEASE))

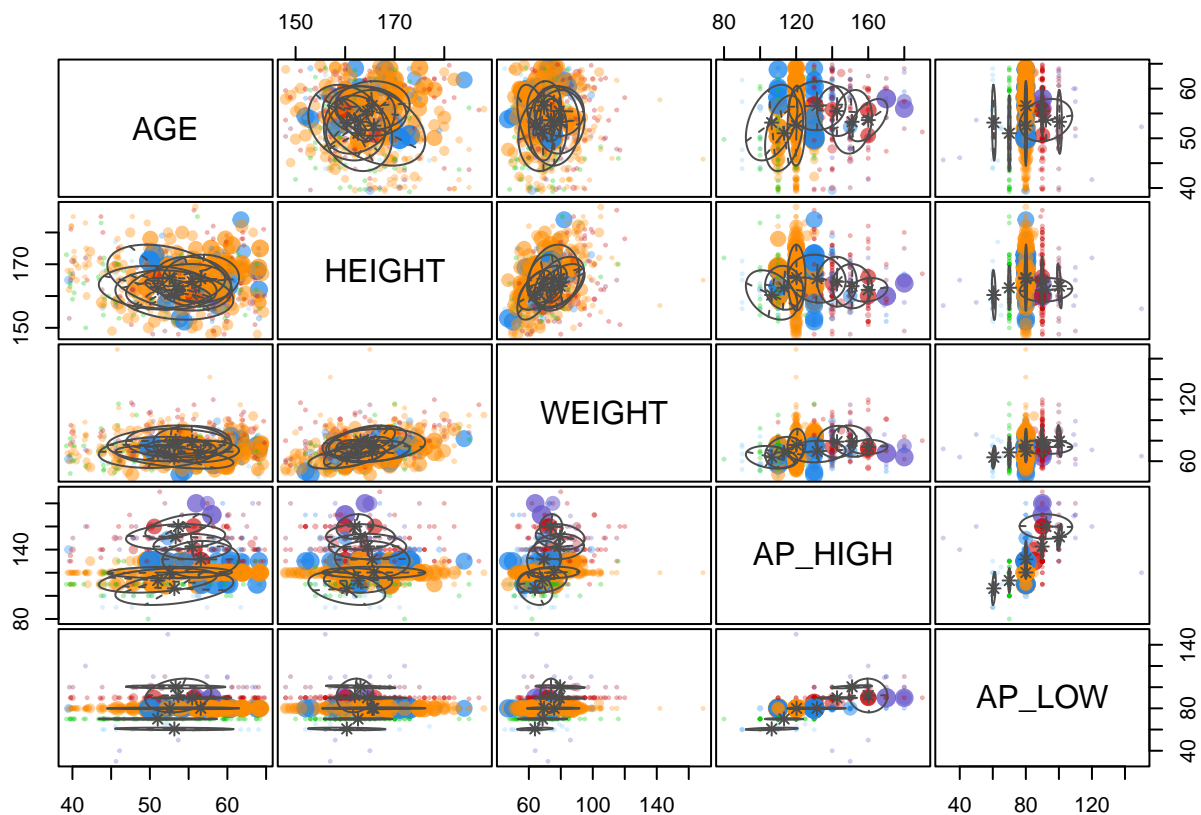
fitt <- Mclust(hhrt)

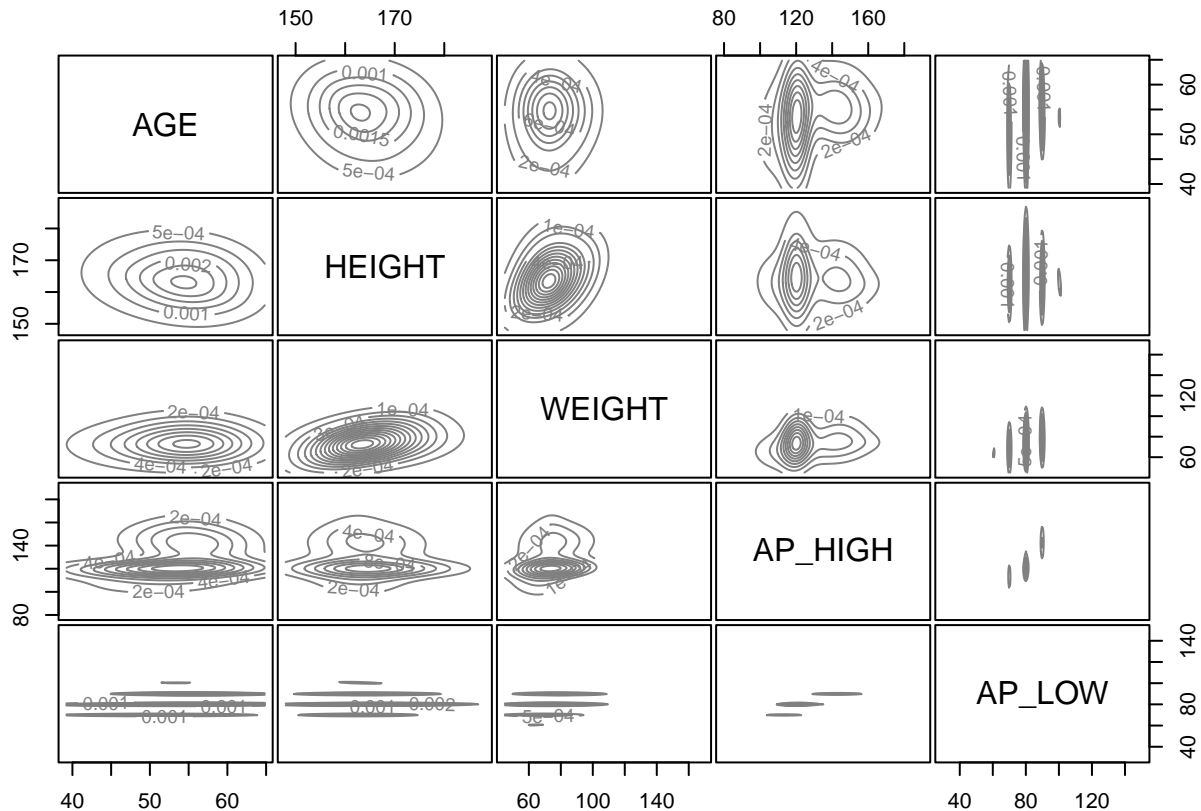
# Plot results

plot(fitt)
```









```
# Display the best model
```

```
summary(fitt)
```

```
## -----
## Gaussian finite mixture model fitted by EM algorithm
## -----
##
## Mclust EEV (ellipsoidal, equal volume and shape) model with 7 components:
##
## log-likelihood  n  df      BIC      ICL
##      -8481.832 500 116 -17684.56 -17765.93
##
## Clustering table:
##   1  2  3  4  5  6  7
##  50 113 76 13 198 21 29
```

```
cat("The first one is the BIC graph, it suggests EEV with 7 groups, based on the
\nhighest BIC value and the number of components it intersects with. The summary
\nof the model also suggests the same thing. Classification, uncertainty, and
\ndensity all have a very similar looking graphs. They just show correlation
\non a scatterplot matrix. Unsurprisingly, age and height seem to have very
\nlittle correlation, due to the age range of the data (40 - 65), and systolic
\nand diastolic blood pressure have the most correlation.")
```

```
## The first one is the BIC graph, it suggests EEV with 7 groups, based on the
##
## highest BIC value and the number of components it intersects with. The summary
##
## of the model also suggests the same thing. Classification, uncertainty, and
##
## density all have a very similar looking graphs. They just show correlation
##
## on a scatterplot matrix. Unsurprisingly, age and height seem to have very
##
## little correlation, due to the age range of the data (40 - 65), and systolic
##
## and diastolic blood pressure have the most correlation.
```