28/2/2022

Clustering in Medical Data

Mihalis Galanakis

Clustering in Medical Data

Mihalis Galanakis

28/2/2022

Objective

The data in the folder named data.txt refer to counts from different variables in a population of women, aiming to gain useful insights and explore different methods of clustering.

Read in the data

```
data <- read.table('C:/Users/mihal/OneDrive/data.txt',sep=",")
# Statistical Learning Project

# The following Dataset involves predicting the onset of diabetes withi
n 5 years in a women population given medical details.
# It is a binary (2-class) classification problem. The number of observ
ations for each class is not balanced.
# There are 768 observations with 8 input variables and 1 output variab
le. Missing values are believed to be encoded with zero values.
# The variable names are as follows:

# Number of times pregnant.
# Plasma glucose concentration a 2 hours in an oral glucose tolerance t
est.
# Diastolic blood pressure (mm Hg).
# Triceps skinfold thickness (mm).
# 2-Hour serum insulin (mu U/ml).
# Body mass index (weight in kg/(height in m)^2).
# Diabetes pedigree function.
# Age (years).
# Class variable (0 or 1).
```

Descriptive statistics

```
str(data)
```

```
## 'data.frame':    768 obs. of  9 variables:
##  $ V1: int  6 1 8 1 0 5 3 10 2 8 ...
##  $ V2: int  148 85 183 89 137 116 78 115 197 125 ...
##  $ V3: int  72 66 64 66 40 74 50 0 70 96 ...
##  $ V4: int  35 29 0 23 35 0 32 0 45 0 ...
##  $ V5: int  0 0 0 94 168 0 88 0 543 0 ...
##  $ V6: num  33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...
##  $ V7: num  0.627 0.351 0.672 0.167 2.288 ...
##  $ V8: int  50 31 32 21 33 30 26 29 53 54 ...
##  $ V9: int  1 0 1 0 1 0 1 0 1 1 ...
```

```
dim(data)
```

```
## [1] 768   9
```

```
summary(data)
```

```
##        V1               V2              V3               V4        
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00  
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00  
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00  
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54  
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00  
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00  
##        V5              V6              V7               V8        
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00  
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00  
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00  
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24  
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00  
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00  
##        V9        
##  Min.   :0.000  
##  1st Qu.:0.000  
##  Median :0.000  
##  Mean   :0.349  
##  3rd Qu.:1.000  
##  Max.   :1.000
```

```
colnames(data) <- c("t.pregnant","plasma","bl.press","tr.thick","serum.
ins","bmi","diab","age","class")
head(data,5)
```

```
##   t.pregnant plasma bl.press tr.thick serum.ins  bmi  diab age class
## 1          6    148       72       35         0 33.6 0.627  50     1
## 2          1     85       66       29         0 26.6 0.351  31     0
## 3          8    183       64        0         0 23.3 0.672  32     1
## 4          1     89       66       23        94 28.1 0.167  21     0
## 5          0    137       40       35       168 43.1 2.288  33     1
```

```
tail(data,5)
```

```
##     t.pregnant plasma bl.press tr.thick serum.ins  bmi  diab age cla
ss
## 764         10    101       76       48       180 32.9 0.171  63
0
## 765          2    122       70       27         0 36.8 0.340  27
0
## 766          5    121       72       23       112 26.2 0.245  30
0
## 767          1    126       60        0         0 30.1 0.349  47
1
```

```
## 768          1     93       70       31         0 30.4 0.315  23
0

class <- data$class
t.pregnant <- data$t.pregnant


expl_data <- data[,2:8]
# We assume that the zeros in the variable times.pregnant are not missi
ng, hence we don't replace zeros with "NA"

head(expl_data,10)

##    plasma bl.press tr.thick serum.ins  bmi  diab age
## 1     148       72       35         0 33.6 0.627  50
## 2      85       66       29         0 26.6 0.351  31
## 3     183       64        0         0 23.3 0.672  32
## 4      89       66       23        94 28.1 0.167  21
## 5     137       40       35       168 43.1 2.288  33
## 6     116       74        0         0 25.6 0.201  30
## 7      78       50       32        88 31.0 0.248  26
## 8     115        0        0         0 35.3 0.134  29
## 9     197       70       45       543 30.5 0.158  53
## 10    125       96        0         0  0.0 0.232  54

expl_data[expl_data==0] <- NA
# Replace the zeros with "NA"

df <- data.frame(t.pregnant,expl_data,class)
# Create the transformed dataframe
head(df,5)

##   t.pregnant plasma bl.press tr.thick serum.ins  bmi  diab age class
## 1          6    148       72       35        NA 33.6 0.627  50     1
## 2          1     85       66       29        NA 26.6 0.351  31     0
## 3          8    183       64       NA        NA 23.3 0.672  32     1
## 4          1     89       66       23        94 28.1 0.167  21     0
## 5          0    137       40       35       168 43.1 2.288  33     1

# View the first 5 obs of the df
tail(df,5)

##     t.pregnant plasma bl.press tr.thick serum.ins  bmi  diab age cla
ss
## 764         10    101       76       48       180 32.9 0.171  63
0
## 765          2    122       70       27        NA 36.8 0.340  27
0
## 766          5    121       72       23       112 26.2 0.245  30
0
## 767          1    126       60       NA        NA 30.1 0.349  47
```

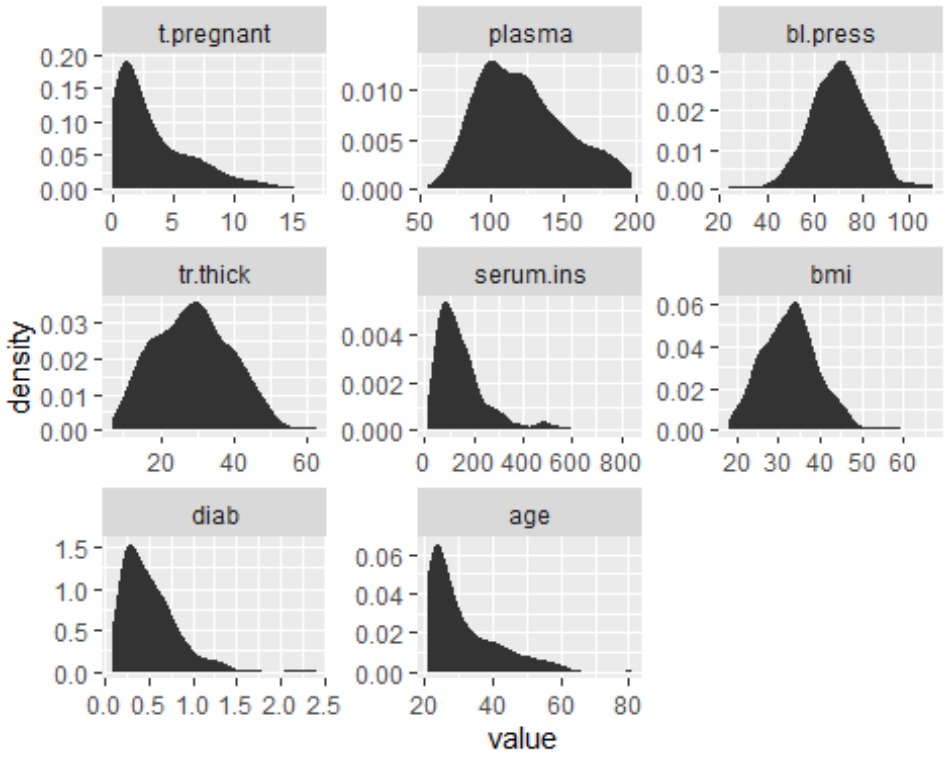
```
1
## 768          1     93       70       31        NA 30.4 0.315  23
0

# View the last 5 obs of the df
dim(df)

## [1] 768   9
```

Since we are not interested in using an imputed dataset we omit the missing values

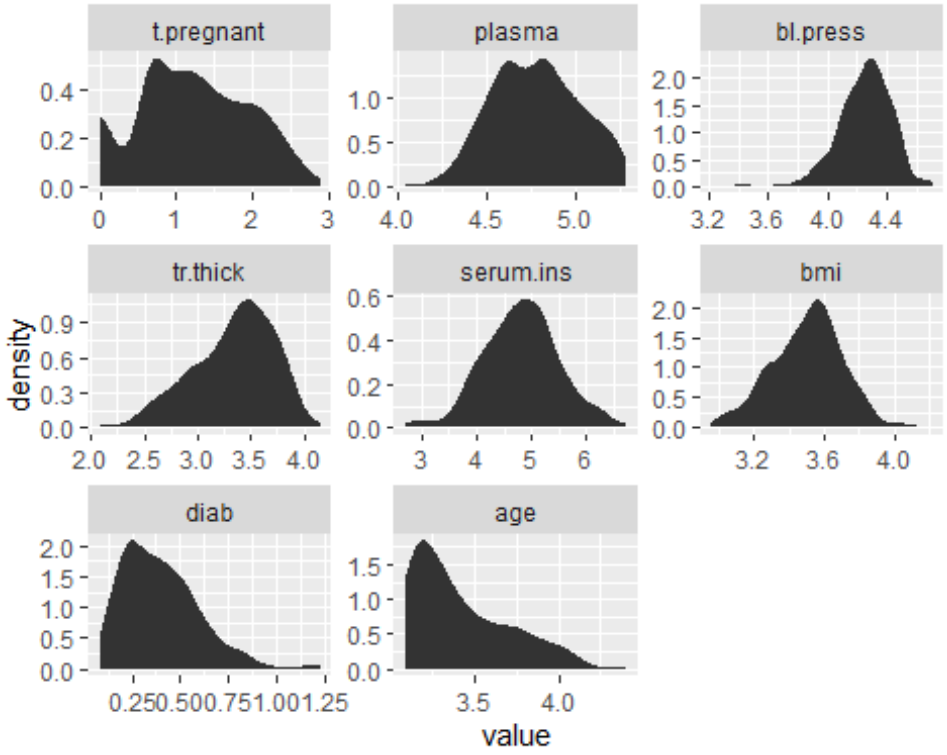
```
newdf <- na.omit(df)
# Create the final dataframe that doesn't include missing values!
str(newdf)

## 'data.frame':    392 obs. of  9 variables:
##  $ t.pregnant: int  1 0 3 2 1 5 0 1 1 3 ...
##  $ plasma    : int  89 137 78 197 189 166 118 103 115 126 ...
##  $ bl.press  : int  66 40 50 70 60 72 84 30 70 88 ...
##  $ tr.thick  : int  23 35 32 45 23 19 47 38 30 41 ...
##  $ serum.ins : int  94 168 88 543 846 175 230 83 96 235 ...
##  $ bmi       : num  28.1 43.1 31 30.5 30.1 25.8 45.8 43.3 34.6 39.3
...
##  $ diab      : num  0.167 2.288 0.248 0.158 0.398 ...
##  $ age       : int  21 33 26 53 59 51 31 33 32 27 ...
##  $ class     : int  0 1 1 1 1 1 1 0 1 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:376] 1 2 3 6 8 10 11 12
13 16 ...
##   ..- attr(*, "names")= chr [1:376] "1" "2" "3" "6" ...

var(newdf)

##              t.pregnant      plasma     bl.press    tr.thick    seru
m.ins
## t.pregnant 10.313247038   19.652043   8.56198131   3.1479331    30.1
44188
## plasma     19.652043426  952.387781  80.99446735  64.5376716  2131.6
62900
## bl.press    8.561981314   80.994467 156.15230440  30.5631557   146.2
95162
## tr.thick    3.147933086   64.537672  30.56315570 110.5951707   227.7
10489
## serum.ins  30.144188110 2131.662900 146.29516154 227.7104885 14123.3
47226
## bmi        -0.572057519   45.439613  26.73217809  49.0997064   189.0
81594
## diab        0.008390234    1.494605  -0.06895125   0.5831391     5.5
80072
## age        22.263309672  108.179694  38.24591576  17.9966922   263.1
63618
```

```
## class       0.388407537    7.502349   1.13497573   1.2687901    16.8
86711
##                    bmi         diab         age       class
## t.pregnant  -0.5720575  0.008390234  22.2633097  0.38840754
## plasma      45.4396132  1.494605381 108.1796936  7.50234877
## bl.press    26.7321781 -0.068951250  38.2459158  1.13497573
## tr.thick    49.0997064  0.583139086  17.9966922  1.26879012
## serum.ins  189.0815935  5.580071585 263.1636176 16.88671121
## bmi         49.3879939  0.385491683   5.0047823  0.89486142
## diab         0.3854917  0.119361988   0.2996635  0.03409215
## age          5.0047823  0.299663513 104.0558419  1.68689650
## class        0.8948614  0.034092150   1.6868965  0.22221932

# Check the variances (diagonal elements) of the variables in the dataf
rame newdf

apply(newdf,2, var)

##   t.pregnant       plasma     bl.press     tr.thick    serum.ins
bmi
## 1.031325e+01 9.523878e+02 1.561523e+02 1.105952e+02 1.412335e+04 4.9
38799e+01
##         diab          age        class
## 1.193620e-01 1.040558e+02 2.222193e-01

# Same as above

n.class <- newdf$class
table(n.class)

## n.class
##   0   1
## 262 130

n.class[n.class==0] <- 2
# Transformation needed for the plots later on

table(n.class)

## n.class
##   1   2
## 130 262

# 1 corresponds to the women that will develop diabetes, 2 refers to th
e women that won't develop diabetes!


library(reshape2)
library(ggplot2)
df1 <- melt(newdf[,-9])
```

```
## No id variables; using all as measure variables

ggplot(data = df1, aes(x = value)) +
  stat_density() +
  facet_wrap(~variable, scales = "free")
```




```
# Small multiple chart

df2 <- log(newdf+1)
new_df2 <- melt(df2[,-9])

## No id variables; using all as measure variables

ggplot(data = new_df2, aes(x = value)) +
  stat_density() +
  facet_wrap(~variable, scales = "free")
```



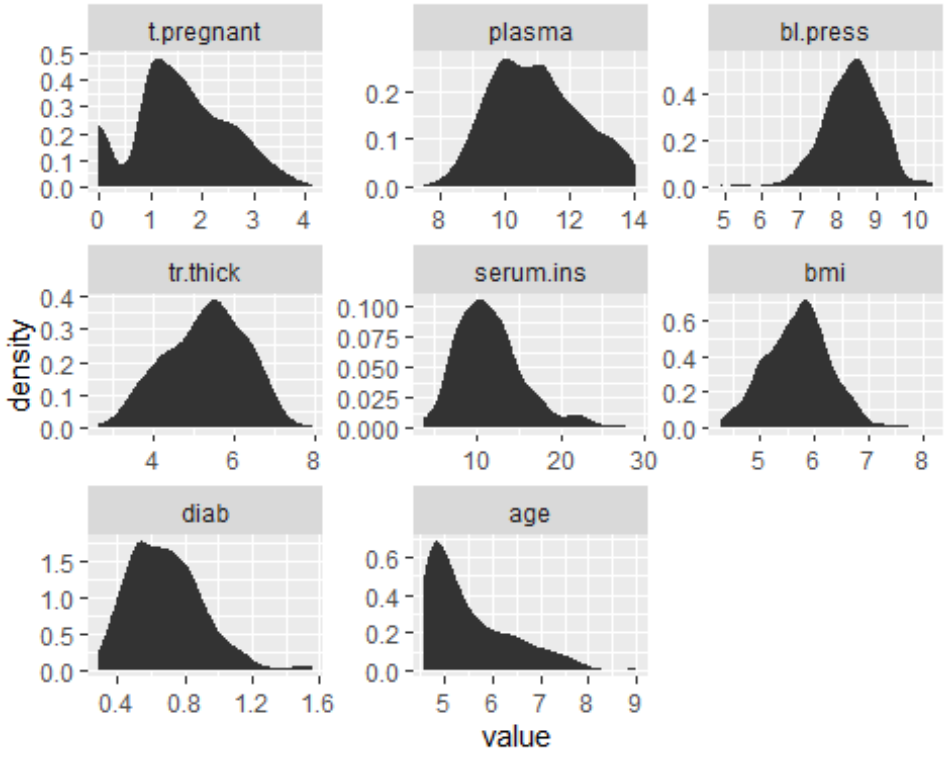

```
# Small multiple chart of the transformed data



df3 <- sqrt(newdf[,-9])
head(df3)
```

```
##    t.pregnant    plasma bl.press tr.thick serum.ins      bmi      di
ab      age
## 4    1.000000  9.433981 8.124038 4.795832  9.695360 5.300943 0.40865
63 4.582576
## 5    0.000000 11.704700 6.324555 5.916080 12.961481 6.565059 1.51261
36 5.744563
## 7    1.732051  8.831761 7.071068 5.656854  9.380832 5.567764 0.49799
60 5.099020
## 9    1.414214 14.035669 8.366600 6.708204 23.302360 5.522681 0.39749
21 7.280110
## 14   1.000000 13.747727 7.745967 4.795832 29.086079 5.486347 0.63087
24 7.681146
## 15   2.236068 12.884099 8.485281 4.358899 13.228757 5.079370 0.76615
93 7.141428
```

```
new_df3 <- data.frame(df3,n.class)
head(new_df3)
```

```
##    t.pregnant    plasma bl.press tr.thick serum.ins      bmi      di
ab      age
```

```
## 4    1.000000  9.433981 8.124038 4.795832  9.695360 5.300943 0.40865
63 4.582576
## 5    0.000000 11.704700 6.324555 5.916080 12.961481 6.565059 1.51261
36 5.744563
## 7    1.732051  8.831761 7.071068 5.656854  9.380832 5.567764 0.49799
60 5.099020
## 9    1.414214 14.035669 8.366600 6.708204 23.302360 5.522681 0.39749
21 7.280110
## 14   1.000000 13.747727 7.745967 4.795832 29.086079 5.486347 0.63087
24 7.681146
## 15   2.236068 12.884099 8.485281 4.358899 13.228757 5.079370 0.76615
93 7.141428
##    n.class
## 4        2
## 5        1
## 7        1
## 9        1
## 14       1
## 15       1

n_df3 <- melt(new_df3[,-9])

## No id variables; using all as measure variables

ggplot(data = n_df3, aes(x = value)) +
  stat_density() +
  facet_wrap(~variable, scales = "free")
```



```
# Small multiple chart of the transformed data


# We observe that when using the square root transformation the variabl
e's distributions are slightly improved as
# regards normality hence we stick with this transformation! (the trans
formed dataset we will work with is called new_df3)

cor(new_df3[,-9])
```

```
##              t.pregnant    plasma    bl.press  tr.thick serum.ins
bmi
## t.pregnant  1.000000000 0.1611331  0.14584736 0.0427007 0.0877161 -0
.10170030
## plasma      0.161133135 1.0000000  0.21380087 0.2059071 0.6240914  0
.21641944
## bl.press    0.145847362 0.2138009  1.00000000 0.2176316 0.1181296  0
.28160708
## tr.thick    0.042700702 0.2059071  0.21763164 1.0000000 0.2018678  0
.67161365
## serum.ins   0.087716104 0.6240914  0.11812961 0.2018678 1.0000000  0
.26074828
## bmi        -0.101700301 0.2164194  0.28160708 0.6716136 0.2607483  1
.00000000
## diab        0.001185833 0.1233397 -0.02419759 0.1243165 0.1228579  0
.14156045
## age         0.637056715 0.3481860  0.30397119 0.1874931 0.2402571  0
.09239483
##                    diab        age
## t.pregnant  0.001185833 0.63705672
## plasma      0.123339709 0.34818604
## bl.press   -0.024197590 0.30397119
## tr.thick    0.124316497 0.18749313
## serum.ins   0.122857896 0.24025708
## bmi         0.141560449 0.09239483
## diab        1.000000000 0.09617298
## age         0.096172982 1.00000000
```

```
# Correlations of the transformed dataset
```

Implementation of Hierarchical clustering using different linkages with proximity measure the euclidian distance

```
d1 <- dist(scale(new_df3[,-9]), method = "euclidian")
# Scaled dissimilarity matrix using as distance the euclidian one
head(d1)
```

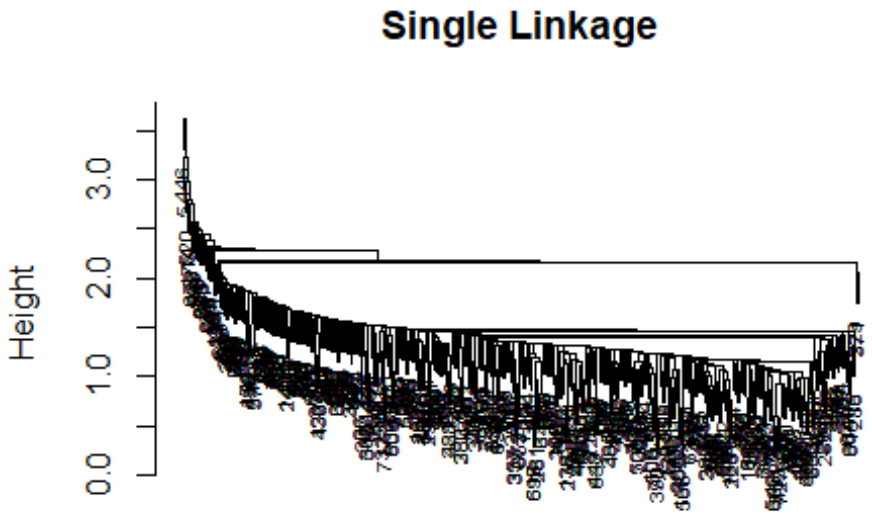
```
## [1] 6.584076 2.044797 5.991233 6.785725 4.581059 4.426926
```

```
library(stats)
library(cluster)
```

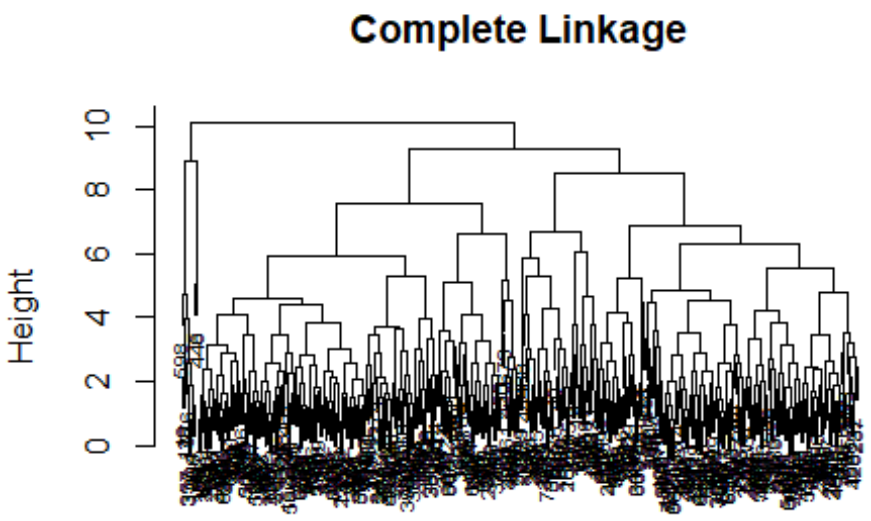
```
# Hierarchical method, different linkage methods
fit1.s <- hclust(d1, method = "single")
fit1.c <- hclust(d1, method = "complete")
fit1.a <- hclust(d1, method = "average")
fit1.d <- hclust(d1,method="ward.D")
fit1.d2 <- hclust(d1,method="ward.D2")
fit1.m <- hclust(d1,method="mcquitty")
fit1.me <- hclust(d1,method="median")
fit1.centr <- hclust(d1,method="centroid")
```

Let us plot the resulting dendrograms

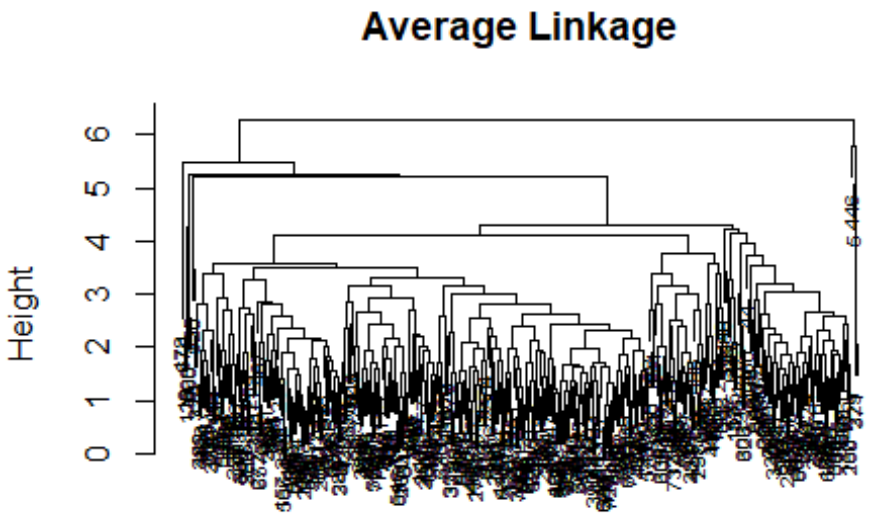
```
plot(fit1.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



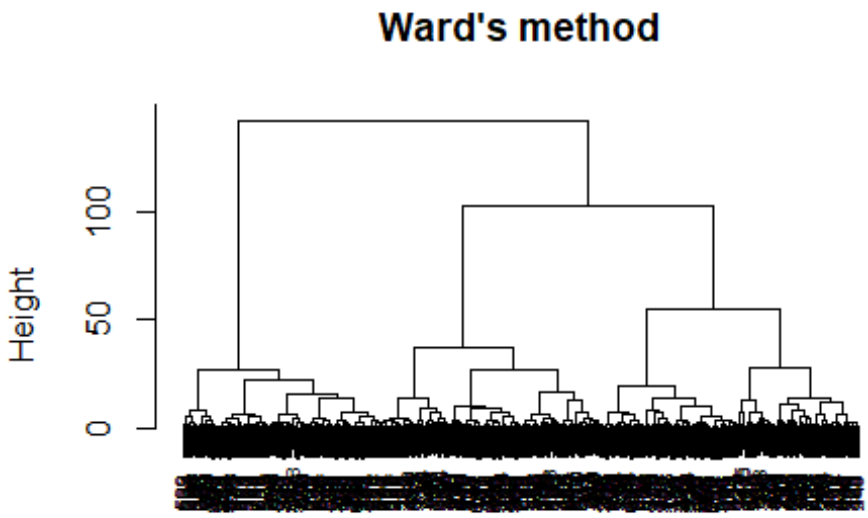

```
# Default label: the increasing number (row number)
# Let us observe the first split! (conservative way to tell how many gr
oups we have!)
# We observe that the single linkage (or nearest neighbor has fallen in
the trap of the chain effect! It's known
# that nearest linkage is prone to the chain effect, yet it's useful to
identify potential outliers!
plot(fit1.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```




```r
# It indicates that there are probably 3 groups, row 446 has to be chec
ked, potential outlier!
plot(fit1.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



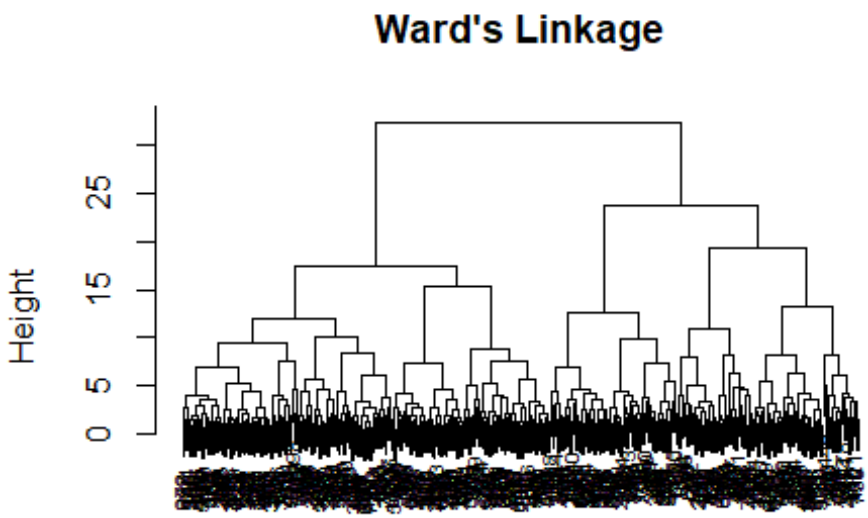
```
# Average linkage seems to be a bit more reasonable in general, it indi
cates that there are 3 groups as well
plot(fit1.d, main = "Ward's method", sub = "", xlab = "", cex=.6)
```



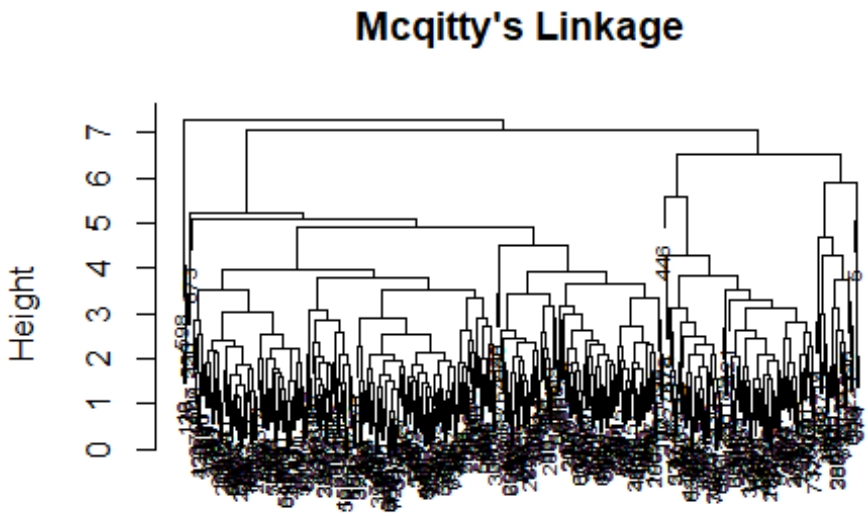

```
# The plot indicates that there are 3 groups (conservative approach)
```

More dentrogram plots

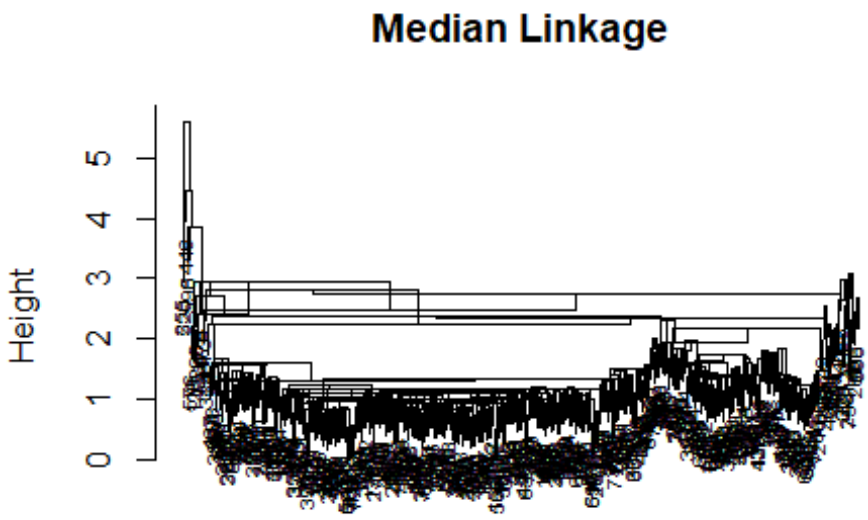
```
# More linkage methods to check their output in order to gain some insi
ghts as regards the clusters
plot(fit1.d2, main = "Ward's Linkage", sub = "", xlab = "", cex=.6)
```



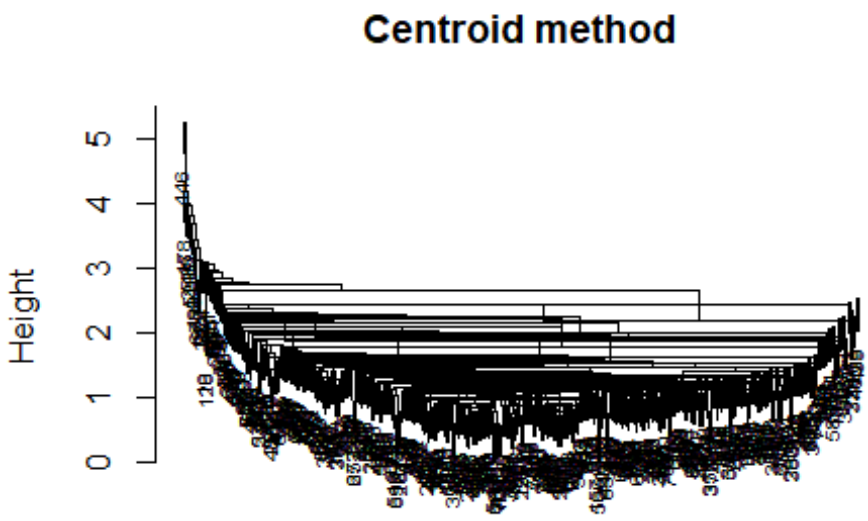

```r
# This method indicates that there are 4 groups!
plot(fit1.m, main = "Mcqitty's Linkage", sub = "", xlab = "", cex=.6)
```



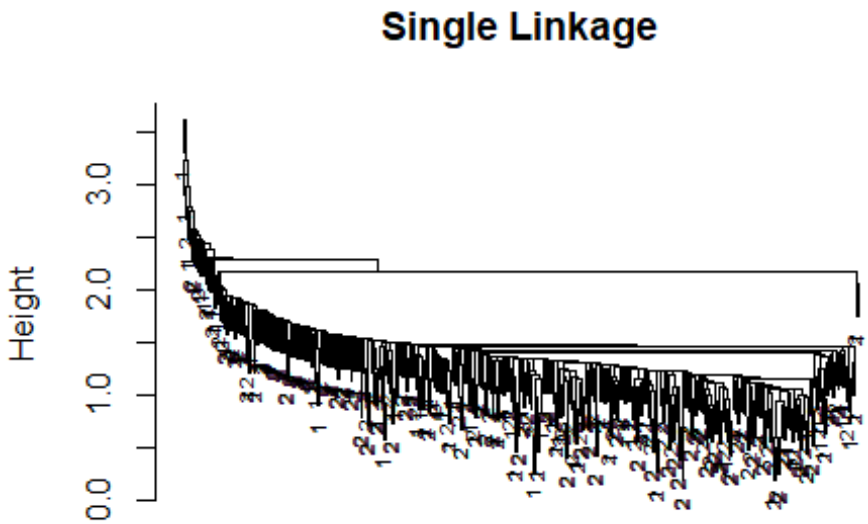
```r
# Default label: the increasing number (row number)
plot(fit1.me, main = "Median Linkage", sub = "", xlab = "", cex=.6)
```



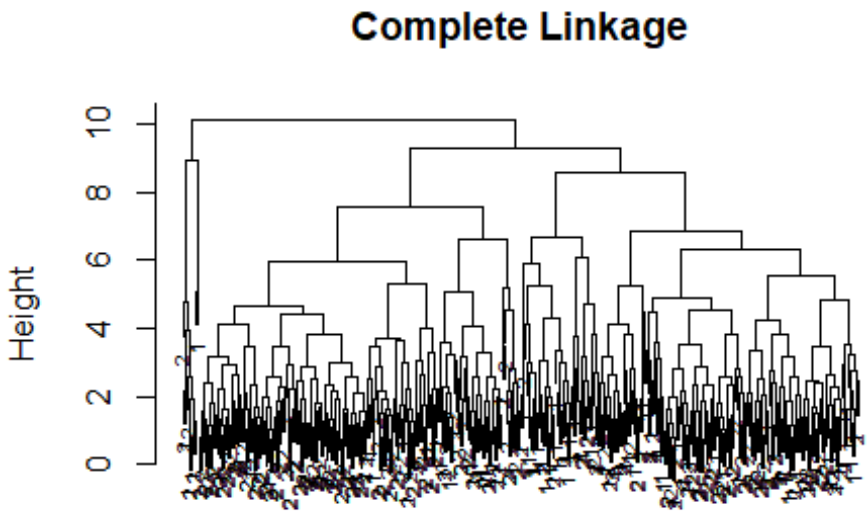

```r
plot(fit1.centr, main = "Centroid method", sub = "", xlab = "", cex=.6)
```



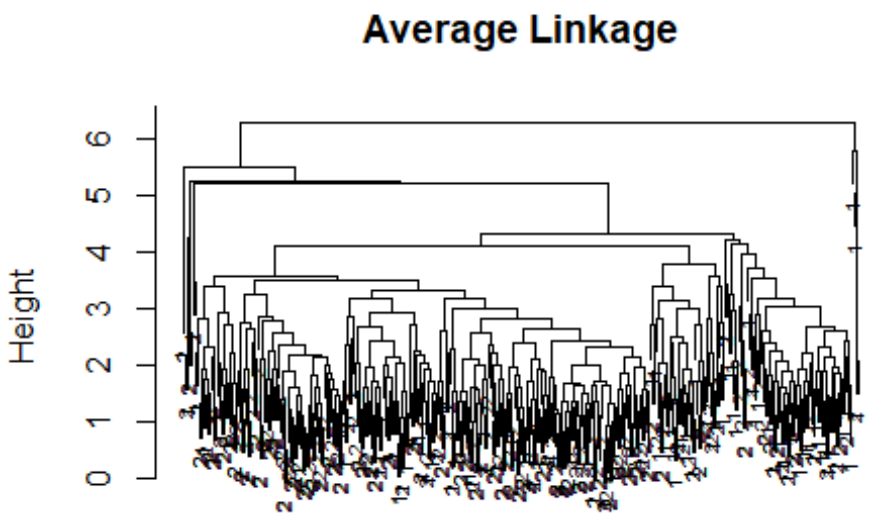

```
plot(fit1.s, main = "Single Linkage", sub = "", xlab = "", labels=new_d
f3[,9], cex=.6)
```



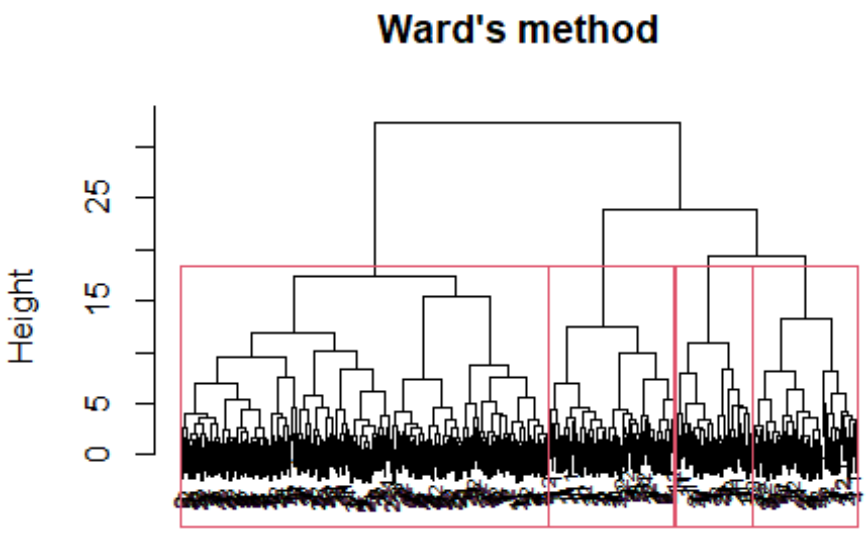
```r
# Label is the 9th column, which refers to class (diabetes-non diabetes that have corresponding values 1,0 respectively)
# Is there a relationship between the medical data and the class?
# It's not compulsory to exist. We seek to cluster the medical data. Provided we manage to cluster them, that should
# mean that medical data have somehow alike composition, yet that's not of top importance
plot(fit1.c, main = "Complete Linkage", sub = "", xlab = "", labels=new_df3[,9], cex=.6)
```




```r
plot(fit1.a, main = "Average Linkage", sub = "", xlab = "", labels=new_df3[,9], cex=.6)
```



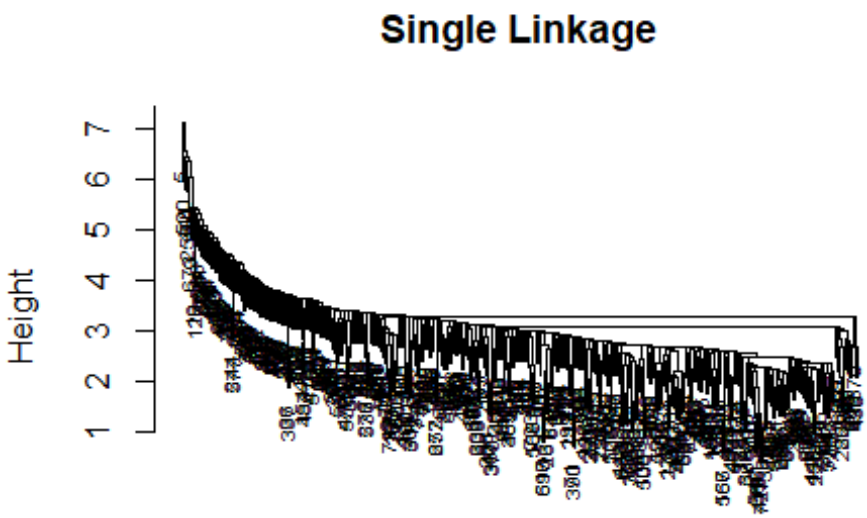

```
plot(fit1.d2, main = "Ward's method", sub = "", xlab = "", labels=new_d
f3[,9], cex=.6)
# We should be aware of the fact that Ward's linkage may impose equal s
ize to clusters as in our case
rect.hclust(fit1.d2, 4)
```



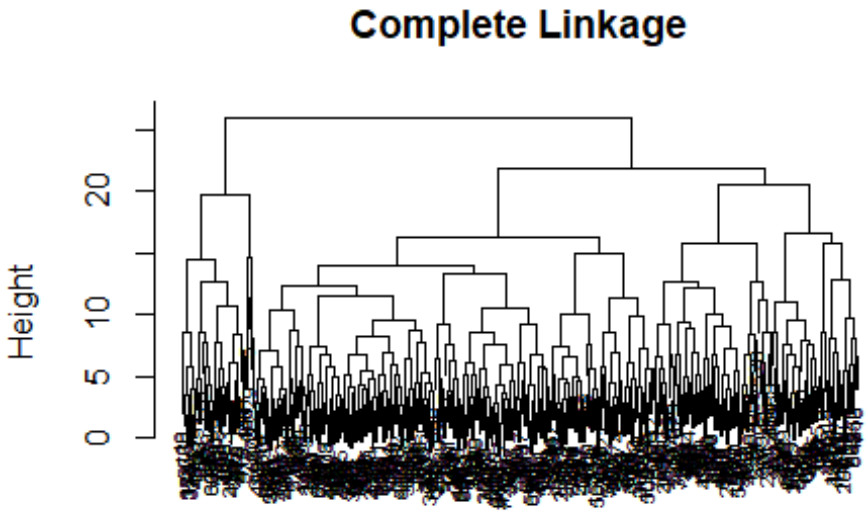

```
# Split a dendrogram in 4 rectangulars
```

Implementation of Hierarchical clustering using different linkages with proximity measure the manhattan distance

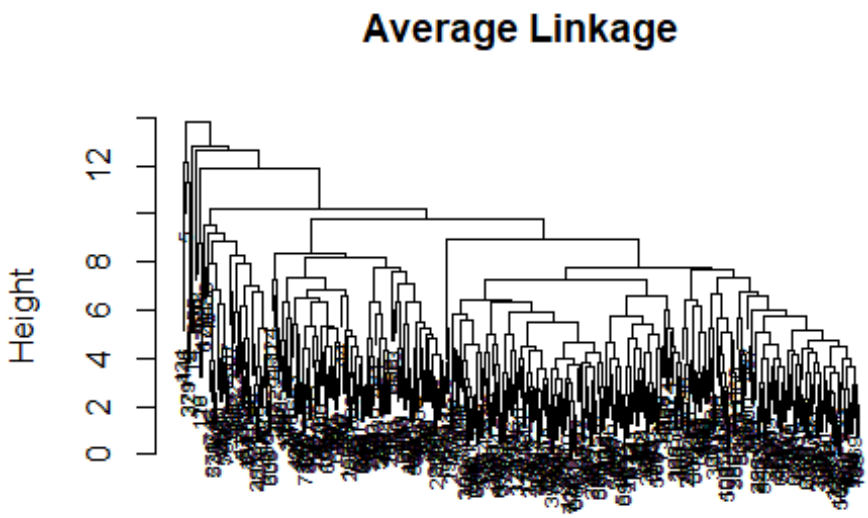
```
d2 <- dist(scale(new_df3[,-9]), method = "manhattan")
# Scaled dissimilarity matrix, now with manhattan distance
# Does there change something????

# Hierarchical method, different methods
fit2.s <- hclust(d2, method = "single")
fit2.c <- hclust(d2, method = "complete")
fit2.a <- hclust(d2, method = "average")
fit2.d <-hclust(d2, method="ward.D")
fit2.d2 <-hclust(d2, method="ward.D2")

# Plot the resulting dendrograms
plot(fit2.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



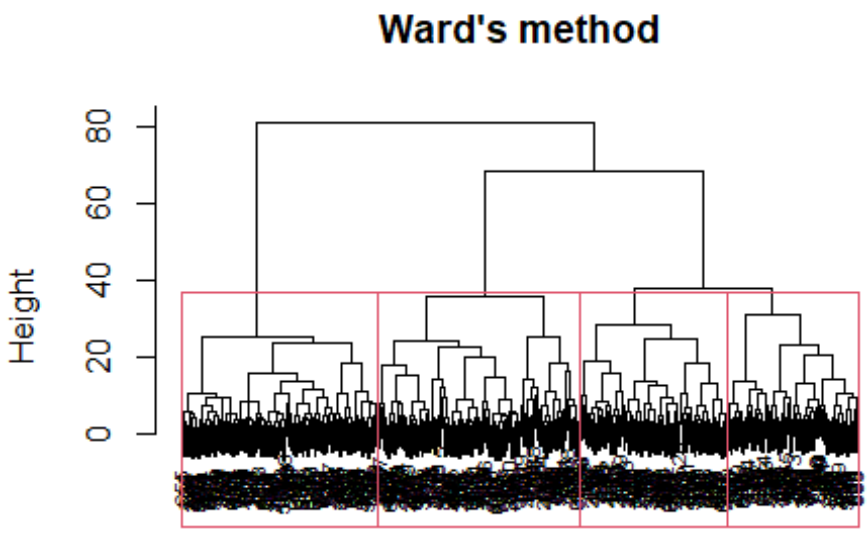

```r
# Default label: the increasing number
plot(fit2.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```



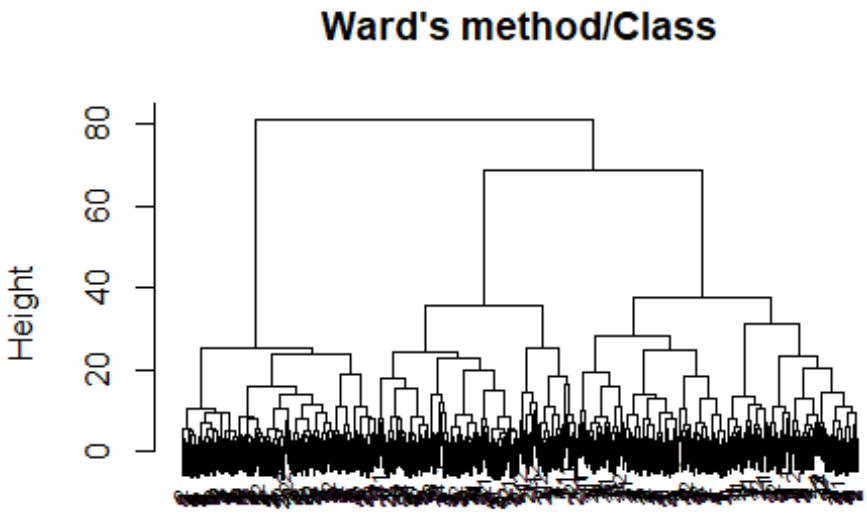
```r
# The plot indicates that there are 4 groups
plot(fit2.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



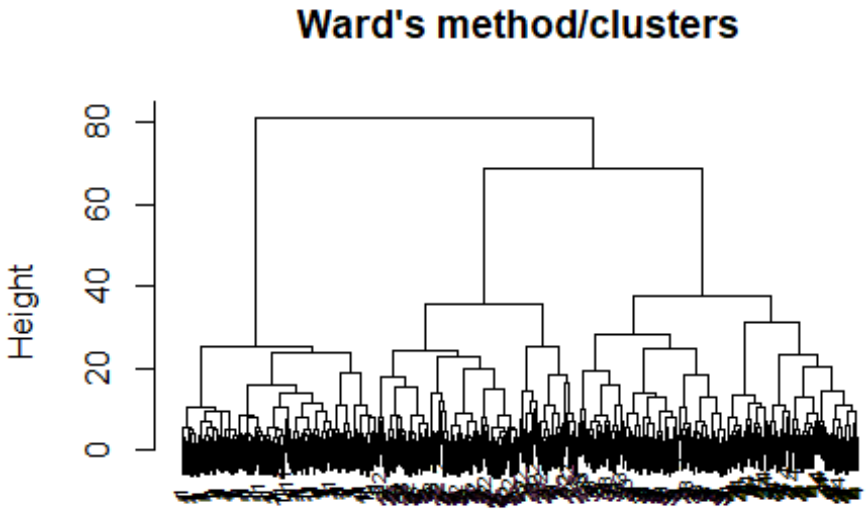

```r
plot(fit2.d2, main = "Ward's method", sub = "", xlab = "", cex=.6)
# The plot indicates that there are 3 groups
rect.hclust(fit2.d2, 4)
```




```
# Split a dendrogram in 4 rectangulars


cluster.d2 <- cutree(fit2.d2,4)
# Split the tree in 4 branches!
cluster.d2[1:10]

##  4  5  7  9 14 15 17 19 20 21
##  1  2  1  3  3  4  2  2  2  2

# Gives the cluster membership for the first 10 observations of the dat
aset.
plot(fit2.d2, main = "Ward's method/Class", sub = "", xlab = "",labels=
new_df3[,9], cex=.6)
```




```
# We observe  3 clusters probably
plot(fit2.d2, main = "Ward's method/clusters", sub = "", xlab = "", lab
els=cluster.d2, cex=.6)
```



```r
# We observe 3 clusters probably

table(cluster.d2, new_df3[,9])
```

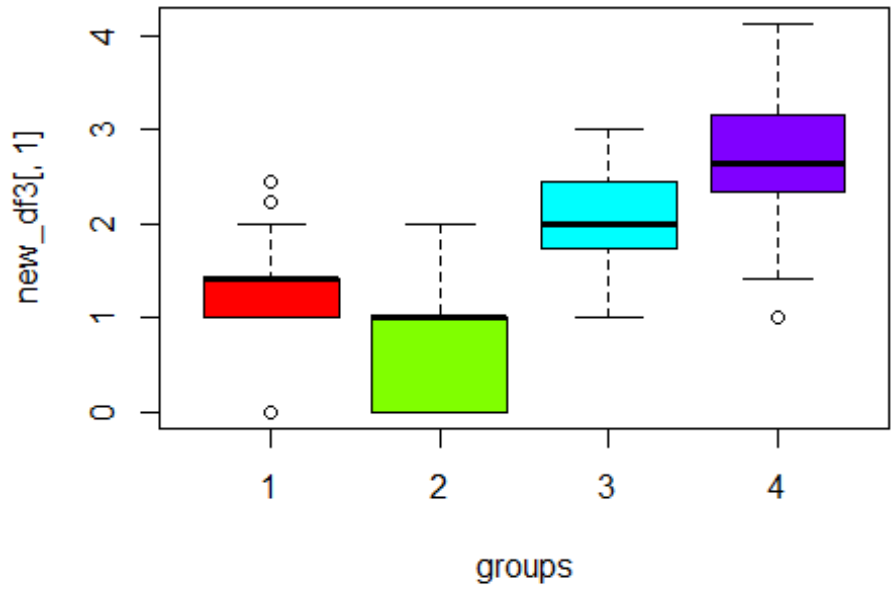
```
##
## cluster.d2   1   2
##          1   4 109
##          2  41  76
##          3  46  40
##          4  39  37
```

```r
# What is the agreement of the diabetes-non diabetes?? Reminder: 1 refe
rs to diabetes development, 2 refers to the women that won't develop di
abetes
# Non-diabetes women have been split in 4 groups. 109 in the first clus
ter, 76 in the second cluster, 40 in the third
# cluster, 37 in the fourth cluster.
# Women with diabetes have been split in 4 groups. 4 in the first clust
er, 41 in the second cluster, 46 in the third
# cluster and 39 in the fourth cluster.
# Contingency table!
# Possible outliers cause the problem of groups merging one another (or
the number of groups is not the best!)

cbind(new_df3[1:10,],cluster.d2[1:10])
```

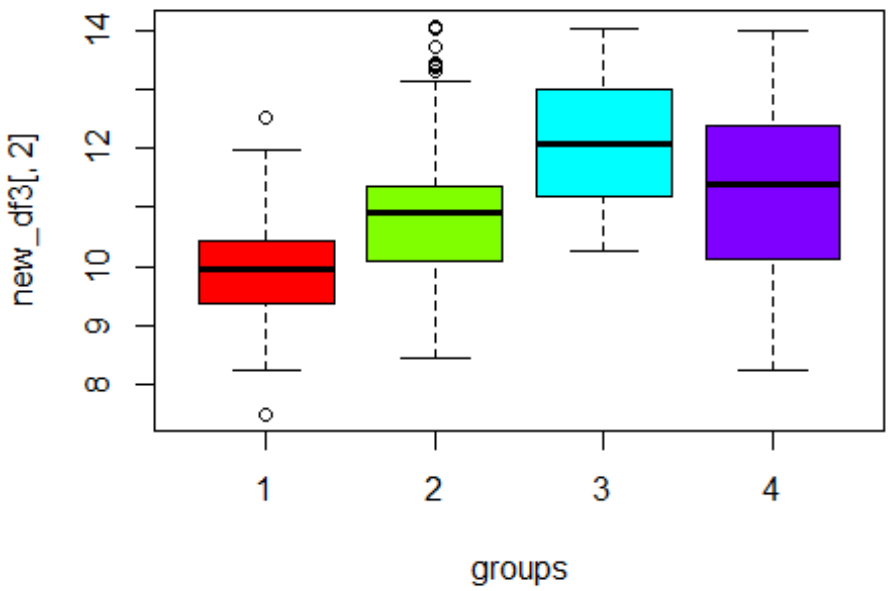
```
##    t.pregnant    plasma bl.press tr.thick serum.ins      bmi      di
ab      age
## 4    1.000000  9.433981 8.124038 4.795832  9.695360 5.300943 0.40865
63 4.582576
## 5    0.000000 11.704700 6.324555 5.916080 12.961481 6.565059 1.51261
36 5.744563
## 7    1.732051  8.831761 7.071068 5.656854  9.380832 5.567764 0.49799
60 5.099020
## 9    1.414214 14.035669 8.366600 6.708204 23.302360 5.522681 0.39749
21 7.280110
## 14   1.000000 13.747727 7.745967 4.795832 29.086079 5.486347 0.63087
24 7.681146
## 15   2.236068 12.884099 8.485281 4.358899 13.228757 5.079370 0.76615
93 7.141428
## 17   0.000000 10.862780 9.165151 6.855655 15.165751 6.767570 0.74229
37 5.567764
## 19   1.000000 10.148892 5.477226 6.164414  9.110434 6.580274 0.42778
50 5.744563
## 20   1.000000 10.723805 8.366600 5.477226  9.797959 5.882176 0.72732
39 5.656854
## 21   1.732051 11.224972 9.380832 6.403124 15.329710 6.268971 0.83904
71 5.196152
##    n.class cluster.d2[1:10]
## 4        2                1
```

```
## 5         1                 2
## 7         1                 1
## 9         1                 3
## 14        1                 3
## 15        1                 4
## 17        1                 2
## 19        2                 2
## 20        1                 2
## 21        2                 2
```

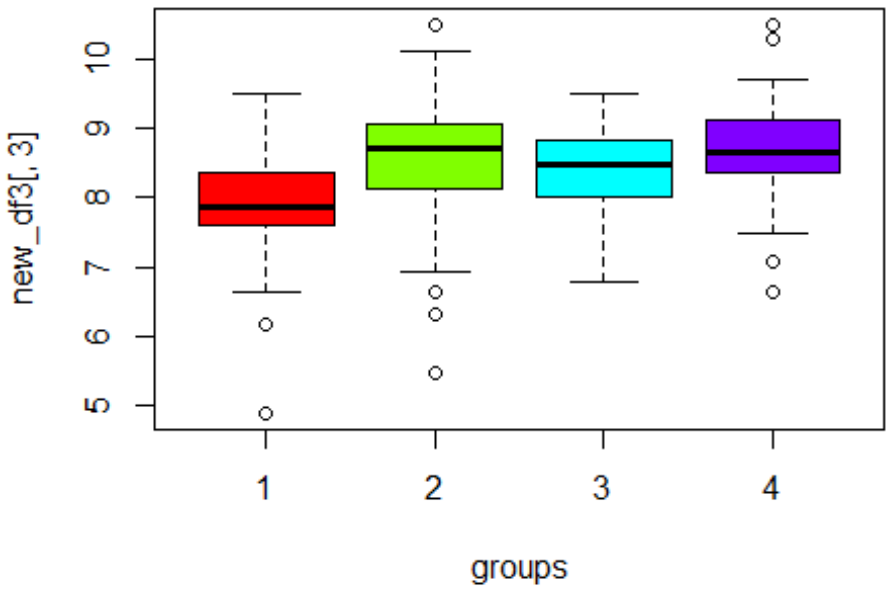
```
# Each observation , in which cluster it belongs

# Let us check if the clusters we created make sense
groups <- as.factor(cluster.d2)

# Let us observe their common characteristics with respect to the varia
bles 1:8
boxplot(new_df3[,1]~groups, col=rainbow(4))
```



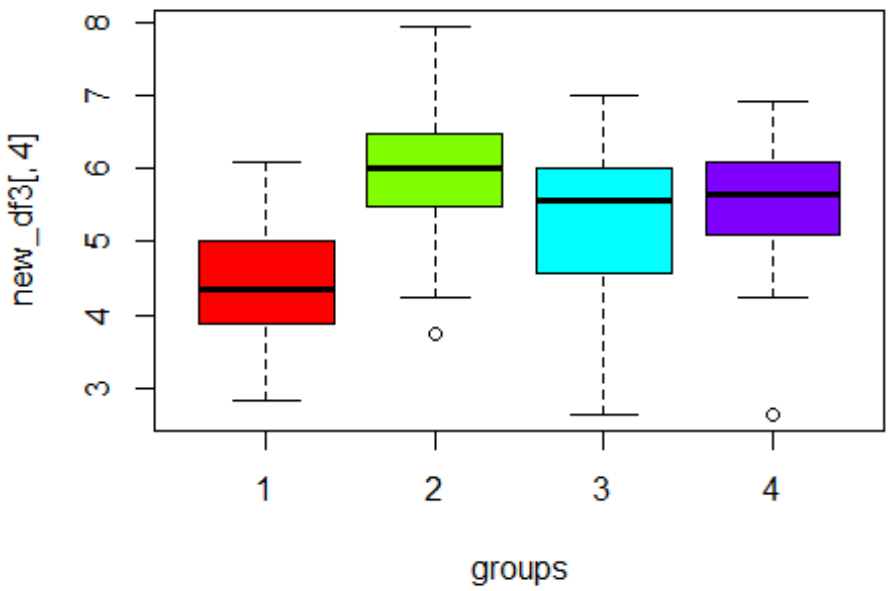

```
boxplot(new_df3[,2]~groups, col=rainbow(4))
```



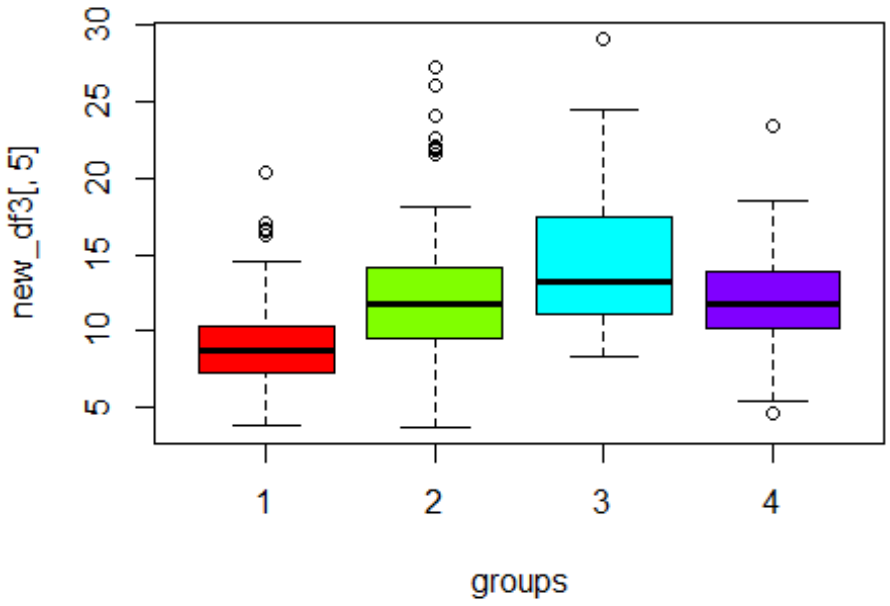

```
boxplot(new_df3[,3]~groups, col=rainbow(4))
```



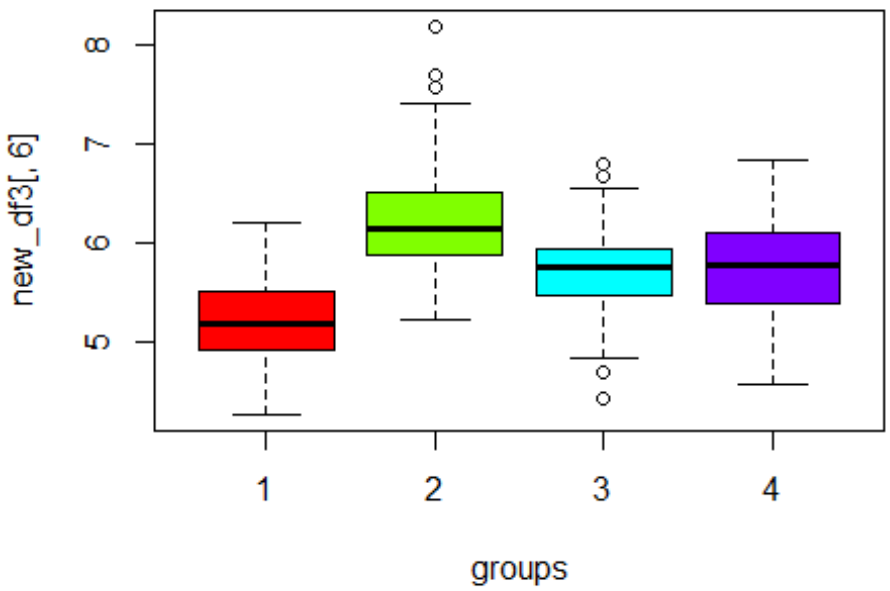

```
boxplot(new_df3[,4]~groups, col=rainbow(4))
```



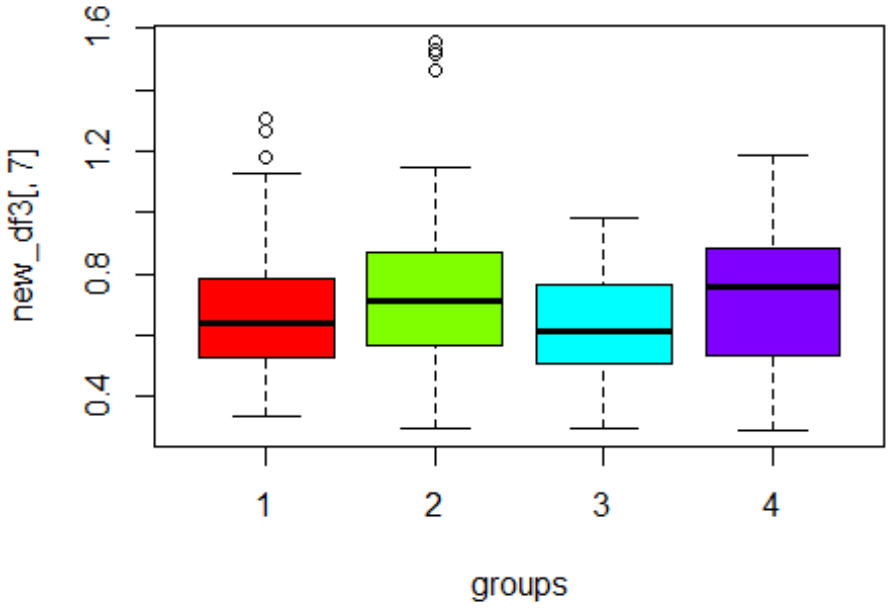

```
boxplot(new_df3[,5]~groups, col=rainbow(4))
```



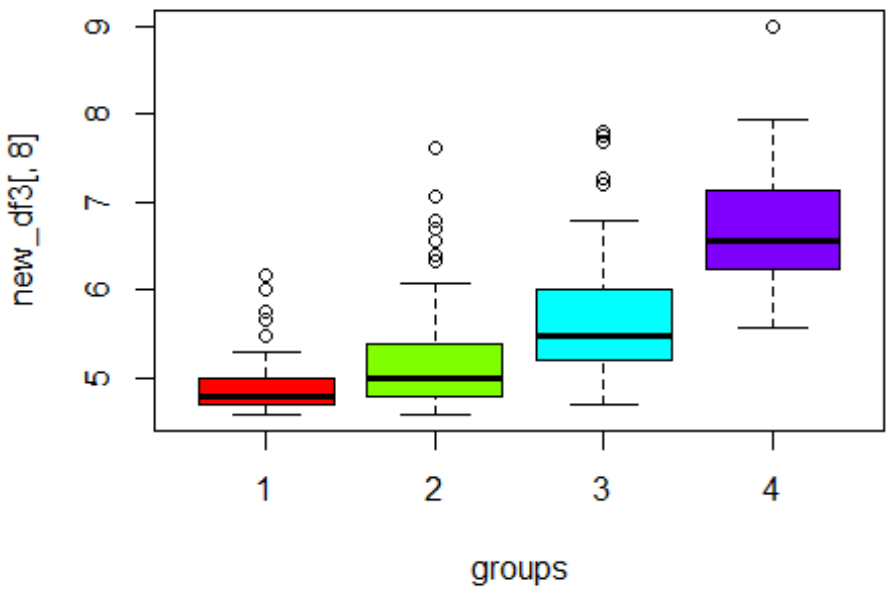

```
boxplot(new_df3[,6]~groups, col=rainbow(4))
```




```
boxplot(new_df3[,7]~groups, col=rainbow(4))
```




```
boxplot(new_df3[,8]~groups, col=rainbow(4))
```



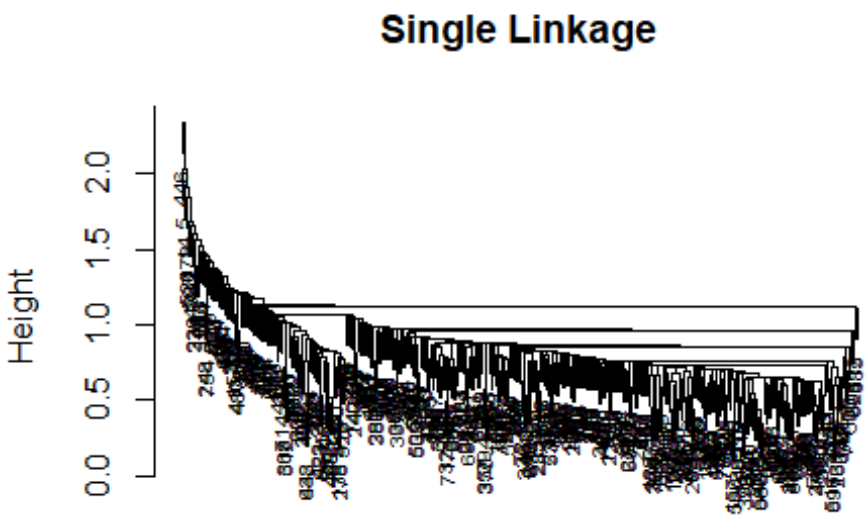

```
# On basis of the original variables we observe that the groups do not
differ!
# Should we have decided about the groups, by creating the multiple box
plots we can gather the range of each group for
# each variable (so we can tell their common characteristics!)
```

Implementation of Hierarchical clustering using different linkages with proximity measure the maximum distance

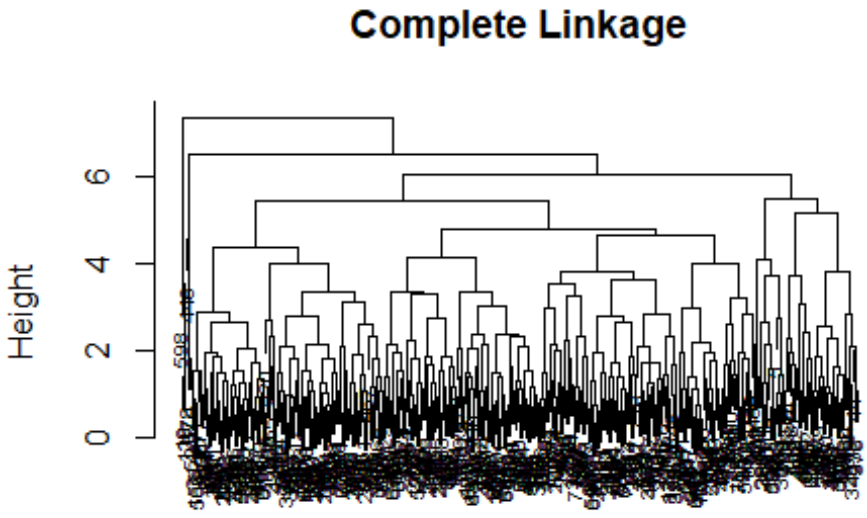
```
d3 <- dist(scale(new_df3[,-9]), method = "maximum")
# Scaled dissimilarity matrix, now with manhattan distance
# Does there change something????

# Hierarchical method, different methods
fit3.s <- hclust(d3, method = "single")
fit3.c <- hclust(d3, method = "complete")
fit3.a <- hclust(d3, method = "average")
fit3.d <- hclust(d3, method="ward.D")
fit3.d2 <- hclust(d3, method="ward.D2")
```

Let us plot the resulting dendrograms

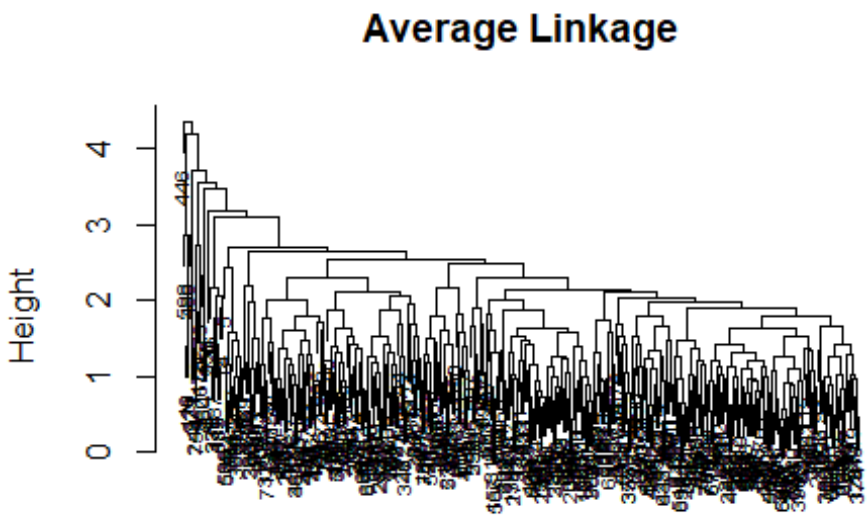
```
# Plot the resulting dendrograms
plot(fit3.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



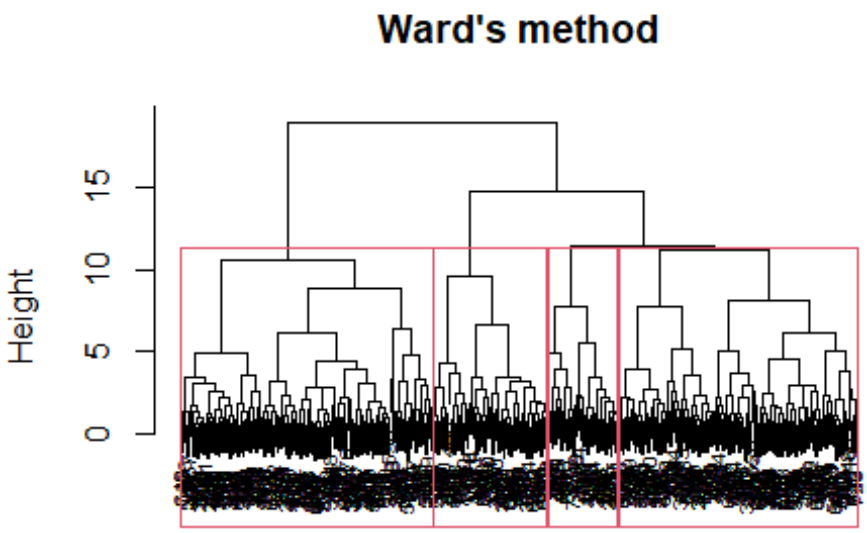

```r
# Default label: the increasing number
plot(fit3.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```



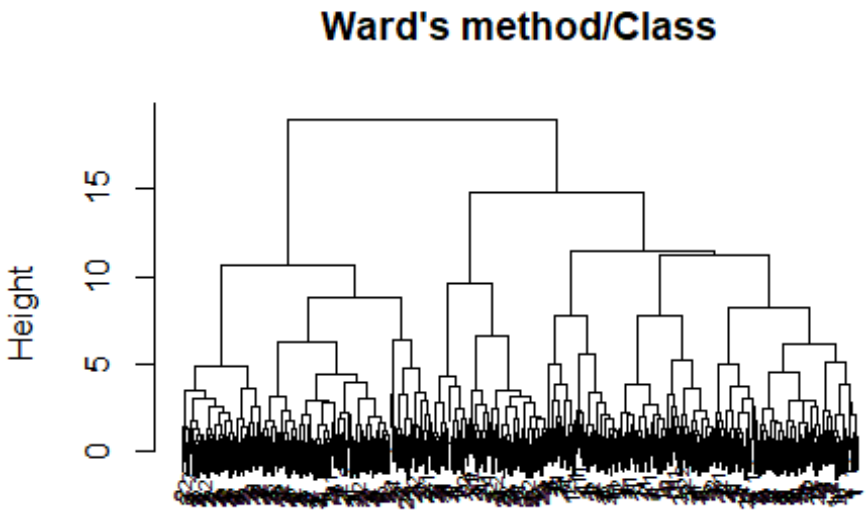
```r
# The plot indicates that there are probably 5 groups (2 singletons!)
plot(fit3.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



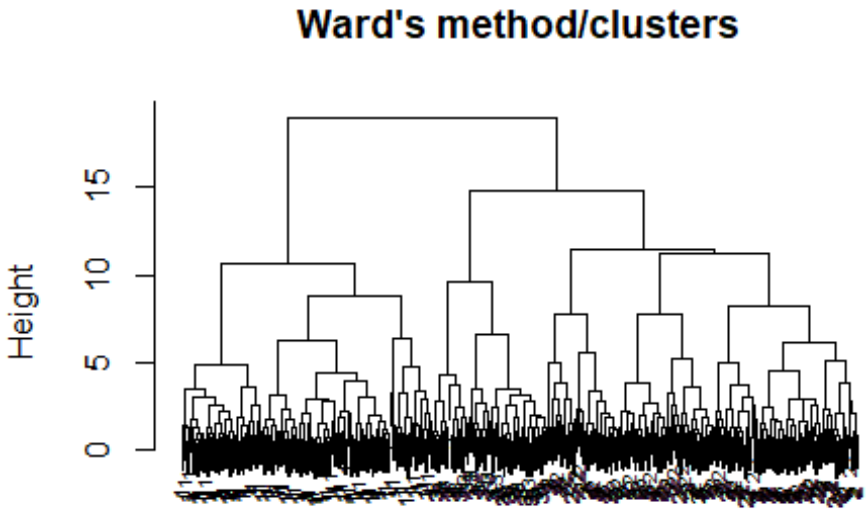

```r
plot(fit3.d2, main = "Ward's method", sub = "", xlab = "", cex=.6)
# The plot indicates that there are 3 groups
rect.hclust(fit3.d2, 4)
```




```
# Split a dendrogram in 4 rectangulars

cluster.d3 <- cutree(fit3.d2,3)
# Split the tree in 3 branches now
cluster.d3[1:10]

##  4  5  7  9 14 15 17 19 20 21
##  1  1  1  2  2  2  3  1  1  3

# Gives the cluster membership for the first 10 observations of the dat
aset.
plot(fit3.d2, main = "Ward's method/Class", sub = "", xlab = "",labels=
new_df3[,9], cex=.6)
```

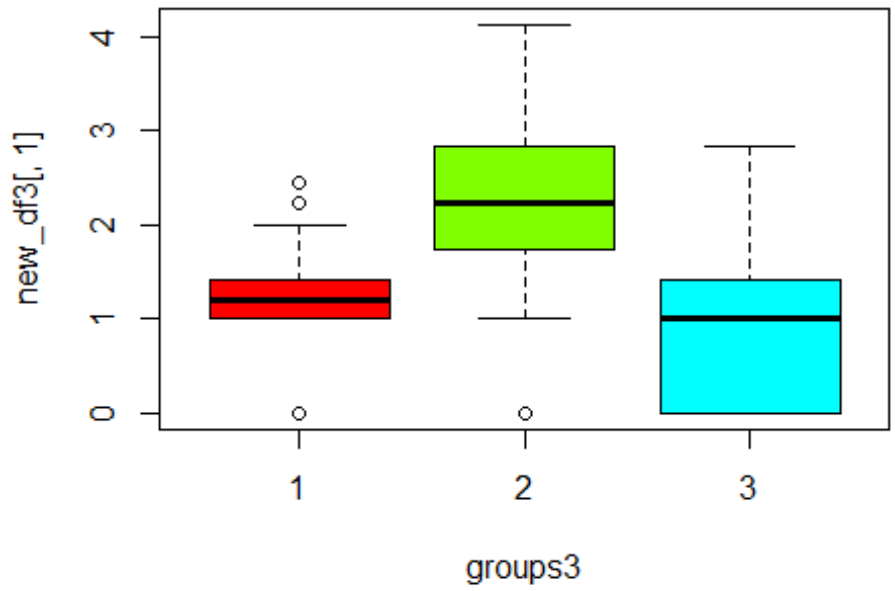



```
# We observe 3 clusters probably (conservative approach)
plot(fit3.d2, main = "Ward's method/clusters", sub = "", xlab = "", lab
els=cluster.d3, cex=.6)
```

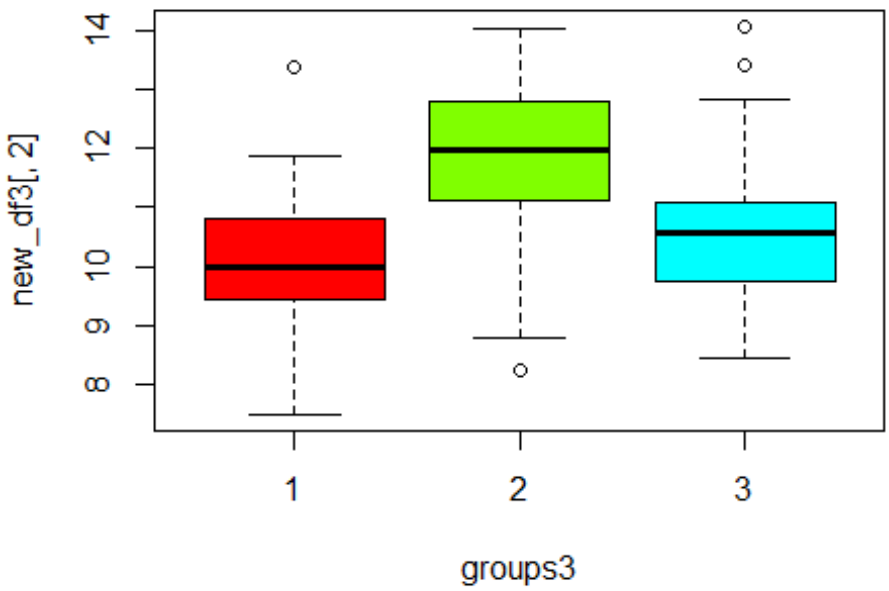


```r
# We observe 3 clusters probably
table(cluster.d3, new_df3[,9])

##
## cluster.d3   1   2
##          1  19 127
##          2  90  90
##          3  21  45

# What is the agreement of the diabetes-non diabetes??
# Non-diabetes women have been split in 3 groups. 127 in the first clus
ter, 90 in the second cluster, 45 in the third cluster
# Women with diabetes have been split in 3 groups. 19 in the first clus
ter, 90 in the second cluster, 21 in the third cluster
# Contingency table!
# Possible outliers cause the problem of groups merging one another (or
there's a better partition for the number of groups)


cbind(new_df3[1:10,],cluster.d3[1:10])

##    t.pregnant    plasma bl.press tr.thick serum.ins      bmi      di
ab      age
## 4    1.000000  9.433981 8.124038 4.795832  9.695360 5.300943 0.40865
63 4.582576
## 5    0.000000 11.704700 6.324555 5.916080 12.961481 6.565059 1.51261
36 5.744563
## 7    1.732051  8.831761 7.071068 5.656854  9.380832 5.567764 0.49799
60 5.099020
## 9    1.414214 14.035669 8.366600 6.708204 23.302360 5.522681 0.39749
21 7.280110
## 14   1.000000 13.747727 7.745967 4.795832 29.086079 5.486347 0.63087
24 7.681146
## 15   2.236068 12.884099 8.485281 4.358899 13.228757 5.079370 0.76615
93 7.141428
## 17   0.000000 10.862780 9.165151 6.855655 15.165751 6.767570 0.74229
37 5.567764
## 19   1.000000 10.148892 5.477226 6.164414  9.110434 6.580274 0.42778
50 5.744563
## 20   1.000000 10.723805 8.366600 5.477226  9.797959 5.882176 0.72732
39 5.656854
## 21   1.732051 11.224972 9.380832 6.403124 15.329710 6.268971 0.83904
71 5.196152
##    n.class cluster.d3[1:10]
## 4        2                1
## 5        1                1
## 7        1                1
## 9        1                2
## 14       1                2
## 15       1                2
```

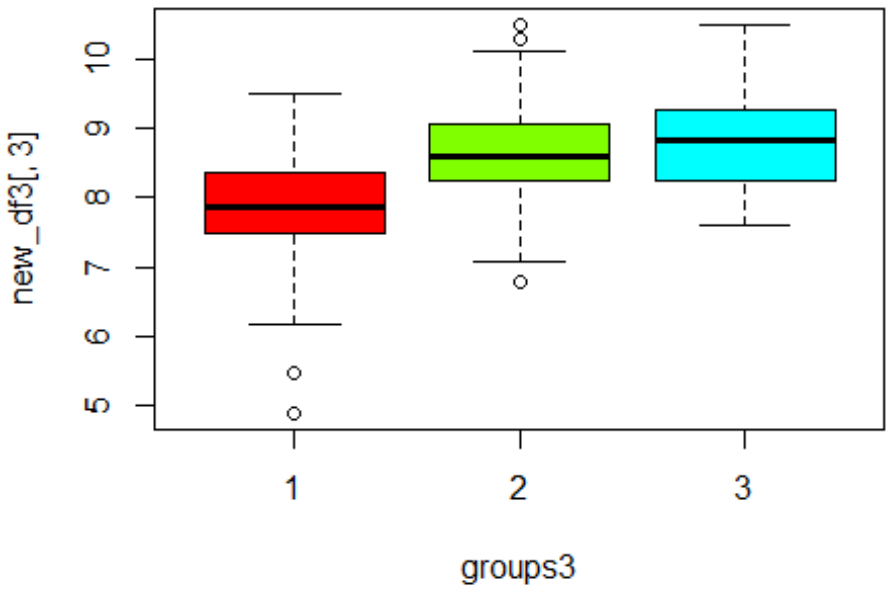
```
## 17       1                3
## 19       2                1
## 20       1                1
## 21       2                3

# Each observation , in which cluster it belongs

# Let us check if the clusters we created make sense
groups3 <- as.factor(cluster.d3)

# Let us observe their common characteristics with respect to the varia
bles 1:8
boxplot(new_df3[,1]~groups3, col=rainbow(4))
```



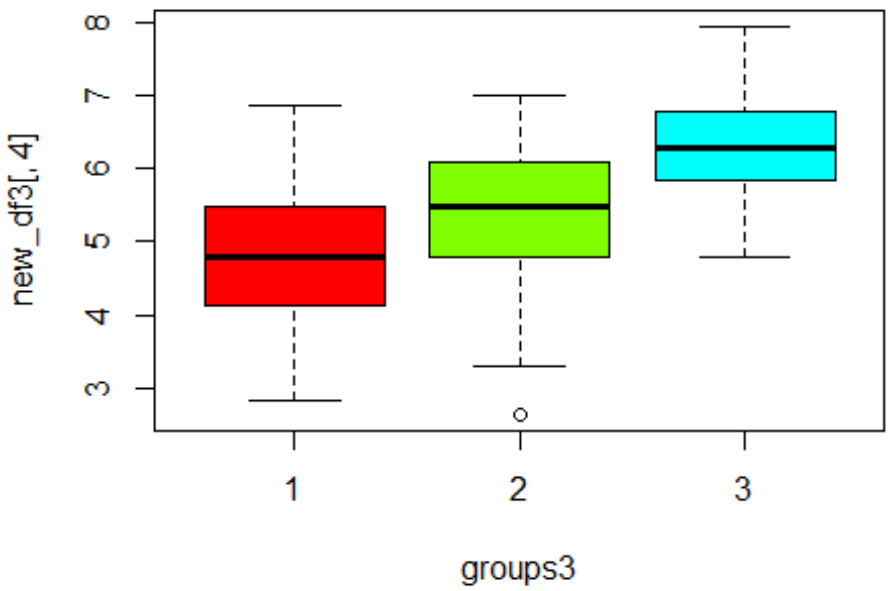

```
boxplot(new_df3[,2]~groups3, col=rainbow(4))
```



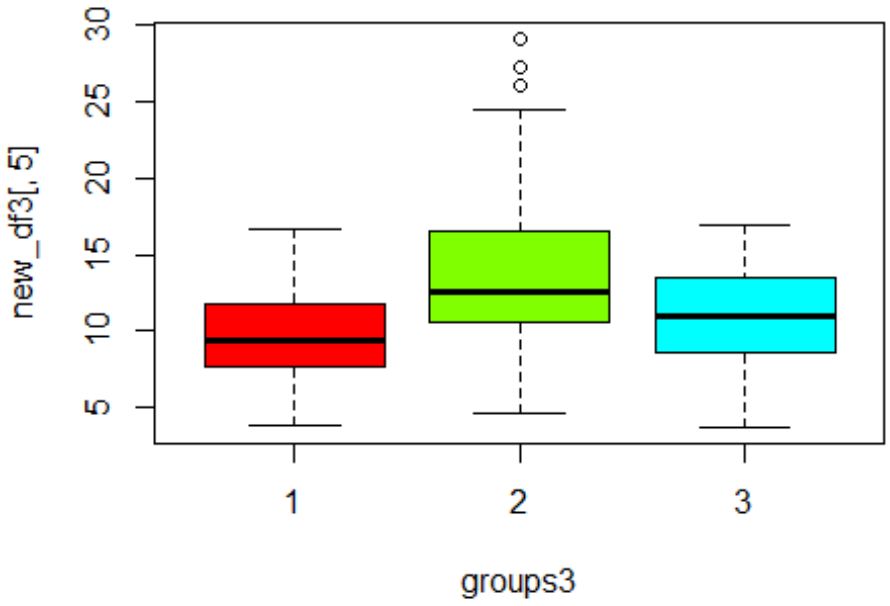

```
boxplot(new_df3[,3]~groups3, col=rainbow(4))
```



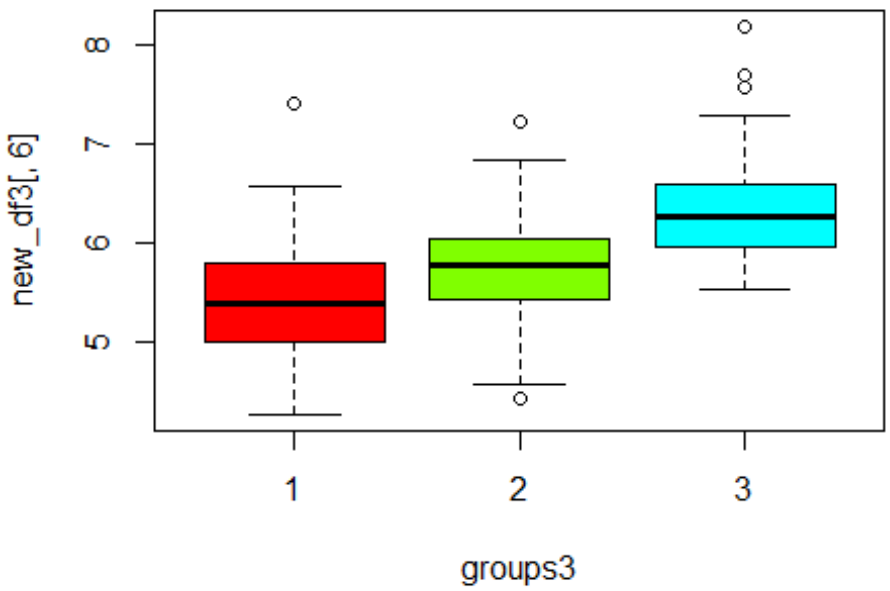

```
boxplot(new_df3[,4]~groups3, col=rainbow(4))
```



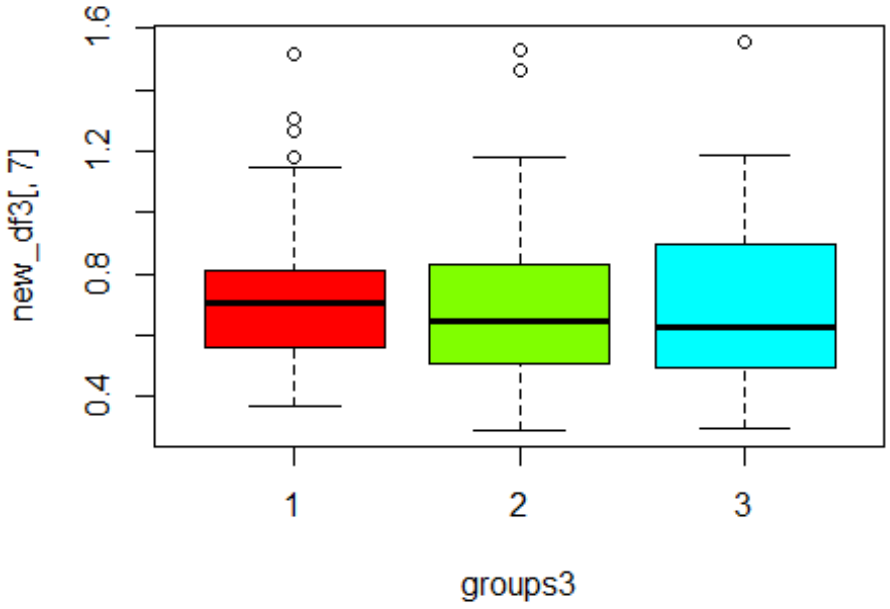

```
boxplot(new_df3[,5]~groups3, col=rainbow(4))
```



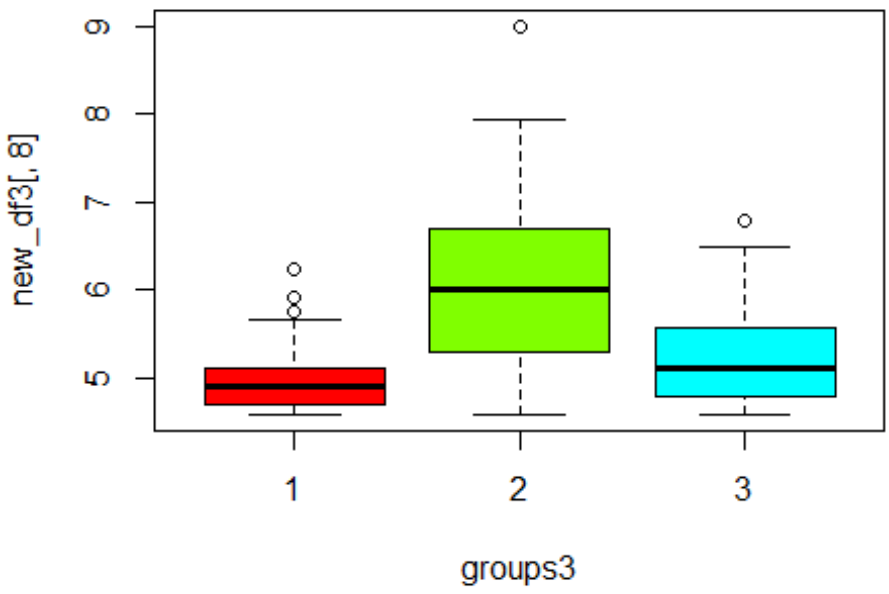

```
boxplot(new_df3[,6]~groups3, col=rainbow(4))
```




```
boxplot(new_df3[,7]~groups3, col=rainbow(4))
```




```
boxplot(new_df3[,8]~groups3, col=rainbow(4))
```

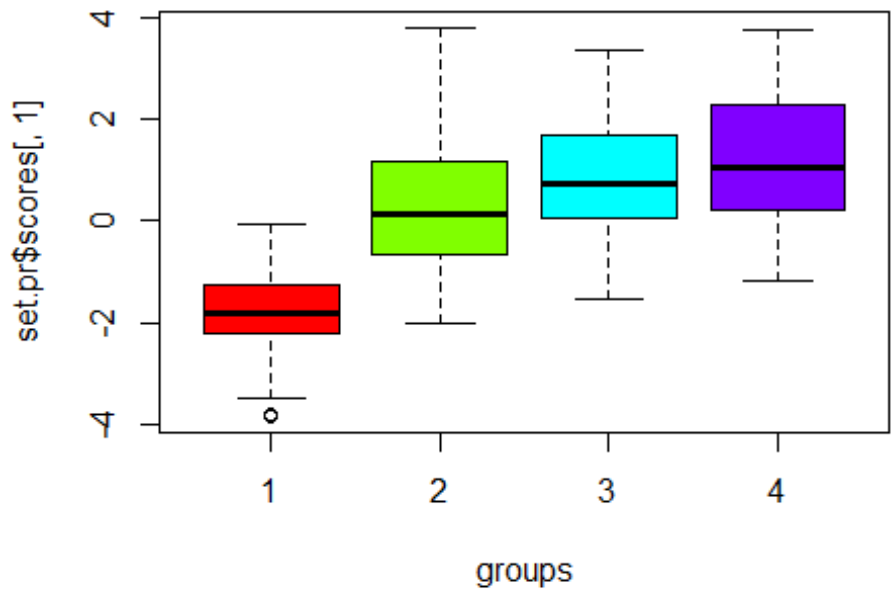



```
# Overall, on basis of the original variables we observe that the group
s do not differ!
```

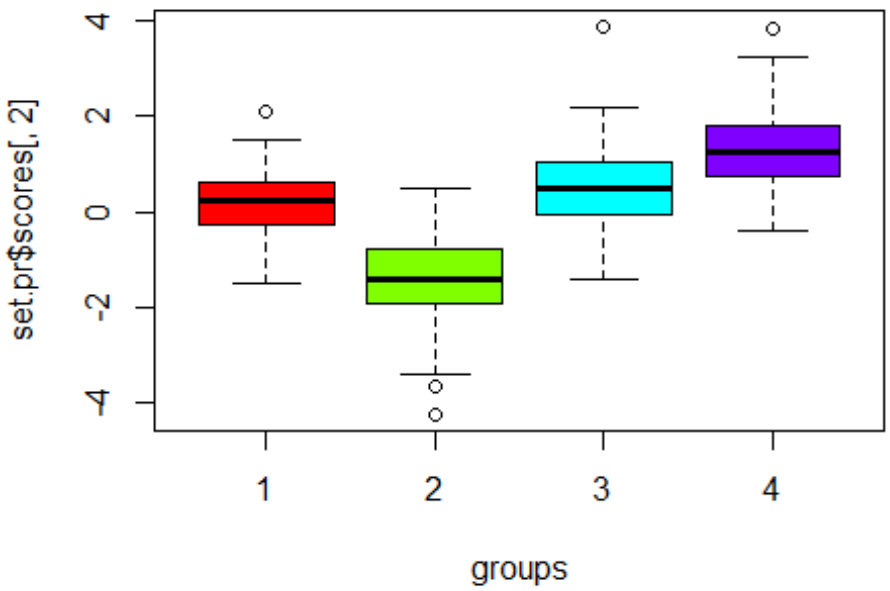
Principal Component Analysis

```
# Creating PCs for two reasons:
# 1) Visualization of the groups on the PCs
# 2) We might prefer to work with the PCs instead of the initial data!
set.pr <- princomp(scale(new_df3[, -9]))
# Exclude the labels, which are in the 9th column

summary(set.pr)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     C
omp.5
## Standard deviation     1.5944967 1.2534822 1.0758981 0.9879298 0.867
53179
## Proportion of Variance 0.3186153 0.1969045 0.1450647 0.1223127 0.094
31703
## Cumulative Proportion  0.3186153 0.5155198 0.6605844 0.7828971 0.877
21417
##                            Comp.6     Comp.7     Comp.8
## Standard deviation     0.60955302 0.56283988 0.53984927
## Proportion of Variance 0.04656314 0.03969987 0.03652282
## Cumulative Proportion  0.92377731 0.96347718 1.00000000
```

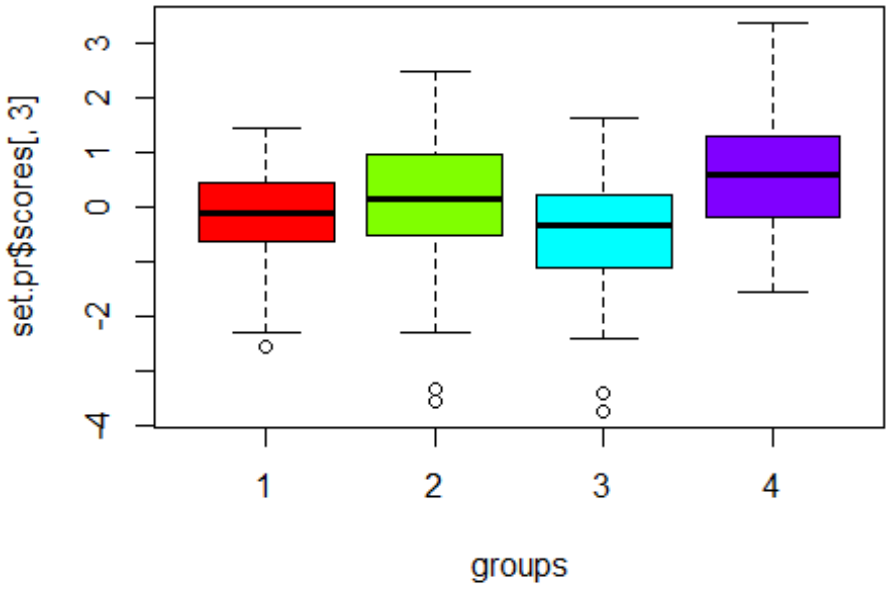
```r
# The 1st PC explains the 31.9% of the total variability
# The first 2 PCs explain the 51.5% of the total variability
# The first 3 PCs explain the 66% of the total variability
# The first 4 PCs explain the 78.3% of the total variability
# We observe that the standard deviatons with value more than 1, are th
e first 3 ones (the fourth is valued at 0.987
# almost 1,hence we have to take that into consideration!)

names(set.pr)

## [1] "sdev"     "loadings" "center"   "scale"    "n.obs"    "scores"
"call"

boxplot(set.pr$scores[,1]~groups, col=rainbow(4))
```



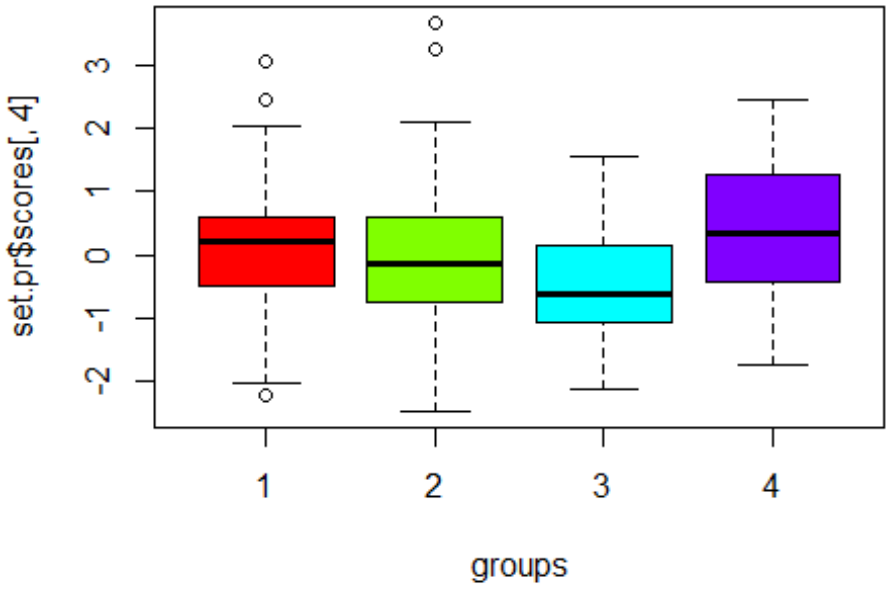

```r
# We observe that group 1 differs from the rest but groups 2,3,4 do not
differ between themselves
boxplot(set.pr$scores[,2]~groups, col=rainbow(4))
```



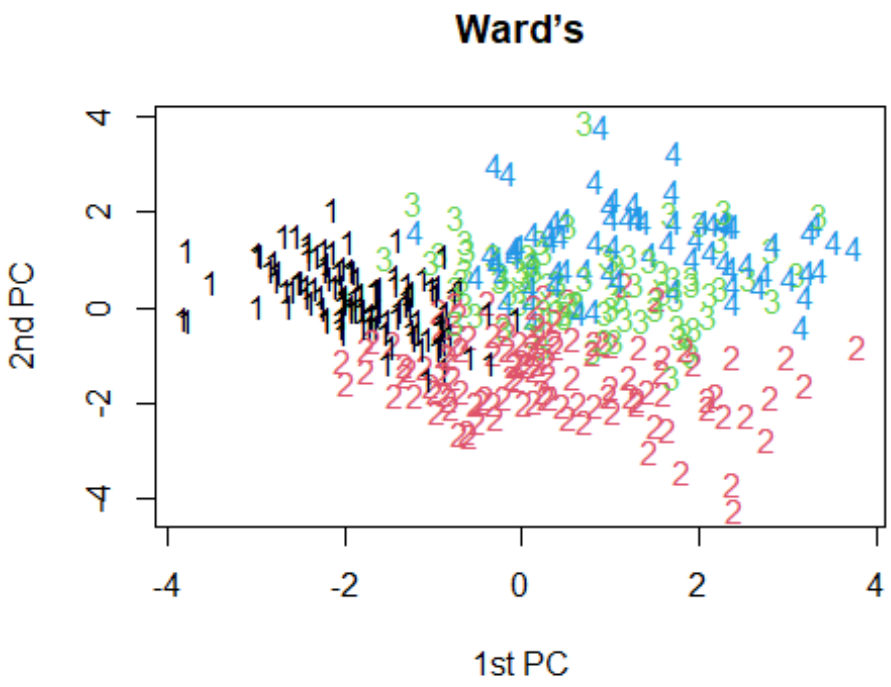

```
# We observe that groups 2,3,4 differ between themselves
boxplot(set.pr$scores[,3]~groups, col=rainbow(4))
```



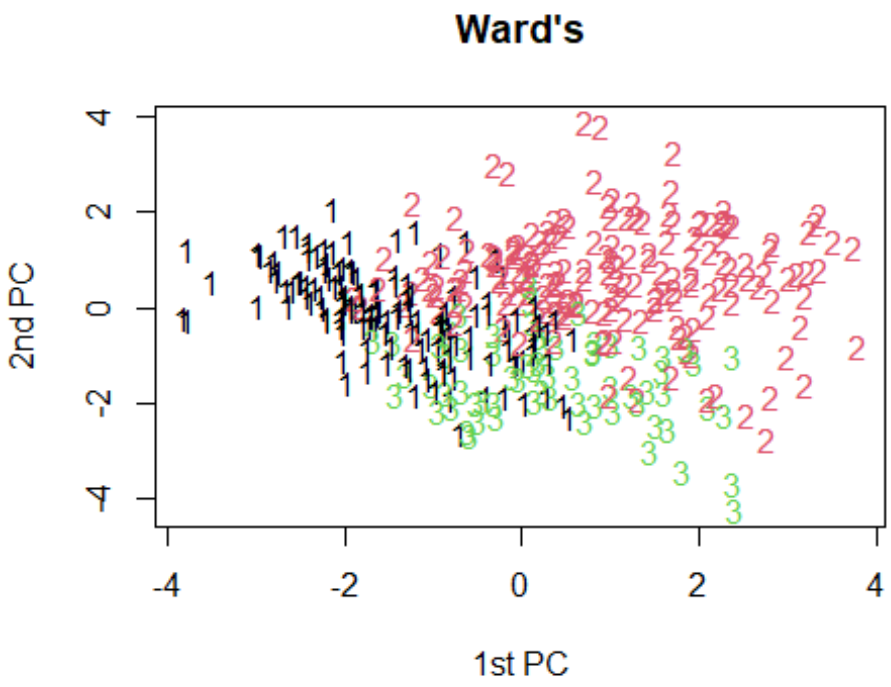
```r
# We observe that groups do not differ in the case of the 3rd PC
boxplot(set.pr$scores[,4]~groups, col=rainbow(4))
```



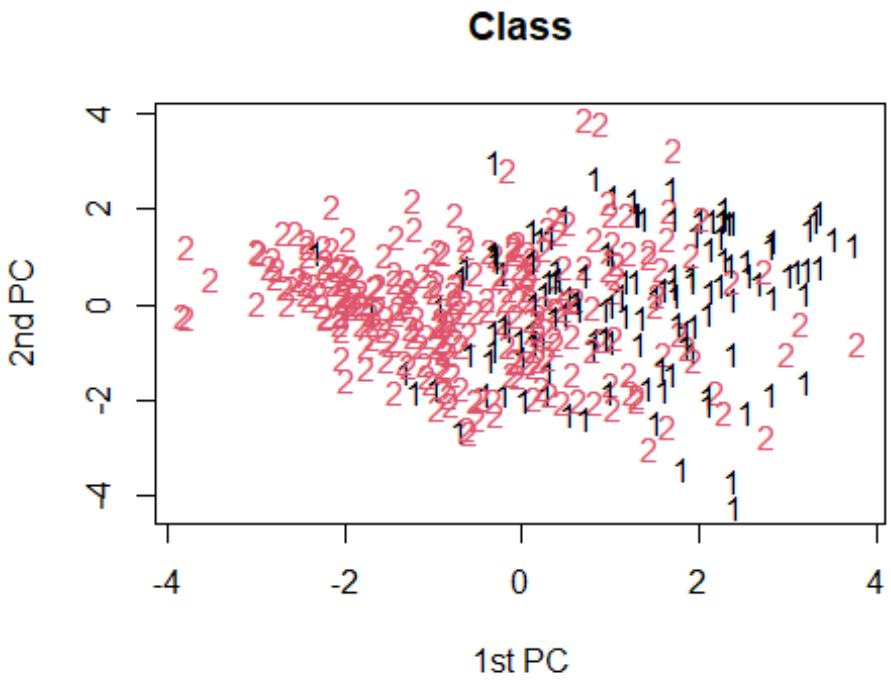

```r
# We observe that groups do not differ in the case of the 4th PC
# We are okay with the first 2 PCs!
# On the negative point of view, the first 2 PCs explain the 51.5% of t
he total variability



# Visualize the result of a clustering technique using PCs
# Initially we couldn't plot the data as the dimension was 8
# Now that's feasible due to using PCs!
plot(set.pr$scores[,1], set.pr$scores[,2], main ="Ward's", xlab="1st PC
", ylab="2nd PC",  type='n')
text(set.pr$scores[,1], set.pr$scores[,2], xlab="1st PC", ylab="2nd PC"
, label=cluster.d2, col=cluster.d2)
```



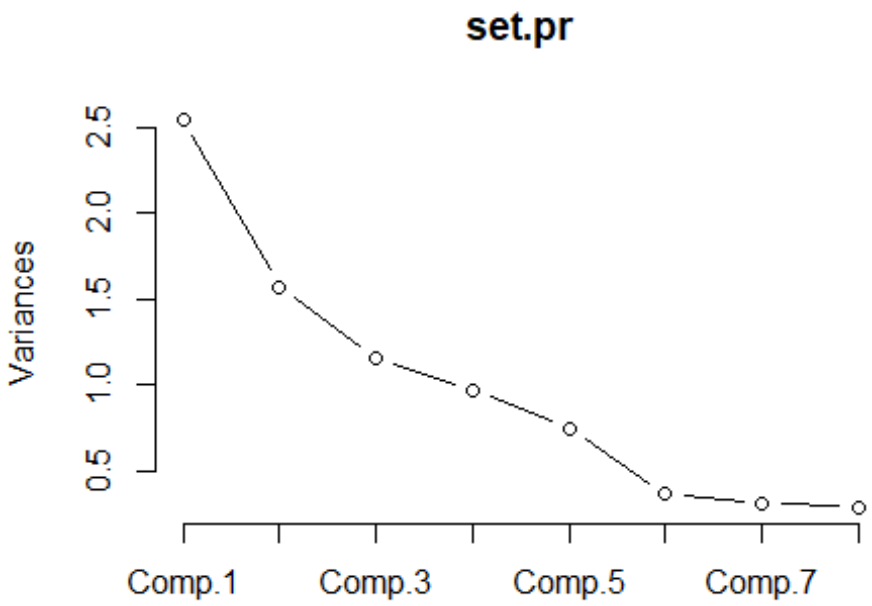

```
# Plot of the 51.5% of the total variability that is expressed by the f
irst 2 PCs
# We observe that there is overlap, maybe the groups we have taken are
not the best possible!
# The partition of group 2 from 4 does make sense, as well as the parti
tion of group 1 from 2 (or group 1 from 4).
# However there's overlap between groups 3 and 4, fact that indicates t
hat this partitions does not seem to make much sense!
# Removing outliers is the next step to be taken in order to make impro
vements!



plot(set.pr$scores[,1], set.pr$scores[,2], main ="Ward's", xlab="1st PC
", ylab="2nd PC", type='n')
text(set.pr$scores[,1], set.pr$scores[,2], xlab="1st PC", ylab="2nd PC"
,  label=cluster.d3, col=cluster.d3)
```



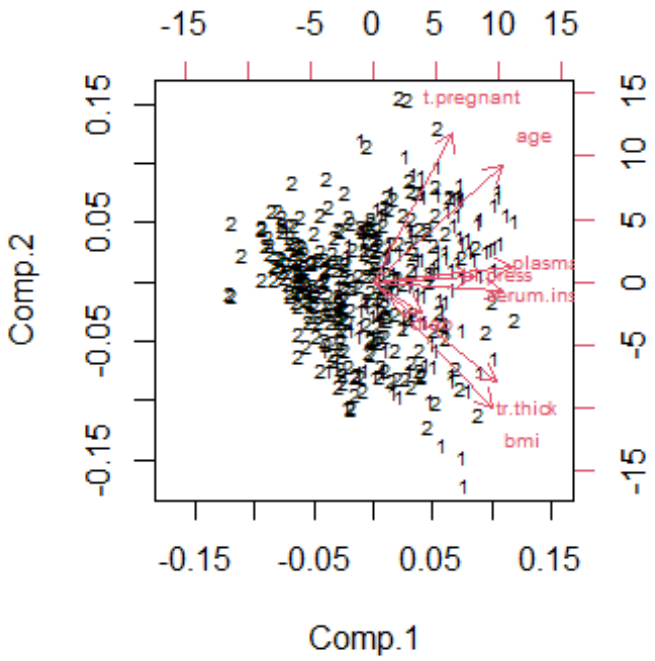

```
# Different label now!

plot(set.pr$scores[,1], set.pr$scores[,2], main ="Class", xlab="1st PC"
, ylab="2nd PC", type='n')
text(set.pr$scores[,1], set.pr$scores[,2], xlab="1st PC", ylab="2nd PC"
,  label=new_df3[,9], col=new_df3[,9])
```



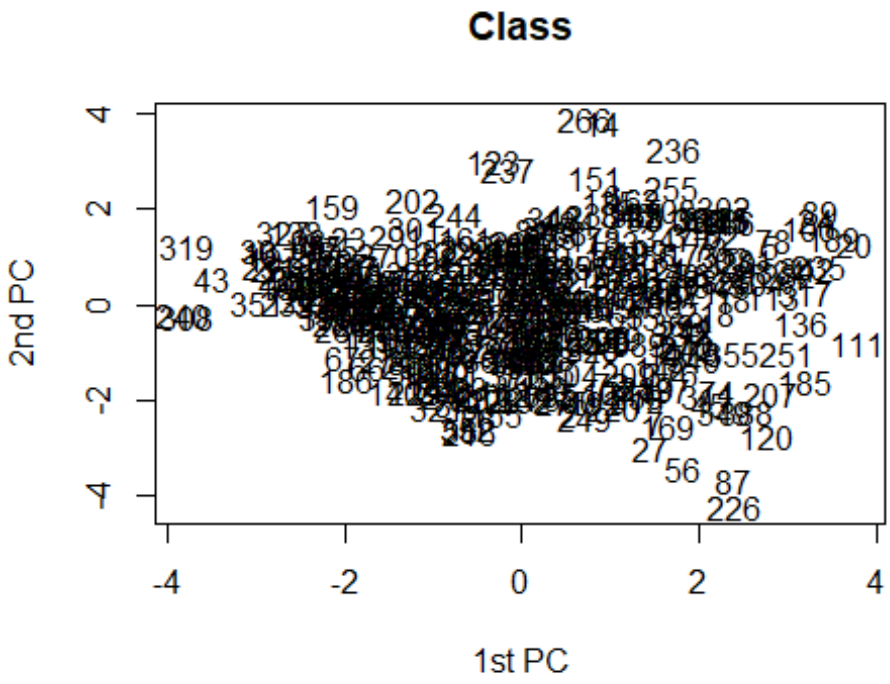

```
# Labels: The class! ( 1 refers to the women that will develop diabetes
, 2 refer to the women that won't develop diabetes)
# We observe that there's not much agreement between the results of War
ds and Class on this case either!


screeplot(set.pr,type="lines")
```



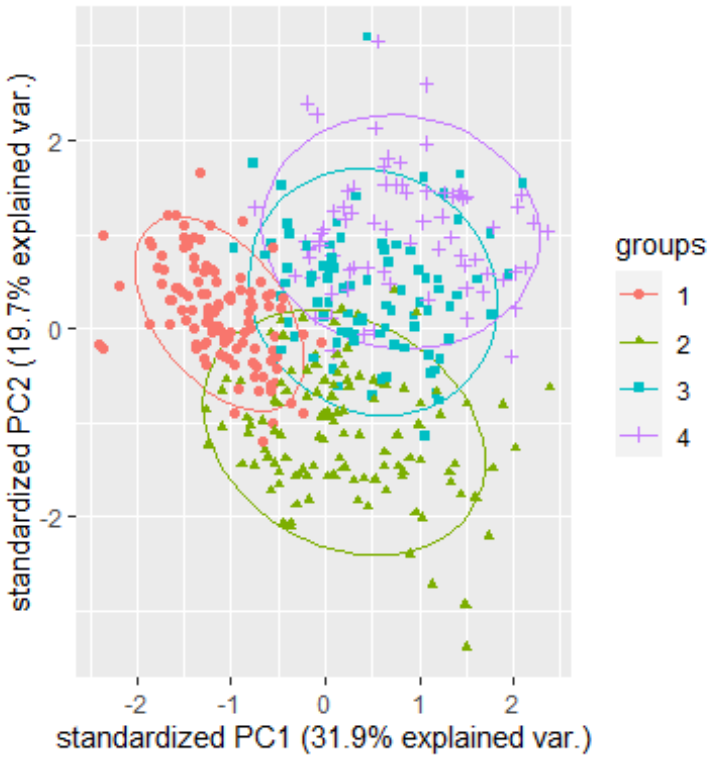

```
# The screeplot indicates to use the first 2 PCs as the greatest angle
is located between the first and the second component

biplot(set.pr, choices=c(1,2), xlabs=new_df3[,9], cex=.6)
```



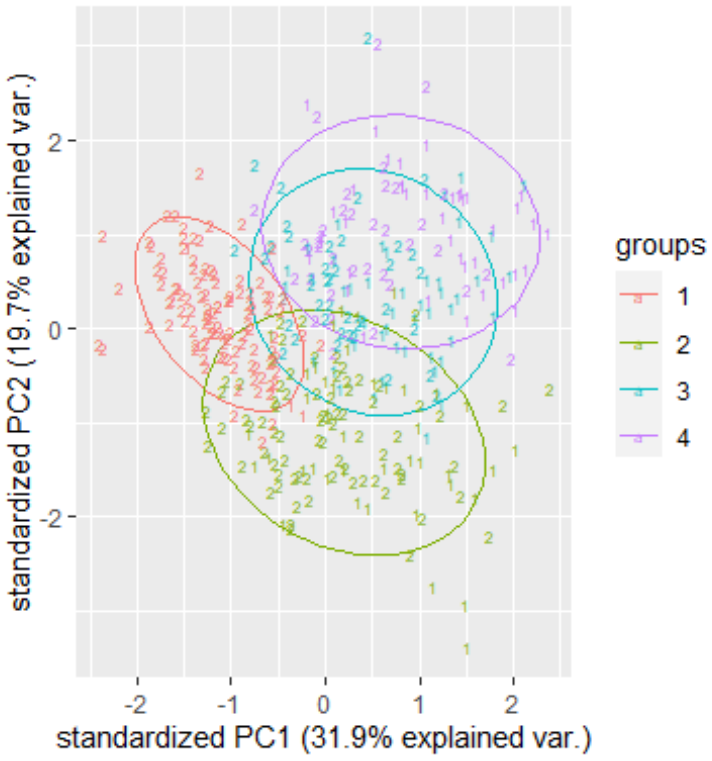

```
# Biplot offers useful insights about the variables!

# In order to spot on the possible outliers!
plot(set.pr$scores[,1], set.pr$scores[,2], main ="Class", xlab="1st PC"
, ylab="2nd PC", type='n')
text(set.pr$scores[,1], set.pr$scores[,2], xlab="1st PC", ylab="2nd PC"
)
```



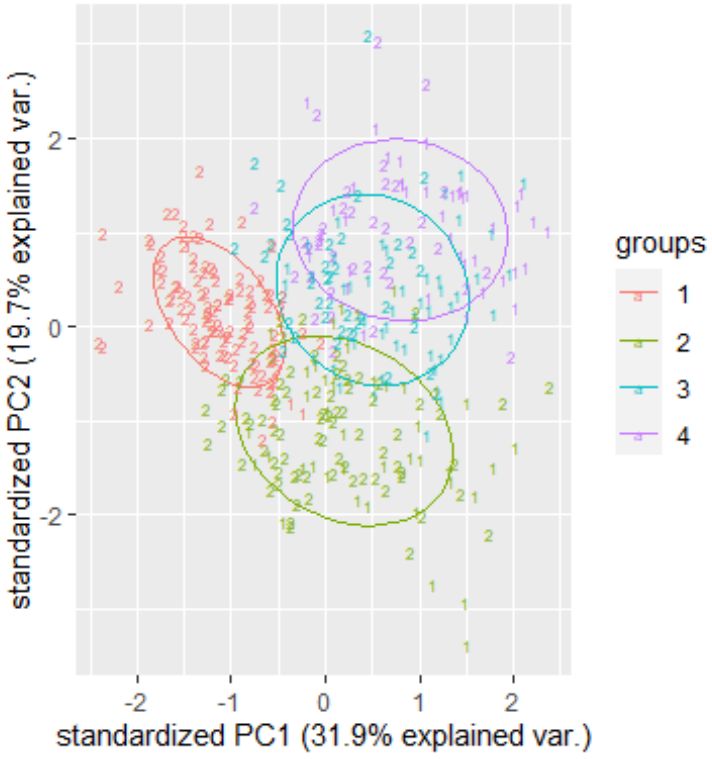

```
# Quite handy


library(ggbiplot)

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid

g <- ggbiplot(set.pr, choices = c(1,2), pc.biplot = TRUE, groups = as.f
actor(cluster.d2), ellipse = TRUE,
ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, alpha=0)
g <- g+geom_point(aes(colour=groups,shape=groups),size=1.3)
g <- g+scale_color_discrete(name = 'groups')
g
```




```
# We observe that there is too much overlap between groups 3 and 4!


g1 <- ggbiplot(set.pr, choices = c(1,2), pc.biplot = TRUE, groups = as.
factor(cluster.d2), ellipse = TRUE,
ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, alpha=0)
g1 <- g1+geom_text(aes(colour=groups, label=new_df3[,9]),size=2)
g1 <- g1+scale_color_discrete(name = 'groups')
g1
```




```
# Same conclusions as before

g2 <- ggbiplot(set.pr, choices = c(1,2), pc.biplot = TRUE, groups = as.
factor(cluster.d2), ellipse = TRUE,
var.axes=FALSE, varname.size = 4, alpha=0)
g2 <- g2+geom_text(aes(colour=groups, label=new_df3[,9]),size=2)
g2 <- g2+scale_color_discrete(name = 'groups')
g2
```




```
# Same conclusions as before
```

Removing outliers

```
# Dig into the outliers
new_df3[226,]

##     t.pregnant   plasma bl.press tr.thick serum.ins     bmi     diab
age
## 446          0 13.41641 8.831761 7.937254  3.741657 7.70714 1.555635
5
##     n.class
## 446       1

# It's line 446 which was indicated before in the single linkage method
as an outlier!
# We shall remove this line and try again the clustering!
new_df4 <- new_df3[-226,]
# New dataset to work with!
```

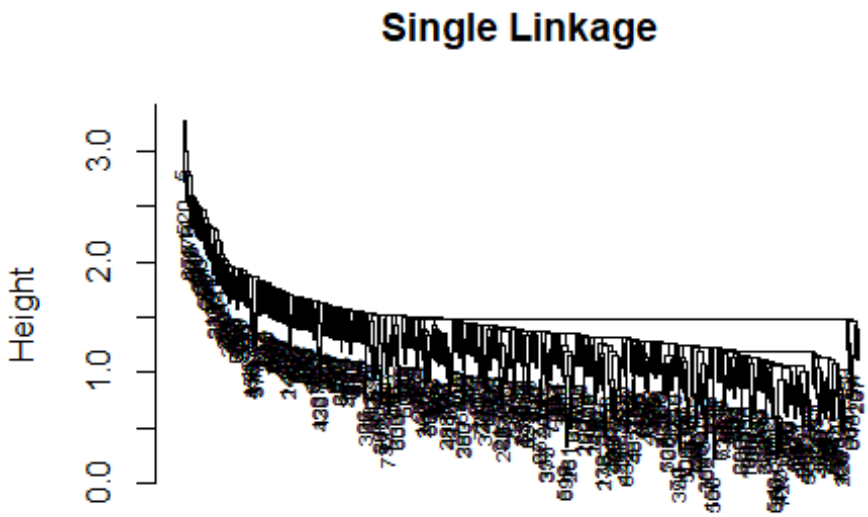
Implementation of Hierarchical clustering using different linkages with proximity measure the euclidian distance on the new dataset now

```
d4 <- dist(scale(new_df4[,-9]), method = "euclidian")
# Scaled dissimilarity matrix using as distance the euclidian one
head(d4)

## [1] 6.672468 2.050452 6.006153 6.802444 4.596916 4.463918
```

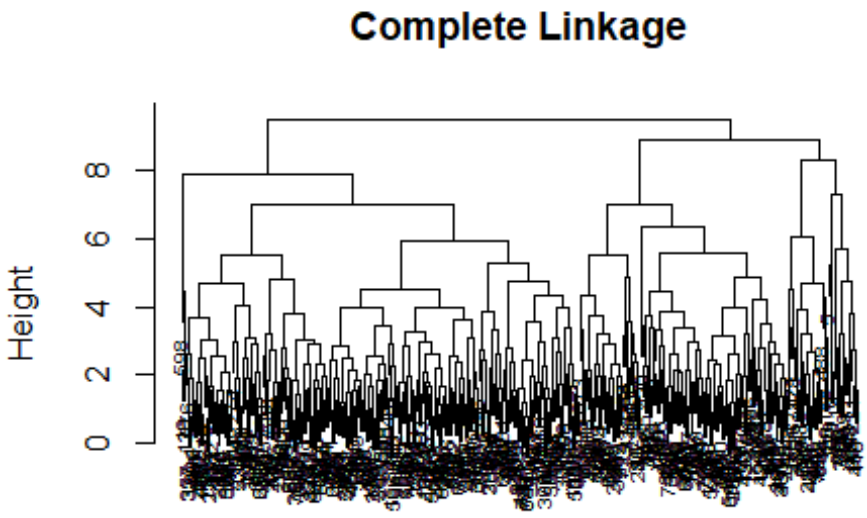
```
# Hierarchical method, different linkage methods
fit4.s <- hclust(d4, method = "single")
fit4.c <- hclust(d4, method = "complete")
fit4.a <- hclust(d4, method = "average")
fit4.d <- hclust(d4, method="ward.D")
fit4.d2 <- hclust(d4, method="ward.D2")
```

Let us plot the resulting dendrograms

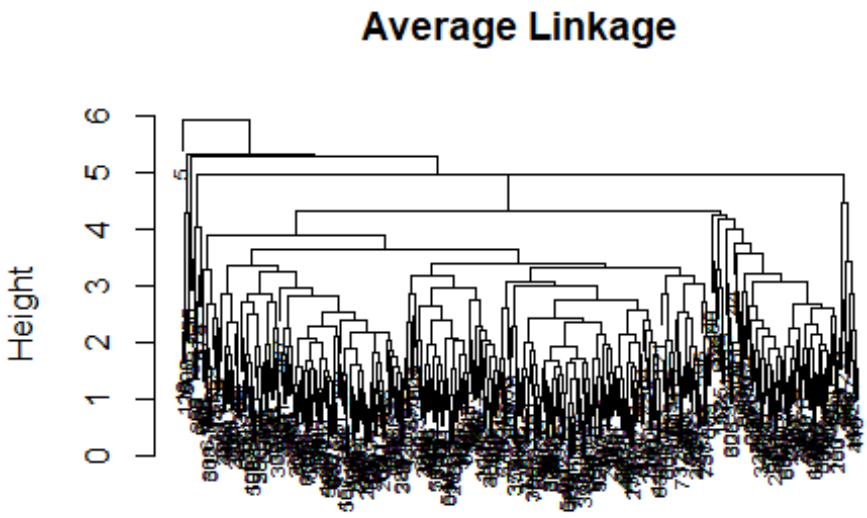
```
# Let us plot the resulting dendrograms
plot(fit4.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



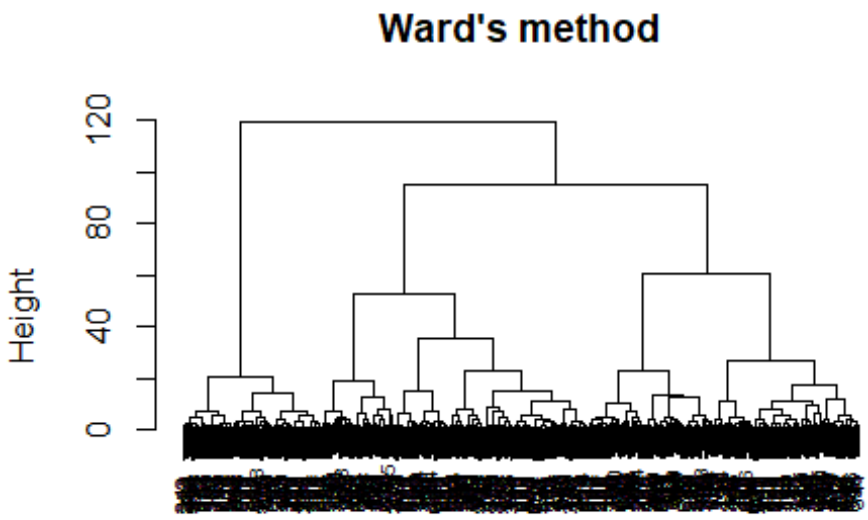

```
# Default label: the increasing number (row number)
# Let us observe the first split! (conservative way to tell how many gr
oups we have!)
# We observe that the single linkage (or nearest neighbor has fallen in
the trap of the chain effect! It's known
# that nearest linkage is prone to the chain effect, yet it's useful to
identify potential outliers!
plot(fit4.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```




```
# It indicates that there are probably 3 groups, row 5 has to be checke
d, potential outlier!(or singleton)
plot(fit4.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



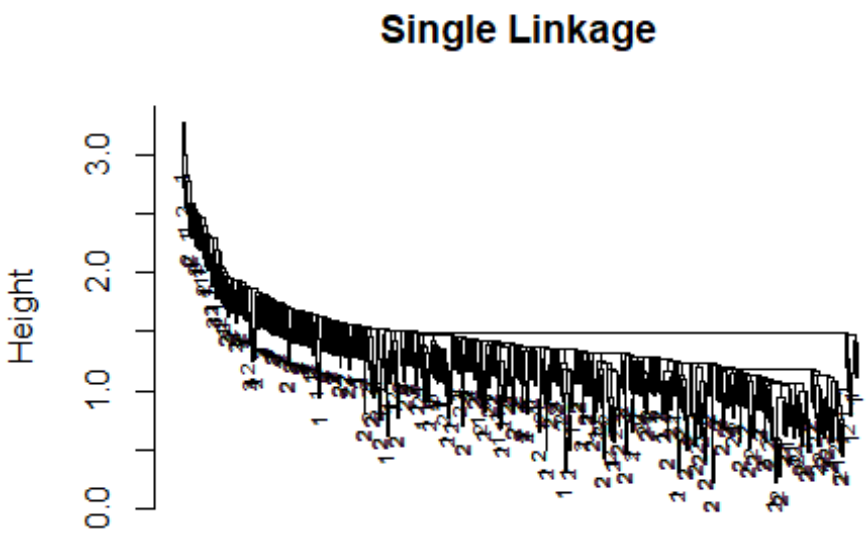
```
# Average linkage seems to be a bit more reasonable in general
plot(fit4.d, main = "Ward's method", sub = "", xlab = "", cex=.6)
```



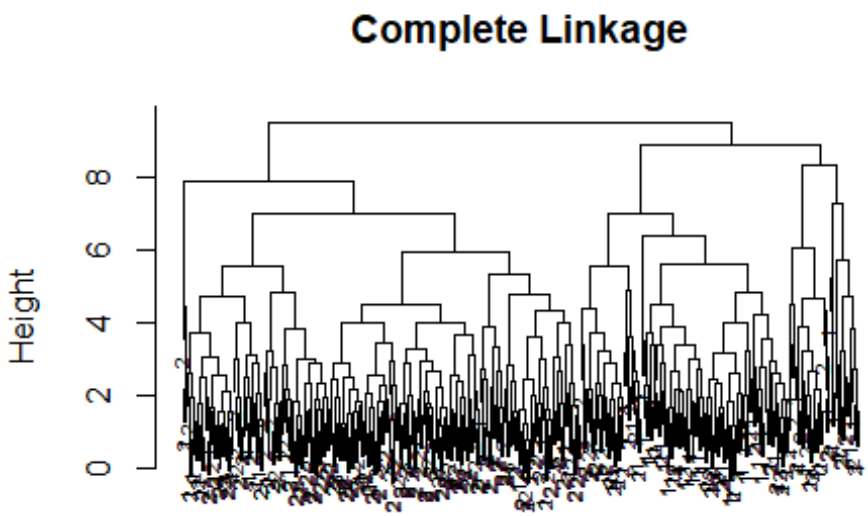

```
# The plot indicates that there are 3 groups (conservative approach)
# Row 5 has to be checked!
```

More dendrogram plots

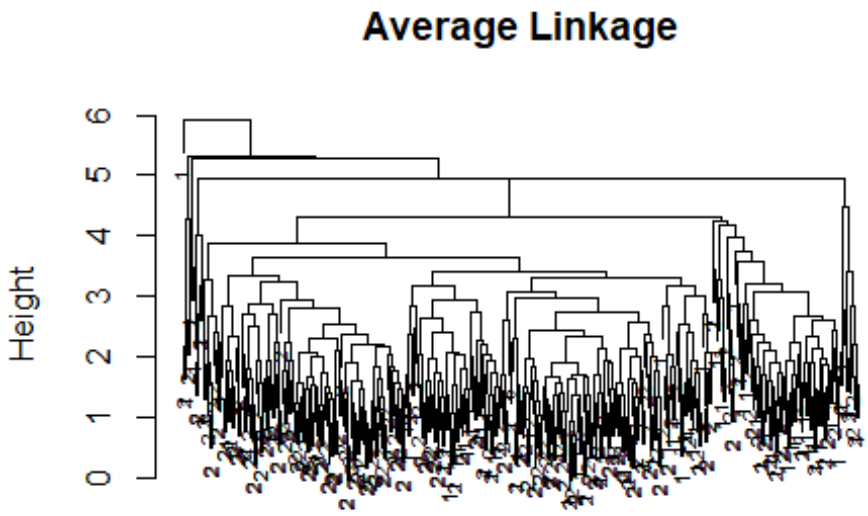
```
plot(fit4.s, main = "Single Linkage", sub = "", xlab = "", labels=new_d
f4[,9], cex=.6)
```



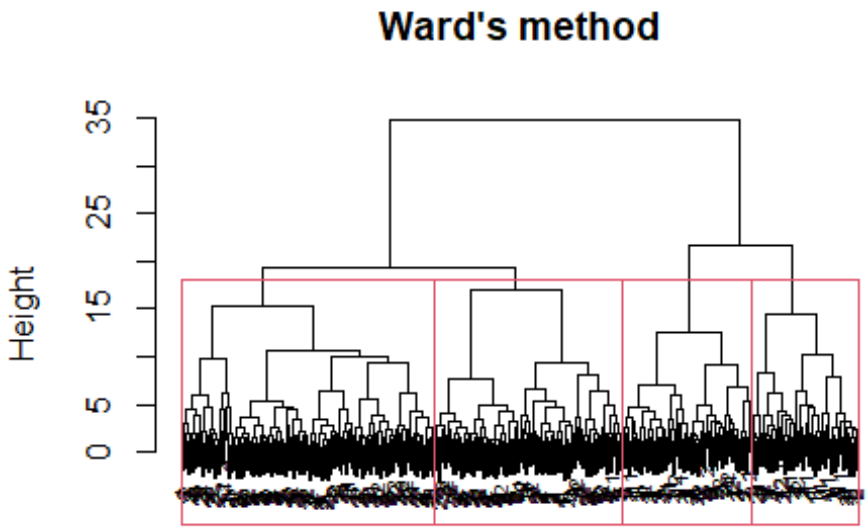

```
# Label is the 9th column, which refers to class (diabetes-non diabetes
that have corresponding values 1,2 respectively)
plot(fit4.c, main = "Complete Linkage", sub = "", xlab = "", labels=new
_df4[,9], cex=.6)
```



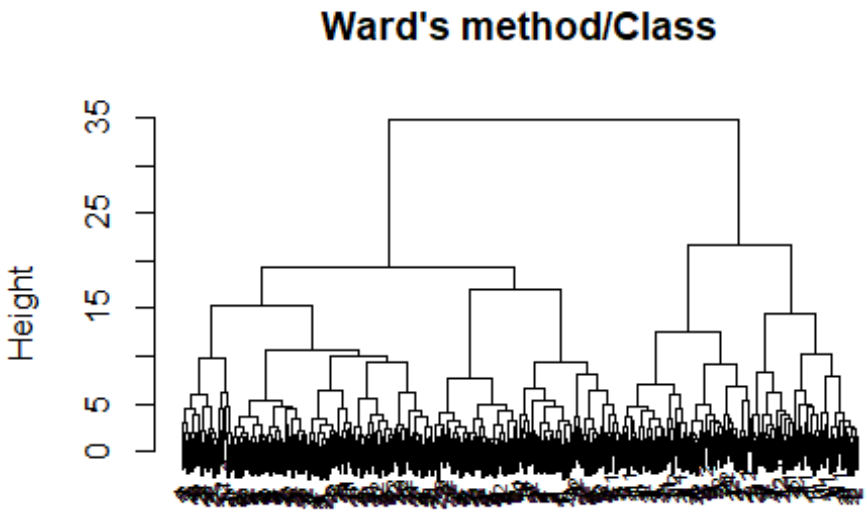

```
plot(fit4.a, main = "Average Linkage", sub = "", xlab = "", labels=new_
df4[,9], cex=.6)
```



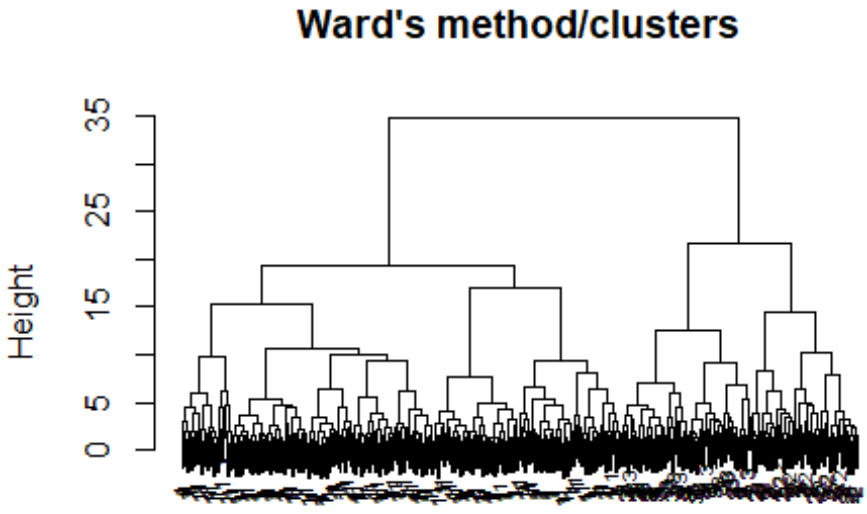
```
plot(fit4.d2, main = "Ward's method", sub = "", xlab = "", labels=new_d
f4[,9], cex=.6)
# We should be aware of the fact that Ward's linkage may impose equal s
ize to clusters as in our case.
rect.hclust(fit4.d2, 4)
```



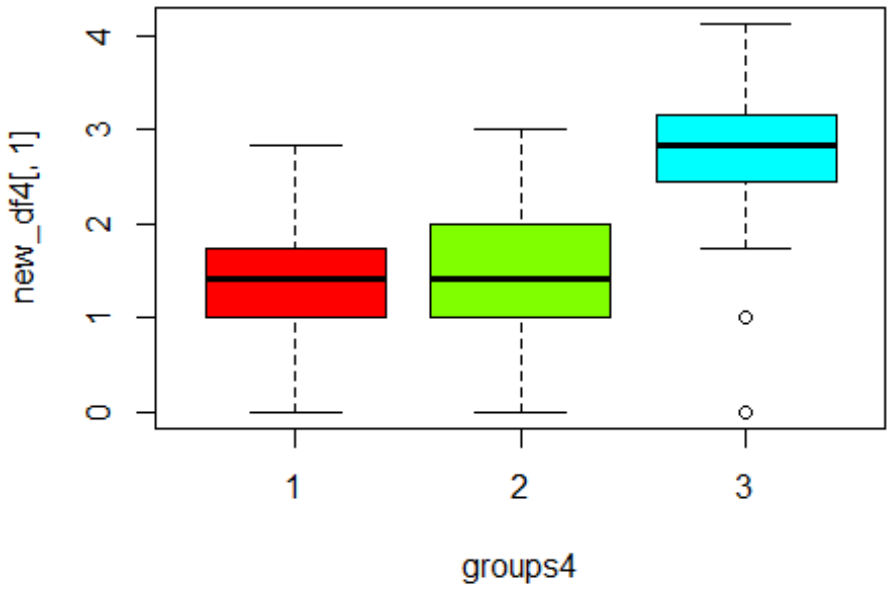

```
# Split a dendrogram in 4 rectangulars

cluster.d4 <- cutree(fit4.d2,3)
# Split the tree in 3 branches
cluster.d4[1:10]

##  4  5  7  9 14 15 17 19 20 21
##  1  1  1  2  2  3  2  1  1  2

# Gives the cluster membership for the first 10 observations of the dat
aset.
par(mfrow=c(1,1))
plot(fit4.d2, main = "Ward's method/Class", sub = "", xlab = "",labels=
new_df4[,9], cex=.6)
```



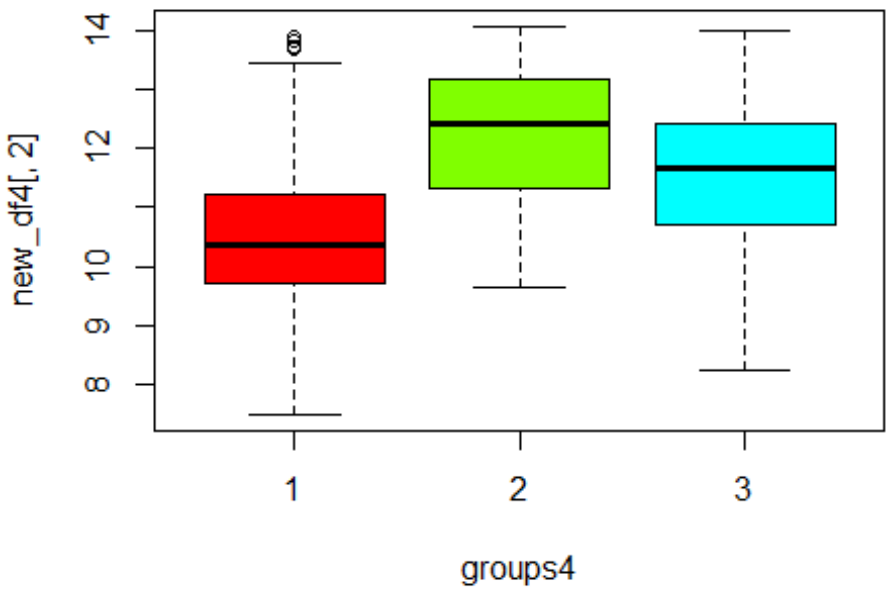

```
# We observe  3 clusters probably
plot(fit4.d2, main = "Ward's method/clusters", sub = "", xlab = "", lab
els=cluster.d4, cex=.6)
```



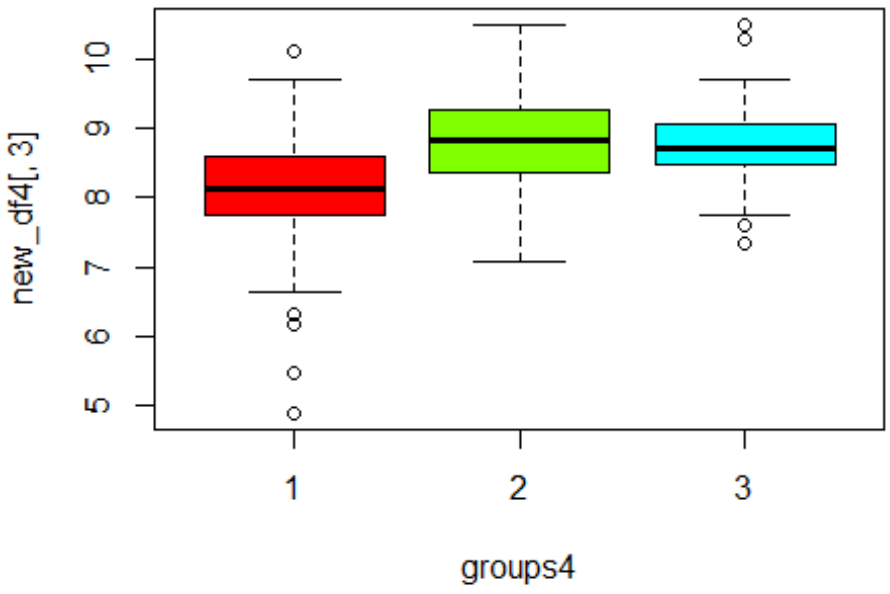
```r
# We observe 3 clusters probably
table(cluster.d4, new_df4[,9])

##
## cluster.d4   1   2
##          1  48 206
##          2  39  23
##          3  42  33

# What is the agreement of the diabetes-non diabetes?? (Reminder: 1 ref
ers to women developing diabetes, 2 refers to non development of diabet
es!)
# Non-diabetes women have been split in 3 groups. 206 in the first clus
ter, 23 in the second cluster, 33 in the third cluster
# Women with diabetes have been split in 3 groups. 48 in the first clus
ter, 39 in the second cluster, 42 in the third cluster
# Contingency table!

# Let us check if the clusters we created make sense
groups4 <- as.factor(cluster.d4)

# Let us observe their common characteristics with respect to the varia
bles 1:8
boxplot(new_df4[,1]~groups4, col=rainbow(4))
```



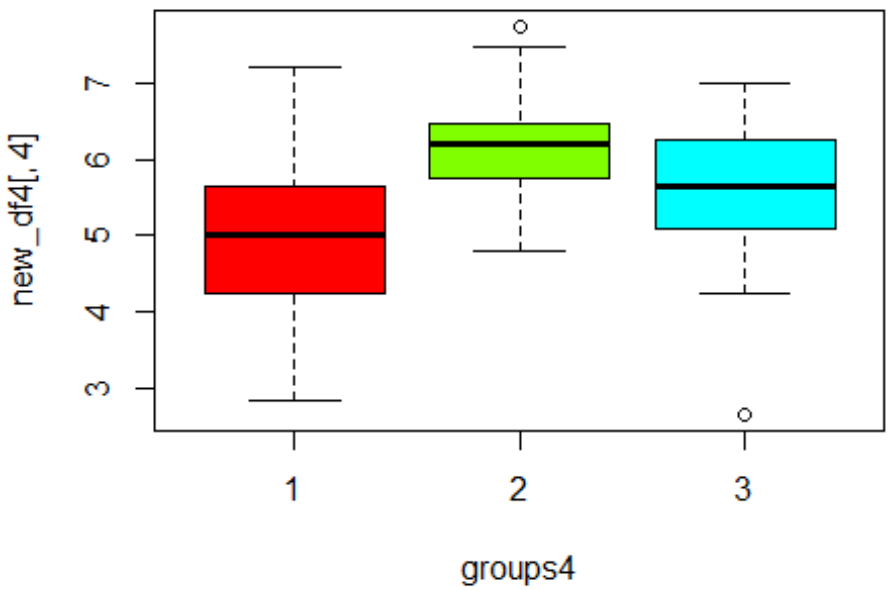

```r
boxplot(new_df4[,2]~groups4, col=rainbow(4))
```



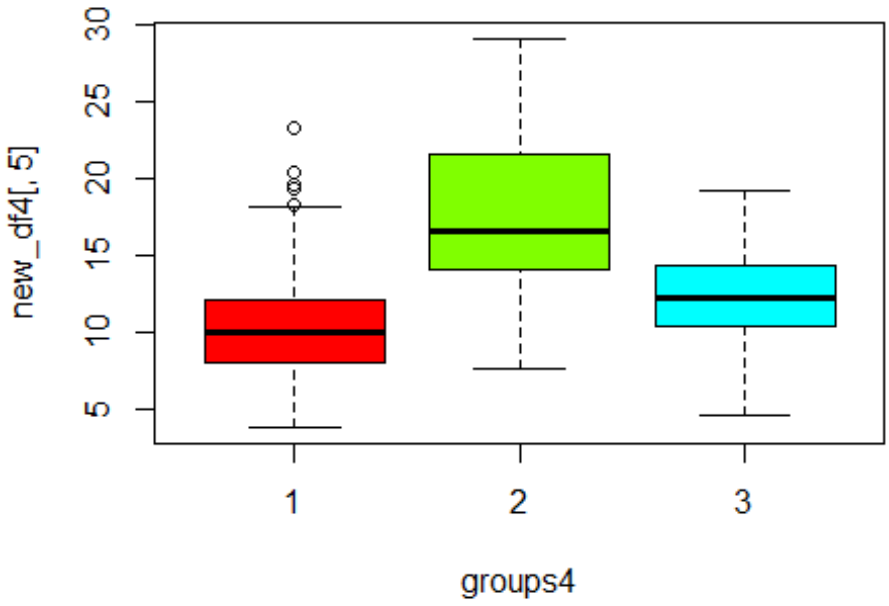

```
boxplot(new_df4[,3]~groups4, col=rainbow(4))
```



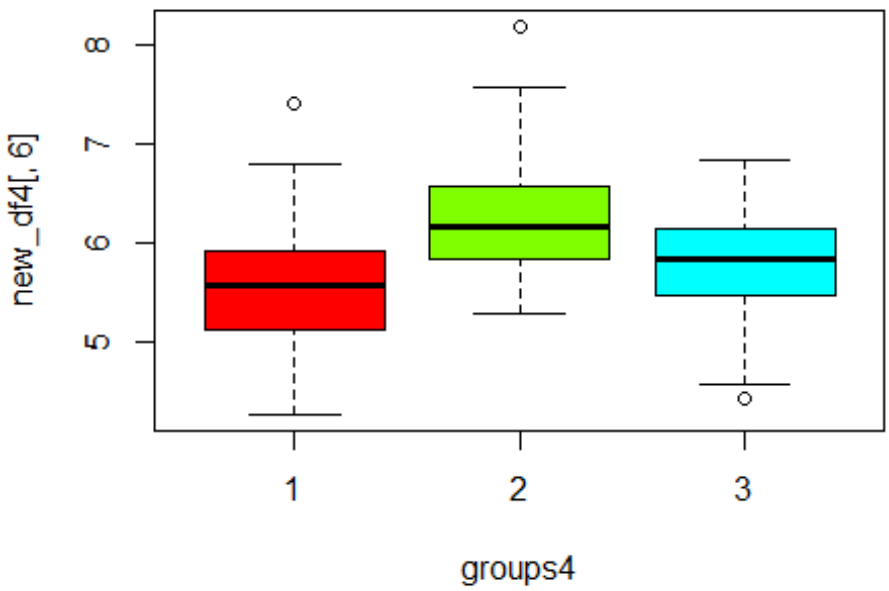

```
boxplot(new_df4[,4]~groups4, col=rainbow(4))
```



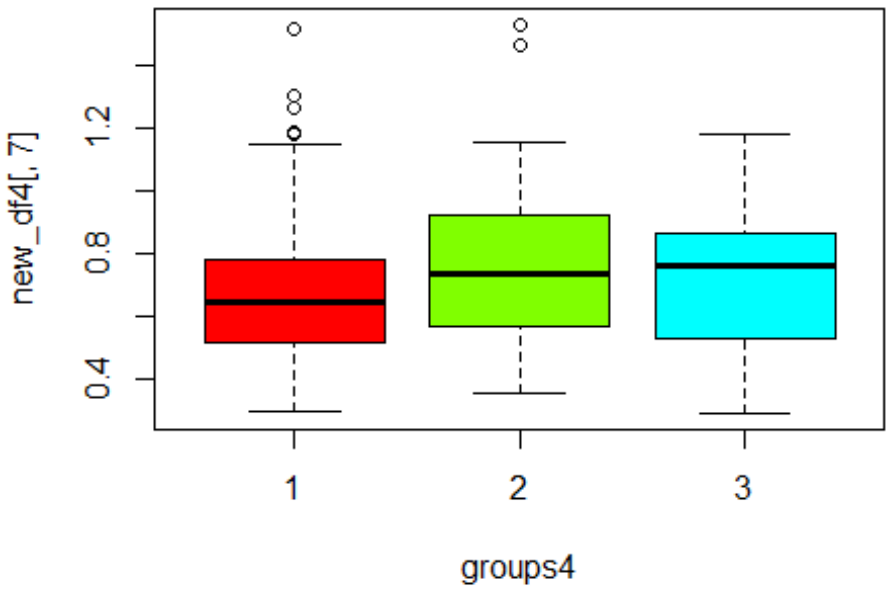

```
boxplot(new_df4[,5]~groups4, col=rainbow(4))
```



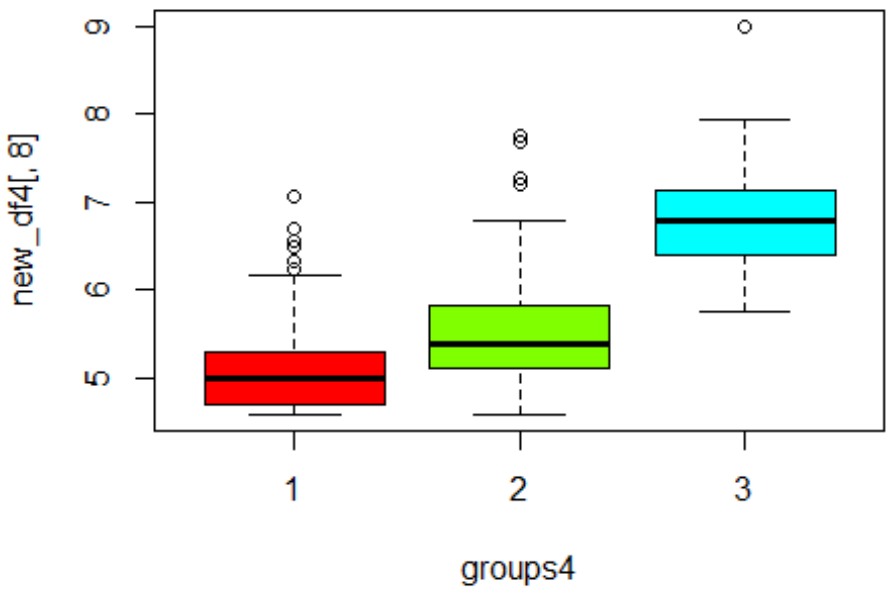

```
boxplot(new_df4[,6]~groups4, col=rainbow(4))
```




```
boxplot(new_df4[,7]~groups4, col=rainbow(4))
```




```
boxplot(new_df4[,8]~groups4, col=rainbow(4))
```

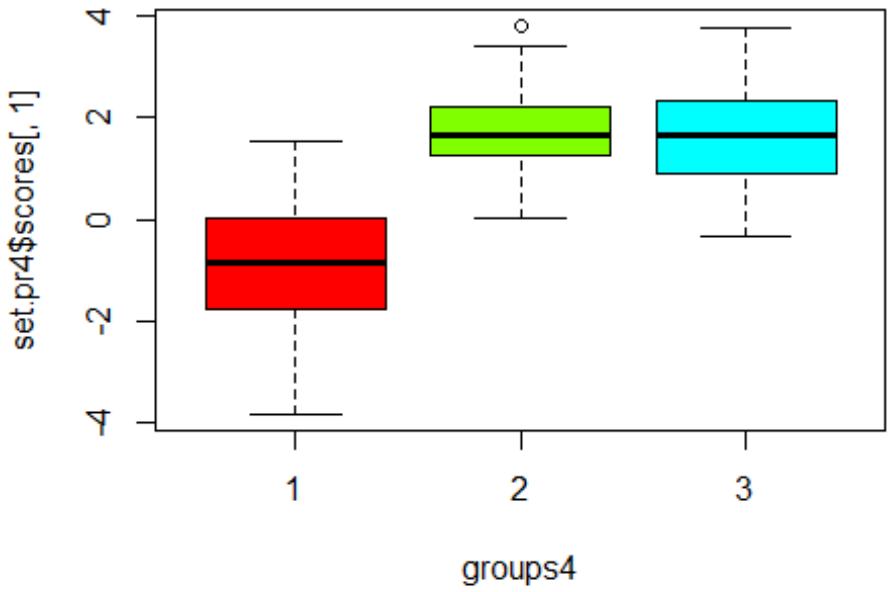



```
# On basis of the original variables we observe that the groups do not
differ!
```

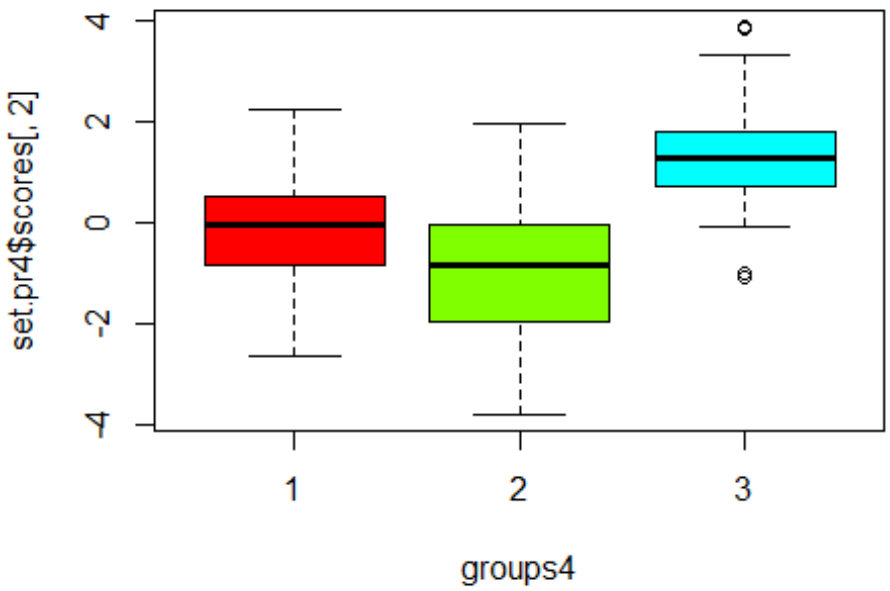
PCA on the new dataset now

```
# Creating PCs once again
set.pr4 <- princomp(scale(new_df4[, -9]))
# Exclude the labels, which are in the 9th column
summary(set.pr4)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     C
omp.5
## Standard deviation     1.5990393 1.2440752 1.0823354 0.9888361 0.869
29709
## Proportion of Variance 0.3204354 0.1939615 0.1468067 0.1225380 0.094
70188
## Cumulative Proportion  0.3204354 0.5143968 0.6612035 0.7837415 0.878
44340
##                            Comp.6     Comp.7     Comp.8
## Standard deviation     0.59891328 0.56492378 0.54049024
## Proportion of Variance 0.04495211 0.03999465 0.03660984
## Cumulative Proportion  0.92339551 0.96339016 1.00000000

# The 1st PC explains the 32% of the total variability
# The first 2 PCs explain the 51.4% of the total variability
# The first 3 PCs explain the 66% of the total variability
```

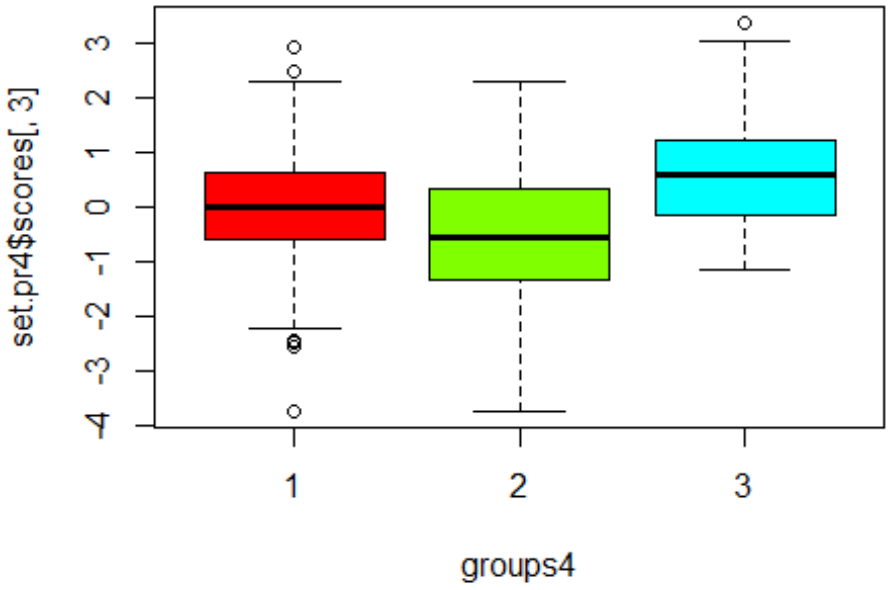
```r
# The first 4 PCs explain the 78.3% of the total variability
# We observe that the standard deviatons with value more than 1, are th
e first 3 ones (the fourth is valued at 0.988
# almost 1,hence we have to take that into consideration!)


boxplot(set.pr4$scores[,1]~groups4, col=rainbow(4))
```



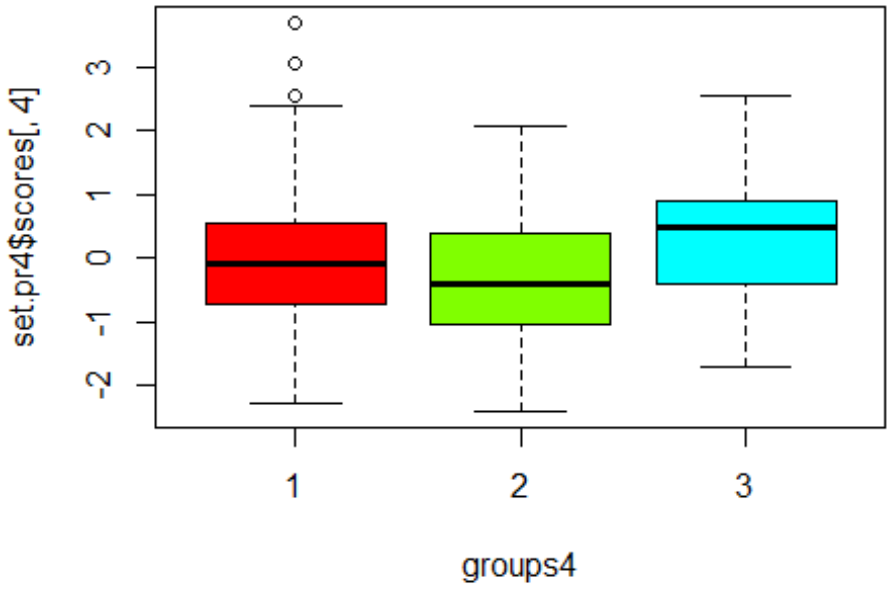

```r
# We observe that group 1 differs from  groups 2,3
boxplot(set.pr4$scores[,2]~groups4, col=rainbow(4))
```



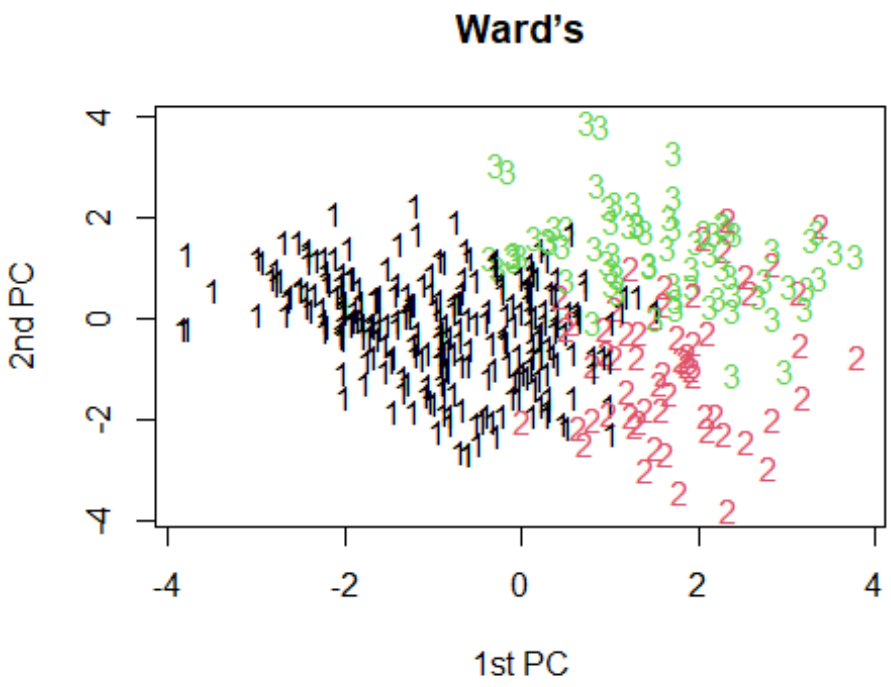

```
# We observe that group 3 differs from the rest
boxplot(set.pr4$scores[,3]~groups4, col=rainbow(4))
```



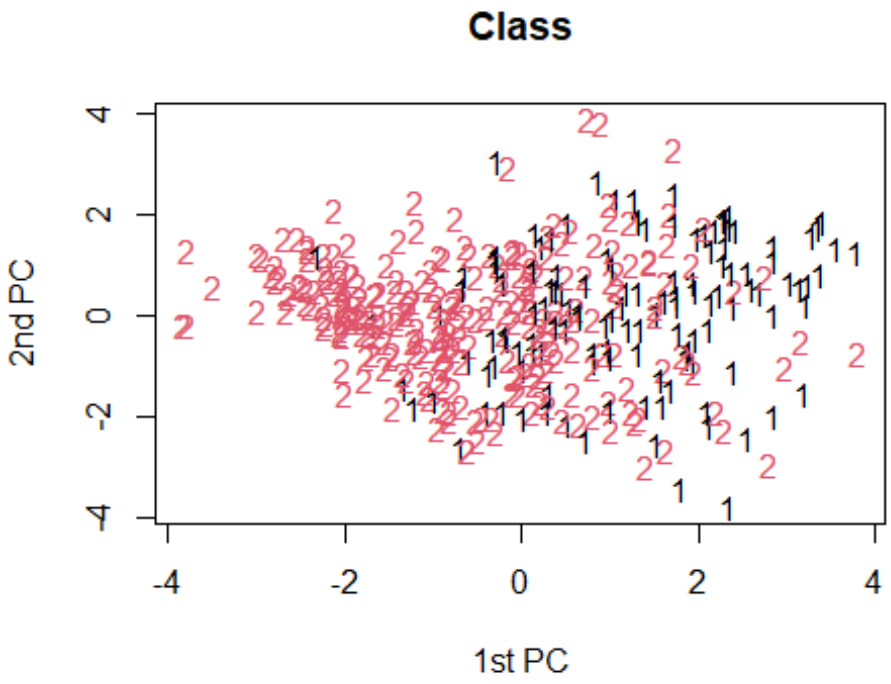
```r
# We observe that groups do not differ in the case of the 3rd PC
boxplot(set.pr4$scores[,4]~groups4, col=rainbow(4))
```



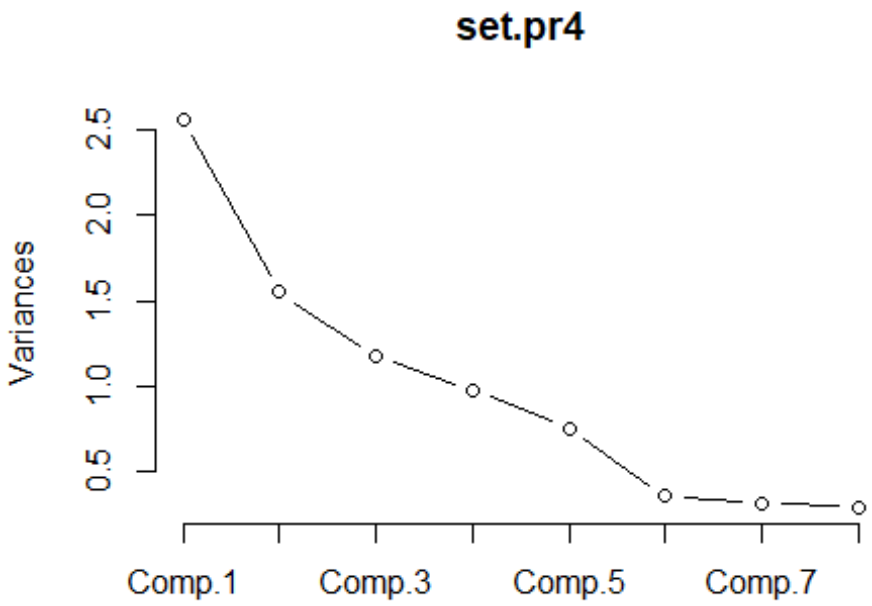

```r
# We observe that groups do not differ in the case of the 4th PC
# The first 2 PCs explain the 51.4% of the total variability


# Visualize the result of a clustering technique using PCs
plot(set.pr4$scores[,1], set.pr4$scores[,2], main ="Ward's", xlab="1st
PC", ylab="2nd PC",  type='n')
text(set.pr4$scores[,1], set.pr4$scores[,2], xlab="1st PC", ylab="2nd P
C", label=cluster.d4, col=cluster.d4)
```



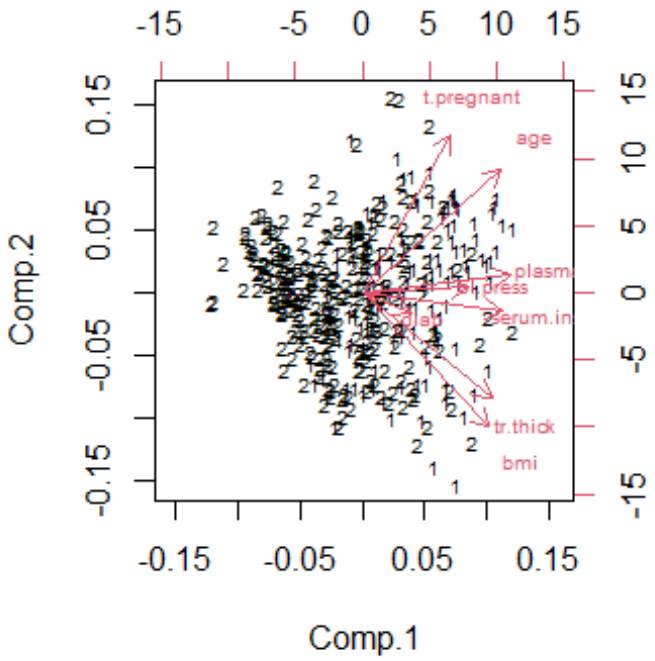

```
# Plot of the 51.4% of the total variability that is expressed by the f
irst 2 PCs
# We observe that the situation is definately improved in comparison wi
th the 4 groups.
# The overlap, is not as much as it was before!
# Removing more outliers is the next step to be taken in order to make
improvements!


plot(set.pr4$scores[,1], set.pr4$scores[,2], main ="Class", xlab="1st P
C", ylab="2nd PC", type='n')
text(set.pr4$scores[,1], set.pr4$scores[,2], xlab="1st PC", ylab="2nd P
C",  label=new_df4[,9], col=new_df4[,9])
```



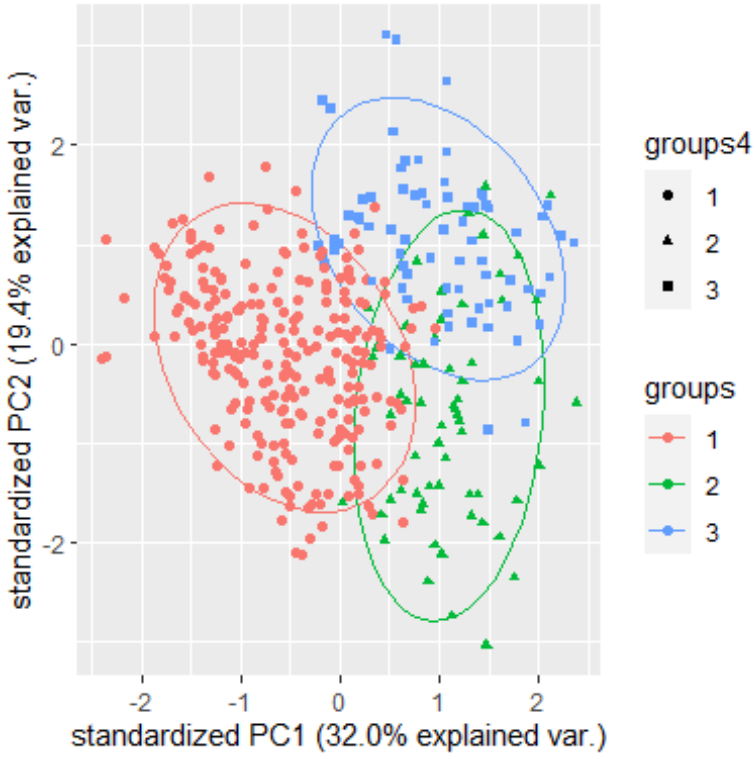

```
# Labels: The class! ( 1 refers to the women that will develop diabetes
, 2 refer to the women that won't develop diabetes)
# We observe that there's not much agreement between the results of War
ds and Class on this case either!


screeplot(set.pr4,type="lines")
```




```
# The screeplot indicates to use the first 2 PC's as the greatest angle
is located between the first and the second component

biplot(set.pr4, choices=c(1,2), xlabs=new_df4[,9], cex=.6)
```




```
# Useful for interpretation! Provides insights about the loadings of ea
ch variable, as well as it aids in detecting outliers!


g4 <- ggbiplot(set.pr4, choices = c(1,2), pc.biplot = TRUE, groups = as
.factor(cluster.d4), ellipse = TRUE,
ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, alpha=0)
g4 <- g4+geom_point(aes(colour=groups4,shape=groups4),size=1.3)
g4 <- g4+scale_color_discrete(name = 'groups')
g4
```




```
# We observe a much better partitioning than before when we had 3 group
s!
# Let us try to make things even better
```

Removing outliers

```
# Dig into the outliers of the dataset new_df4, which is the last one w
e worked with
new_df4[2,]
```

```
##   t.pregnant  plasma bl.press tr.thick serum.ins      bmi     diab
age
## 5          0 11.7047 6.324555  5.91608  12.96148 6.565059 1.512614 5
.744563
##   n.class
## 5       1
```

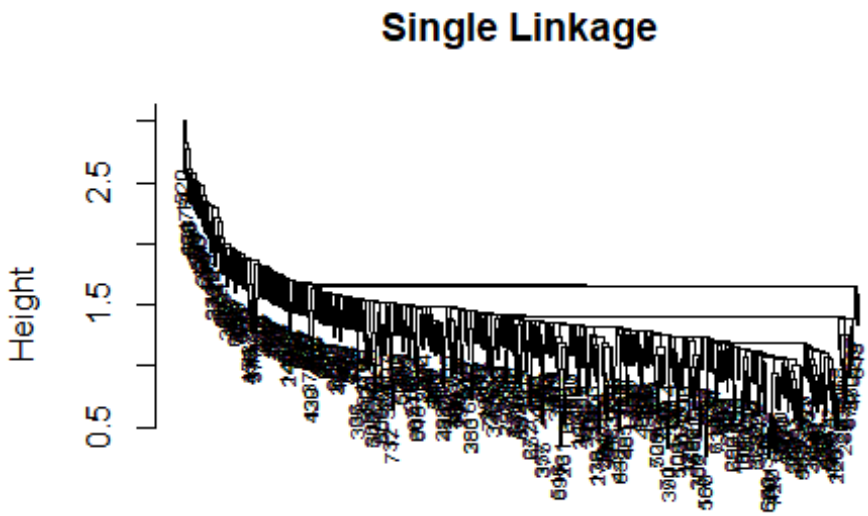
```
# It's line 5 which was indicated before in the single linkage method a
s an outlier!
# We shall remove this line and try again the clustering!
new_df5 <- new_df4[-2,]
# Creating the new dataset to work with!
```

Implementation of Hierarchical clustering using different linkages with proximity measure the euclidian distance on the new dataset

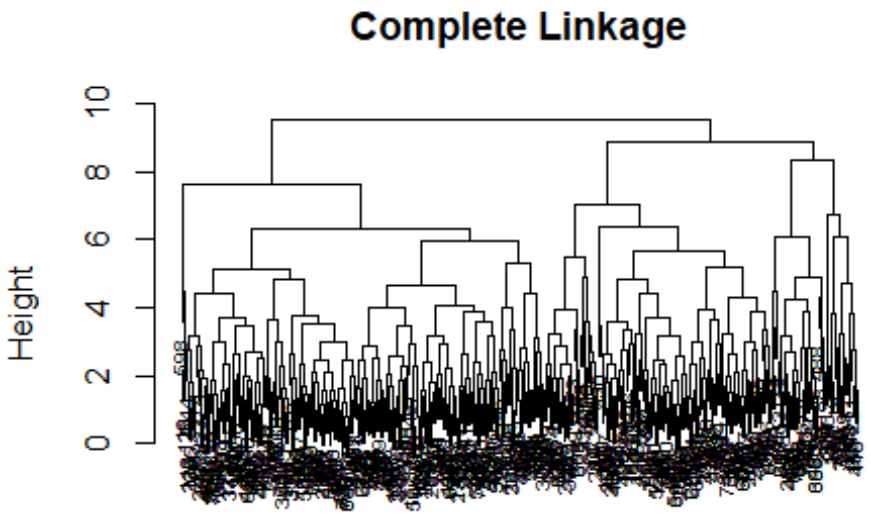
```
d5 <- dist(scale(new_df5[,-9]), method = "euclidian")
# Scaled dissimilarity matrix using as distance the euclidian one
head(d5)

## [1] 2.059820 6.000130 6.798353 4.605894 4.478172 4.570903

# Hierarchical method, different linkage methods
fit5.s <- hclust(d5, method = "single")
fit5.c <- hclust(d5, method = "complete")
fit5.a <- hclust(d5, method = "average")
fit5.d <- hclust(d5, method="ward.D")
fit5.d2 <- hclust(d5, method="ward.D2")
```

Let us plot the resulting dendrograms

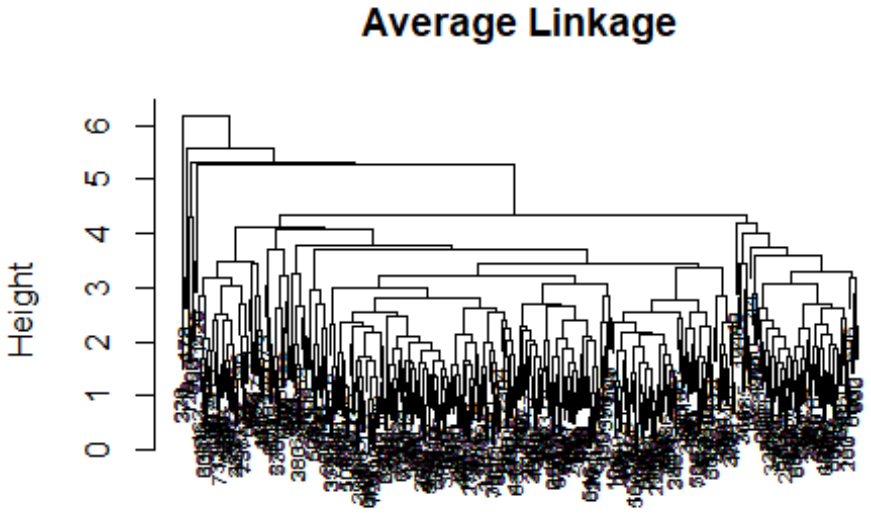
```
plot(fit5.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



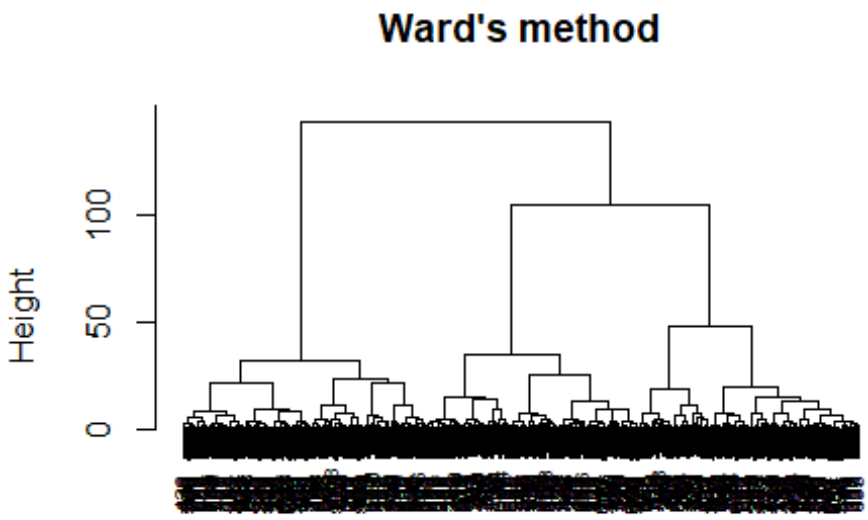

```
# Default label: the increasing number (row number)
# Let us observe the first split! (conservative way to tell how many gr
oups we have!)
# We observe that the single linkage (or nearest neighbor has fallen in
the trap of the chain effect! It's known
# that nearest linkage is prone to the chain effect, however it's usefu
l to identify potential outliers!
plot(fit5.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```



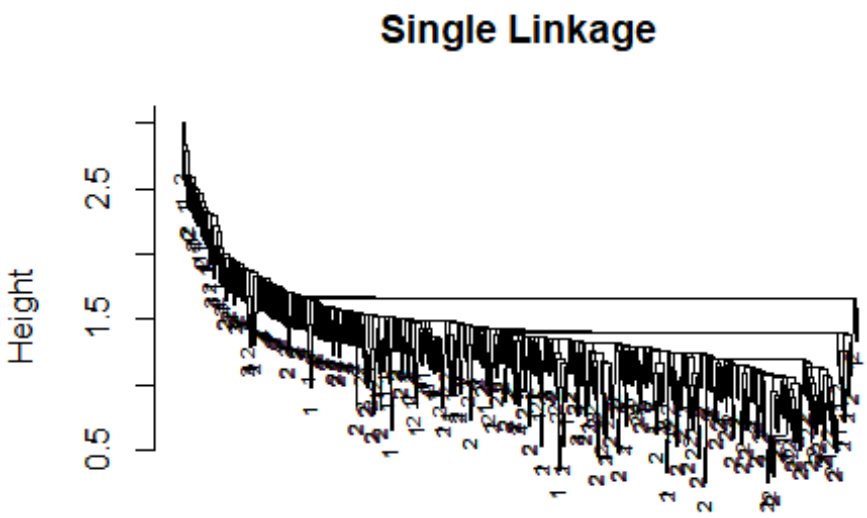

```r
# It indicates that there are probably 3 groups, row 598 has to be chec
ked, potential outlier!(or singleton)
plot(fit5.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



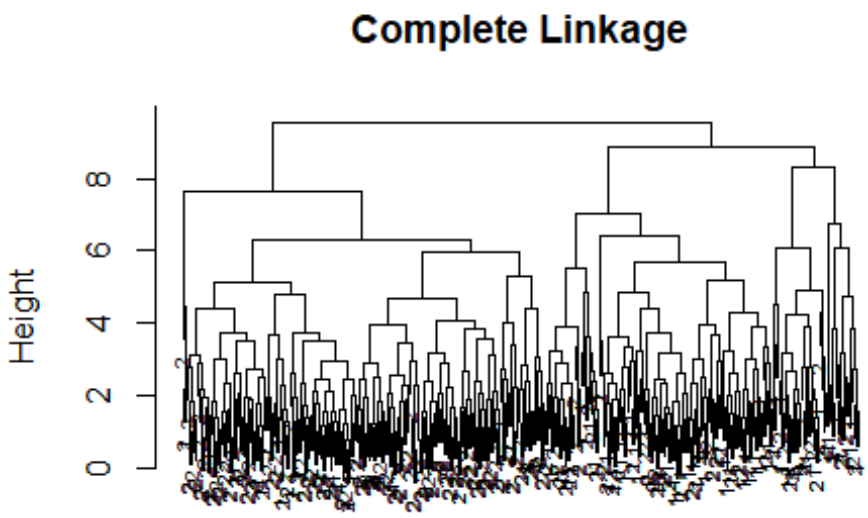
```r
# Average linkage seems to be a bit more reasonable in general
plot(fit5.d, main = "Ward's method", sub = "", xlab = "", cex=.6)
```



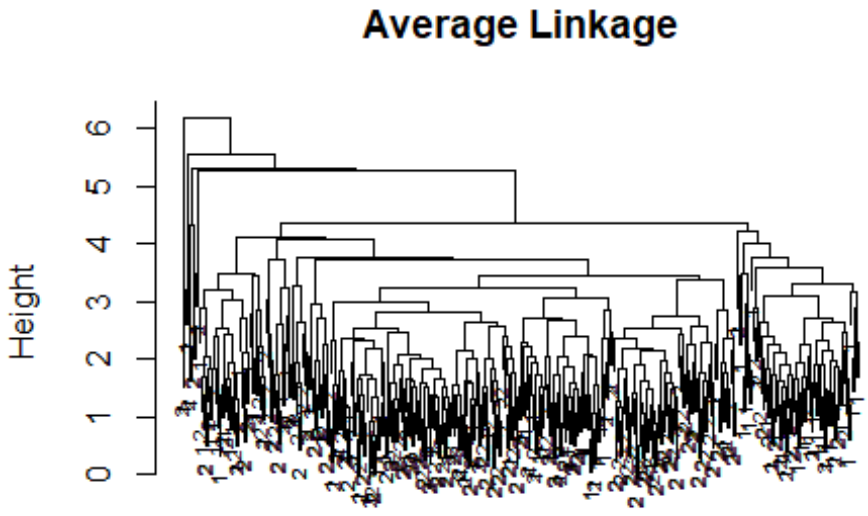

```r
# The plot indicates that there are 3 groups (conservative approach)


plot(fit5.s, main = "Single Linkage", sub = "", xlab = "", labels=new_d
f5[,9], cex=.6)
```



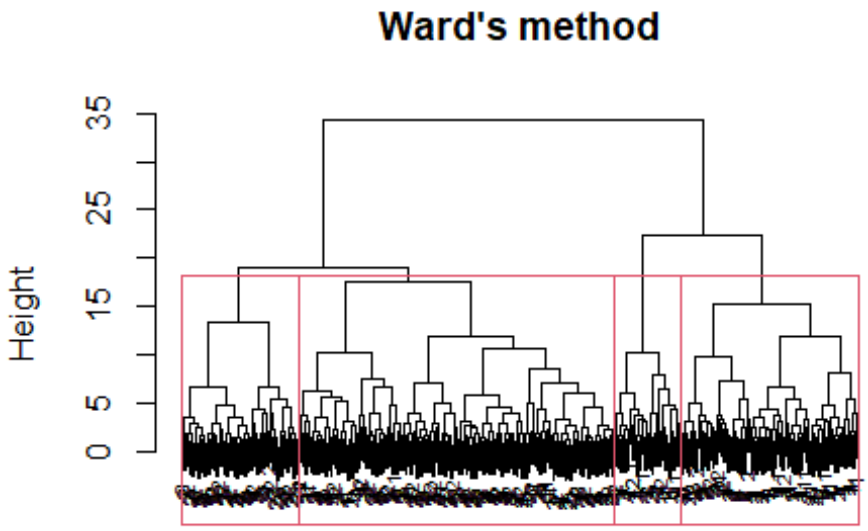

```
# Label is the 9th column, which refers to class (diabetes-non diabetes
that have corresponding values 1,2 respectively)
plot(fit5.c, main = "Complete Linkage", sub = "", xlab = "", labels=new
_df5[,9], cex=.6)
```



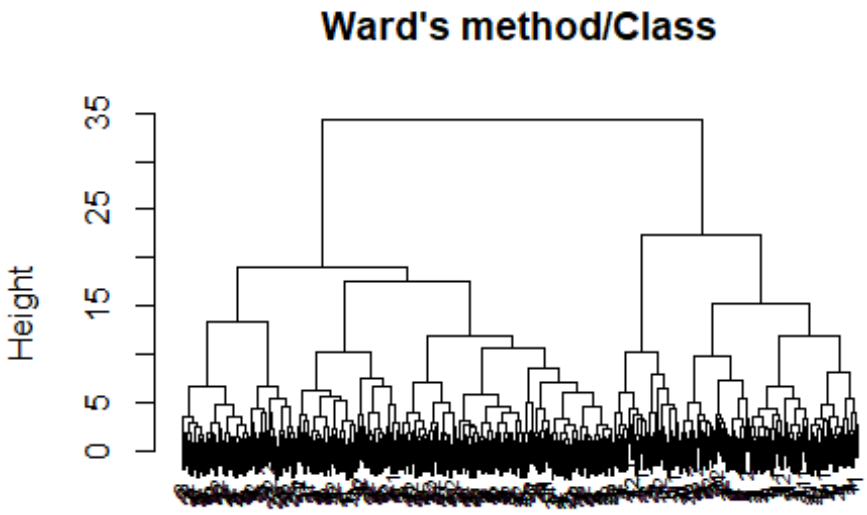

```
plot(fit5.a, main = "Average Linkage", sub = "", xlab = "", labels=new_
df5[,9], cex=.6)
```



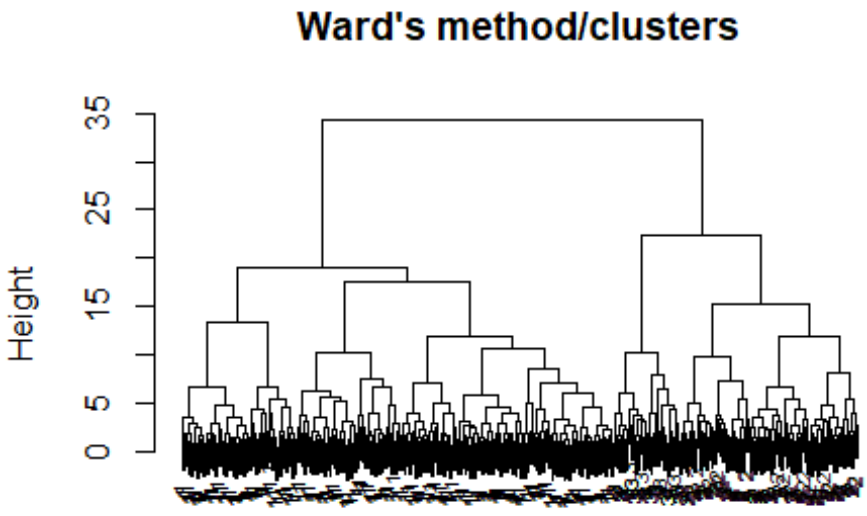
```
plot(fit5.d2, main = "Ward's method", sub = "", xlab = "", labels=new_d
f5[,9], cex=.6)
# We should be aware of the fact that Ward's linkage may impose equal s
ize to clusters as in our case
rect.hclust(fit5.d2, 4)
```



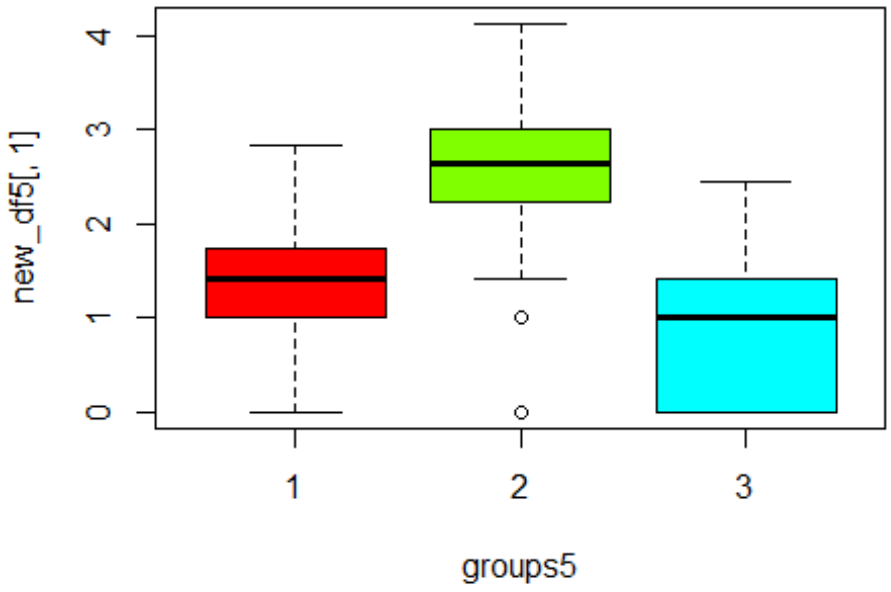

```
# Split a dendrogram in 4 rectangulars


cluster.d5 <- cutree(fit5.d2,3)
# Split the tree in 3 branches
cluster.d5[1:10]

##  4  7  9 14 15 17 19 20 21 25
##  1  1  2  2  2  3  1  1  3  2

# Gives the cluster membership for the first 10 observations of the dat
aset.
plot(fit5.d2, main = "Ward's method/Class", sub = "", xlab = "",labels=
new_df5[,9], cex=.6)
```



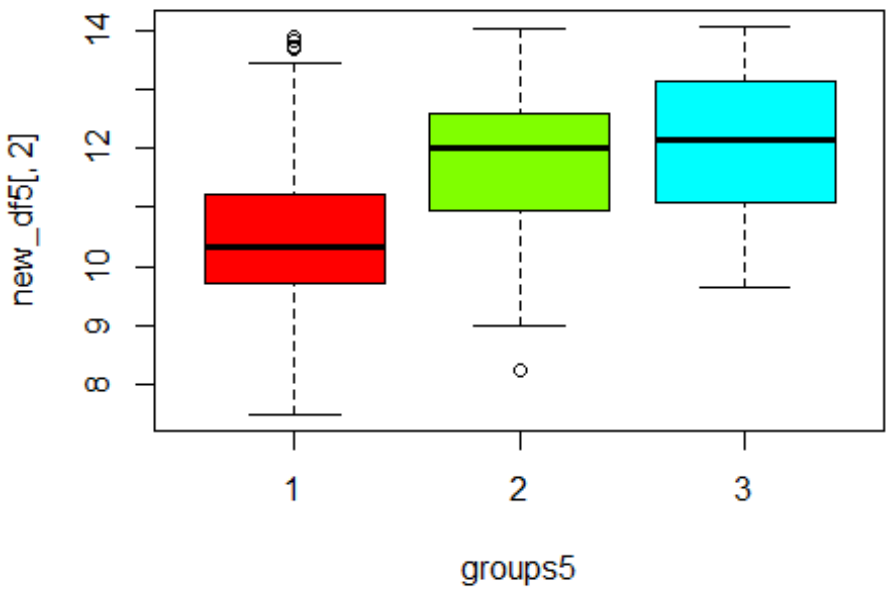

```
# We observe  3 clusters probably (conservative approach)
plot(fit5.d2, main = "Ward's method/clusters", sub = "", xlab = "", lab
els=cluster.d5, cex=.6)
```



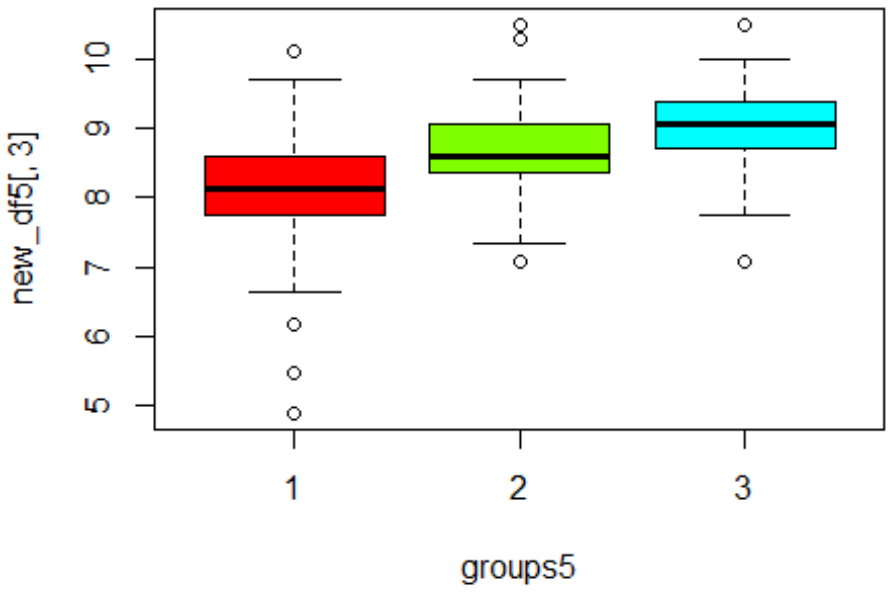
```r
# We observe 3 clusters probably
table(cluster.d5, new_df5[,9])

##
## cluster.d5   1   2
##          1  47 202
##          2  59  44
##          3  22  16

# What is the agreement of the diabetes-non diabetes?? (Reminder: 1 ref
ers to women developing diabetes, 2 refers to non development of diabet
es!)
# Non-diabetes women have been split in 3 groups. 202 in the first clus
ter, 44 in the second cluster, 16 in the third cluster
# Women with diabetes have been split in 3 groups. 47 in the first clus
ter, 59 in the second cluster, 22 in the third cluster
# Contigency table!

# Let us check if the clusters we created make sense
groups5 <- as.factor(cluster.d5)

# Let us observe their common characteristics with respect to the varia
bles 1:8
boxplot(new_df5[,1]~groups5, col=rainbow(4))
```



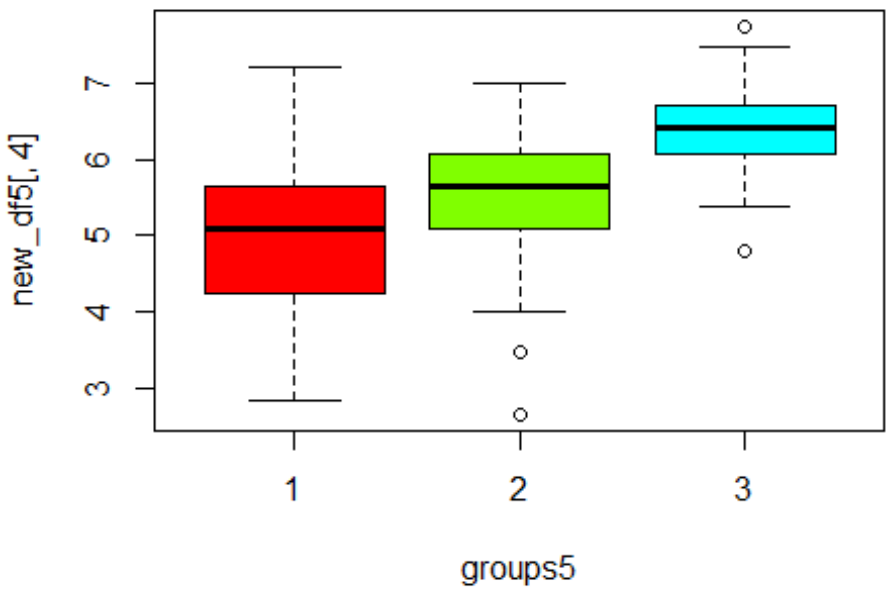

```r
boxplot(new_df5[,2]~groups5, col=rainbow(4))
```



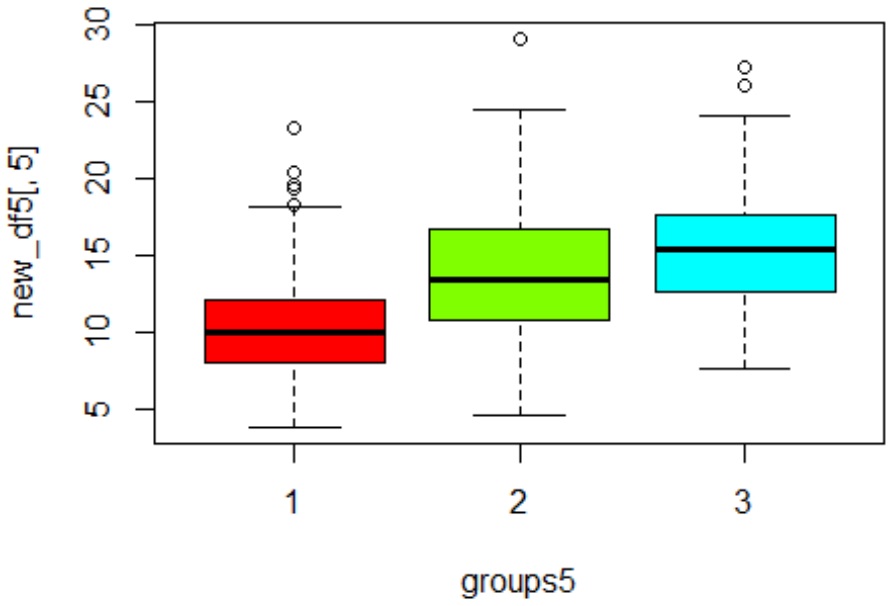

```
boxplot(new_df5[,3]~groups5, col=rainbow(4))
```



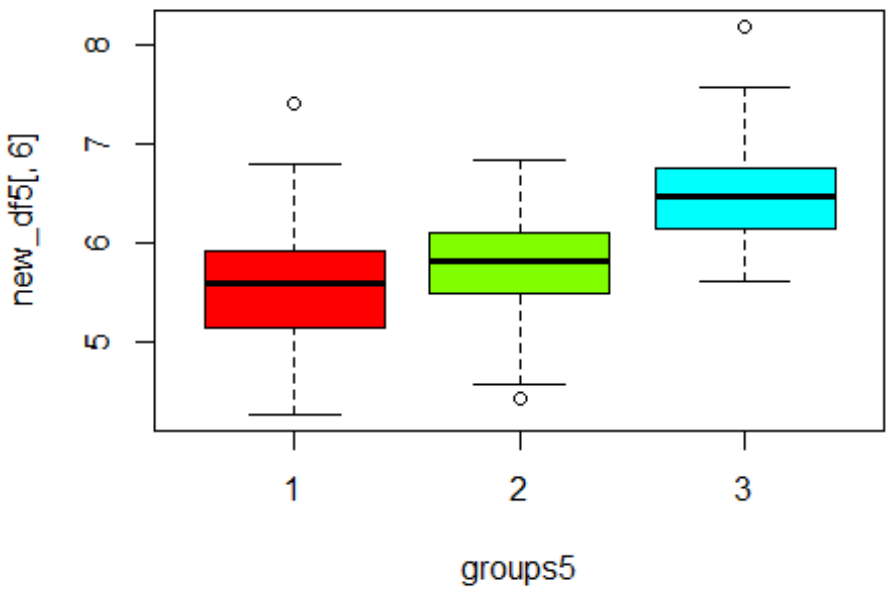

```
boxplot(new_df5[,4]~groups5, col=rainbow(4))
```



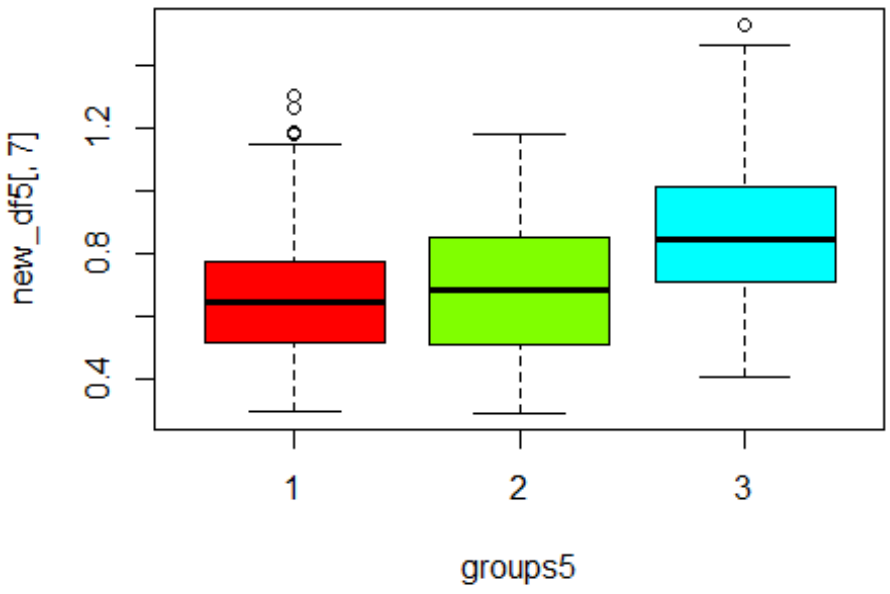

```
boxplot(new_df5[,5]~groups5, col=rainbow(4))
```



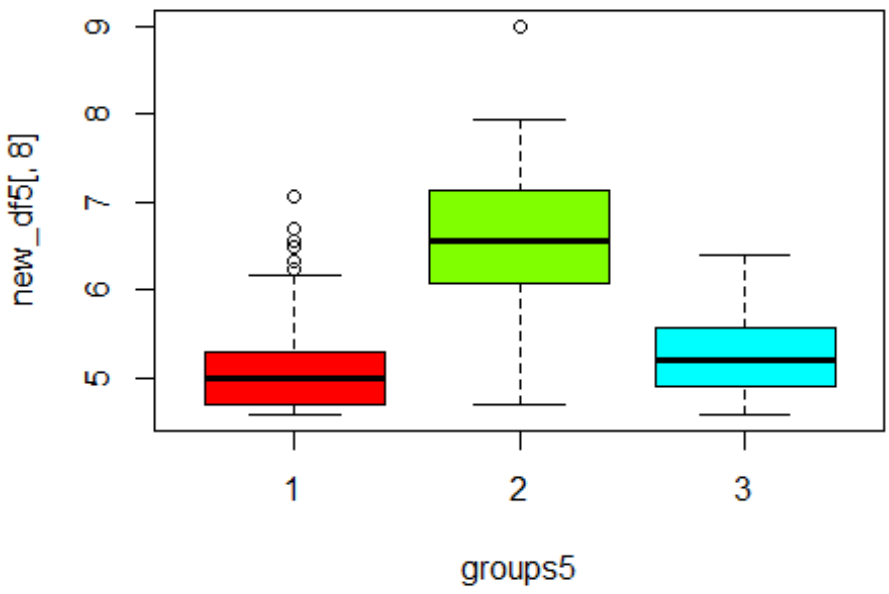

```
boxplot(new_df5[,6]~groups5, col=rainbow(4))
```




```
boxplot(new_df5[,7]~groups5, col=rainbow(4))
```




```
boxplot(new_df5[,8]~groups5, col=rainbow(4))
```

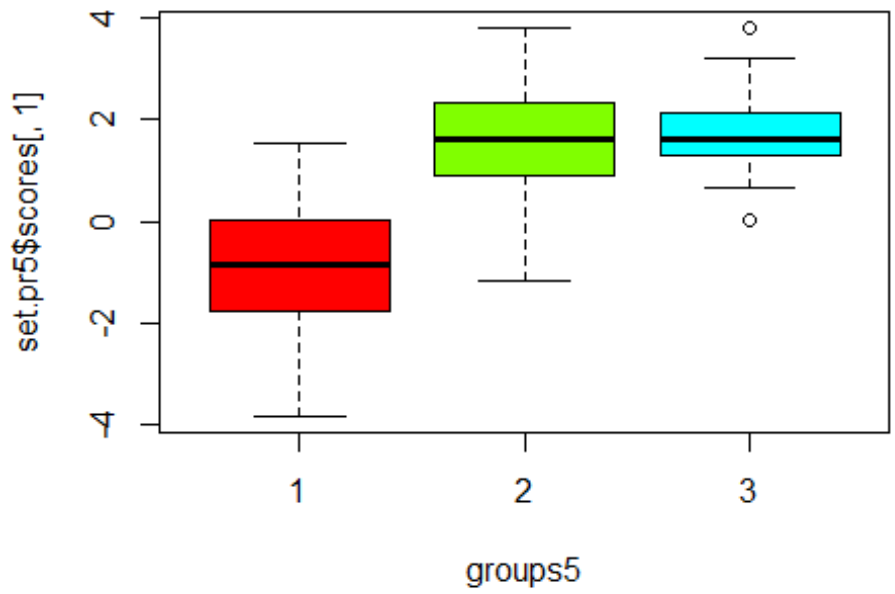



```
# On basis of the original variables we observe that the groups do not
differ!
```

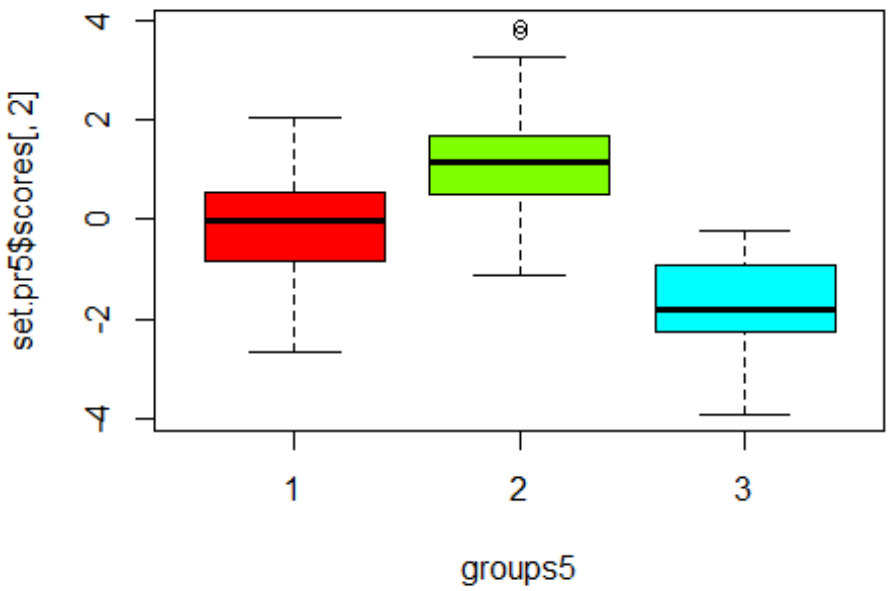
PCA on the new dataset now

```
set.pr5 <- princomp(scale(new_df5[, -9]))
# Exclude the labels, which are in the 9th column
summary(set.pr5)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     C
omp.5
## Standard deviation     1.6022231 1.2423866 1.0792310 0.9853069 0.876
32289
## Proportion of Variance 0.3217148 0.1934366 0.1459667 0.1216657 0.096
23949
## Cumulative Proportion  0.3217148 0.5151513 0.6611180 0.7827837 0.879
02322
##                            Comp.6     Comp.7    Comp.8
## Standard deviation     0.59832737 0.56086513 0.5410798
## Proportion of Variance 0.04486449 0.03942229 0.0366900
## Cumulative Proportion  0.92388771 0.96331000 1.0000000

# The 1st PC explains the 32.1% of the total variability
# The first 2 PCs explain the 51.4% of the total variability
# The first 3 PCs explain the 66.1% of the total variability
# The first 4 PCs explain the 78.3% of the total variability
```

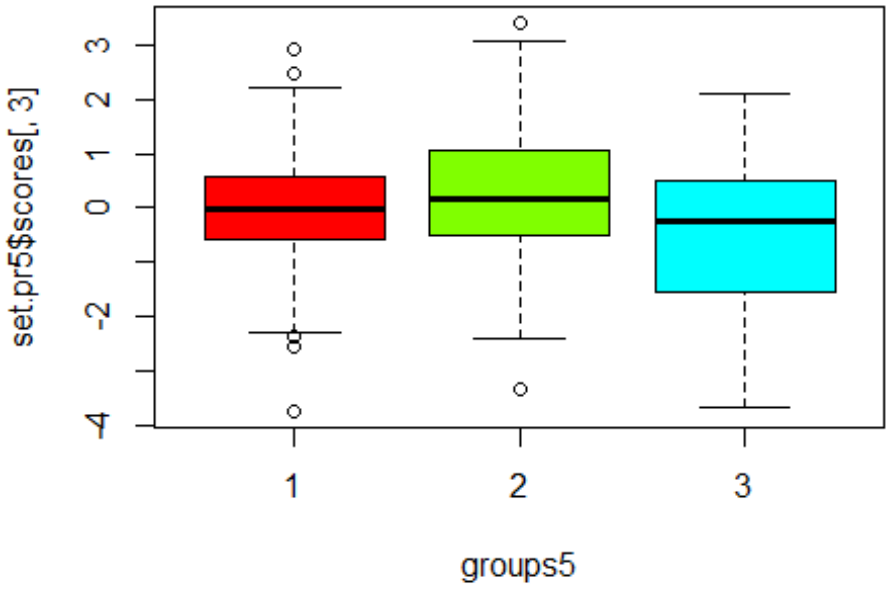
```
# We observe that the standard deviatons with value more than 1, are th
e first 3 ones (the fourth is valued at 0.985
# almost 1,hence we have to take that into consideration!)

boxplot(set.pr5$scores[,1]~groups5, col=rainbow(4))
```



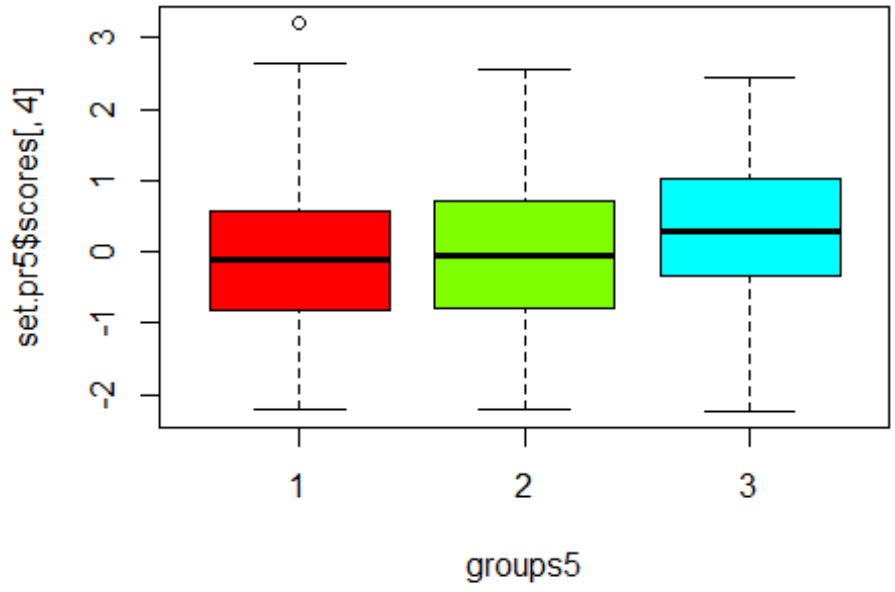

```
# We observe that group 1 differs from  groups 2 and 3
boxplot(set.pr5$scores[,2]~groups5, col=rainbow(4))
```



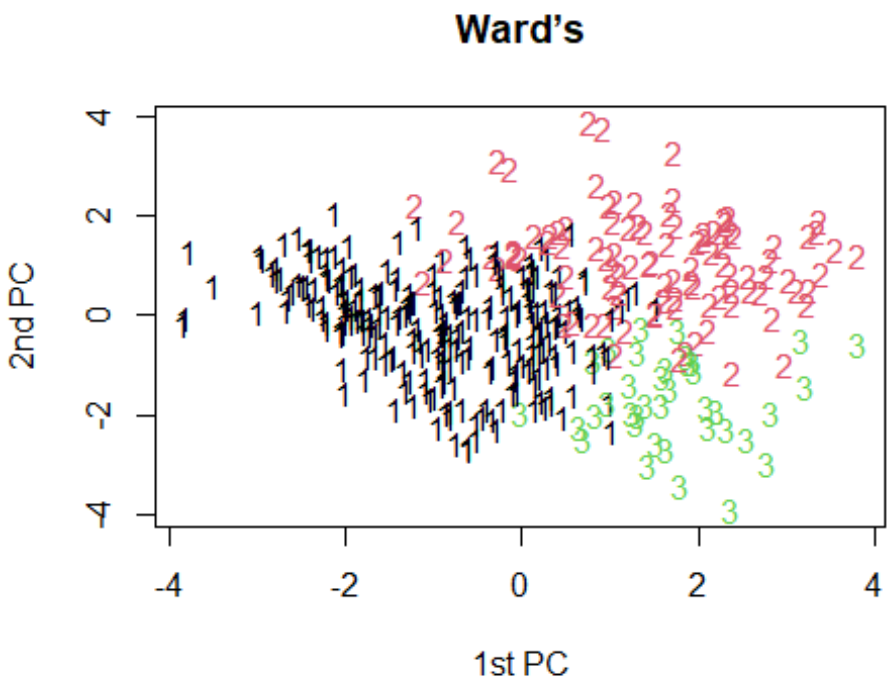

```
# We observe that group 3 differs from the rest
boxplot(set.pr5$scores[,3]~groups5, col=rainbow(4))
```



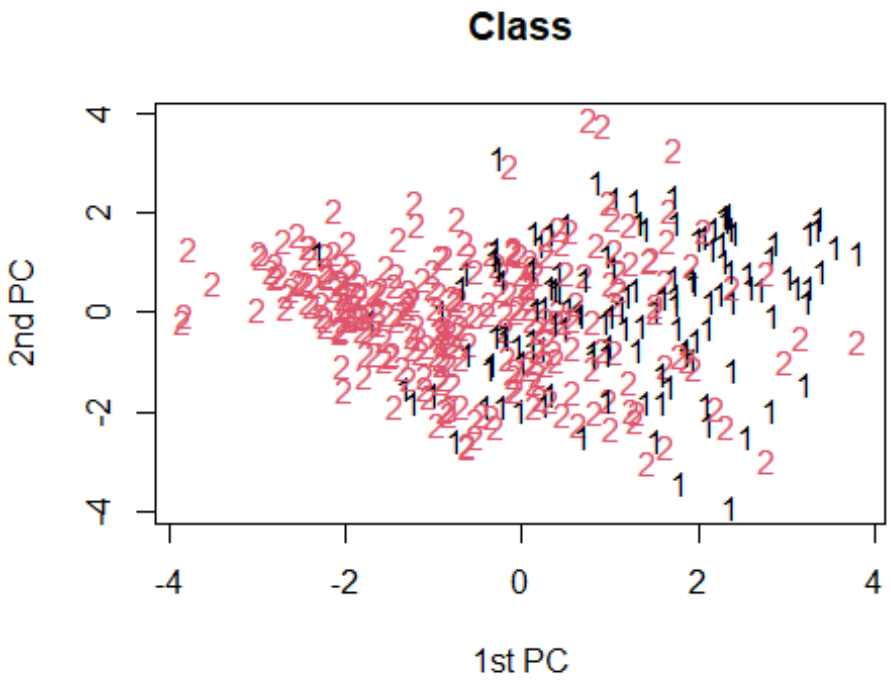
```r
# We observe that groups do not differ in the case of the 3rd PC
boxplot(set.pr5$scores[,4]~groups5, col=rainbow(4))
```



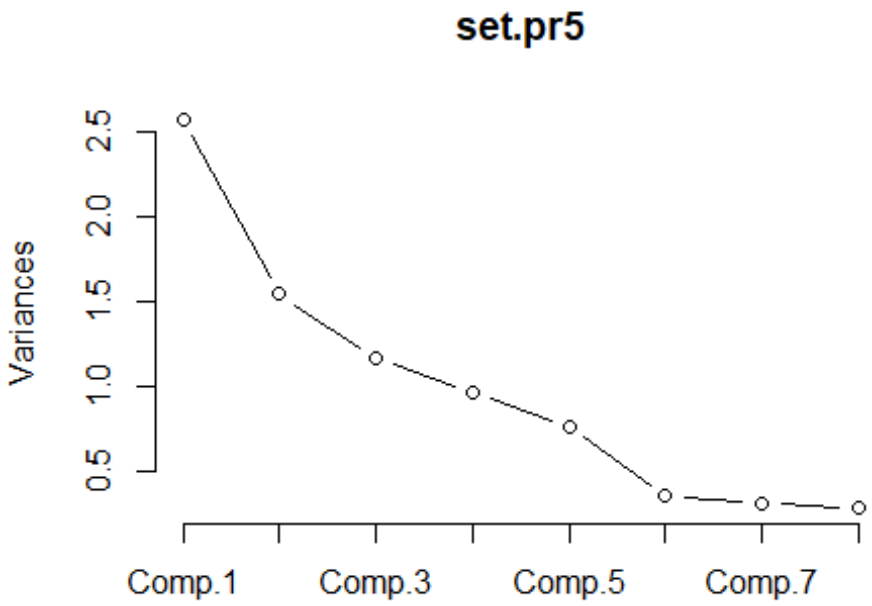

```r
# We observe that groups do not differ in the case of the 4th PC
# The first 2 PCs explain the 51.5% of the total variability


# Visualize the result of a clustering technique using PCs
plot(set.pr5$scores[,1], set.pr5$scores[,2], main ="Ward's", xlab="1st
PC", ylab="2nd PC",  type='n')
text(set.pr5$scores[,1], set.pr5$scores[,2], xlab="1st PC", ylab="2nd P
C", label=cluster.d5, col=cluster.d5)
```



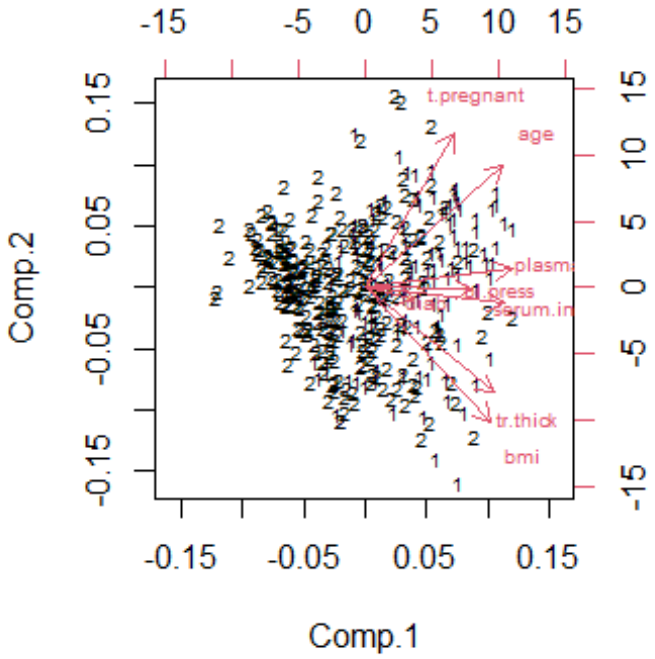

```
# Plot of the 51.5% of the total variability that is expressed by the f
irst 2 PCs
# We observe that the situation is definately improved in comparison wi
th the 4 groups.
# The overlap, is not as much as it was before!
# Removing more outliers is the next step to be taken in order to make
improvements!


plot(set.pr5$scores[,1], set.pr5$scores[,2], main ="Class", xlab="1st P
C", ylab="2nd PC", type='n')
text(set.pr5$scores[,1], set.pr5$scores[,2], xlab="1st PC", ylab="2nd P
C",  label=new_df5[,9], col=new_df5[,9])
```



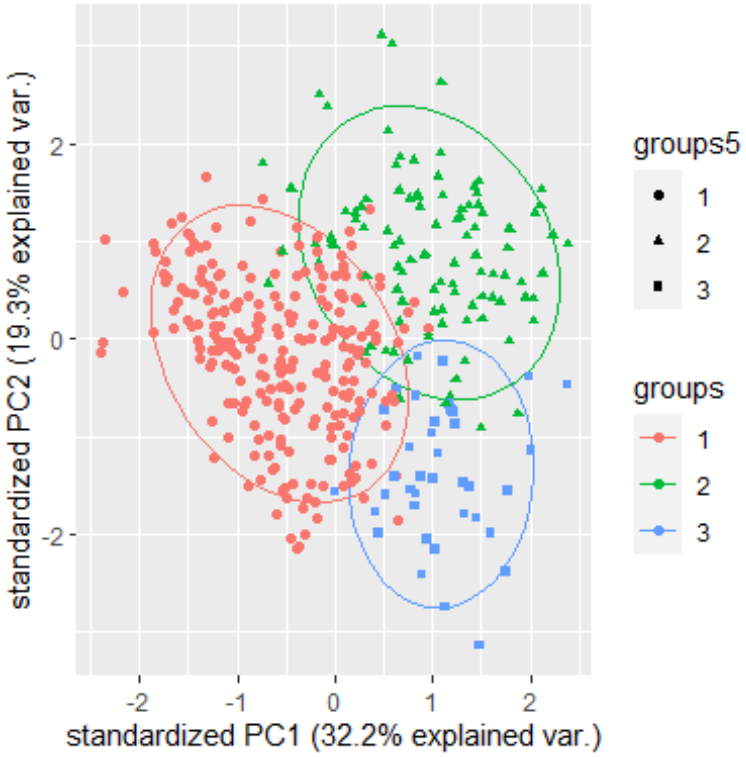

```
# Labels: The class! (1 refers to the women that will develop diabetes,
2 refer to the women that won't develop diabetes)
# We observe that there's not much agreement between the results of War
ds and Class on this case either!


screeplot(set.pr5,type="lines")
```




```
# The screeplot indicates to use the first 2 PC's as the greatest angle
is located between the first and the second component

biplot(set.pr5, choices=c(1,2), xlabs=new_df5[,9], cex=.6)
```




```
# Useful for interpretation! Provides insights about the loadings of ea
ch variable, as well as it aids in detecting outliers!

library(ggbiplot)
g5 <- ggbiplot(set.pr5, choices = c(1,2), pc.biplot = TRUE, groups = as
.factor(cluster.d5), ellipse = TRUE,
               ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, a
lpha=0)
g5 <- g5+geom_point(aes(colour=groups5,shape=groups5),size=1.3)
g5 <- g5+scale_color_discrete(name = 'groups')
g5
```




```
# We observe a better partitioning than before when we had 3 groups! Th
e overlap has been reduced!
# Let us try to make things even better
```

Removing outliers

```
# Dig into the outliers of the dataset new_df5, which is the last one w
e worked with
new_df5[306,]
```

```
##     t.pregnant   plasma bl.press tr.thick serum.ins      bmi     dia
b      age
## 598          1 9.433981 4.898979 4.358899         5 5.272571 0.74766
3 4.582576
##     n.class
## 598       2
```

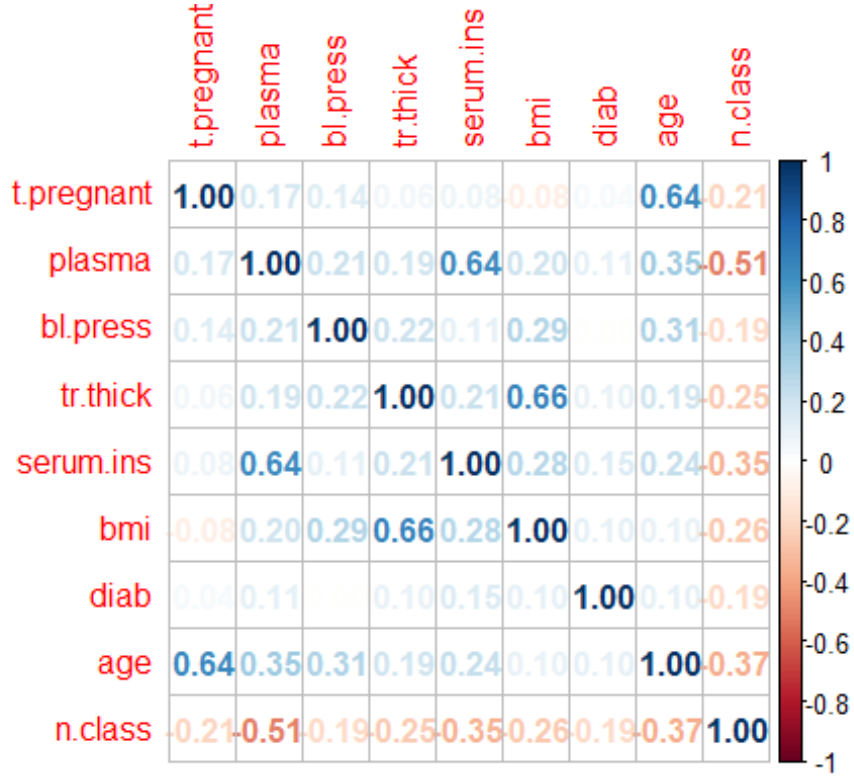
```
# It's line 598 which was indicated before in the single linkage method
as an outlier!
# We shall remove this line and try again the clustering!
new_df6 <- new_df5[-306,]
# Creating the new dataset to work with!
```

Heatmap of the correlations of our final dataset!

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrmatrix <- cor(new_df6)
corrplot(corrmatrix, method = 'number')
```

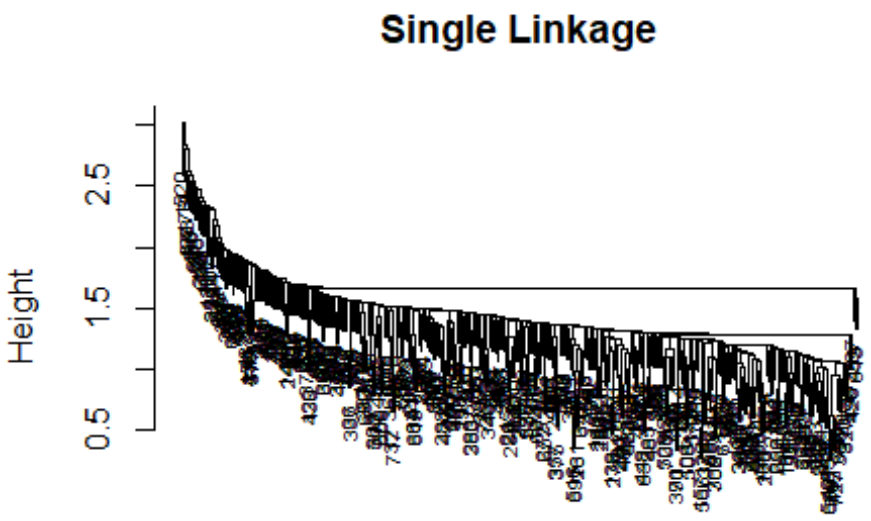


Implementation of Hierarchical clustering using different linkages with proximity measure the euclidian distance on the new dataset

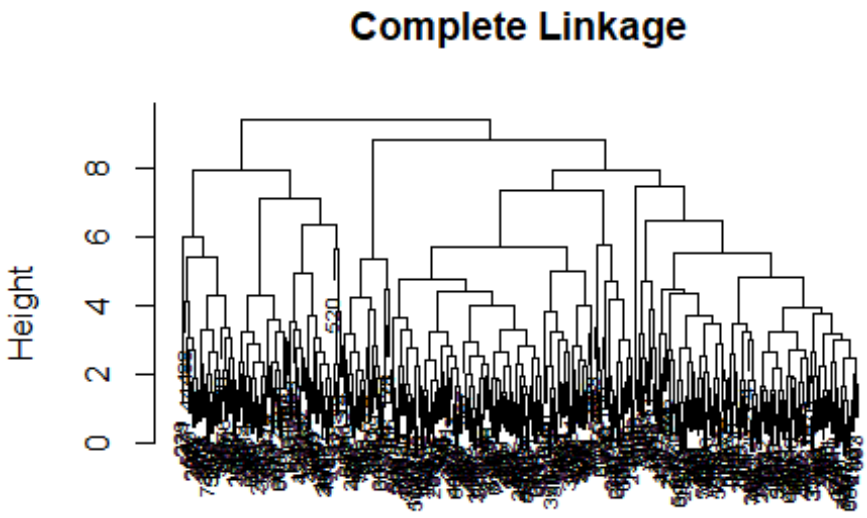
```
d6 <- dist(scale(new_df6[,-9]), method = "euclidian")
# Scaled dissimilarity matrix using as distance the euclidian one
head(d6)

## [1] 2.085105 6.005273 6.806931 4.607321 4.488946 4.643321

# Hierarchical method, different linkage methods
fit6.s <- hclust(d6, method = "single")
fit6.c <- hclust(d6, method = "complete")
fit6.a <- hclust(d6, method = "average")
fit6.d <- hclust(d6, method = "ward.D")
fit6.d2 <- hclust(d6, method = "ward.D2")
```

Let us plot the resulting dendrograms

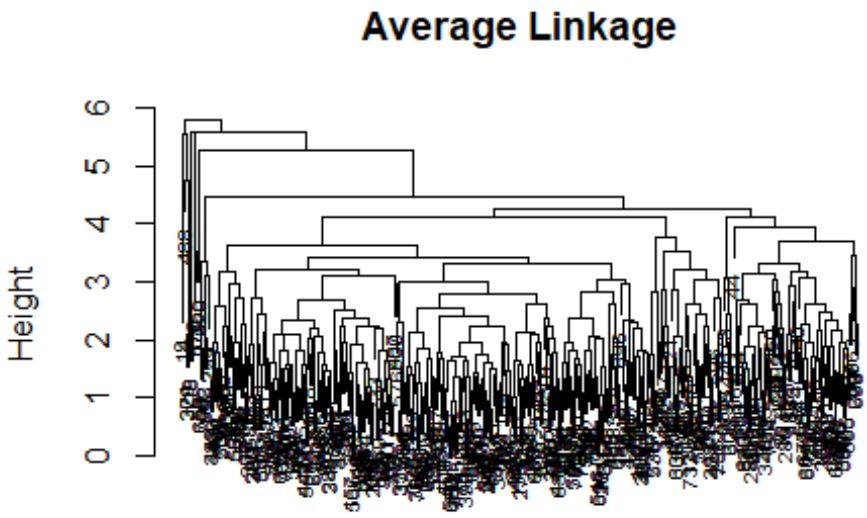
```
plot(fit6.s, main = "Single Linkage", sub = "", xlab = "", cex=.6)
```



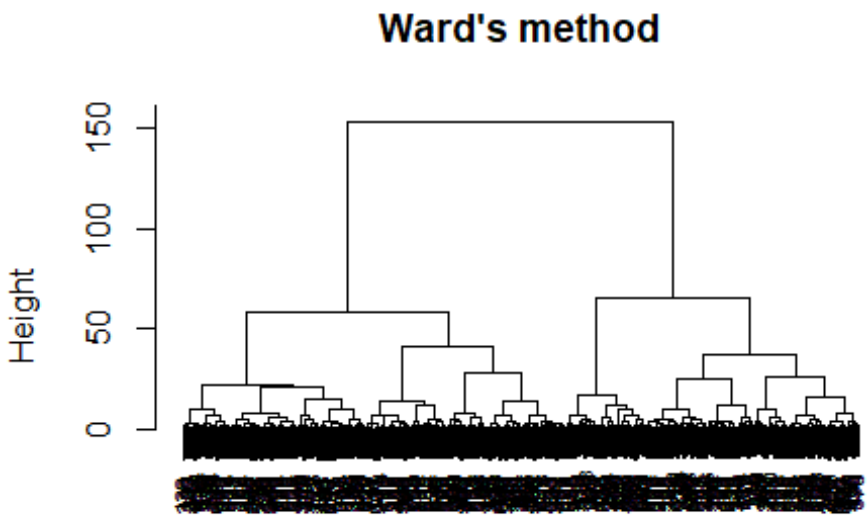

```
# Default label: the increasing number (row number)
# Let us observe the first split! (conservative way to tell how many gr
oups we have!)
# We observe that the single linkage (or nearest neighbor has fallen in
the trap of the chain effect! It's known
# that nearest linkage is prone to the chain effect, however it's usefu
l to identify potential outliers!
plot(fit6.c, main = "Complete Linkage", sub = "", xlab = "", cex=.6)
```



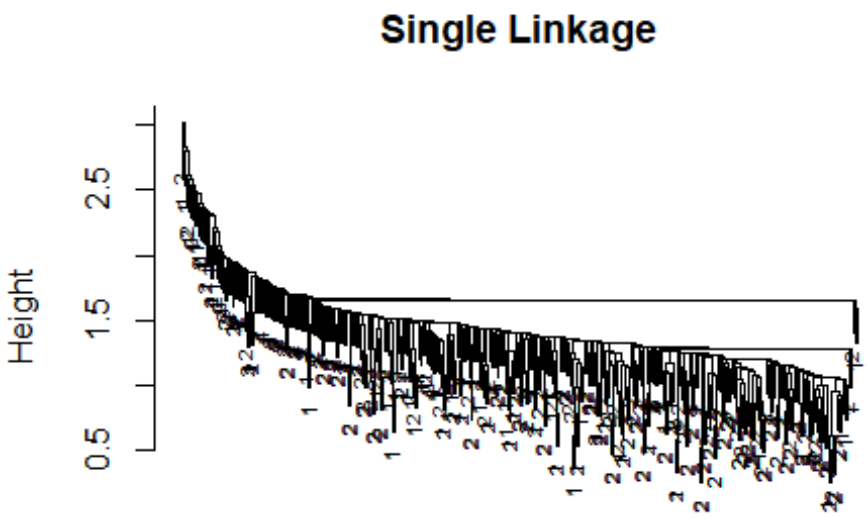

```
# It indicates that there are probably 3 groups (conservative approach)
plot(fit6.a, main = "Average Linkage", sub = "", xlab = "", cex=.6)
```



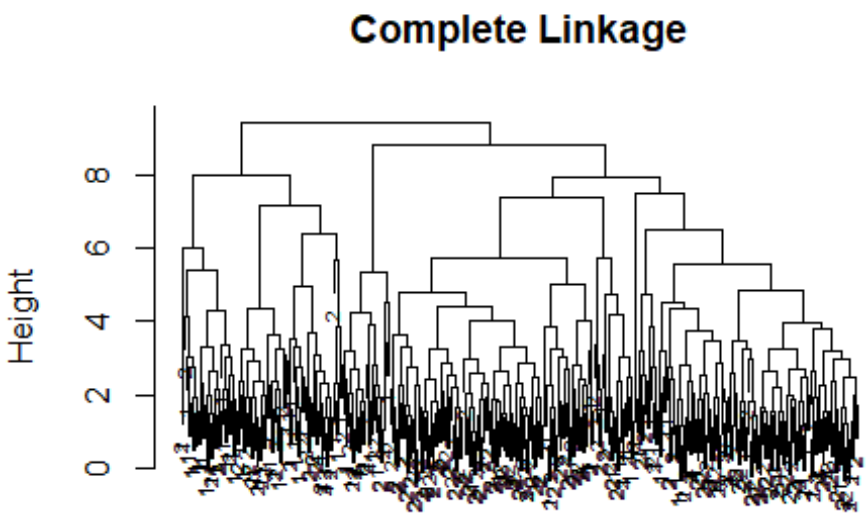
```
# Average linkage seems to be a bit more reasonable in general
plot(fit6.d, main = "Ward's method", sub = "", xlab = "", cex=.6)
```



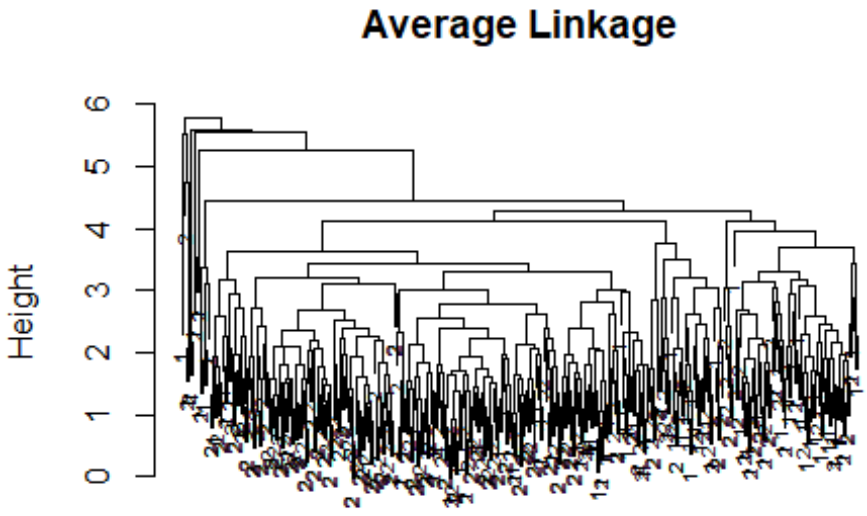

```
# The plot indicates that there are 2 groups probably (conservative app
roach) , maybe 4 as well (needs more exploration!)

plot(fit6.s, main = "Single Linkage", sub = "", xlab = "", labels=new_d
f6[,9], cex=.6)
```



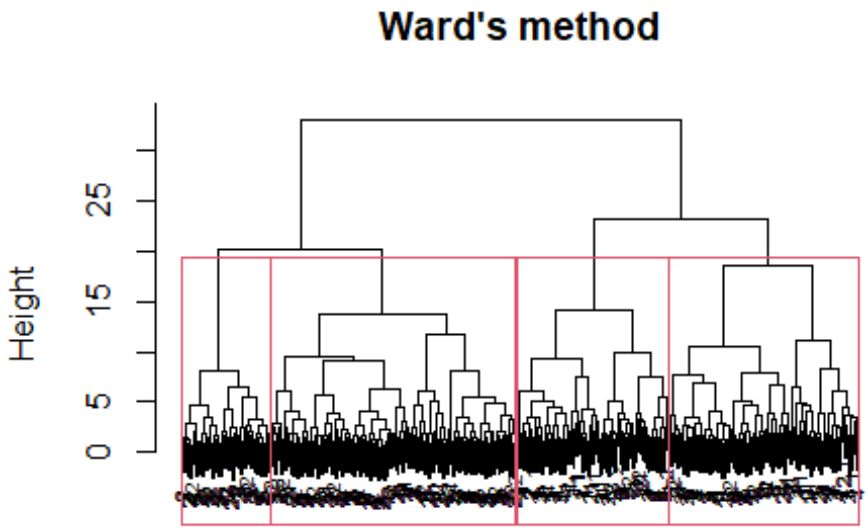

```
# Label is the 9th column, which refers to class (diabetes-non diabetes
that have corresponding values 1,2 respectively)
plot(fit6.c, main = "Complete Linkage", sub = "", xlab = "", labels=new
_df6[,9], cex=.6)
```



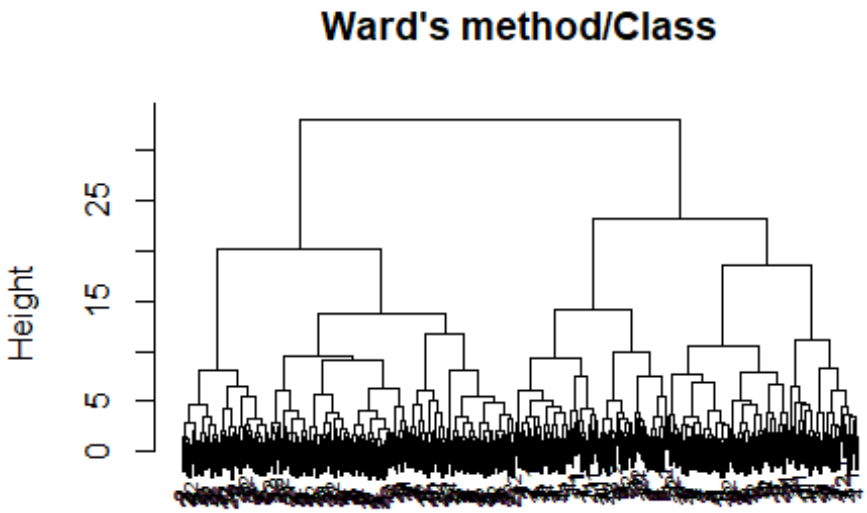

```
plot(fit6.a, main = "Average Linkage", sub = "", xlab = "", labels=new_
df6[,9], cex=.6)
```



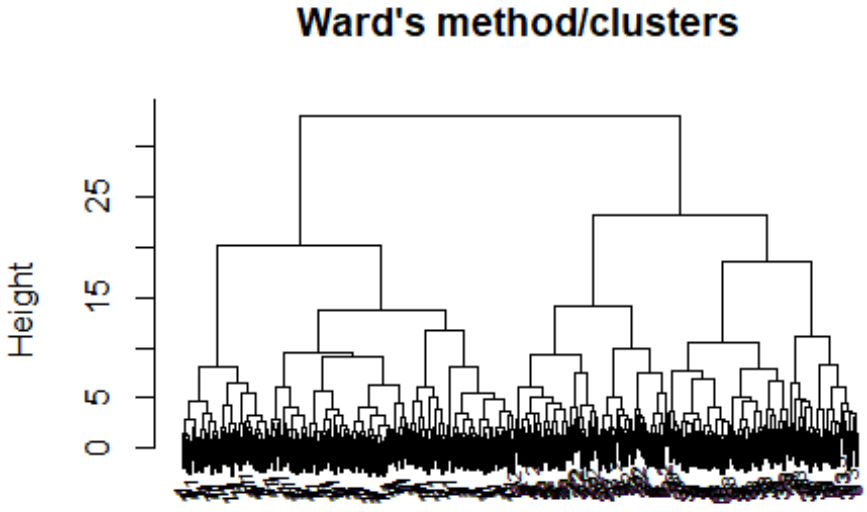
```
plot(fit6.d2, main = "Ward's method", sub = "", xlab = "", labels=new_d
f6[,9], cex=.6)
# We should be aware of the fact that Ward's linkage may impose equal s
ize to clusters as in our case
rect.hclust(fit6.d2, 4)
```



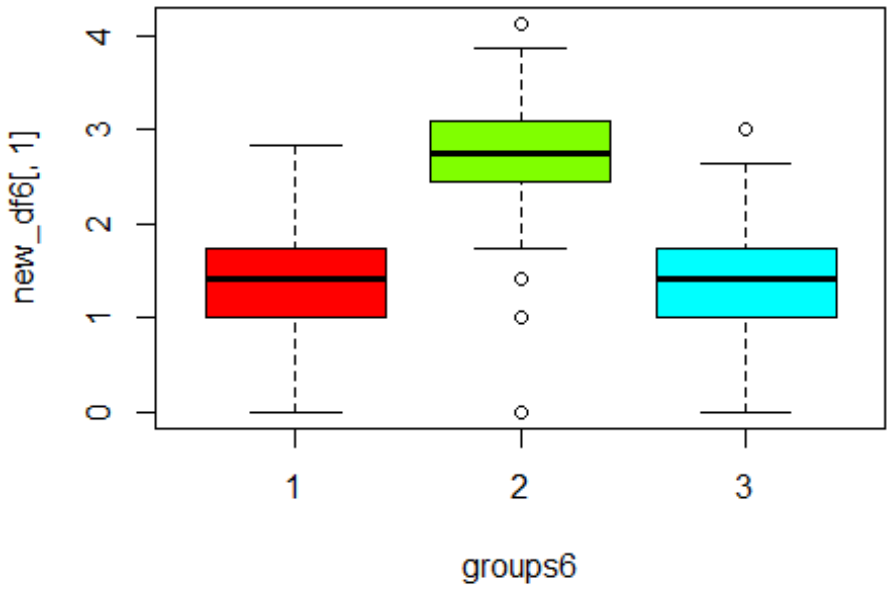

```
# Split a dendrogram in 4 rectangulars

cluster.d6 <- cutree(fit6.d2,3)
# Split the tree in 3 branches
cluster.d6[1:10]

##  4  7  9 14 15 17 19 20 21 25
##  1  1  2  2  2  3  1  1  3  2

# Gives the cluster membership for the first 10 observations of the dat
aset.
par(mfrow=c(1,1))
plot(fit6.d2, main = "Ward's method/Class", sub = "", xlab = "",labels=
new_df6[,9], cex=.6)
```



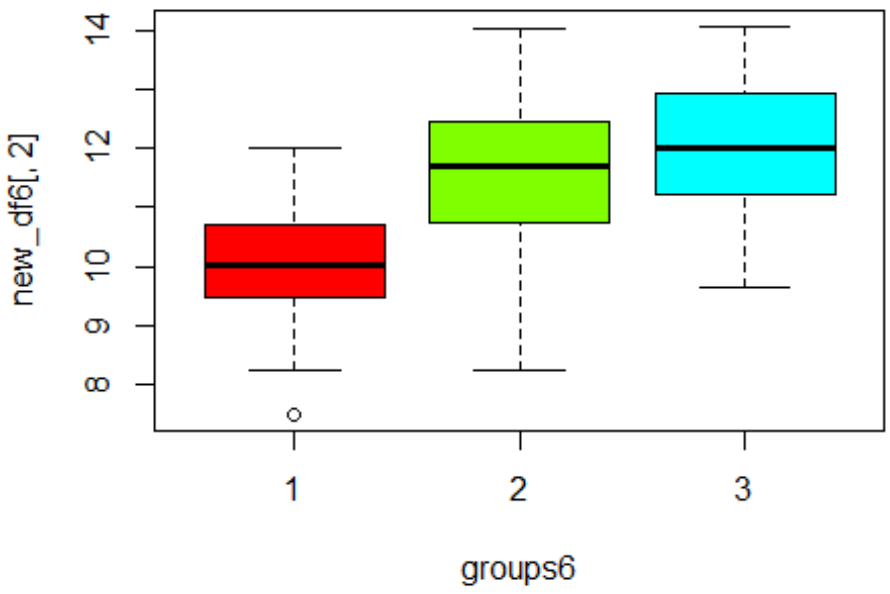

```
# We observe  4 clusters probably (conservative approach)
plot(fit6.d2, main = "Ward's method/clusters", sub = "", xlab = "", lab
els=cluster.d6, cex=.6)
```



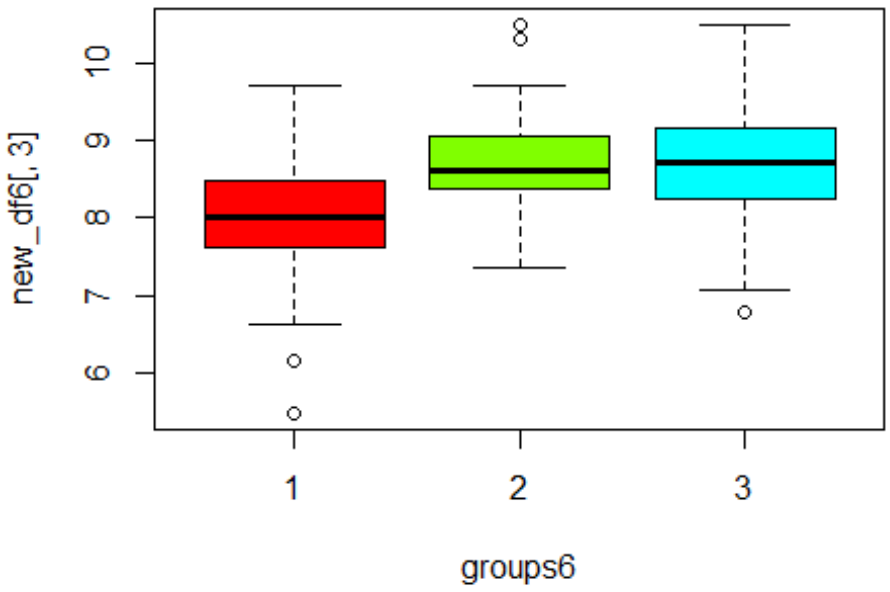
```r
# We observe 4 clusters probably
table(cluster.d6, new_df6[,9])

##
## cluster.d6   1   2
##          1  27 165
##          2  50  38
##          3  51  58

# What is the agreement of the diabetes-non diabetes?? (Reminder: 1 ref
ers to women developing diabetes, 2 refers to non development of diabet
es!)
# Non-diabetes women have been split in 3 groups. 165 in the first clus
ter, 38 in the second cluster, 58 in the third cluster
# Women with diabetes have been split in 3 groups. 27 in the first clus
ter, 50 in the second cluster, 51 in the third cluster
# Contingency table!

# Let us check if the clusters we created make sense
groups6 <- as.factor(cluster.d6)

# Let us observe their common characteristics with respect to the varia
bles 1:8
boxplot(new_df6[,1]~groups6, col=rainbow(4))
```



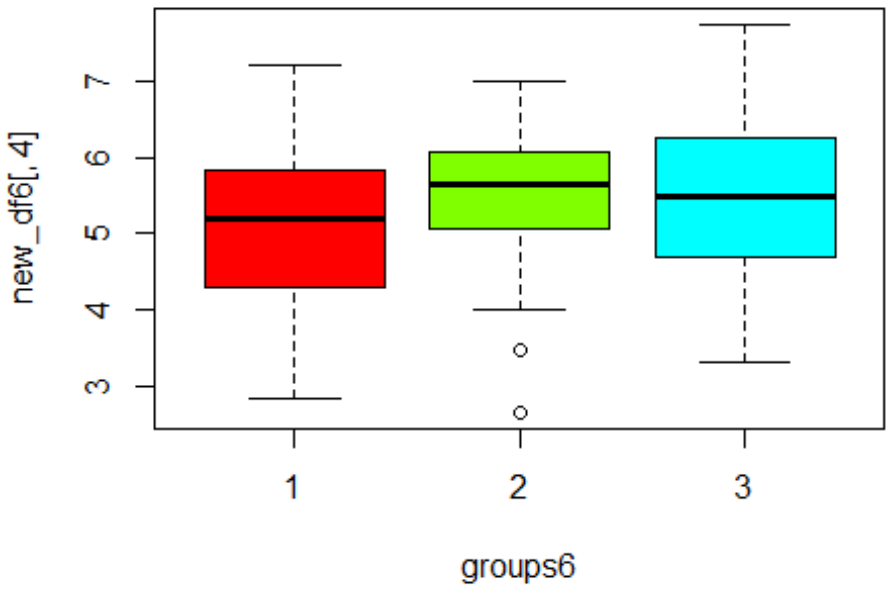

```r
boxplot(new_df6[,2]~groups6, col=rainbow(4))
```



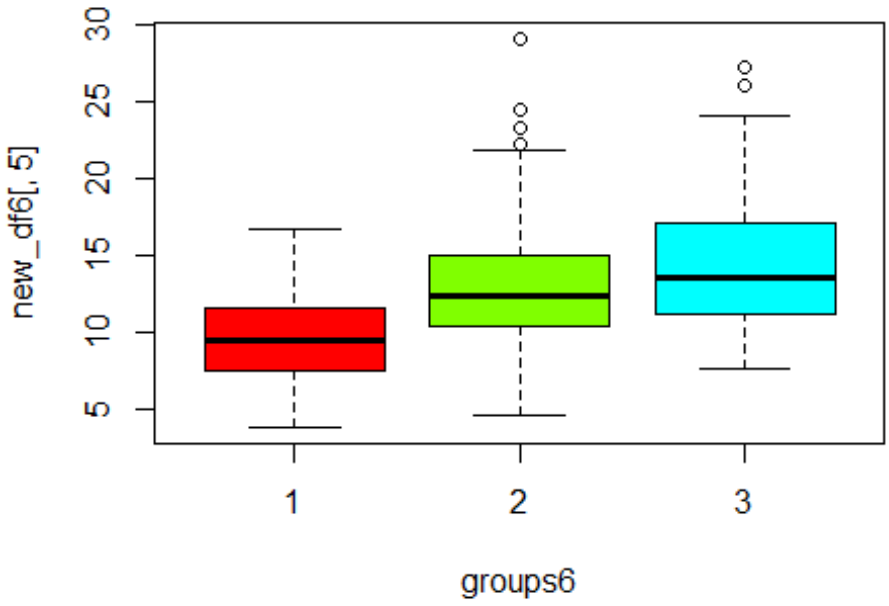

```
boxplot(new_df6[,3]~groups6, col=rainbow(4))
```



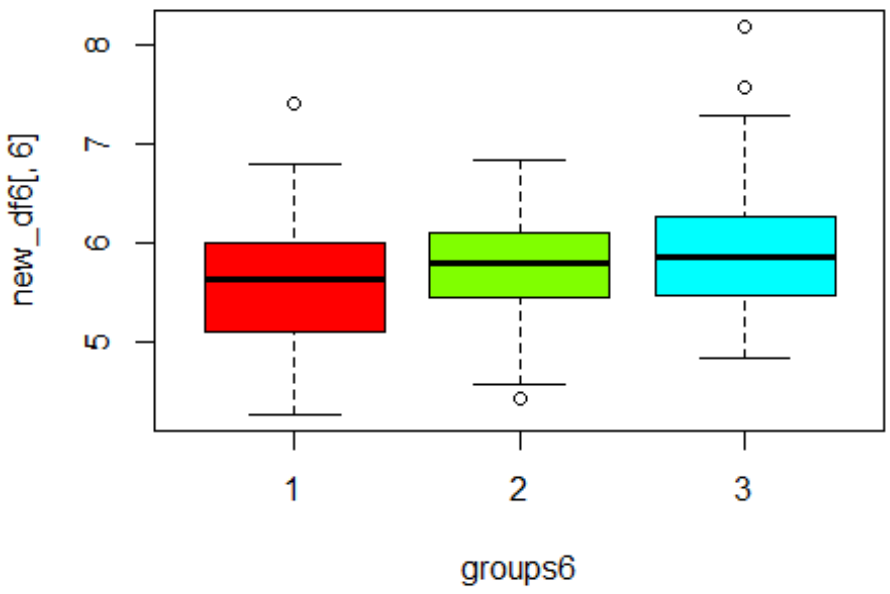

```
boxplot(new_df6[,4]~groups6, col=rainbow(4))
```



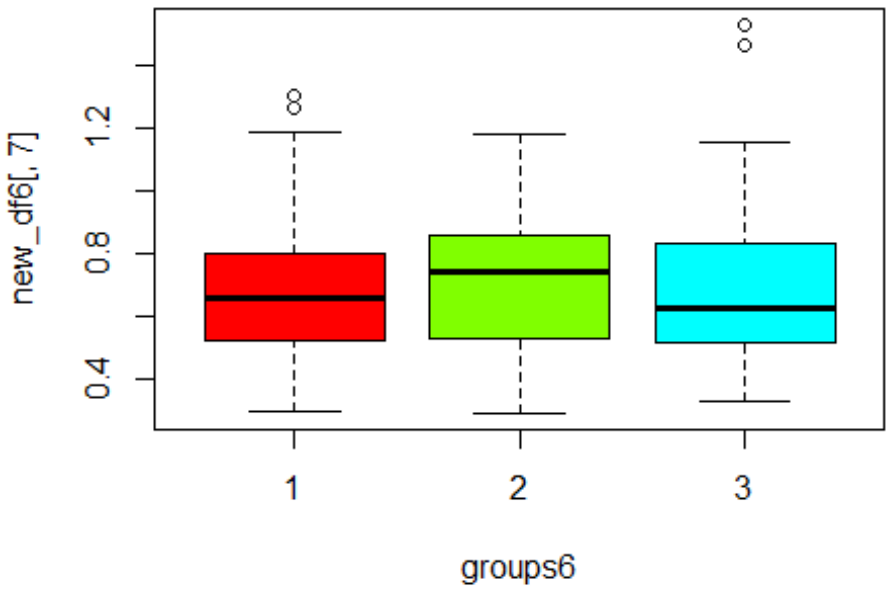

```
boxplot(new_df6[,5]~groups6, col=rainbow(4))
```



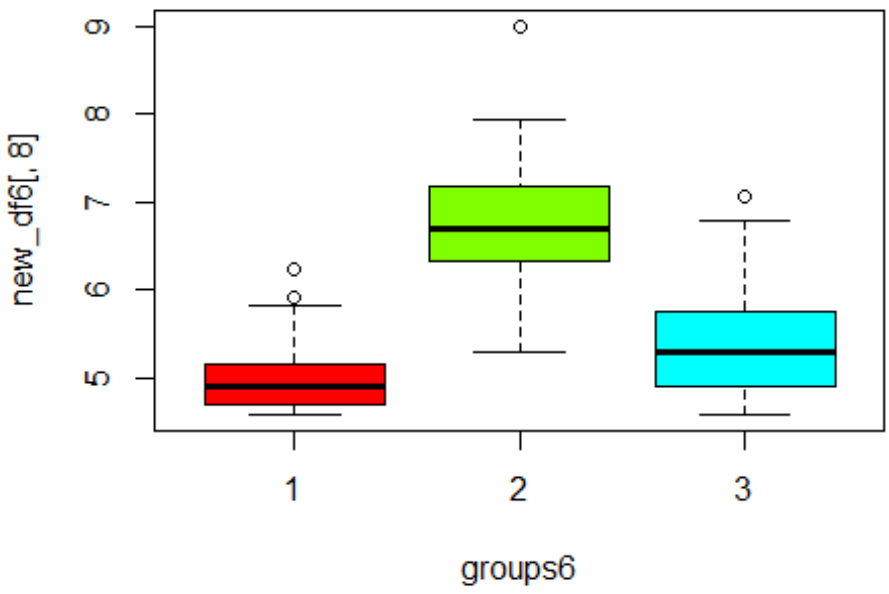

```
boxplot(new_df6[,6]~groups6, col=rainbow(4))
```




```
boxplot(new_df6[,7]~groups6, col=rainbow(4))
```




```
boxplot(new_df6[,8]~groups6, col=rainbow(4))
```

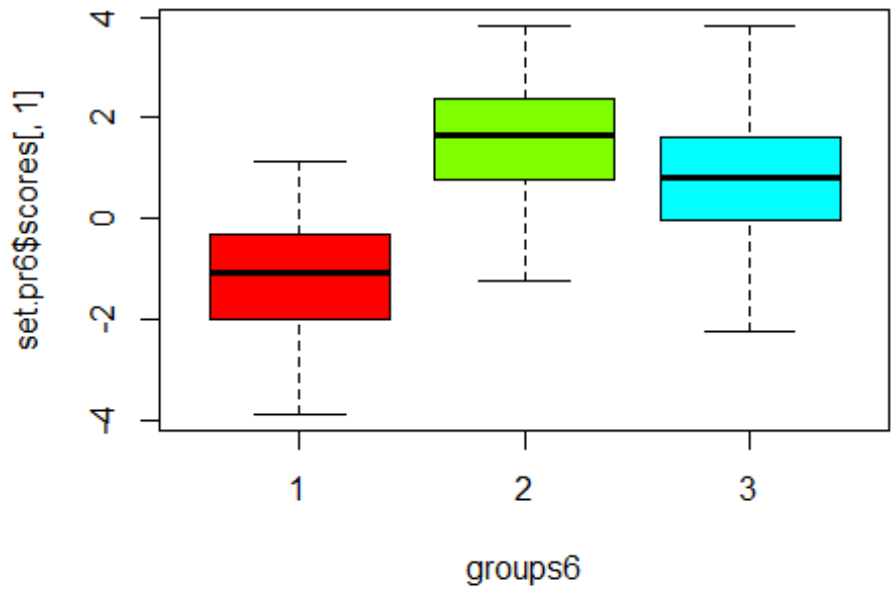



```
# On basis of the original variables we observe that the groups do not
differ!
```

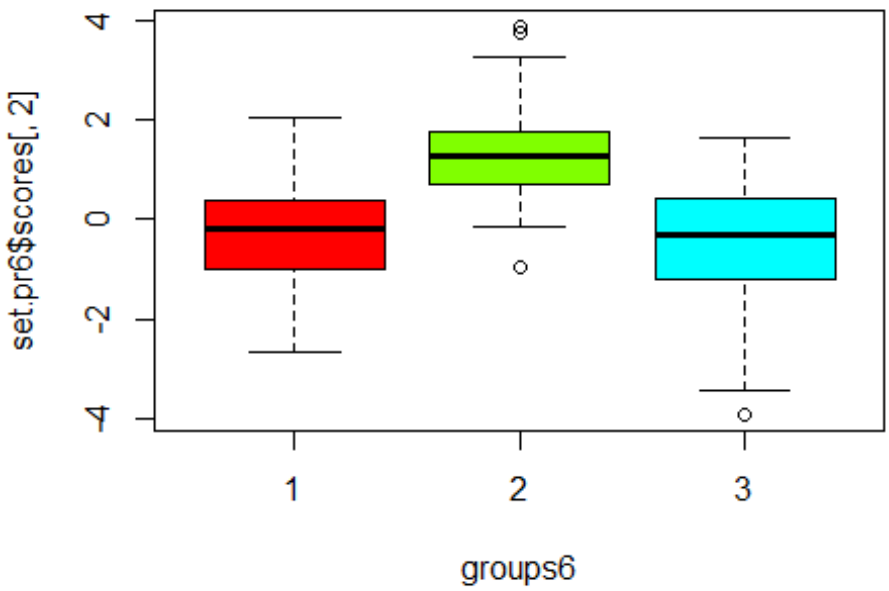
PCA on the final dataset now

```
# Creating PCs once again
set.pr6 <- princomp(scale(new_df6[, -9]))
# Exclude the labels, which are in the 9th column
summary(set.pr6)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     C
omp.5
## Standard deviation     1.5972715 1.2434640 1.0836023 0.9836866 0.880
15023
## Proportion of Variance 0.3197315 0.1937735 0.1471525 0.1212666 0.097
08262
## Cumulative Proportion  0.3197315 0.5135050 0.6606575 0.7819241 0.879
00675
##                           Comp.6     Comp.7    Comp.8
## Standard deviation     0.5989686 0.56075274 0.5406022
## Proportion of Variance 0.0449610 0.03940676 0.0366255
## Cumulative Proportion  0.9239677 0.96337450 1.0000000

# The 1st PC explains the 31.9% of the total variability
# The first 2 PCs explain the 51.4% of the total variability
# The first 3 PCs explain the 66% of the total variability
```

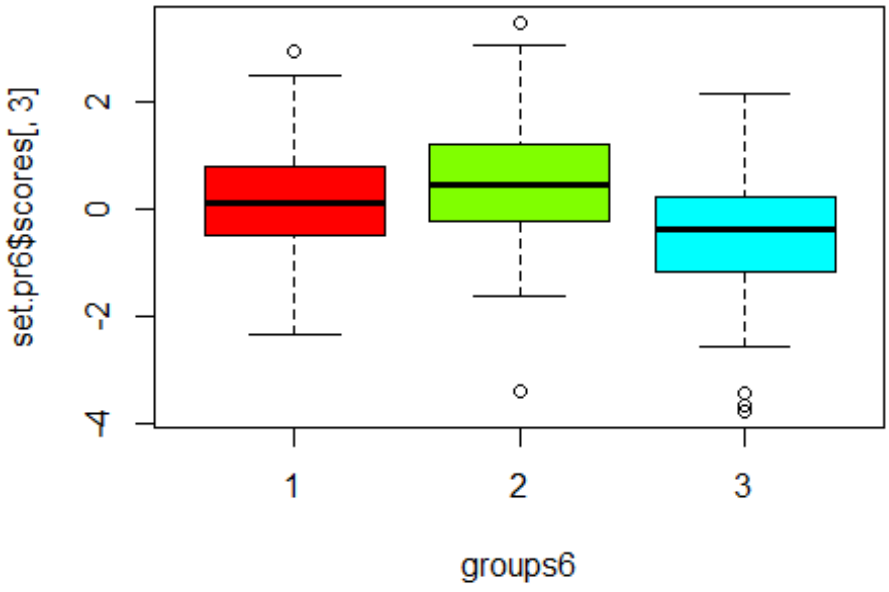
```r
# The first 4 PCs explain the 78.2% of the total variability
# We observe that the standard deviatons with value more than 1, are th
e first 3 ones (the fourth is valued at 0.983
# almost 1,hence we have to take that into consideration!)

boxplot(set.pr6$scores[,1]~groups6, col=rainbow(4))
```



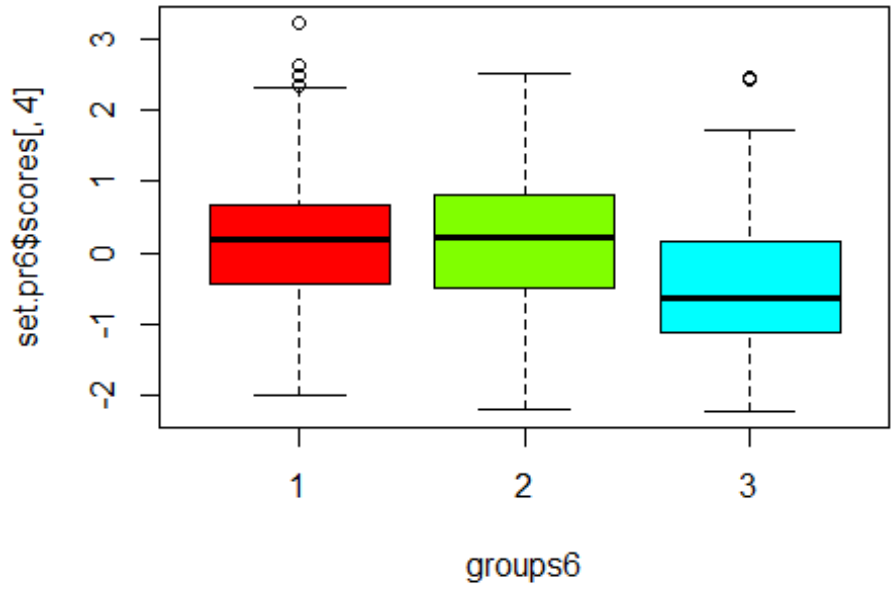

```r
# We observe that group 1 differs from  groups 2 and 3
boxplot(set.pr6$scores[,2]~groups6, col=rainbow(4))
```



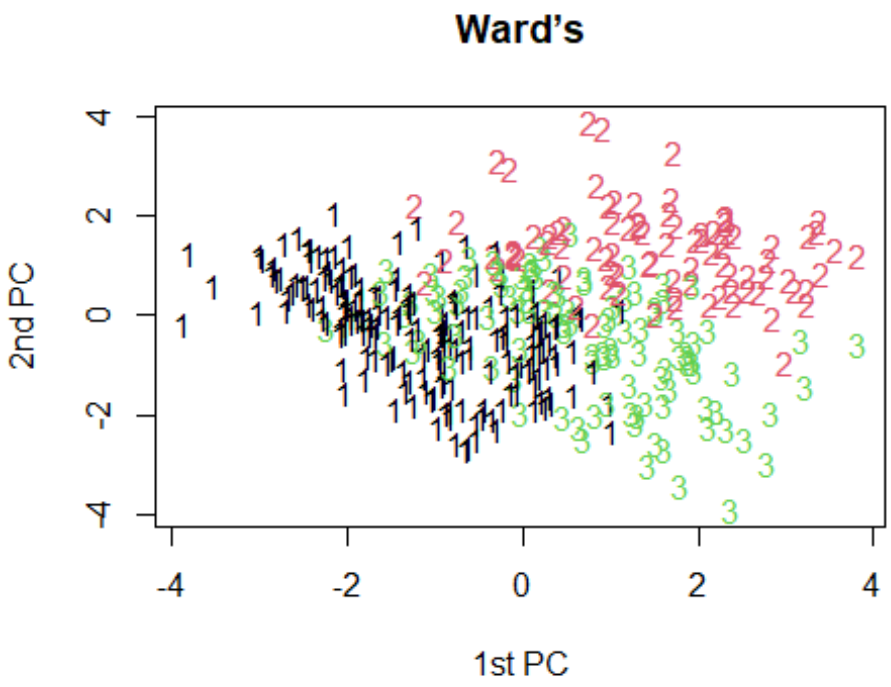

```
# We observe that group 2 differs from the rest
boxplot(set.pr6$scores[,3]~groups6, col=rainbow(4))
```



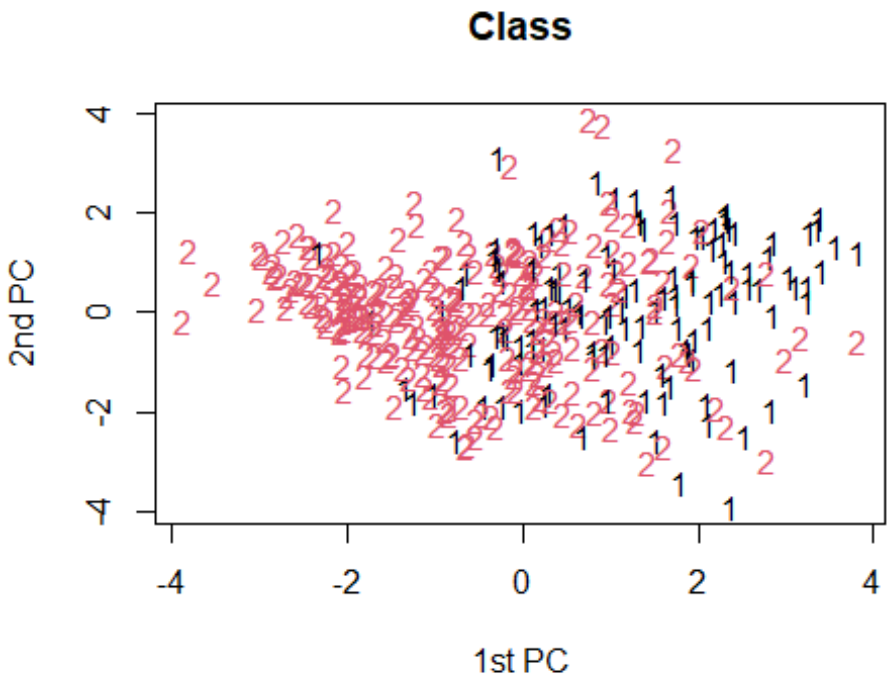
```r
# We observe that groups do not differ in the case of the 3rd PC
boxplot(set.pr6$scores[,4]~groups6, col=rainbow(4))
```



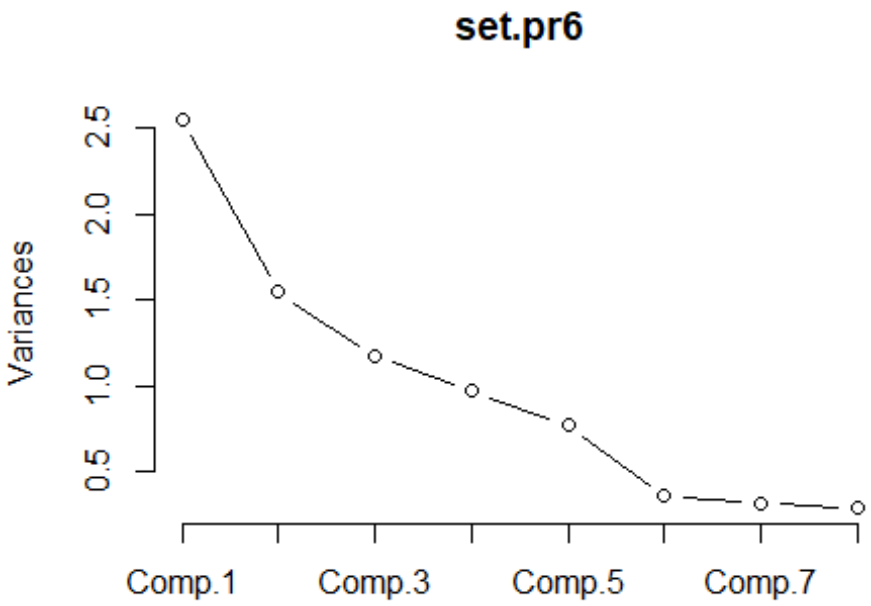

```r
# We observe that groups do not differ in the case of the 4th PC
# The first 2 PCs explain the 51.5% of the total variability


# Visualize the result of a clustering technique using PCs
plot(set.pr6$scores[,1], set.pr6$scores[,2], main ="Ward's", xlab="1st
PC", ylab="2nd PC",  type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], xlab="1st PC", ylab="2nd P
C", label=cluster.d6, col=cluster.d6)
```



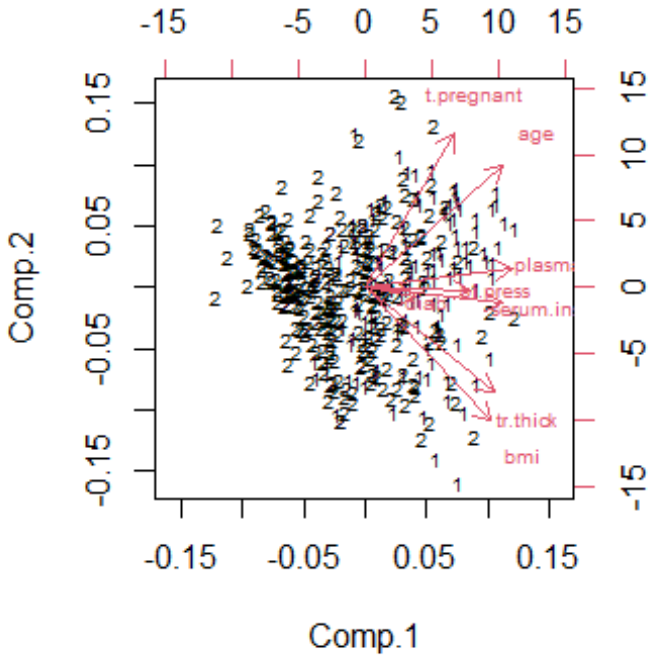

```
# Plot of the 51.3% of the total variability that is expressed by the f
irst 2 PCs


plot(set.pr6$scores[,1], set.pr6$scores[,2], main ="Class", xlab="1st P
C", ylab="2nd PC", type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], xlab="1st PC", ylab="2nd P
C",  label=new_df6[,9], col=new_df6[,9])
```




```
# Labels: The class! (1 refers to the women that will develop diabetes,
2 refer to the women that won't develop diabetes)
# We observe that there's not much agreement between the results of War
ds and Class on this case either!


screeplot(set.pr6,type="lines")
```

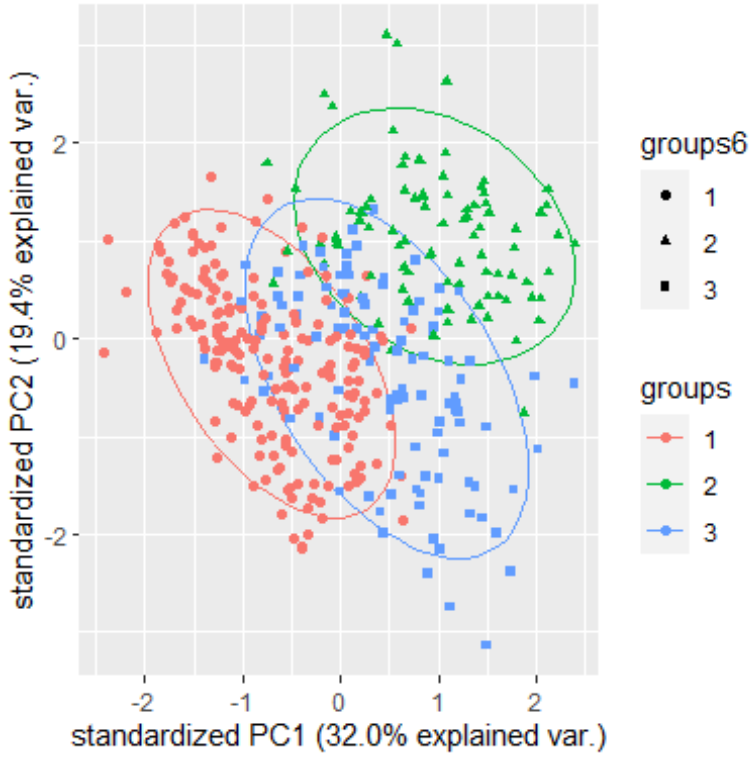



```
# The screeplot indicates to use the first 2 PC's as the greatest angle
is located between the first and the second component

biplot(set.pr6, choices=c(1,2), xlabs=new_df6[,9], cex=.6)
```




```
# Useful for interpretation! Provides insights about the loadings of ea
ch variable, as well as it aids in detecting outliers!

# We observe that variables t.pregnant and age have positive values sho
uld we check their projections on the first PC as well as their project
ions on the second PC.

# We also observe that variables, such as bmi and tr.thick have negativ
e values (we can check that by viewing their corresponding projections
on y'y axis!)
```

```
library(ggbiplot)
g6 <- ggbiplot(set.pr6, choices = c(1,2), pc.biplot = TRUE, groups = as
.factor(cluster.d6), ellipse = TRUE,
ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, alpha=0)
g6 <- g6+geom_point(aes(colour=groups6,shape=groups6),size=1.3)
g6 <- g6+scale_color_discrete(name = 'groups')
g6
```




```
# We observe a worse partitioning than before when we had 3 groups! The
overlap has been inceased so we stick with the previous dataset!
# (before removing row 598)
```

Let us try splitting this (final) dataset in 4 clusters

```
# Contigency table
cluster.d6_v2 <- cutree(fit6.d2,4)
# Split the tree in 4 branches
cluster.d6_v2[1:10]
```

```
##  4  7  9 14 15 17 19 20 21 25
##  1  1  2  2  2  3  1  1  3  2
```

```
# Gives the cluster membership for the first 10 observations of the dat
aset

table(cluster.d6_v2, new_df6[,9])
```

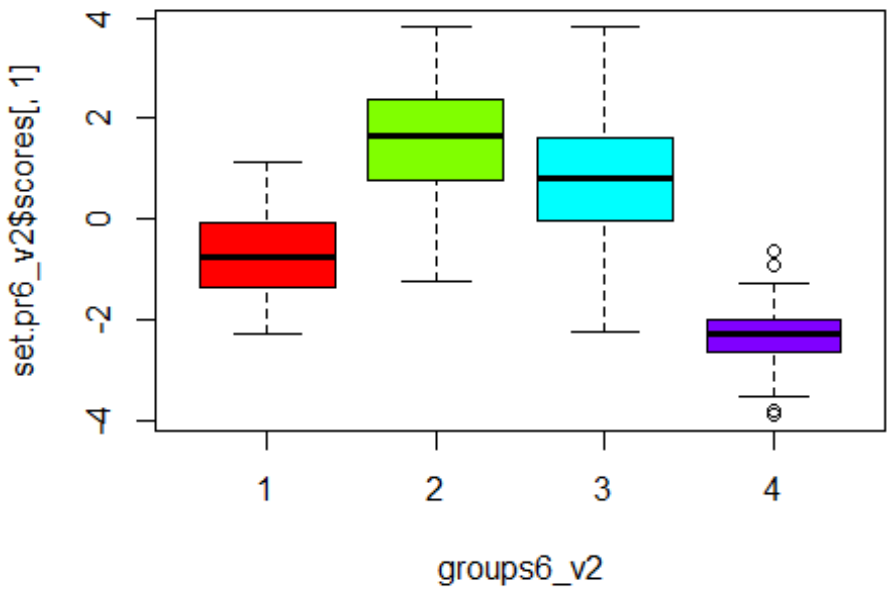
```
##
## cluster.d6_v2   1   2
##             1  26 115
##             2  50  38
##             3  51  58
##             4   1  50
```

```
# What is the agreement of the diabetes-non diabetes?? (Reminder: 1 ref
ers to women developing diabetes, 2 refers to non development of diabet
```

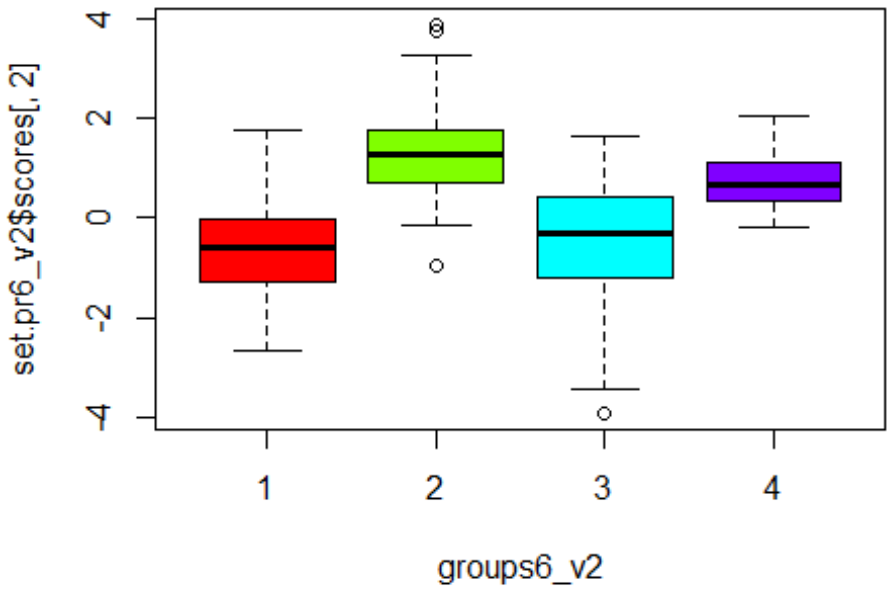
```
es!)
# Non-diabetes women have been split in 3 groups. 165 in the first clus
ter, 38 in the second cluster, 58 in the third cluster
# Women with diabetes have been split in 3 groups. 27 in the first clus
ter, 50 in the second cluster, 51 in the third cluster
groups6_v2 <- as.factor(cluster.d6_v2)
```

Creating PCs once again

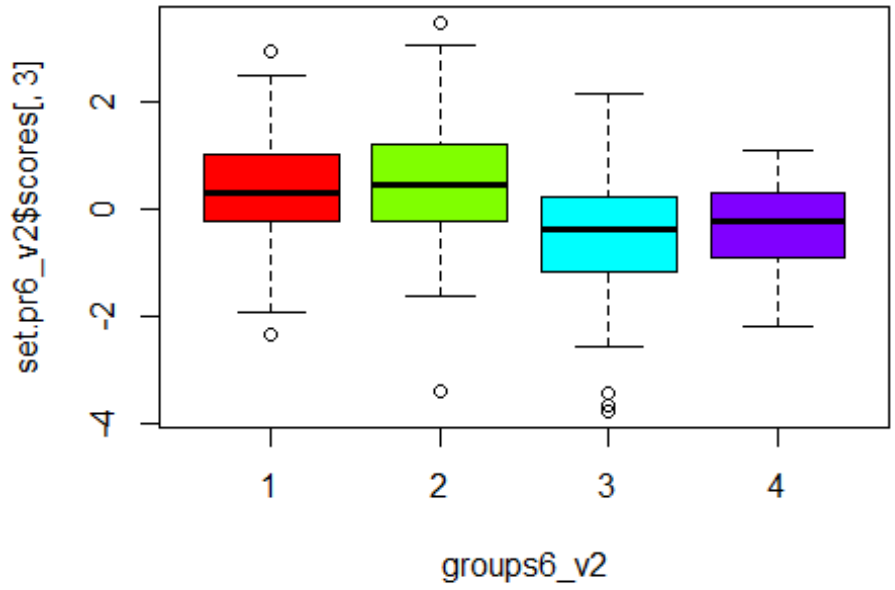
```
set.pr6_v2 <- princomp(scale(new_df6[, -9]))
# Exclude the labels, which are in the 9th column
summary(set.pr6_v2)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     C
omp.5
## Standard deviation     1.5972715 1.2434640 1.0836023 0.9836866 0.880
15023
## Proportion of Variance 0.3197315 0.1937735 0.1471525 0.1212666 0.097
08262
## Cumulative Proportion  0.3197315 0.5135050 0.6606575 0.7819241 0.879
00675
##                           Comp.6     Comp.7    Comp.8
## Standard deviation     0.5989686 0.56075274 0.5406022
## Proportion of Variance 0.0449610 0.03940676 0.0366255
## Cumulative Proportion  0.9239677 0.96337450 1.0000000

# The 1st PC explains the 31.9% of the total variability
# The first 2 PCs explain the 51.4% of the total variability
# The first 3 PCs explain the 66% of the total variability
# The first 4 PCs explain the 78.2% of the total variability
# We observe that the standard deviatons with value more than 1, are th
e first 3 ones (the fourth is valued at 0.983
# almost 1,hence we have to take that into consideration!)

boxplot(set.pr6_v2$scores[,1]~groups6_v2, col=rainbow(4))
```



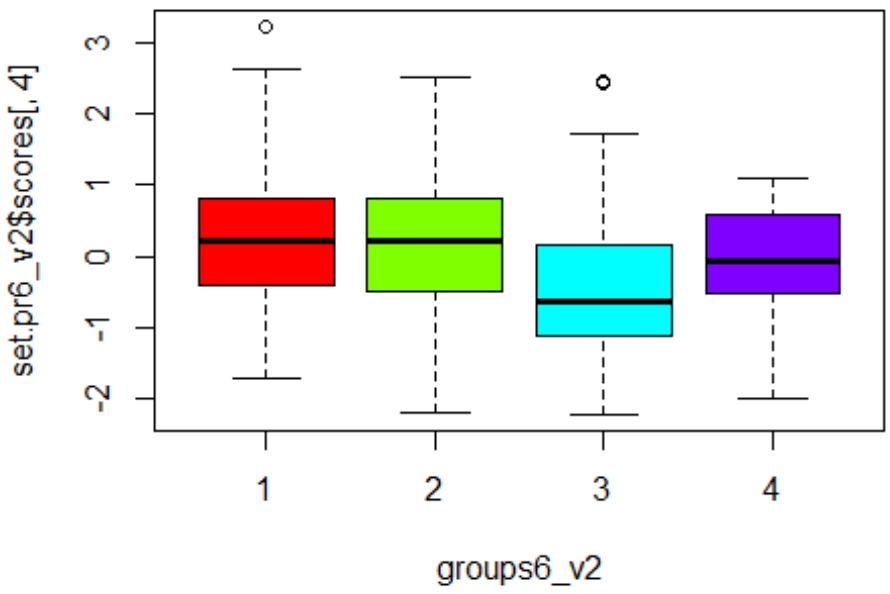

```r
# We observe that group 4 differs from the rest
boxplot(set.pr6_v2$scores[,2]~groups6_v2, col=rainbow(4))
```



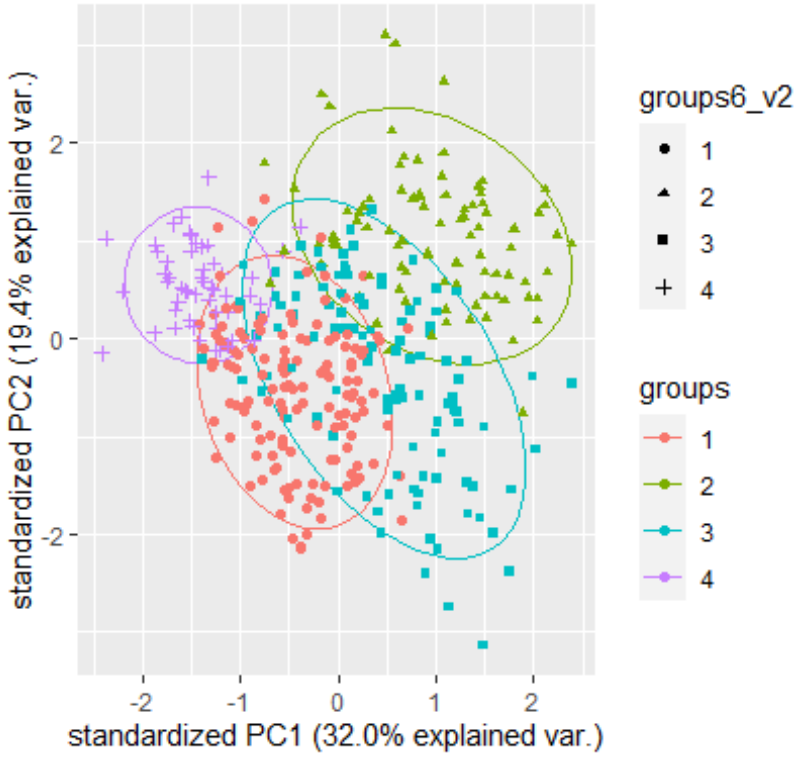
```r
# We observe that group 2 differs from the rest
boxplot(set.pr6_v2$scores[,3]~groups6_v2, col=rainbow(4))
```




```r
# We observe that groups do not differ in the case of the 3rd PC
boxplot(set.pr6_v2$scores[,4]~groups6_v2, col=rainbow(4))
```




```
# We observe that groups do not differ in the case of the 4th PC
# The first 2 PCs explain the 51.3% of the total variability


library(ggbiplot)
g6_v2 <- ggbiplot(set.pr6_v2, choices = c(1,2), pc.biplot = TRUE, group
s = as.factor(cluster.d6_v2), ellipse = TRUE,
ellipse.prob = 0.85, var.axes=FALSE, varname.size = 4, alpha=0)
g6_v2 <- g6_v2+geom_point(aes(colour=groups6_v2,shape=groups6_v2),size=
1.3)
g6_v2 <- g6_v2+scale_color_discrete(name = 'groups')
g6_v2
```




```
# We observe a worse partitioning than before when we had 3 groups! The
overlap has been increased so we stick with the previous dataset!
# (before removing row 598)
```

K Means Clustering

```
# Say 3 clusters is selected
kmeans.3 <- kmeans(scale(new_df6[,-9]), centers = 3, iter.max = 25, tra
ce = TRUE)

## KMNS(*, k=3): iter=  1, indx=0
##  QTRAN(): istep=389, icoun=20
##  QTRAN(): istep=778, icoun=66
##  QTRAN(): istep=1167, icoun=353
## KMNS(*, k=3): iter=  2, indx=389

names(kmeans.3)

## [1] "cluster"      "centers"      "totss"        "withinss"     "tot
.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"

# Parameters that belong to object kmeans.3
table(kmeans.3$cluster)

##
##   1   2   3
## 132 112 145
```

```r
# How many obs belong to each of the 3 clusters
kmeans.3$centers
```

```
##   t.pregnant      plasma   bl.press   tr.thick  serum.ins        bmi
## 1 -0.6106360  0.06652944  0.2228454  0.7270356  0.2043976  0.7874570
## 2  1.0746211  0.66353409  0.3894116  0.2290340  0.4166793  0.1055771
## 3 -0.2741628 -0.57308761 -0.5036530 -0.8387621 -0.5079212 -0.7984066
##          diab        age
## 1  0.05599207 -0.4060297
## 2  0.07131150  1.2544646
## 3 -0.10605408 -0.5993387
```

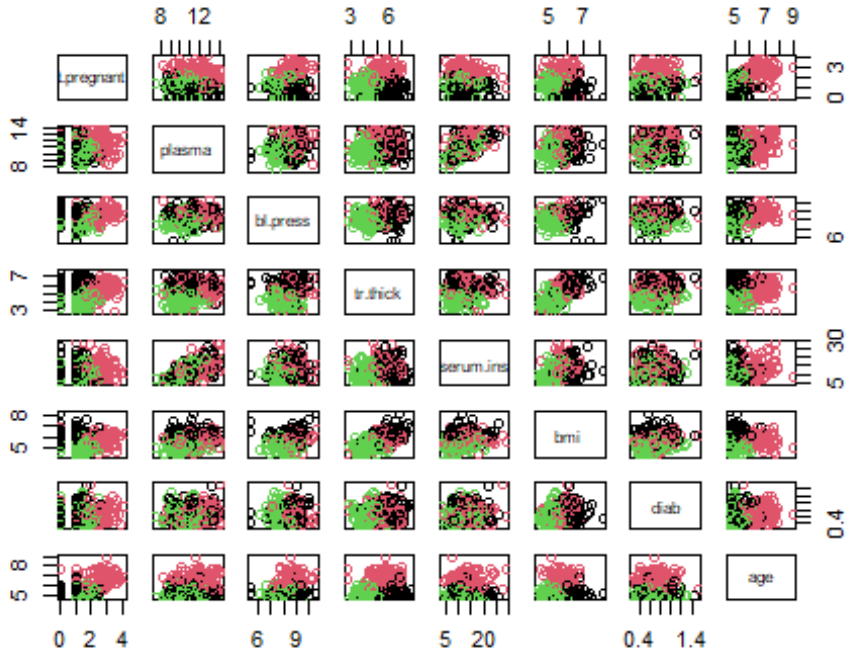
```r
# Mean values for each cluster
kmeans.3$withinss
```

```
## [1] 775.1651 685.1115 628.1573
```

```r
# Sum of squares within groups
kmeans.3$betweenss
```

```
## [1] 1015.566
```

```r
# Sum of square between groups

# Scatterplot for the pairs of the initial variables
pairs(new_df6[,-9], col=c(1:3)[kmeans.3$cluster])
```



```r
# Say 4 clusters is selected
kmeans.4 <- kmeans(scale(new_df6[,-9]), centers = 4, iter.max = 25, tra
ce = TRUE)
```

```
## KMNS(*, k=4): iter=  1, indx=1
##  QTRAN(): istep=389, icoun=13
##  QTRAN(): istep=778, icoun=33
##  QTRAN(): istep=1167, icoun=76
##  QTRAN(): istep=1556, icoun=307
##  QTRAN(): istep=1945, icoun=81
##  QTRAN(): istep=2334, icoun=23
##  QTRAN(): istep=2723, icoun=13
##  QTRAN(): istep=3112, icoun=20
##  QTRAN(): istep=3501, icoun=98
##  QTRAN(): istep=3890, icoun=14
##  QTRAN(): istep=4279, icoun=11
##  QTRAN(): istep=4668, icoun=107
## KMNS(*, k=4): iter=  2, indx=2
##  QTRAN(): istep=389, icoun=25
##  QTRAN(): istep=778, icoun=5
##  QTRAN(): istep=1167, icoun=48
##  QTRAN(): istep=1556, icoun=18
## KMNS(*, k=4): iter=  3, indx=29
##  QTRAN(): istep=389, icoun=14
##  QTRAN(): istep=778, icoun=209
## KMNS(*, k=4): iter=  4, indx=389
```

```r
names(kmeans.4)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot
.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```r
# Parameters that belong to object kmeans.4
table(kmeans.4$cluster)
```

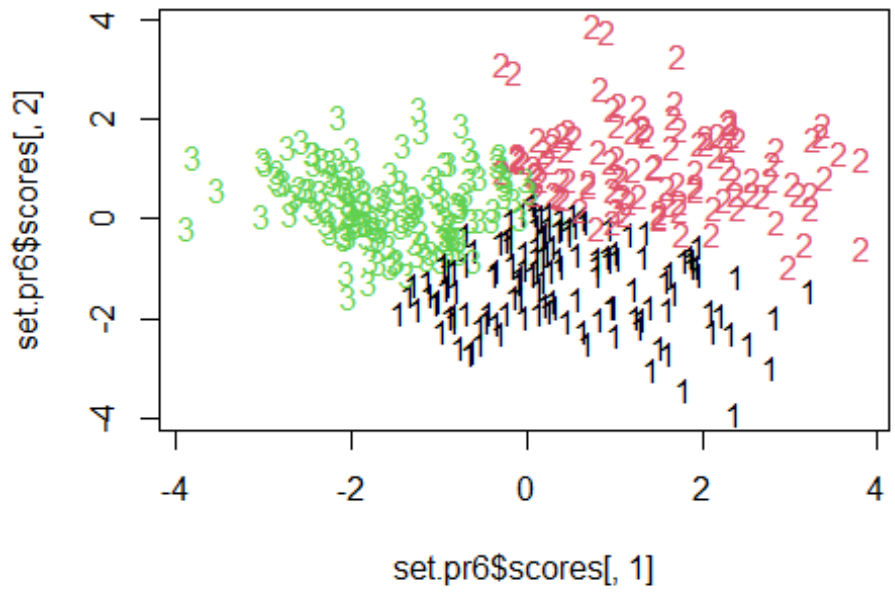
```
##
##   1   2   3   4
## 108 102  64 115
```

```r
# How many obs belong to each of the 4 clusters
kmeans.4$centers
```

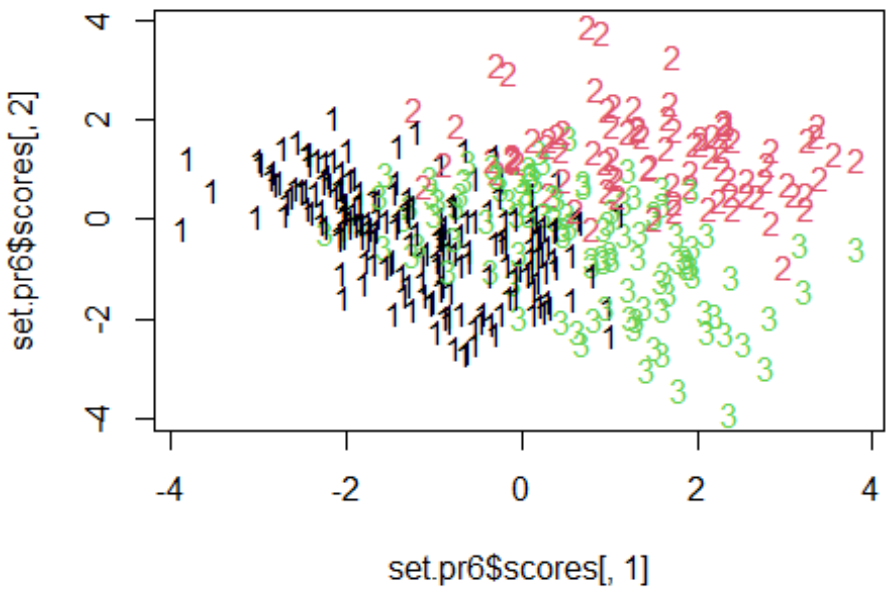
```
##   t.pregnant     plasma   bl.press   tr.thick  serum.ins         bmi
## 1  1.1106067  0.5594713  0.4223935  0.1766636  0.3386401  0.06965084
## 2 -0.2945874 -0.4709976 -0.4462116 -1.1311955 -0.4734205 -1.06360266
## 3 -0.5935915  0.8708431  0.6795439  0.8257253  0.9775965  0.96421655
## 4 -0.4513718 -0.5923052 -0.3790933  0.3778770 -0.4421775  0.34135062
##          diab        age
## 1 -0.01076019  1.2715029
## 2 -0.29865231 -0.5964125
```

```
## 3  0.37978991 -0.2333705
## 4  0.06363549 -0.5352394

# Mean values for each cluster
kmeans.4$withinss

## [1] 619.8494 417.3790 387.5410 492.4328

# Sum of squares within groups
kmeans.4$betweenss

## [1] 1186.798

# Sum of square between groups




# Say 2 clusters is selected
kmeans.2 <- kmeans(scale(new_df6[,-9]), centers = 2, iter.max = 25, tra
ce = TRUE)

## KMNS(*, k=2): iter=  1, indx=0
##  QTRAN(): istep=389, icoun=17
##  QTRAN(): istep=778, icoun=31
##  QTRAN(): istep=1167, icoun=5
##  QTRAN(): istep=1556, icoun=48

names(kmeans.2)

## [1] "cluster"      "centers"      "totss"        "withinss"     "tot
.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"

# Parameters that belong to object kmeans.4
table(kmeans.2$cluster)

##
##   1   2
## 207 182

# How many obs belong to each of the 4 clusters
kmeans.2$centers

##   t.pregnant     plasma   bl.press   tr.thick  serum.ins        bmi
diab
## 1 -0.4134982 -0.5503661 -0.4625387 -0.4136568 -0.4702493 -0.3850385
-0.05982241
## 2  0.4702974  0.6259658  0.5260742  0.4704778  0.5348440  0.4379284
0.06803977
##          age
```

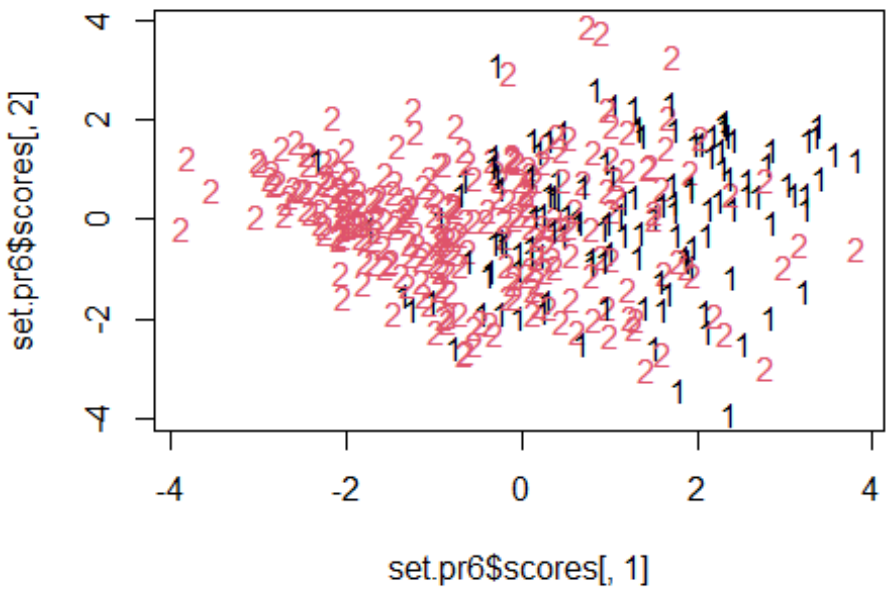
```
## 1 -0.5603742
## 2  0.6373487

# Mean values for each cluster
kmeans.2$withinss

## [1] 1107.409 1312.621

# Sum of squares within groups
kmeans.2$betweenss

## [1] 683.9697

# Sum of square between groups



# Plots
plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=kmeans.3$cluster, co
l=kmeans.3$cluster)
```



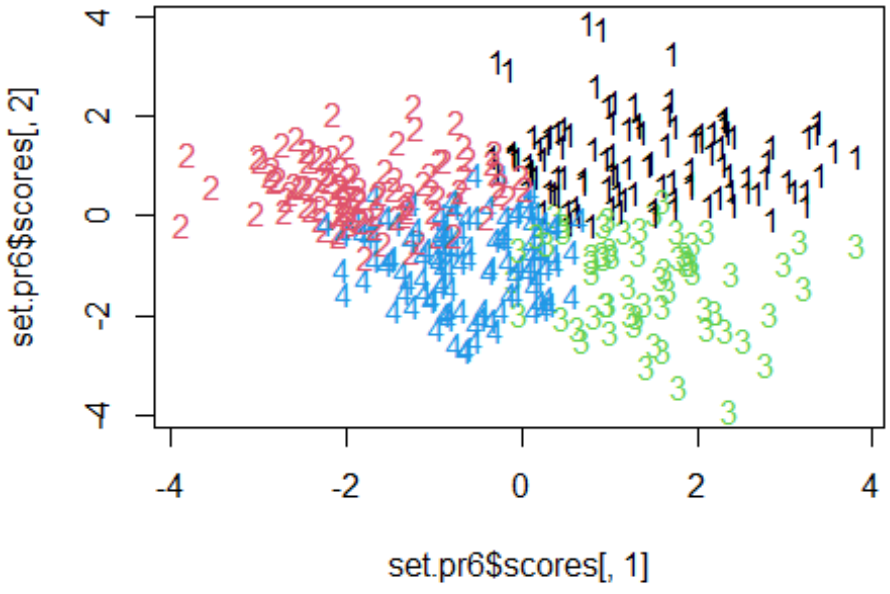

```
# We observe that there is no significant overlap!

plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=cluster.d6, col=clus
ter.d6)
```



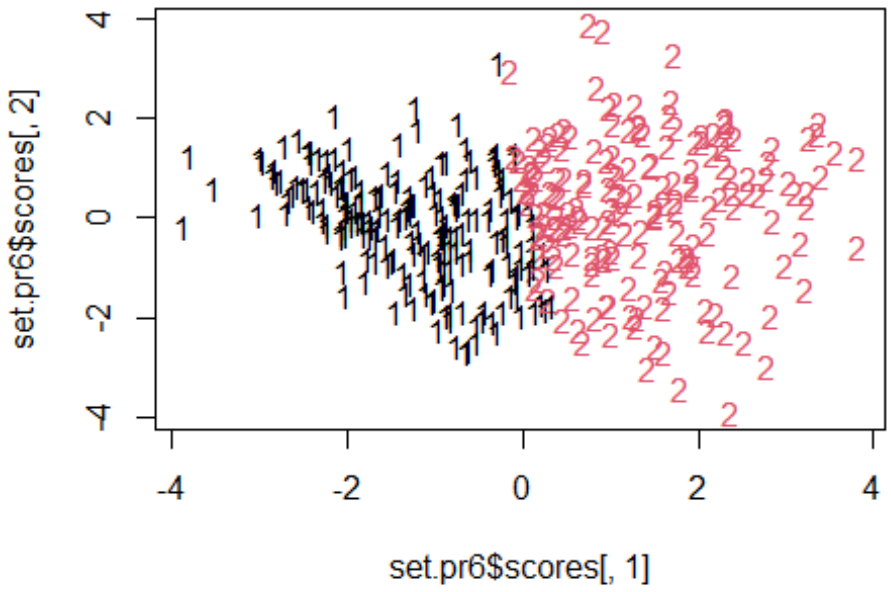

```
# we observe that k.means has provided a much better result compared to
this partitioning!

plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=new_df6[,9], col=new
_df6[,9])
```




```
plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=kmeans.4$cluster, co
l=kmeans.4$cluster)
```



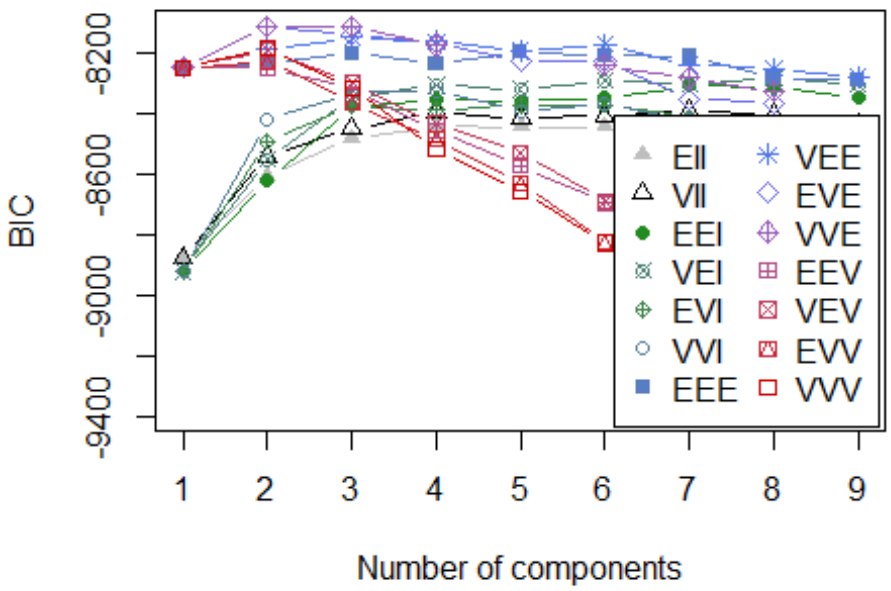
```
# We observe that k means with 4 clusters produces much worse results t
han the ones with 3 clusters!
# Visible much overlap between groups 1 and 2 mostly!

plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=kmeans.2$cluster, co
l=kmeans.2$cluster)
```



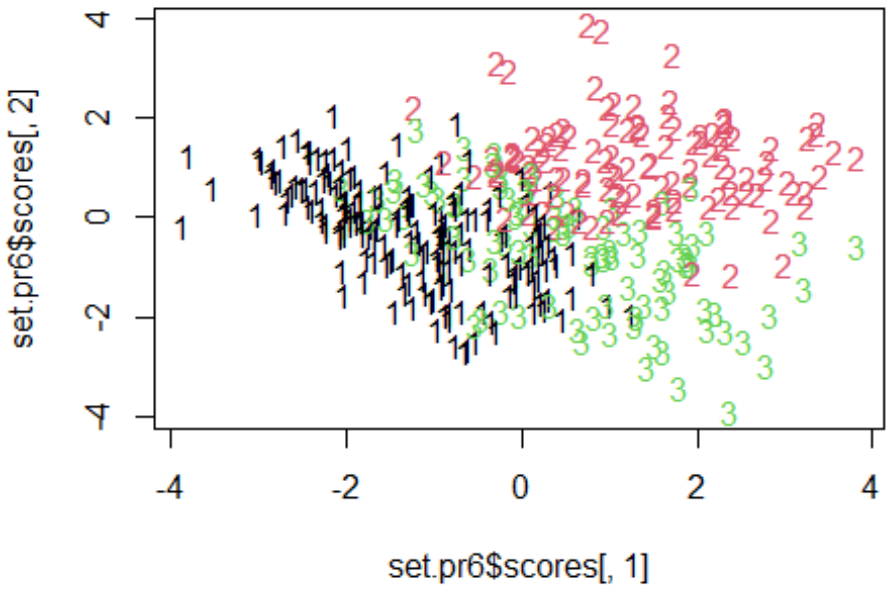

```
# Much better partition than the one with the 4 clusters!
```

Model based method

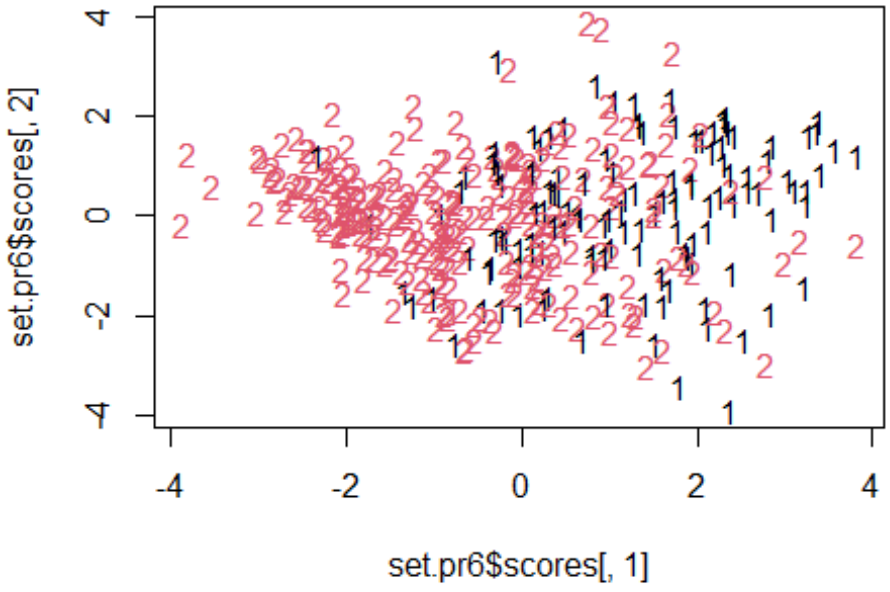
```
library(mclust)

## Package 'mclust' version 5.4.9
## Type 'citation("mclust")' for citing this R package in publications.

fit.model.based <- Mclust(scale(new_df6[,-9]))
names(fit.model.based)

##  [1] "call"           "data"           "modelName"      "n"
##  [5] "d"              "G"              "BIC"            "loglik"
##  [9] "df"             "bic"            "icl"            "hypvol"
## [13] "parameters"     "z"              "classification" "uncertainty
"

plot(fit.model.based, new_df6[,-9], what = 'BIC')
```




```
# The BIC values plotted
print(fit.model.based)

## 'Mclust' model object: (VVE,3)
##
## Available components:
##  [1] "call"           "data"           "modelName"      "n"
##  [5] "d"              "G"              "BIC"            "loglik"
##  [9] "df"             "bic"            "icl"            "hypvol"
## [13] "parameters"     "z"              "classification" "uncertainty
"

# Proposed model VVE (Variable-Variable-Equal) with 3 clusters!
table(fit.model.based$classification)

##
##   1   2   3
## 181 113  95

# Gives the sum of the cluster membership of the observations that belo
ng in the final transformed dataset (new_df6)

plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], main= 'Model Based', label
=fit.model.based$classification,
col=fit.model.based$classification)
```




```
plot(set.pr6$scores[,1], set.pr6$scores[,2], type='n')
text(set.pr6$scores[,1], set.pr6$scores[,2], label=new_df6[,9], col=new
_df6[,9])
```

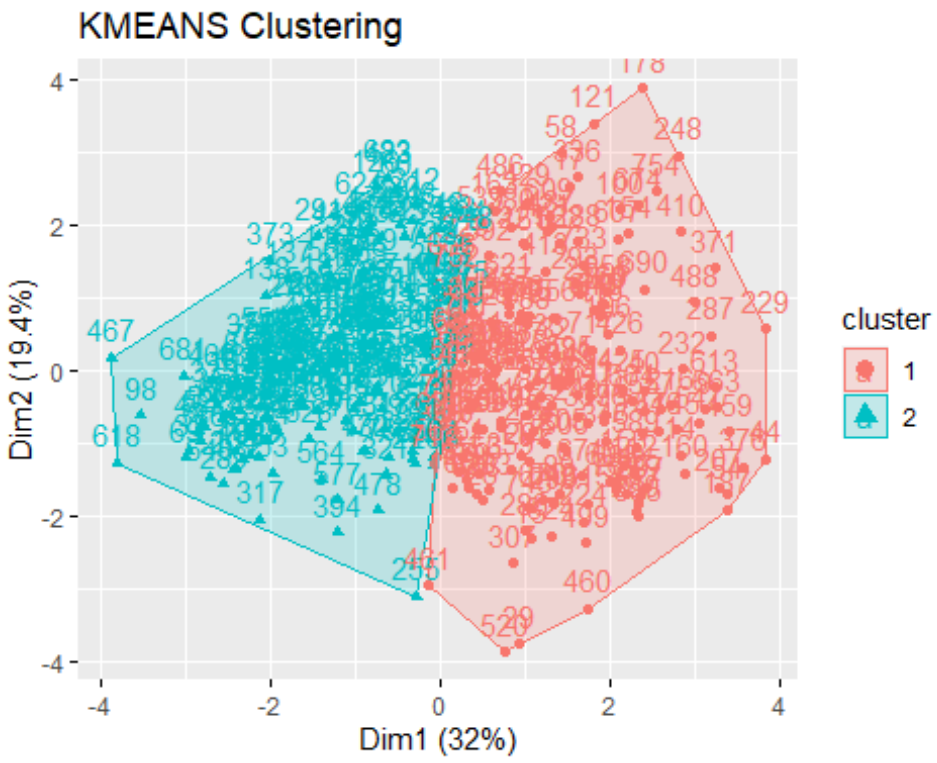


Indices

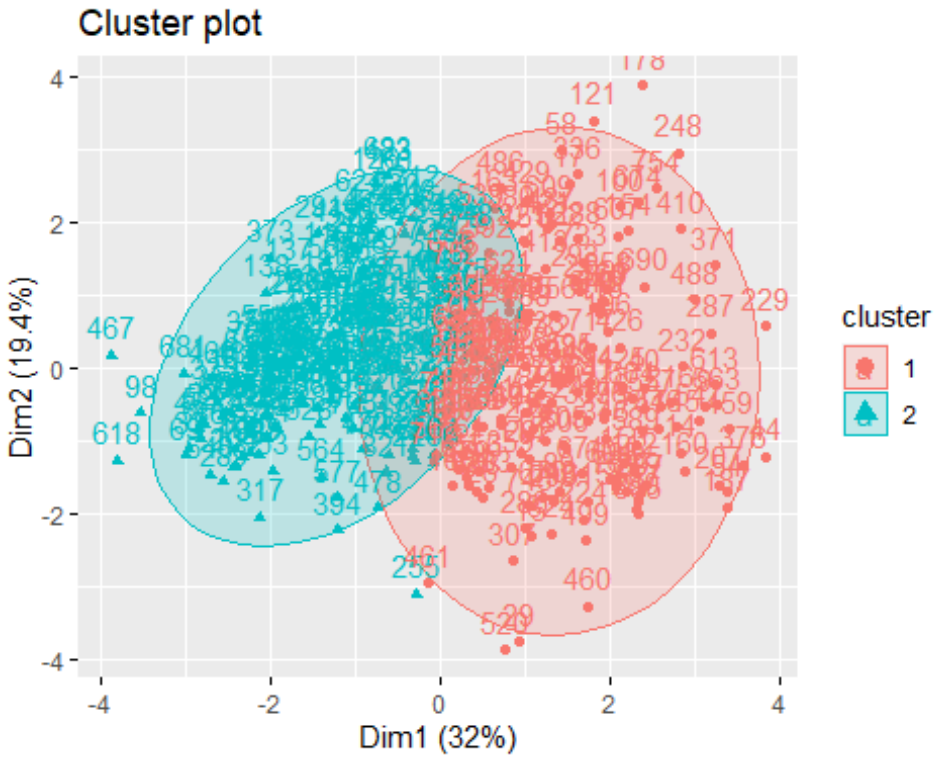
```
# install.packages("profdpm", repos="http://R-Forge.R-project.org")
library(profdpm)

# Indices
pci(cluster.d6,fit.model.based$classification)

##         R        FM       W10       W01         J
## 0.7884743 0.7105731 0.7231474 0.6982175 0.5509452

# R stands for Rand index, FM stands for Fowlkes and Mallows index, W10
stands for Wallace 10 index, W01 stands for
# Wallace 01 index and J stands for Jaccard index
pci(cluster.d6,kmeans.2$cluster)

##         R        FM       W10       W01         J
## 0.6875679 0.6490441 0.5591660 0.7533690 0.4726459

pci(cluster.d6,kmeans.3$cluster)

##         R        FM       W10       W01         J
## 0.6912517 0.5640265 0.5938661 0.5356863 0.3920576

pci(cluster.d6,kmeans.4$cluster)

##         R        FM       W10       W01         J
## 0.7485755 0.6109247 0.7327179 0.5093761 0.4295593
```

Visualizations

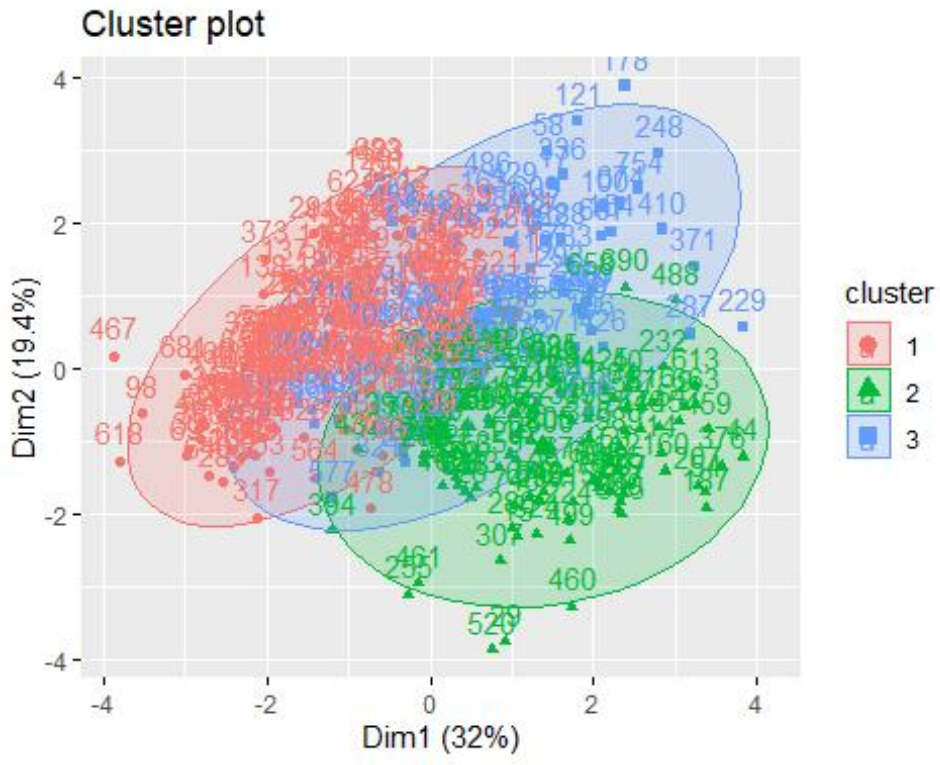
```
library(factoextra)

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

kmeans.cl = eclust(scale(new_df6[,-9]), FUNcluster = "kmeans", hc_metri
c = "euclidean")
```



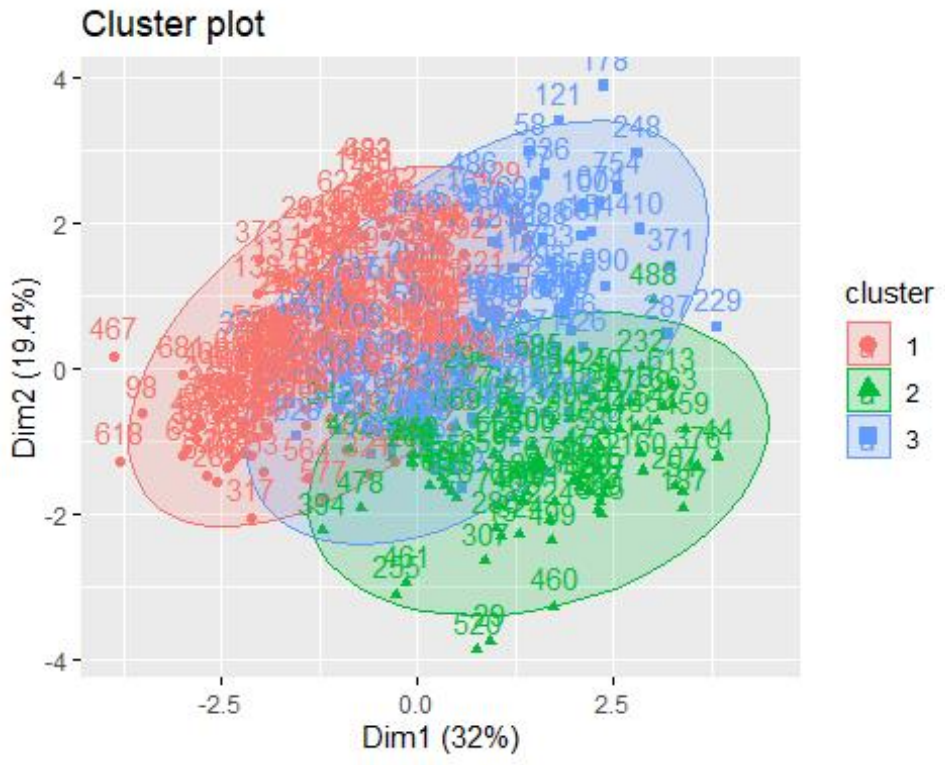

```
fviz_cluster(kmeans.cl, ellipse=TRUE, ellipse.type='norm')
```



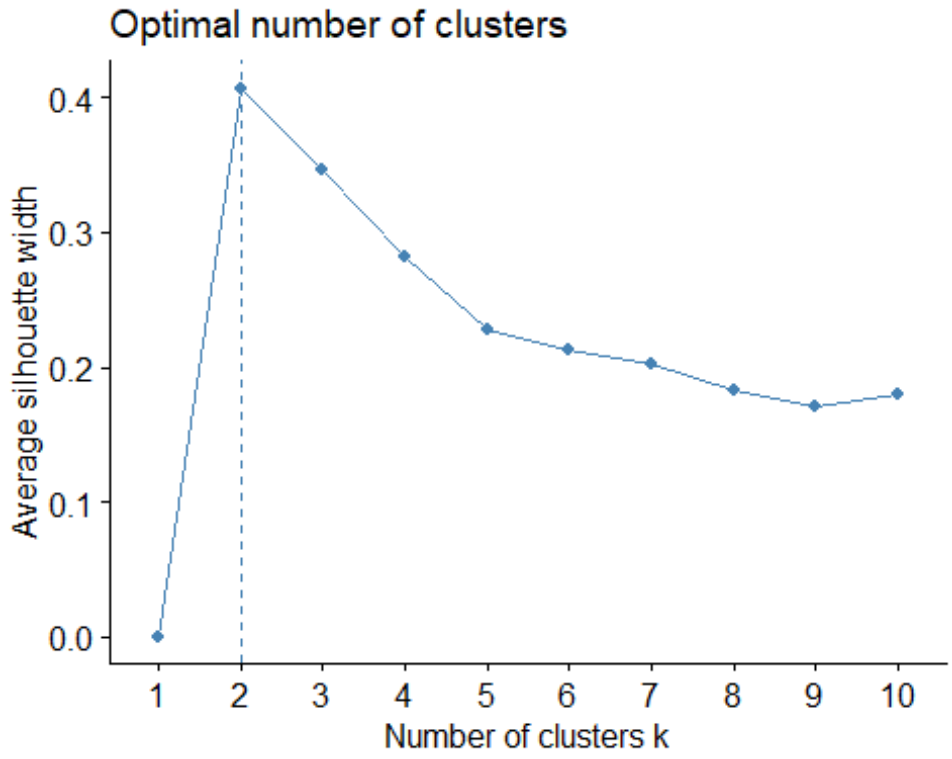

```
fviz_cluster(fit.model.based, ellipse=TRUE, ellipse.type='norm')
```




```
# Similar plot with the one for ggbiplot

hclust_d2 <- eclust(scale(new_df6[,-9]), "hclust", k = 3, hc_metric = "
euclidean", hc_method = "ward.D2",
graph = FALSE)
fviz_cluster(hclust_d2, ellipse=TRUE, ellipse.type='norm')
```

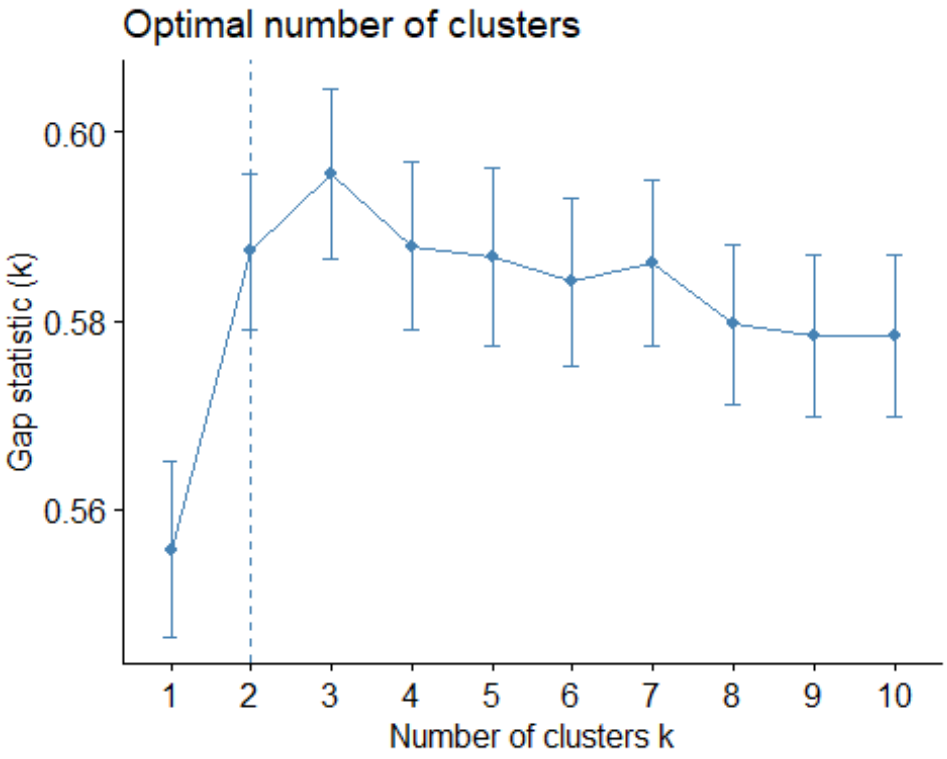



```
fviz_nbclust(new_df6, kmeans , method='silhouette')
```

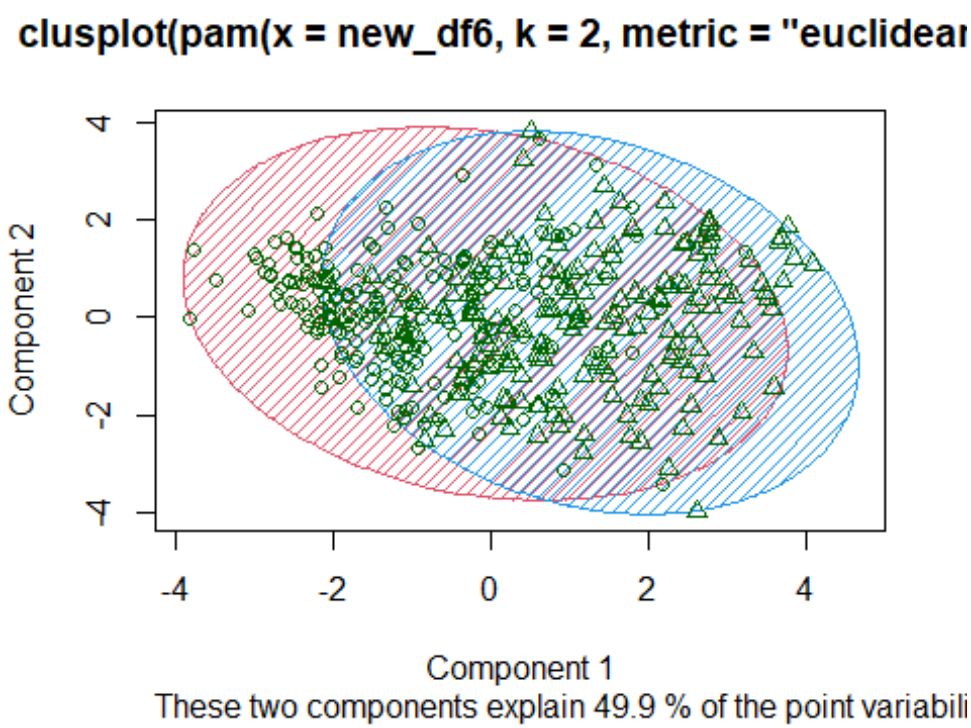


Partitioning Clustering

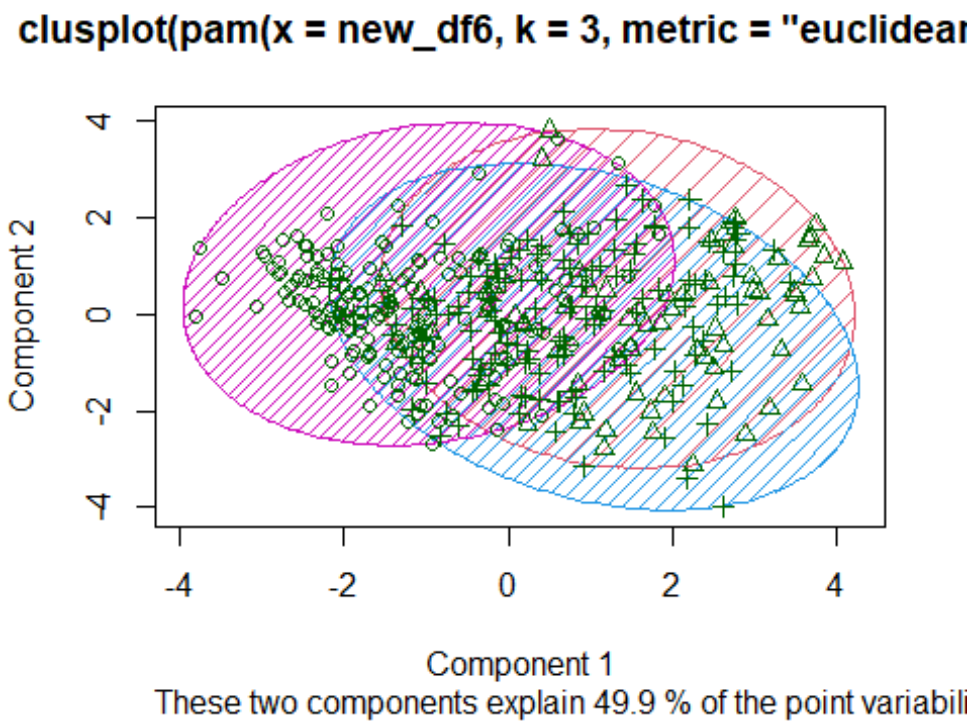
```
library(cluster)
gap_stat <- clusGap(scale(new_df6[,-9]), FUN = kmeans, nstart = 25, K.m
ax = 10, B = 50)
fviz_gap_stat(gap_stat)
```




```
# Visualizing the optimal value for the number of clusters ( k = 3)

part1 <- pam(new_df6,k=2,metric="euclidean")
clusplot(part1,shade=T,color=T)
```



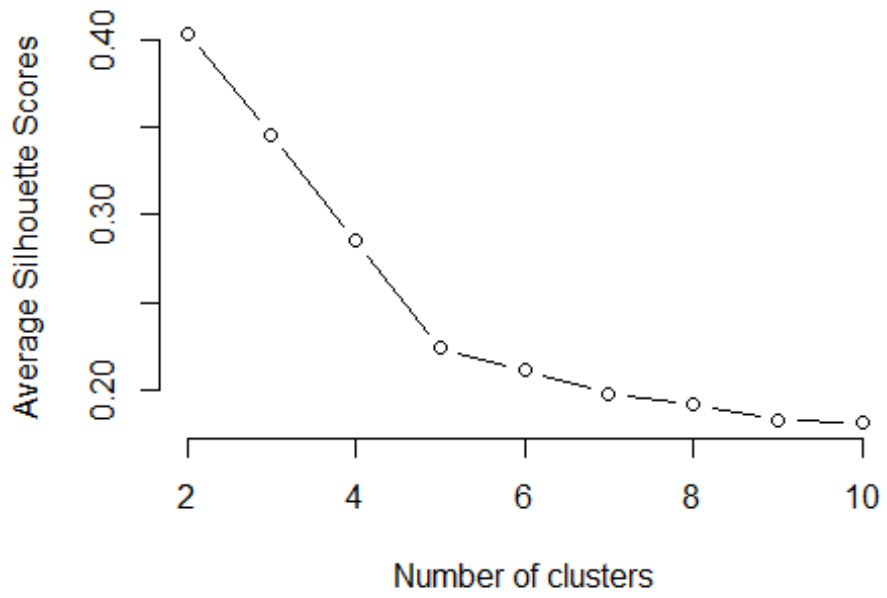

```
# Way too much overlap observed!

part2 <- pam(new_df6,k=3,metric="euclidean")
clusplot(part2,shade=T,color=T)
```




```
# Definately not prefferable the clustering this way!
```

Silhouette for the different number of clusters.

```
# Determining optimal number of clusters (k)
silhouette_score <- function(k){
  km <- kmeans(new_df6, centers = k, nstart=25)
  ss <- silhouette(km$cluster, dist(new_df6))
  mean(ss[, 3])
}
k <- 2:10
avg_sil <- sapply(k, silhouette_score)
plot(k, type='b', avg_sil, xlab='Number of clusters', ylab='Average Sil
houette Scores', frame=FALSE)
```




```
# We observe that this way the optimal number of clusters seems to be 2
!
```

Final suggestions

Following the exploration we conducted, in order to choose the most appropriate way to do the clustering, we have to take into consideration the results of all the methods used above. The reason lies in the fact that the more methods one explores the better picture of the situation he gets! Let's sum up the methods used and what their overall results produced:

- Most methods of the Hierarchical Clustering category indicate to use 3 clusters.
- Kmeans suggests using 2 clusters.
- Model based recommends using 3 clusters.

Therefore, after taking into account all of the above, our final suggestion would be to use 3 groups in order to do the clustering!