BB512/BB612 - Homework I - KEY

Mar 24, 2022

Use the `montpick` eset to perform the required analyses:

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

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

```
library(dendextend)
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
library(Biobase)
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
```

```
##     union, unique, unsplit, which.max, which.min
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
con <- url("http://bowtie-bio.sourceforge.net/recount/ExpressionSets/montpick_eset.RData")
load(file = con)
close(con)

pdata <- pData(montpick.eset)
edata <- exprs(montpick.eset)
fdata <- fData(montpick.eset)
```

Proprocessing, EDA and Clustering

1. [5 pt] Exclude probes with average expression count < 100

```
## exclude probes with ave. expr. < 100
edata <- edata[rowMeans(edata) >= 100, ]
```

2. [5 pt] Perform log2 transformation

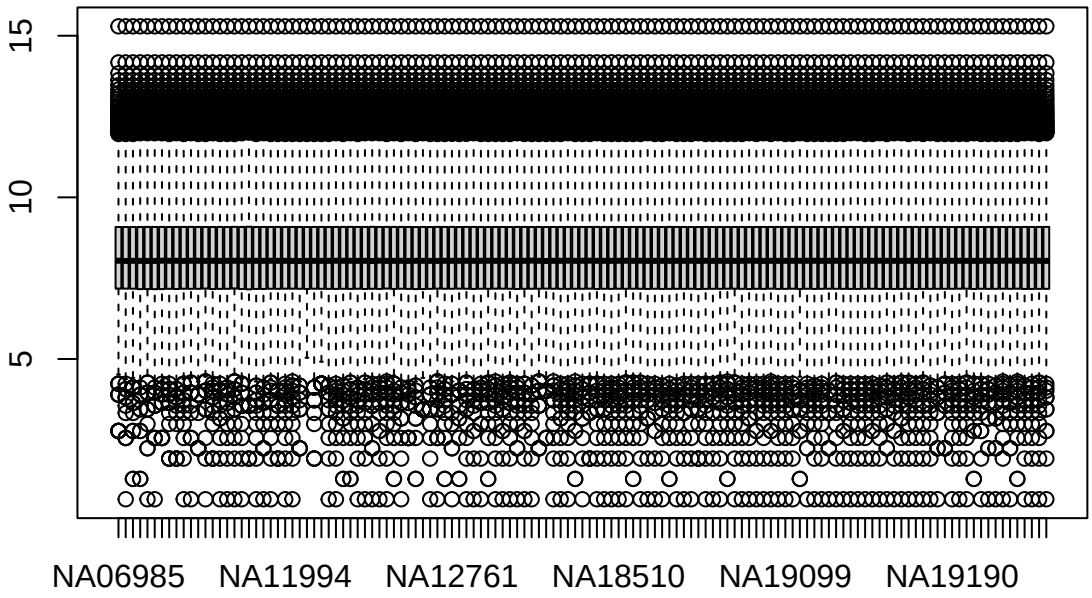
```
## log2 transform
edata <- log2(edata + 1)
```

3. [10 pt] Perform quantile normalization, keeping the row and column names

```
## normalize
norm_data <- preprocessCore::normalize.quantiles(as.matrix(edata))
colnames(norm_data) <- colnames(edata)
rownames(norm_data) <- rownames(edata)
```

4. [5 pt] Check the distributions via a boxplot

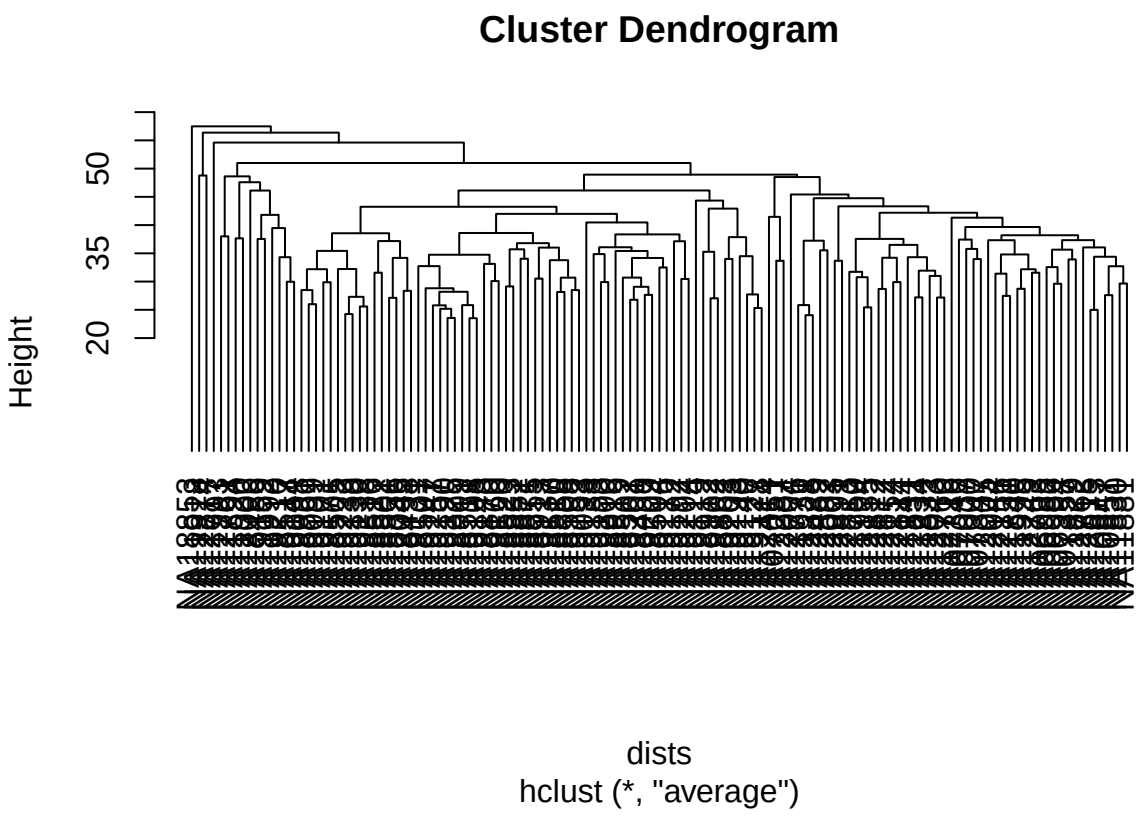
```
## check distibutions by boxplot
boxplot(norm_data)
```



5. [15] Perform hierarchical clustering with average agglomeration (UPGMA) and plot the dendrogram (You may use any appropriate distance metric)

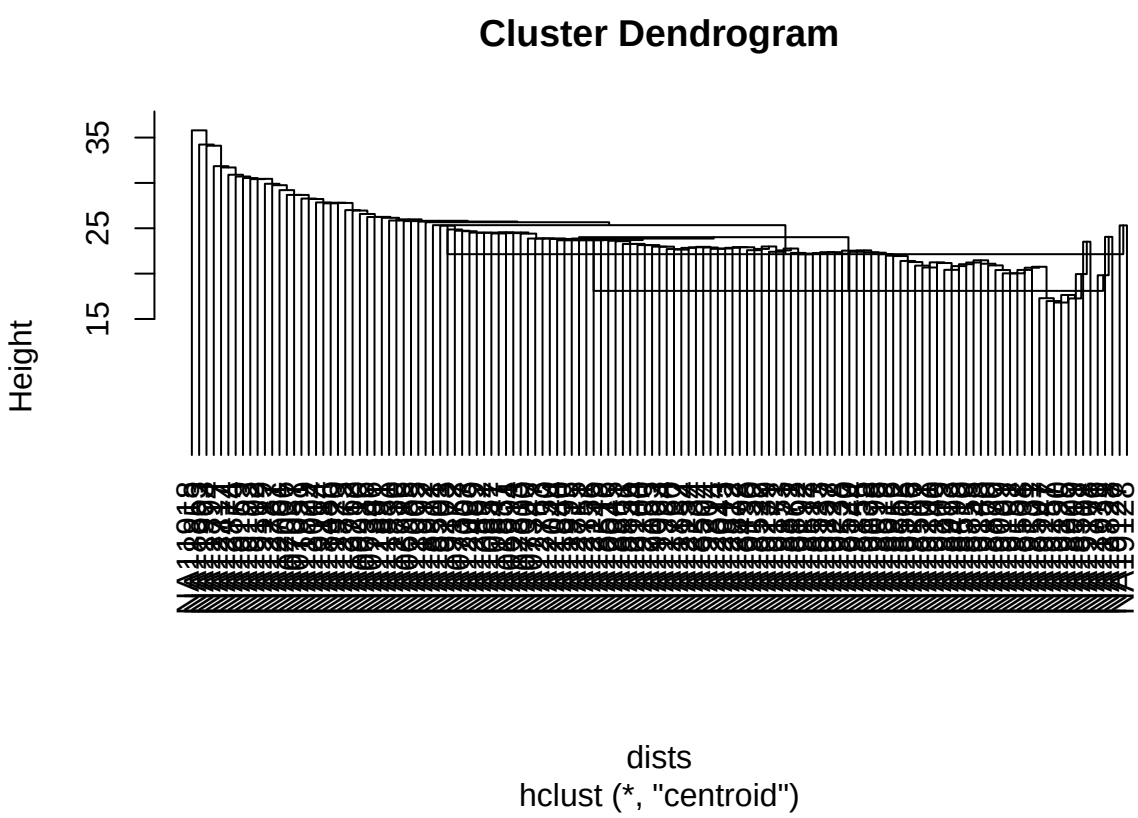
```
dists <- dist(t(norm_data))

## h.clust with average agglomeration (UPGMA) and plot dend.
clu <- hclust(dists, method = "average")
plot(clu, hang = -1)
```



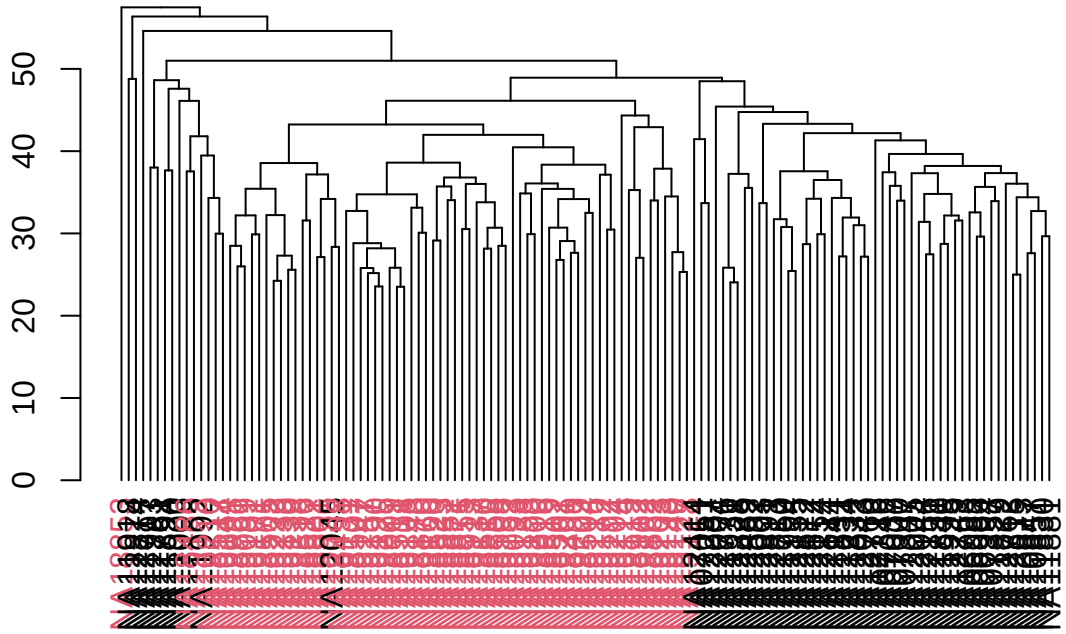
6. [15] Perform hierarchical clustering with centroid (UPGMC) and plot the dendrogram (You may use any appropriate distance metric)

```
## h.clust with centroid agglomeration and plot dend.
clu2 <- hclust(dists, method = "centroid")
plot(clu2, hang = -1)
```



7. [10] Plot the dendrogram of UPGMA, coloring leaves by `population` (in `pdata`)

```
## plot dend. of UPGMA by coloring leaves by population
dend <- as.dendrogram(clu)
labels_colors(dend) <- as.numeric(pdata$population[match(labels(dend), pdata$sample.id)])
plot(dend)
```



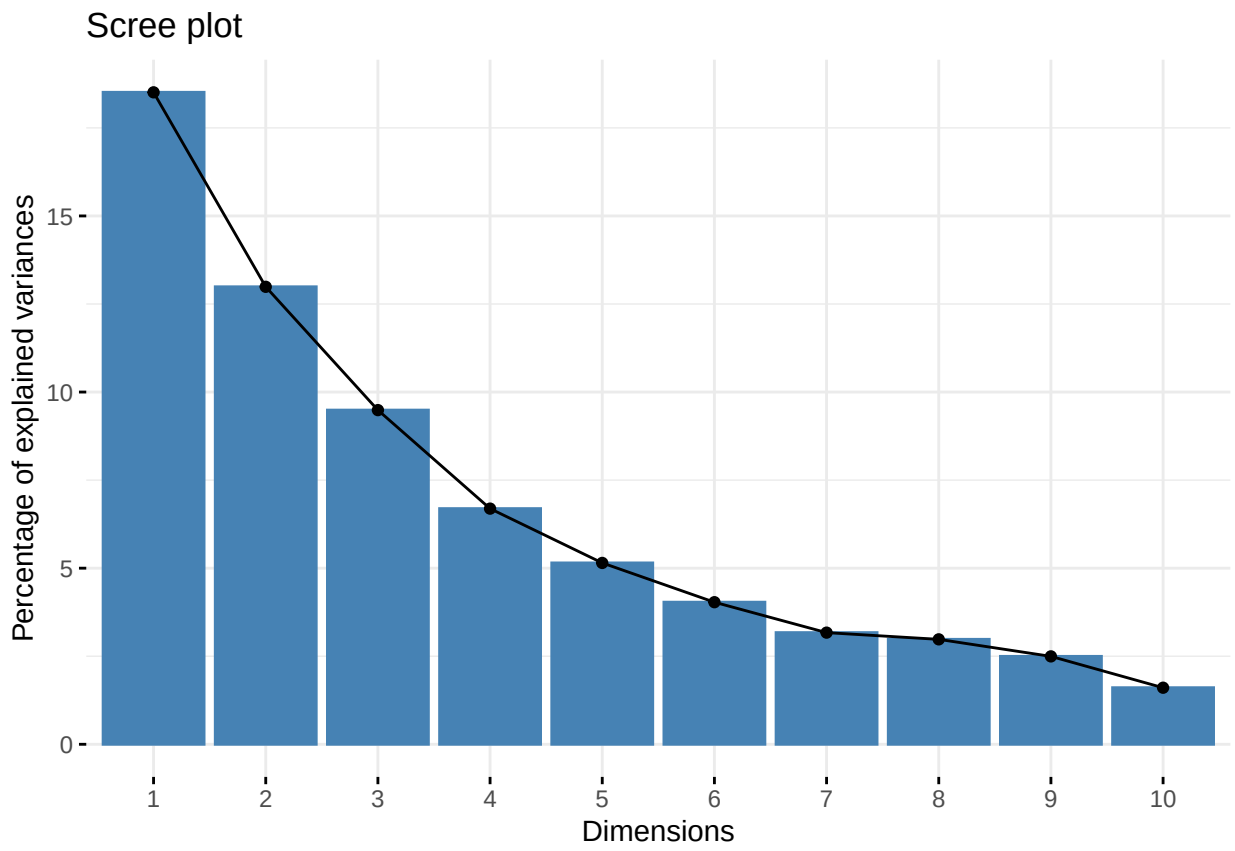
PCA

8. [15 pt] Perform PCA of samples, scaling the variables

```
res.pca <- prcomp(t(norm_data), scale = TRUE)
```

9. [10 pt] Plot the scree plot, showing the percentage of variances explained by each principal component

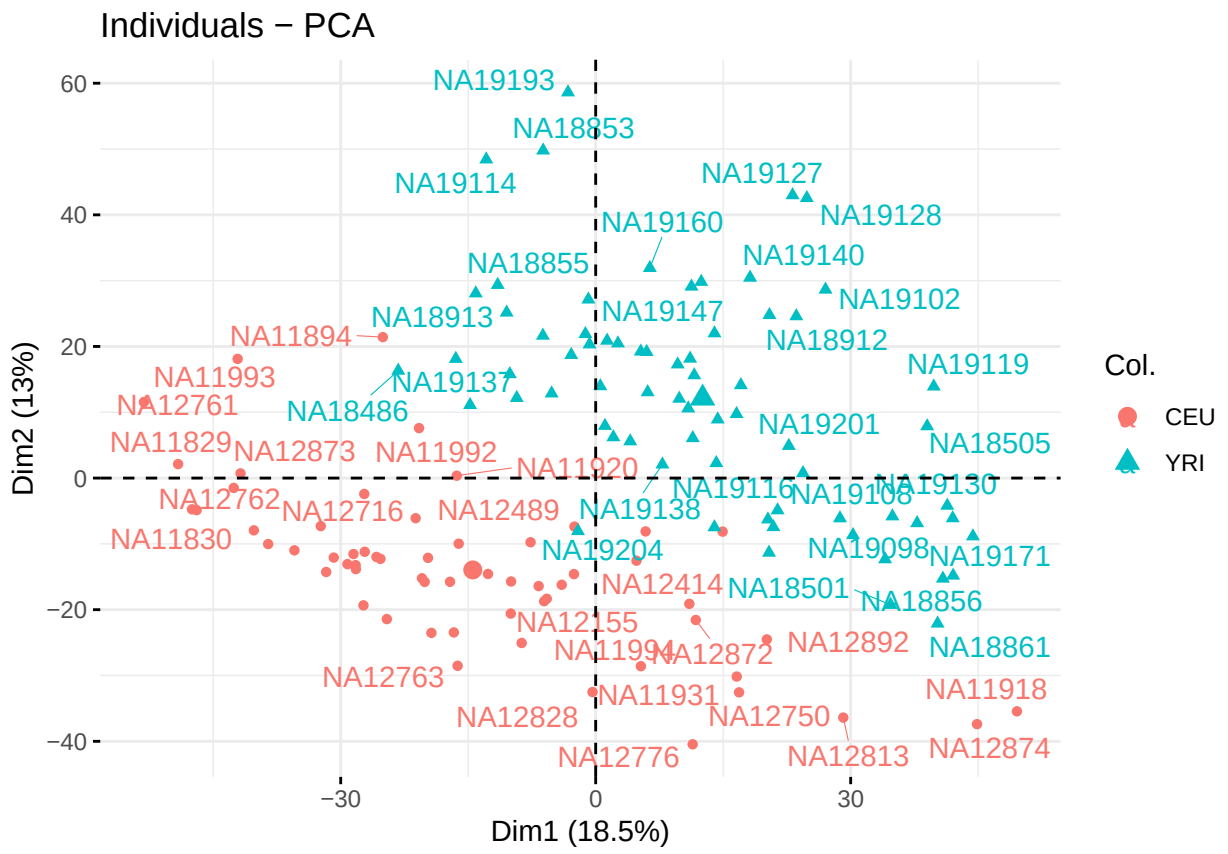
```
## Plot scree plot
fviz_eig(res.pca)
```



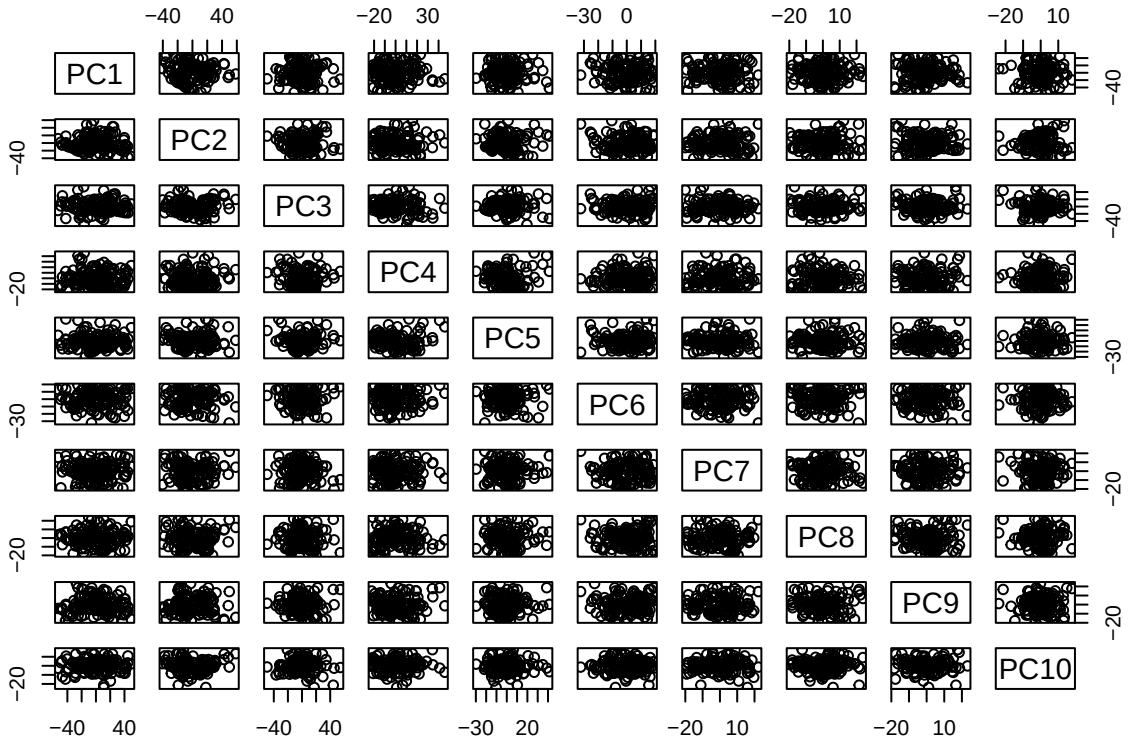
10. [10 pt] Plot PC1 vs. PC2 showing individuals colored by population

```
### Plot PC1 vs PC2 showing indiduals colored by population
fviz_pca_ind(res.pca,
             col.ind = pdata$population,
             repel = TRUE)
```

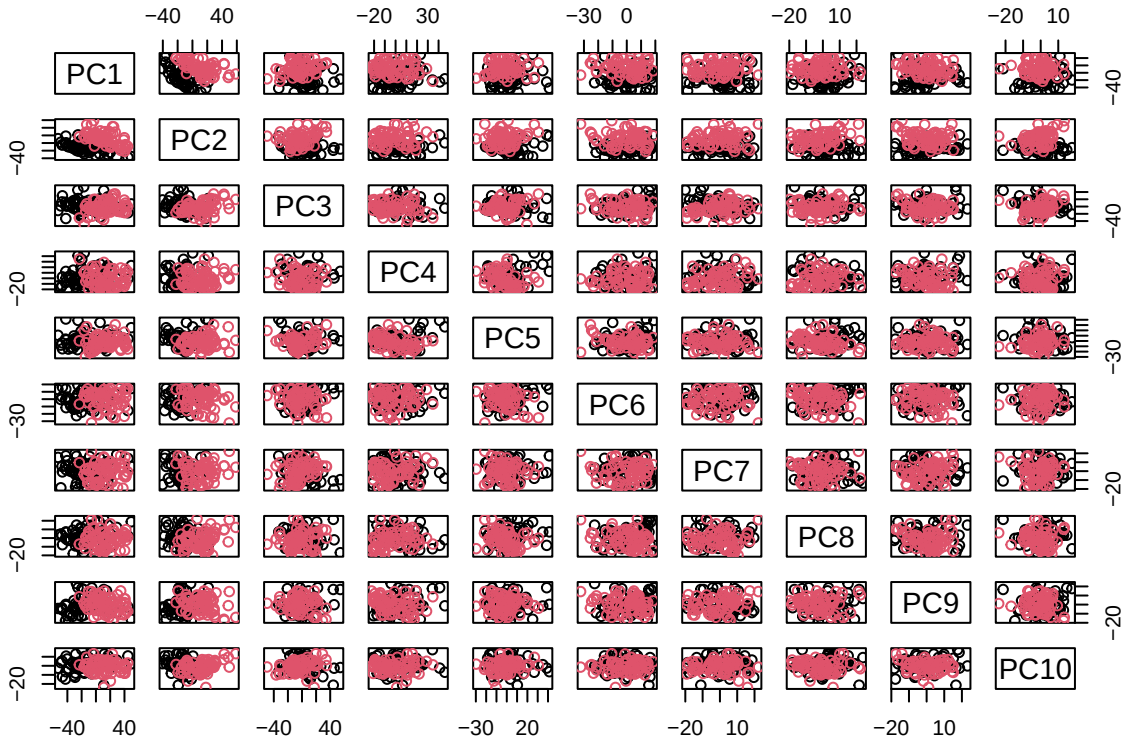
```
## Warning: ggrepel: 78 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
pairs(res.pca$x[, 1:10])
```




```
pairs(res.pca$x[, 1:10], col = pdata$population)
```


PC1
PC2
PC3
PC4
PC5
PC6
PC7
PC8
PC9
PC10
−40 40
−20 30
−30 0
−20 10
−20 10
−40 40
−40 40
−30 20
−20 10
−20 10
−40
−40
−20
−30
−20
−20