ST340 Lab 2: SVD & PCA

2020–21

1: A simple singular value decomposition

(a) Generate a realization of a 4×5 Gaussian random matrix G.

```
set.seed(5)
G = matrix(nrow = 4, ncol = 5)

for(j in 1:5){
  for(i in 1:4){
    G[i,j] = rnorm(1)
  }
}
```

(b) Look at `?svd`.

(c) Set U, d, and V by using `svd`.

```
svd_G = svd(G, nv = 5)
print(svd_G)
```

```
## $d
## [1] 2.9179553 2.2422189 1.8845677 0.9448286
##
## $u
##            [,1]        [,2]        [,3]        [,4]
## [1,]  0.3437357 -0.87424173  0.32926491  0.09556026
## [2,] -0.8106523 -0.46050166 -0.34436523 -0.11042471
## [3,]  0.4406251 -0.14544590 -0.87870654  0.11211533
## [4,] -0.1747515  0.04985085  0.02953019  0.98290629
##
## $v
##            [,1]        [,2]       [,3]        [,4]         [,5]
## [1,] -0.6774350  0.12653270  0.1866162 -0.32284793  0.621301017
## [2,]  0.3358571 -0.52696553  0.6193840 -0.47344657  0.041463612
## [3,]  0.1613625 -0.01439958 -0.6601293 -0.73346320 -0.003978202
## [4,] -0.2370222  0.52003115  0.3375693 -0.36262448 -0.654170093
## [5,]  0.5882725  0.66004323  0.1783653 -0.04639809  0.429315127
```

```
U = svd_G$u
d = svd_G$d
V = svd_G$v

L = diag(svd_G$d)
L = cbind(L, rep(0,4))
```

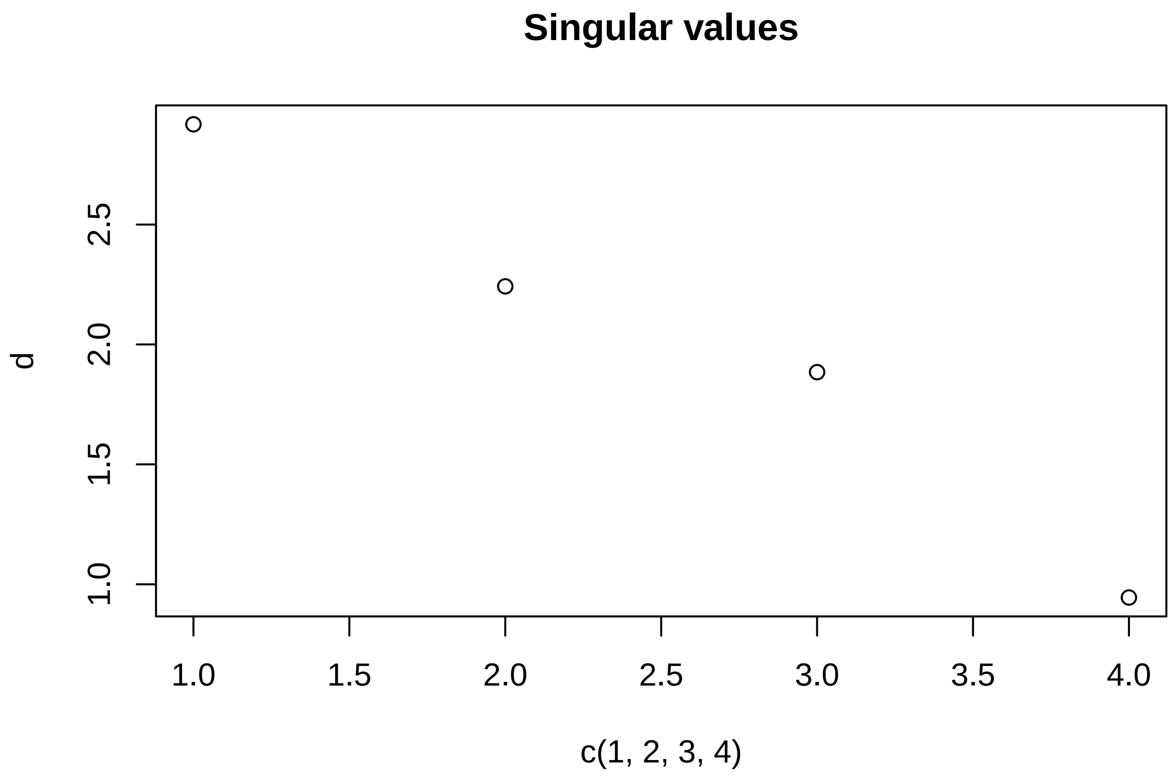
(d) Check that `G` is equal to `U%*%Sigma%*%t(V)` (to machine precision).

```
SVD_calc = (U)%*%L%*%t(V)

all.equal(G, SVD_calc)
```

```
## [1] TRUE
```

(e) Plot the singular values.

```
plot(x = c(1,2,3,4), y = d, main = "Singular values")
```



(f) Compute G_2, the 2-rank approximation of G, and also compute $||G - G_2||_F$.

```
G_2 = d[1]*(U[,1])%*%t(V[,1]) + d[2]*(U[,2])%*%t(V[,2])

G_G_2_frobenius = sqrt(sum((G-G_2)^2))
```

(g) Does the value agree with the theory?

```
all.equal(G_G_2_frobenius, sqrt(sum(d[3:4]^2)))
```

```
## [1] TRUE
```

2: Image compression via the singular value decomposition

```
load("pictures.rdata")
source("svd.image.compression.R")
```

Take a look at `svd.image.compression.R` and understand what the code is doing.Then run `image.compression()` here to see how well we can compress our images.

I have commented on the r file that creates the functions.

3: PCA: Crabs

(a) Load the MASS library to access the crabs data.

```
library(MASS)
library(factoextra)
```

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

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

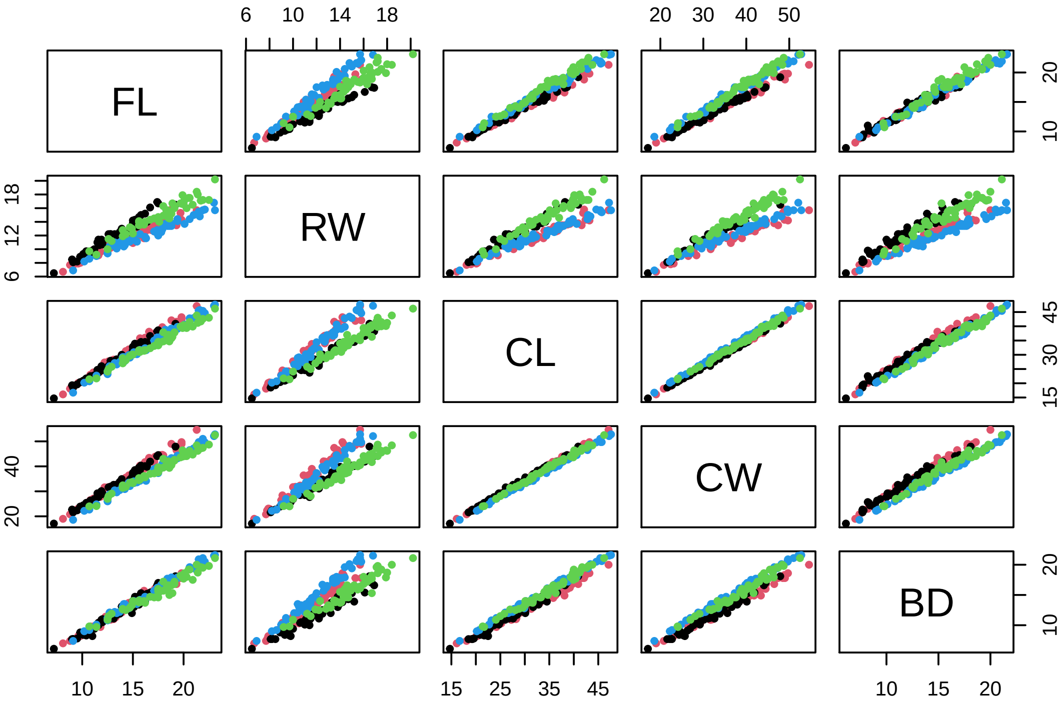
(b) Read `?crabs`.

```
head(crabs)
```

```
##   sp sex index   FL  RW   CL   CW  BD
## 1  B   M     1  8.1 6.7 16.1 19.0 7.0
## 2  B   M     2  8.8 7.7 18.1 20.8 7.4
## 3  B   M     3  9.2 7.8 19.0 22.4 7.7
## 4  B   M     4  9.6 7.9 20.1 23.1 8.2
## 5  B   M     5  9.8 8.0 20.3 23.0 8.2
## 6  B   M     6 10.8 9.0 23.0 26.5 9.8
```

(c) Read in the FL, RW, CL, CW, and BD measurements.

```
Crabs <- crabs[,4:8]
Crabs.class <- factor(paste(crabs[,1],crabs[,2],sep=""))
# Creating factor that combines the species with the sex

plot(Crabs,col=Crabs.class,pch=20)
```

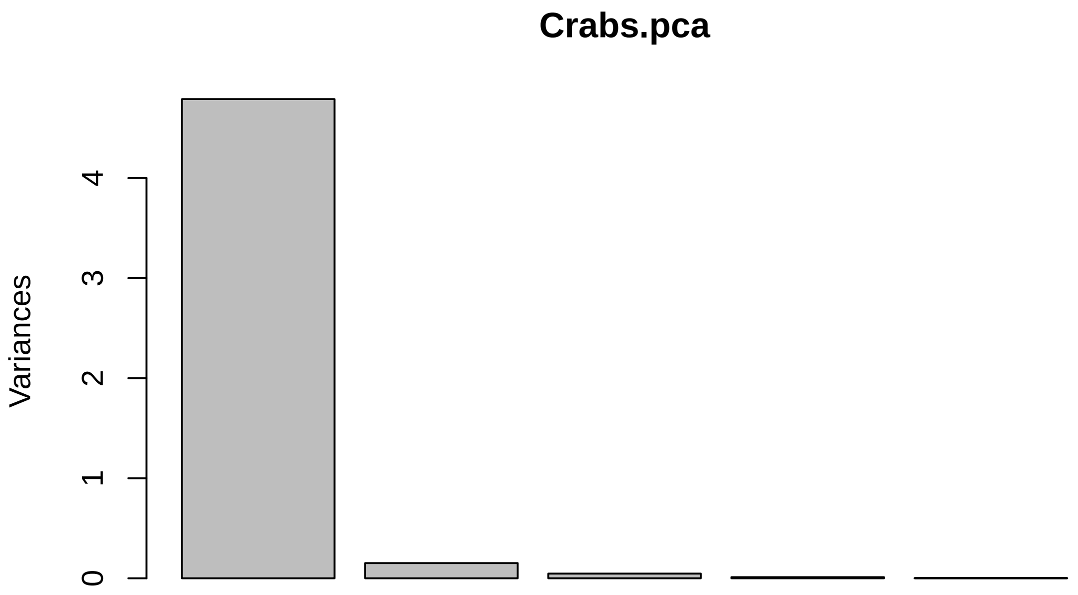


(d) Read `?prcomp` and use it to obtain the principal components of a centred and scaled version of `Crabs`. Call the output of `prcomp Crabs.pca`'.

```
Crabs.pca = prcomp(Crabs, center = T, scale. = T)
```

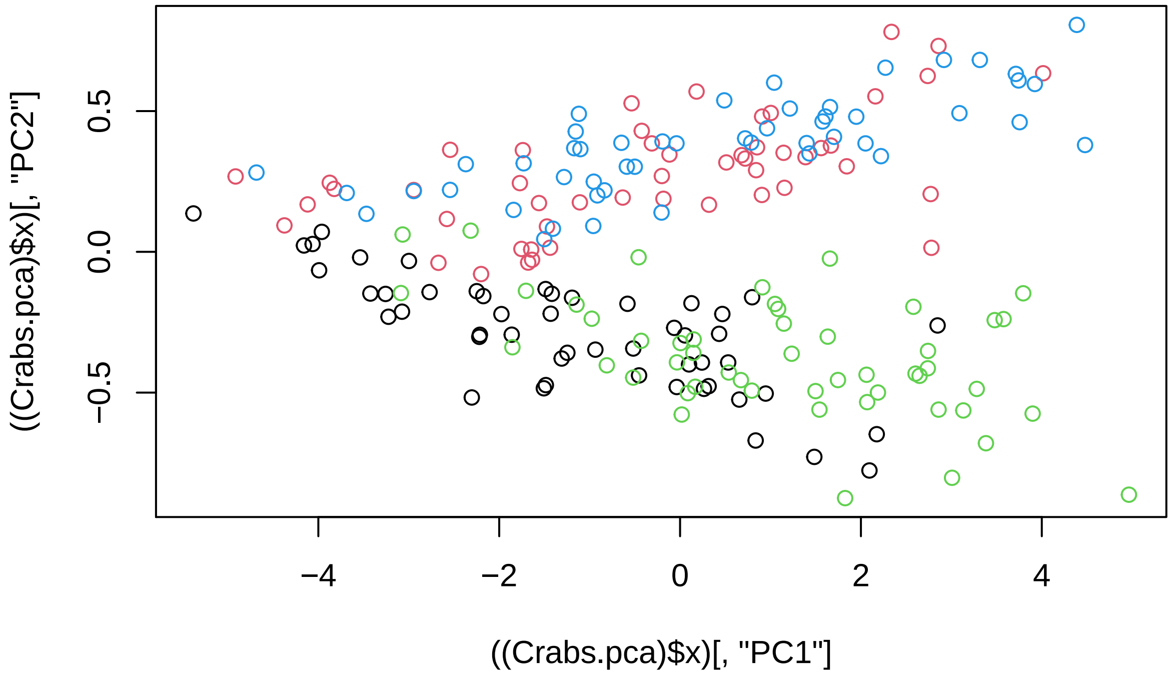
(e) If you `plot(Crabs.pca)` it visualizes the variances associated with the components.

```
plot(Crabs.pca)
```



(f) Plot PC2 against PC1.

```
plot(((Crabs.pca)$x)[,'PC1'], ((Crabs.pca)$x)[,'PC2'], col=Crabs.class)
```

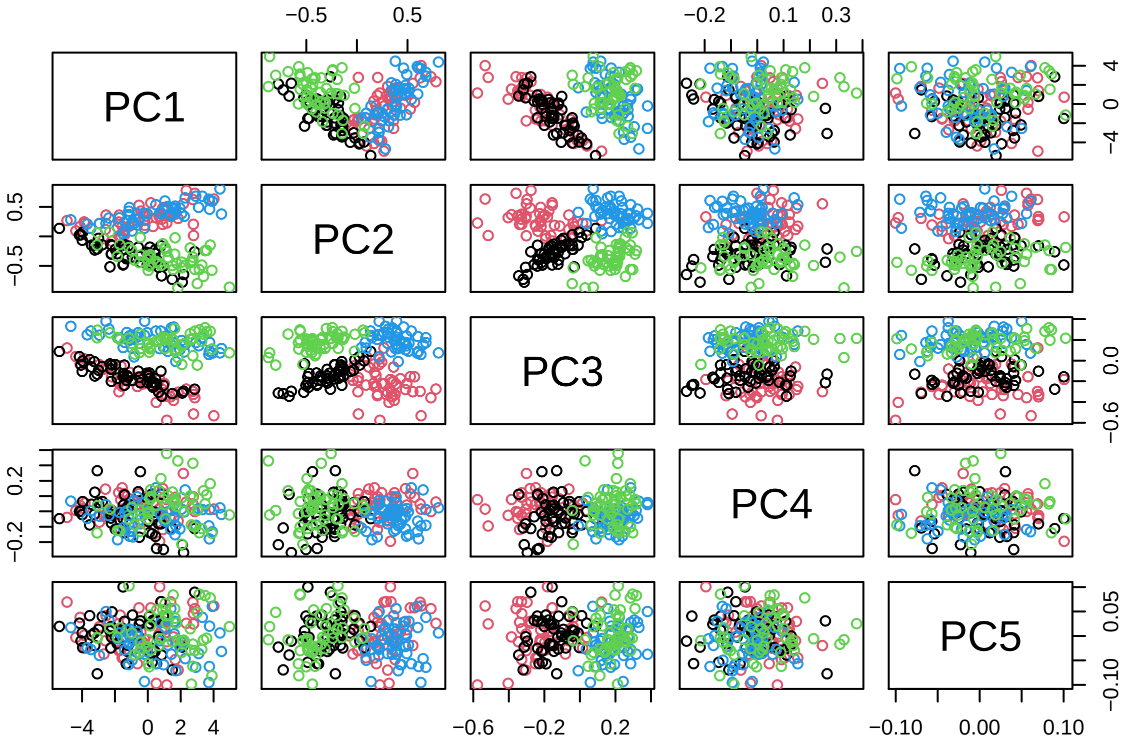


```
str(Crabs.pca)
```

```
## List of 5
##  $ sdev    : num [1:5] 2.1883 0.3895 0.2159 0.1055 0.0414
##  $ rotation: num [1:5, 1:5] 0.452 0.428 0.453 0.451 0.451 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "FL" "RW" "CL" "CW" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 15.6 12.7 32.1 36.4 14
##   ..- attr(*, "names")= chr [1:5] "FL" "RW" "CL" "CW" ...
##  $ scale   : Named num [1:5] 3.5 2.57 7.12 7.87 3.42
##   ..- attr(*, "names")= chr [1:5] "FL" "RW" "CL" "CW" ...
##  $ x       : num [1:200, 1:5] -4.92 -4.38 -4.12 -3.87 -3.82 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:200] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

(g) Read `?pairs` and use it to find a pair of components with good separation of the classes.

```
pairs(Crabs.pca$x, col=Crabs.class)
```



(h) Read `?scale`. Check that you can obtain the principal components by using the singular value decomposition on a centred and scaled version of `Crabs`.

4: PCA: Viruses

This is a dataset on 61 viruses with rod-shaped particles affecting various crops (tobacco, tomato, cucumber and others) described by Fauquet *et al.* (1988) and analysed by Eslava-Gómez (1989). There are 18 measurements on each virus, the number of amino acid residues per molecule of coat protein.

```
load("viruses.rdata")
```

(a) Obtain the principal components of a centred and scaled version of allviruses.

```
groups <- rep(0,61)
groups[1:3] <- 1
groups[4:9] <- 2
groups[10:48] <- 3
groups[49:61] <- 4
group.names <- c("Hordeviruses","Tobraviruses","Tobamoviruses","furoviruses")

head(allviruses)
```

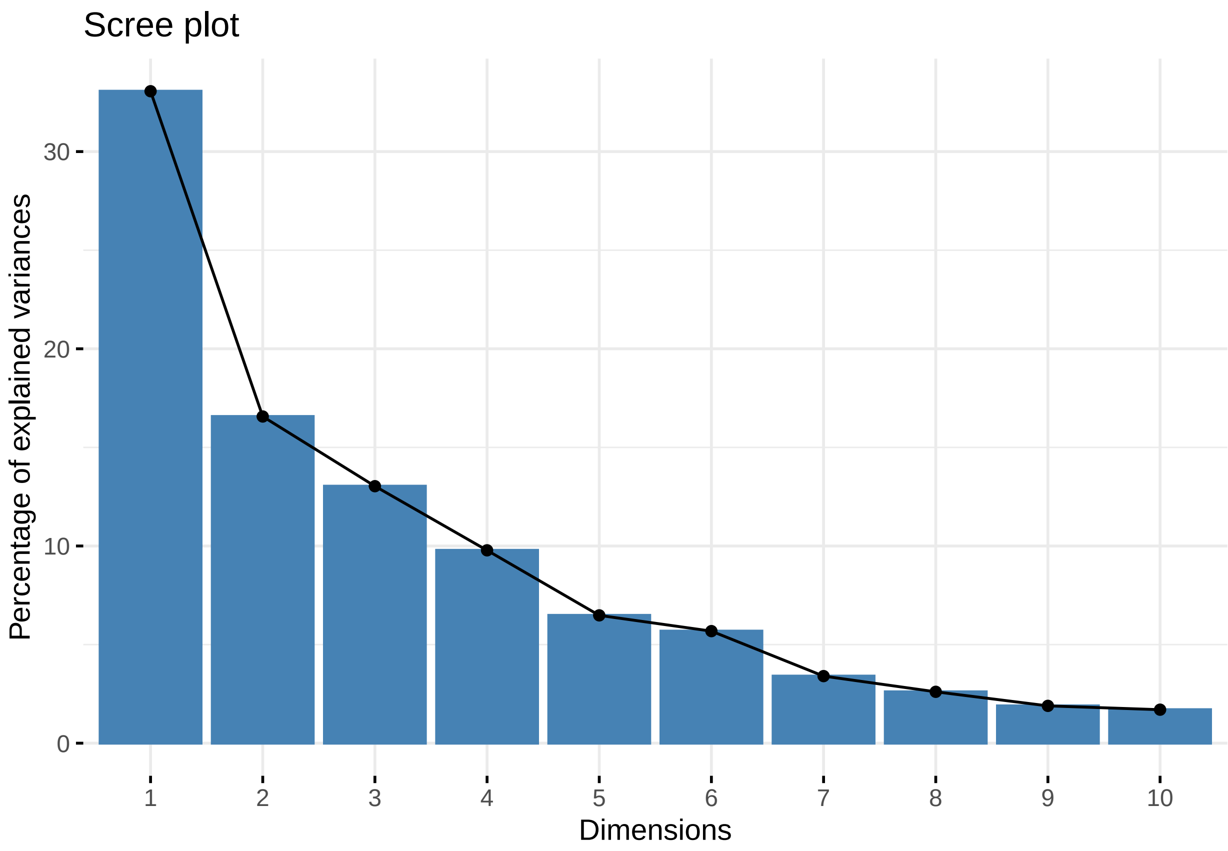
```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]   25    9    9   19   12    8   20    0   10     0     6    21     8     7
## [2,]   26    9    9   20   13    8   20    0   10     0     6    21     8     7
## [3,]   25    9    9   22   10   10   23    0   13     0     6    19     5     6
## [4,]   15   10   21   13   18   12   22    1    9     2     4    11     5    10
## [5,]   17   11   22   15   14   10   23    1   11     2     4    11     5     9
## [6,]   22   17   17   16   10   15   13    1    7     2     3    14     9     9
##      [,15] [,16] [,17] [,18]
## [1,]     4     7    17     5
## [2,]     4     7    17     5
## [3,]     4     8    16     5
## [4,]     1    14     8     2
## [5,]     1    13     9     1
## [6,]     2    12     6     2
```

```
allviruses.PCA = prcomp(allviruses, center = T, scale. = T)
```

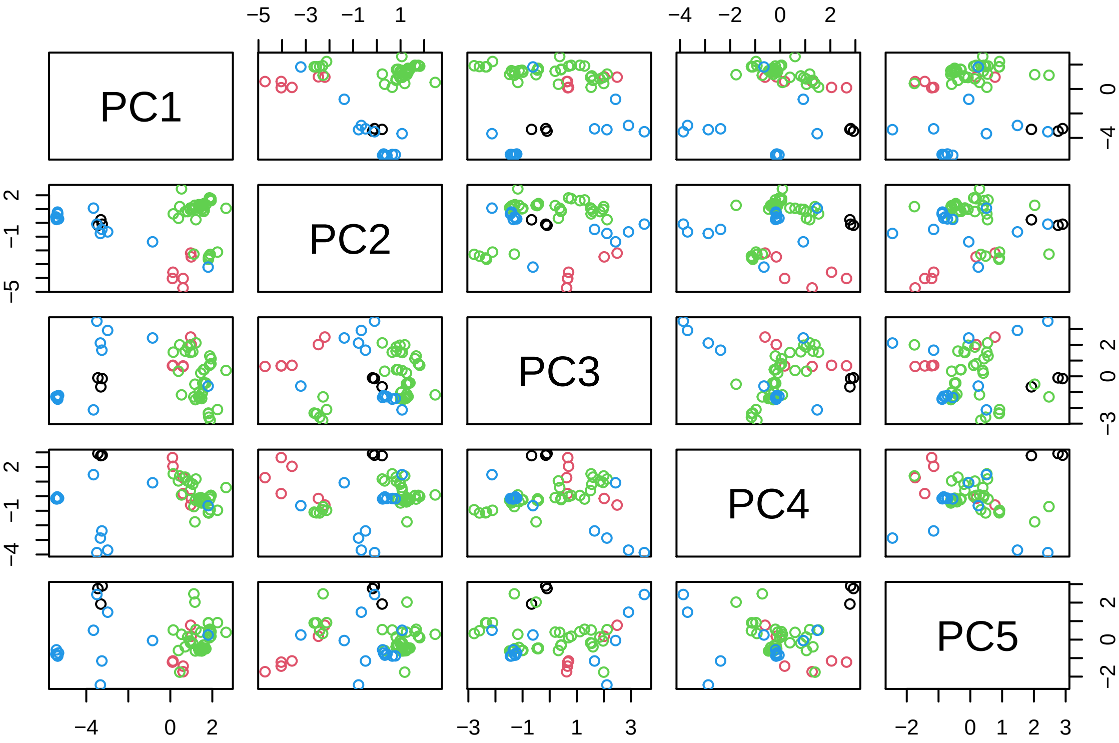
If you colour by groups (i.e. `col=groups` in plot) then black is horde, red is tobra, green is tobamo, blue is furo.

(b) Do the principal components show some separation between the viruses?

```
fviz_eig(allviruses.PCA)
```

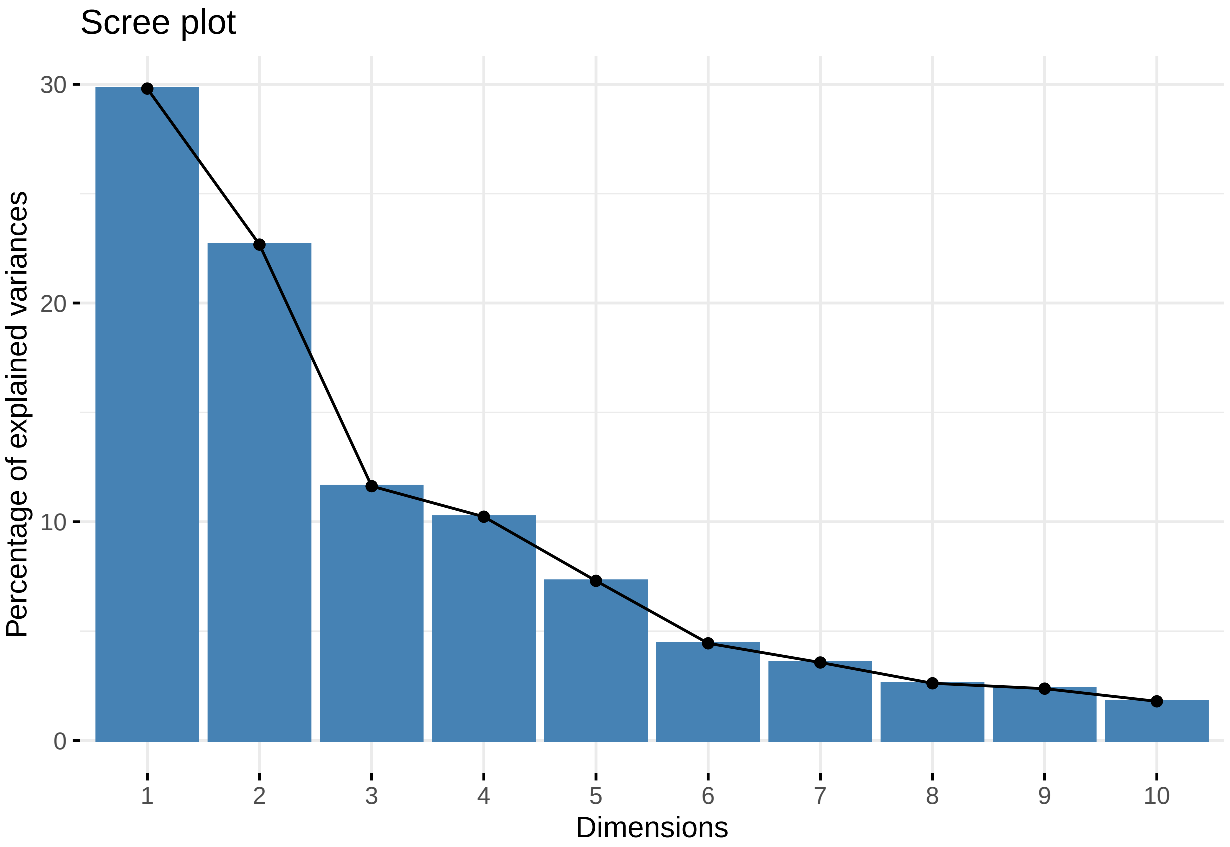


```
pairs(allviruses.PCA$x[,1:5], col = groups)
```

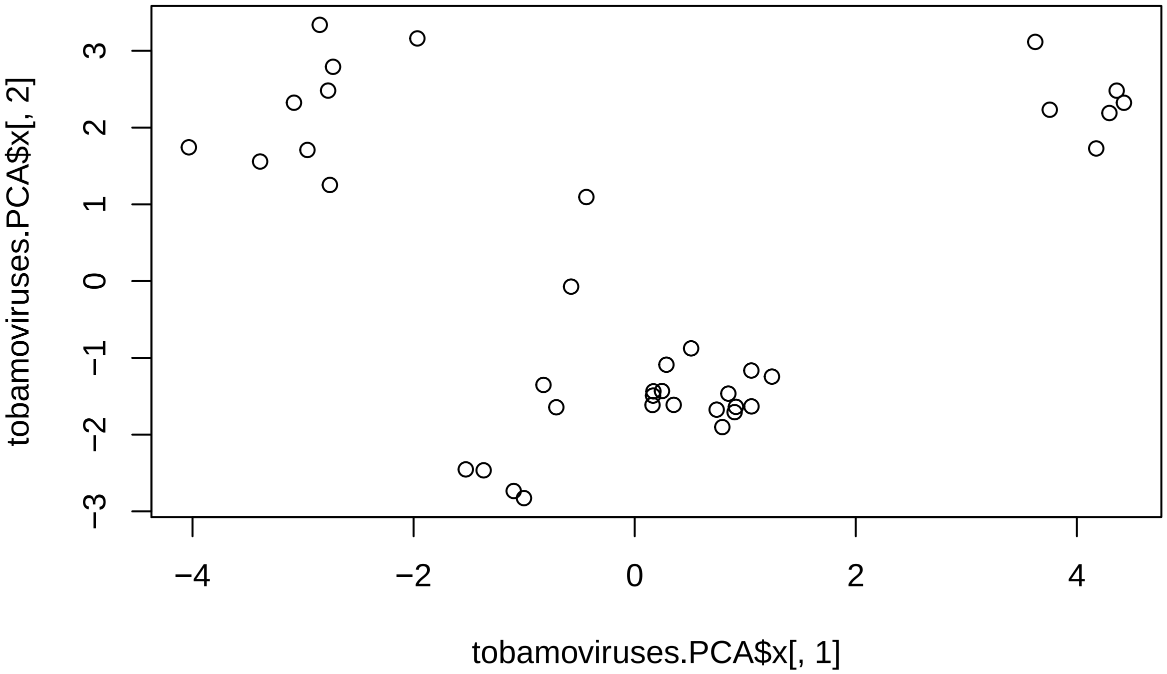


(c) The largest group of viruses is the tobamoviruses. Does a principal component analysis suggest there might be subgroups within this group of viruses?

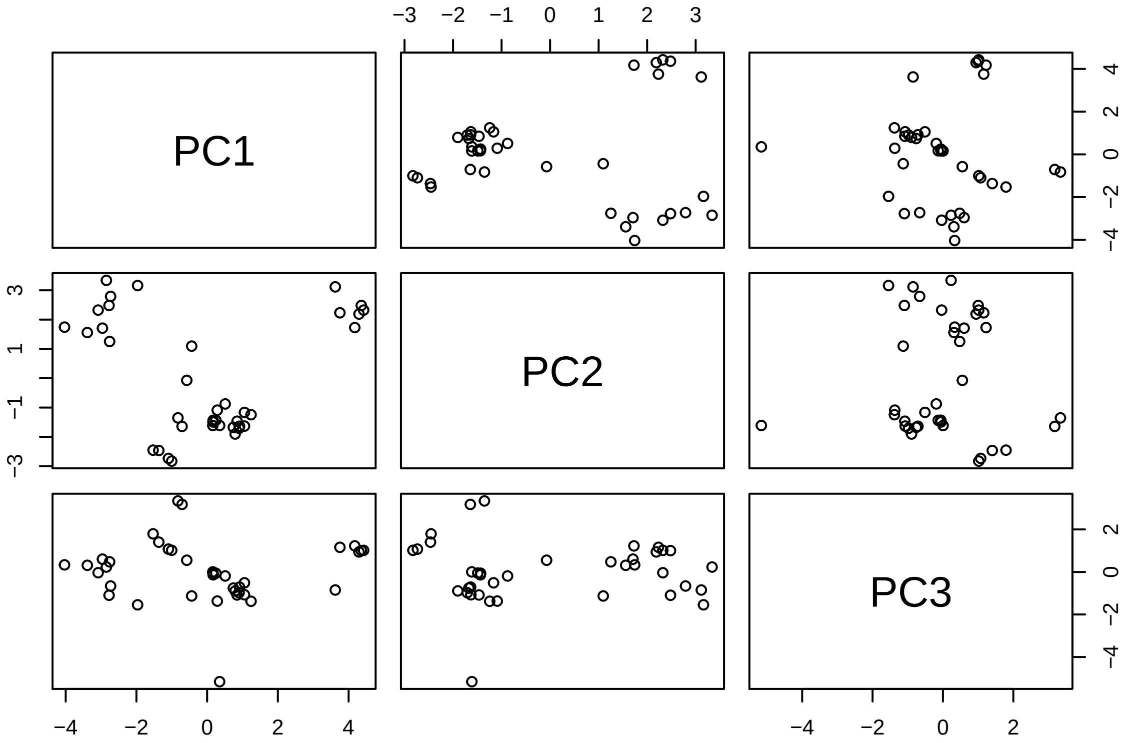
```
tobamoviruses.PCA = prcomp(tobamoviruses, center = T, scale. = T)

fviz_eig(tobamoviruses.PCA)
```



```
plot(tobamoviruses.PCA$x[,1], tobamoviruses.PCA$x[,2])
```



```r
pairs(tobamoviruses.PCA$x[,1:3])
```



From the plot it looks like we have 3 clusters, meaning that there are maybe groups that have different characteristics within the Tobamoviruses group of viruses.