Final project: PCA

Jose Antonio Alvarez Ocete

For fractions:

```
library(MASS)
options(digits=2)
```

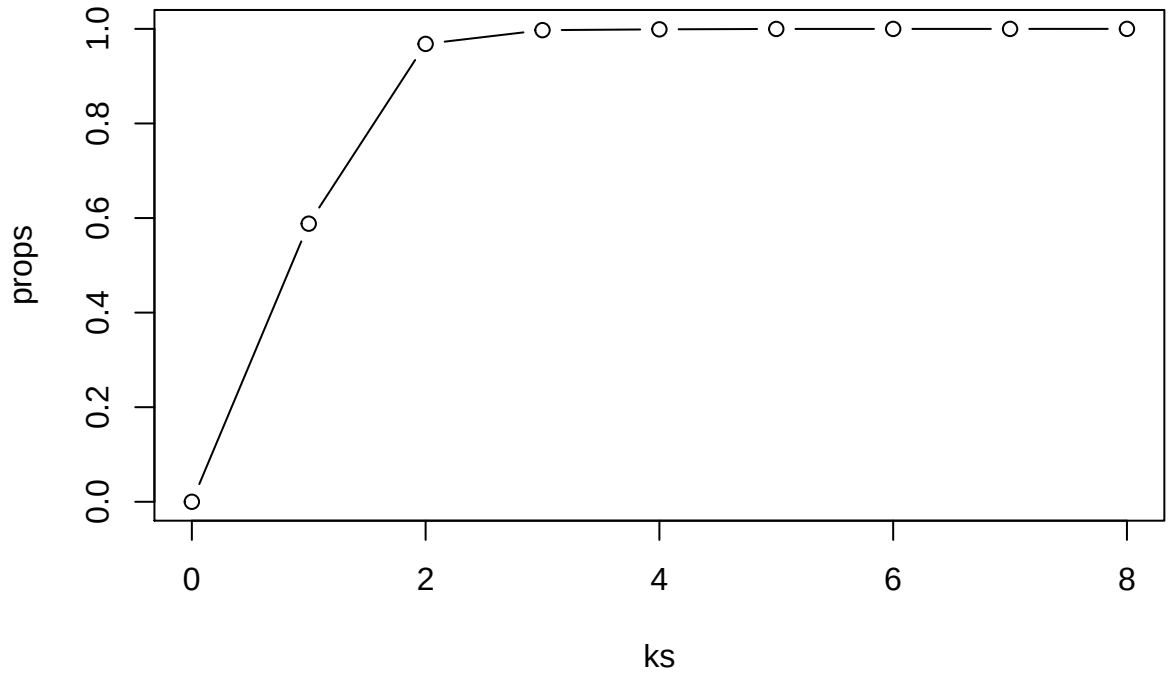
```
#import data, table T5-8.dat, see example 5.8
data <- read.table("data/cereal.dat")

r <- 8

#set up X
X <- data.matrix(data[,3:10])
colnames(X) <- c('Calories','Protein','Fat','Sodium','Fiber','Carbohydrates','Sugar','Potassium')
classes <- data.matrix(data[,11])
n <- length(X[,1])
```

(1.b) Determine the proportion of total sample variance due to the first sample principale component.

```
plotProportions<-function(eigen_values, print=FALSE) {
  ks <- c(0)
  props <- c(0)
  RR <- sum(eigen_values)

  for (k in 1:(r)) {
    ks <- c(ks, k)
    LL_r <- sum(eigen_values[1:k])
    props <- c(props, LL_r/RR)

    if (print)
      cat('k =',k,':', fractions(LL_r/RR), '=', LL_r, '/', RR, fill=TRUE)
  }
  plot(ks, props, type="b",xlim=c(0,r))
}
```

```
# Compute sample mean and S
Ones <- rep(1,n)
x_sample_mean <- 1/n * t(X)%*%Ones
S <- 1/(n-1) * t(X - Ones%*%t(x_sample_mean))%*%(X - Ones%*%t(x_sample_mean))

# eigens
ev <- eigen(S)
eigen_values <- ev$values
V <- ev$vectors

plotProportions(eigen_values, print=TRUE)
```

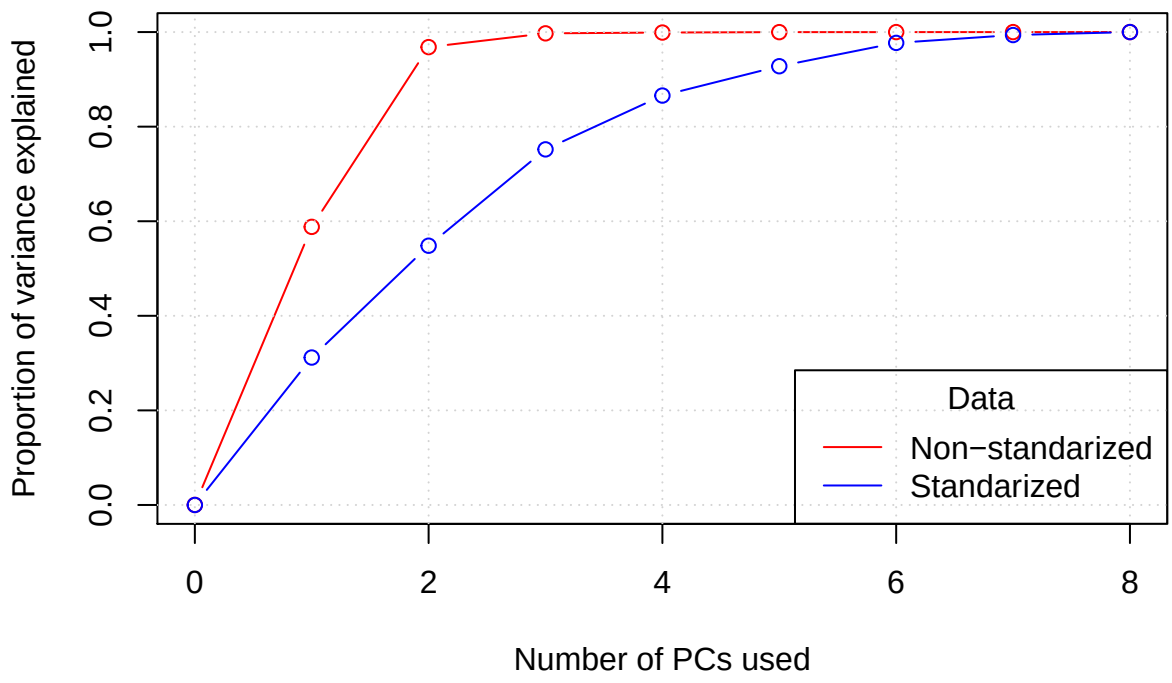
```
## k = 1 : 0.59 = 6500 / 11052
## k = 2 : 0.97 = 10701 / 11052
## k = 3 : 1 = 11022 / 11052
## k = 4 : 1 = 11041 / 11052
## k = 5 : 1 = 11051 / 11052
## k = 6 : 1 = 11051 / 11052
## k = 7 : 1 = 11052 / 11052
## k = 8 : 1 = 11052 / 11052
```



(1.e) Repeat with the data standarized. Aka, use R instead of S for the analysis.

```r
plotProportionsTogether<-function(eigen_values, eigen_values_z) {
  ks <- c(0)
  props <- c(0)
  props_z <- c(0)
  RR <- sum(eigen_values)
  RR_z <- sum(eigen_values_z)

  for (k in 1:(r+1)) {
    ks <- c(ks, k)
    LL_r <- sum(eigen_values[1:k])
    props <- c(props, LL_r/RR)
    LL_r_z <- sum(eigen_values_z[1:k])
    props_z <- c(props_z, LL_r_z/RR_z)
  }

  plot(ks, props, col="red", type="b", xlab = 'Number of PCs used', ylab = 'Proportion of variance expla
```

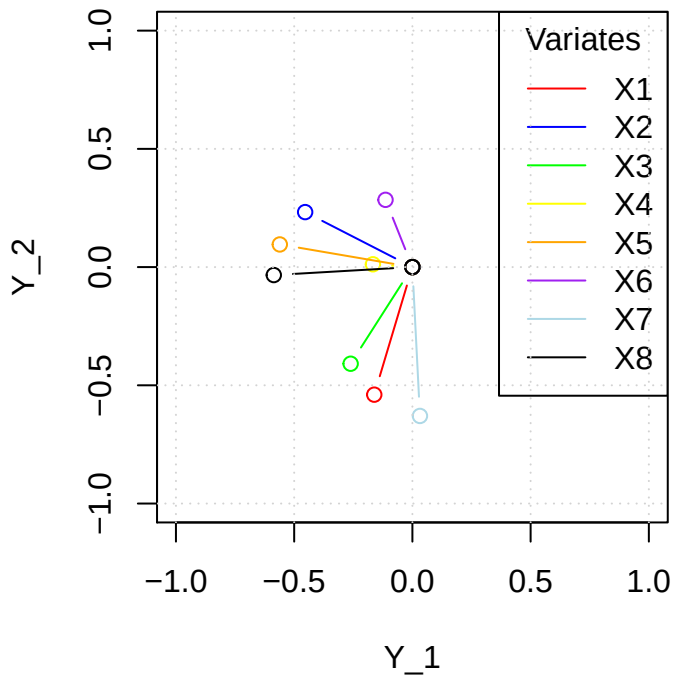
```
  grid()
  points(ks, props_z, type="b", col="blue")
  legend('bottomright', title='Data', legend=c('Non-standarized', 'Standarized'),
       col=c('red','blue'), lty=1)
}

# Com
R <- cor(X)

# eigens
ev_z <- eigen(R)
eigen_values_z <- ev_z$values
V_z <- ev_z$vectors

#plotProportions(eigen_values_z, print=TRUE)
plotProportionsTogether(eigen_values, eigen_values_z)
```



Plot PCs contributions

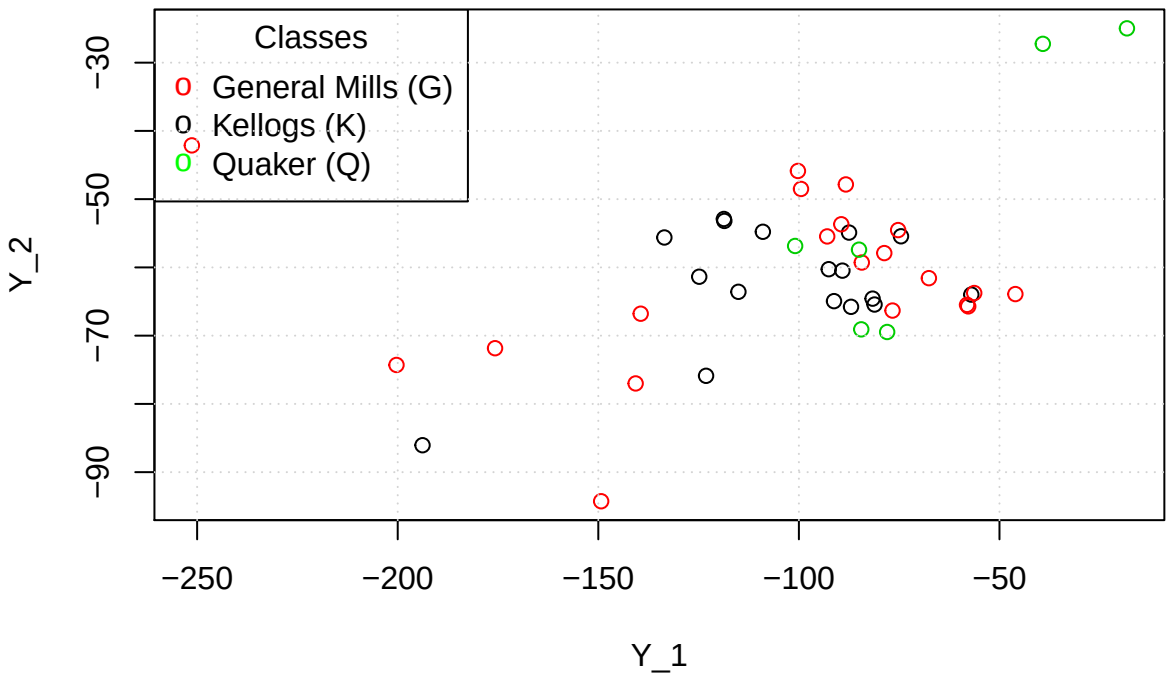
```
color <- c('red', 'blue', 'green', 'yellow', 'orange', 'purple', 'light blue', 'black')
par(pty="s")

for (i in 1:r) {
  vector_X <- c(0, V_z[i,1])
  vector_Y <- c(0, V_z[i,2])
```

```
  if (i==1)
    plot(vector_X, vector_Y, type='b', xlab='Y_1', ylab='Y_2', xlim=c(-1,1), ylim=c(-1,1), col=color[i])
  else
    lines(vector_X, vector_Y, type='b', col=color[i])
}
points(c(0), c(0))
legend("topright", legend=c('X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8'),
       col=color, lty=1, title="Variates")

grid()
```



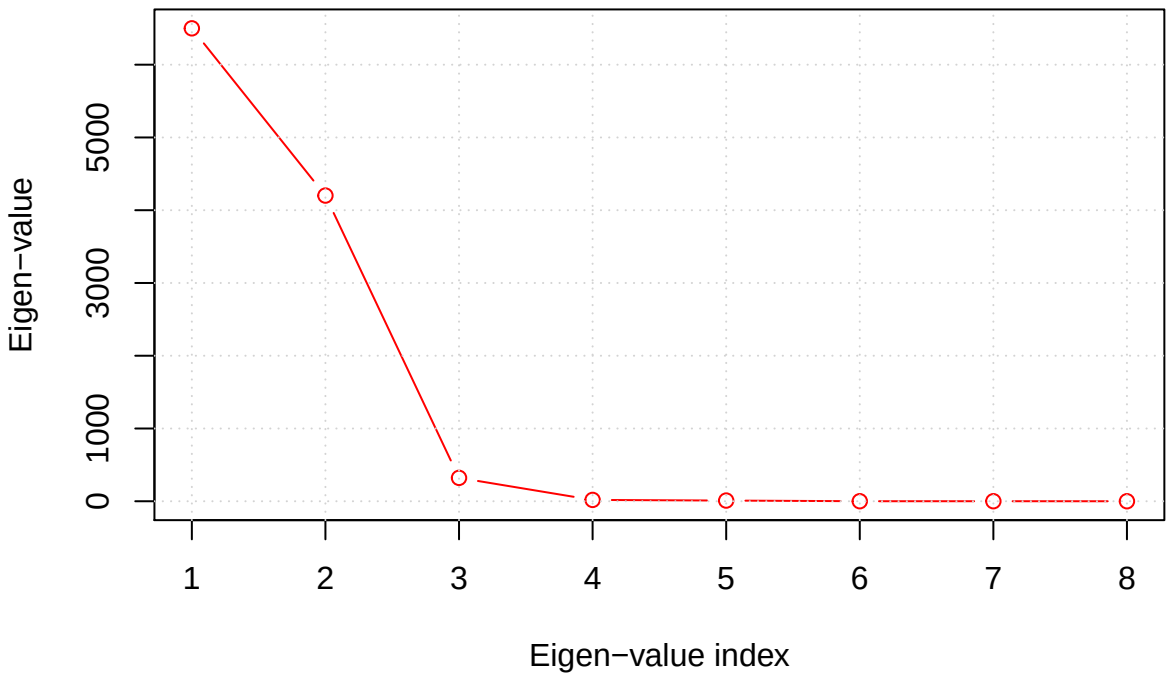
Plot PC scores

```
Y_scores <- X %*% V_z[,1:2]

#replace(classes, rep(length(classes)), c(rep('red',17), rep('blue',20), rep('green', 6)))
plot(Y_scores[,1], Y_scores[,2], xlab='Y_1', ylab='Y_2', col=classes)
legend("topleft", legend=c('General Mills (G)', 'Kellogs (K)', 'Quaker (Q)'),
       col=c('red', 'black', 'green'), pch='o', lty=0, title="Classes")
grid()
```

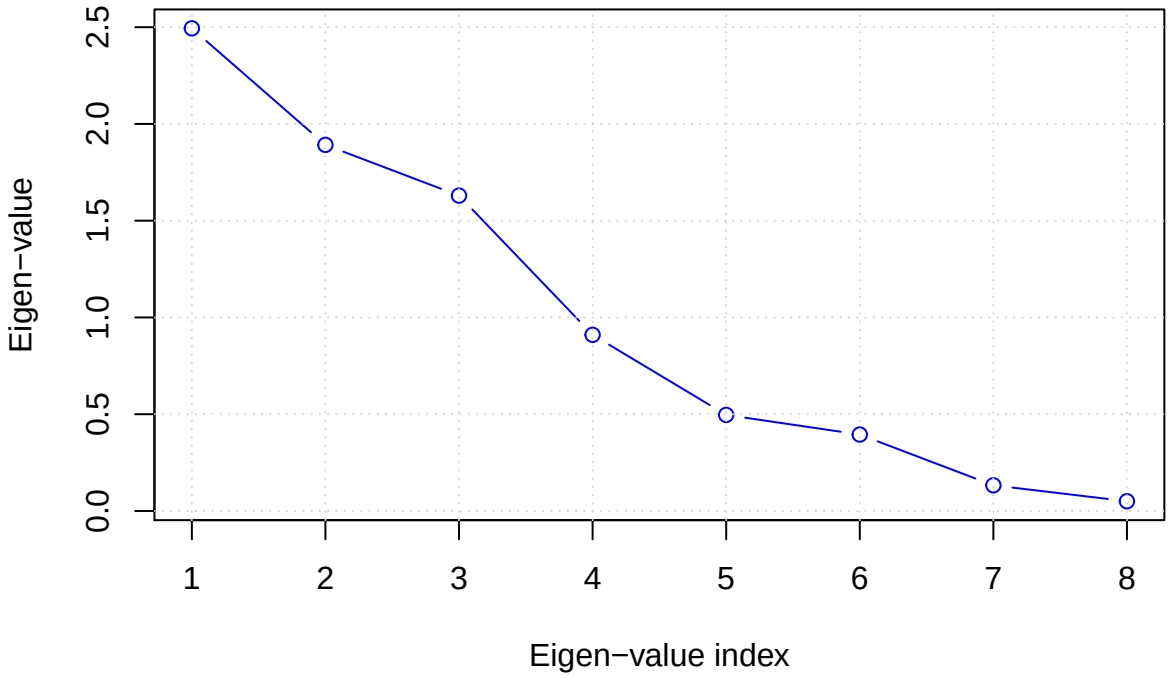


Plot eigen values

```
plot(rep(1:r), eigen_values, type='b', col='red', xlab='Eigen-value index', ylab='Eigen-value')
grid()
```




```
plot(rep(1:r), eigen_values_z, type='b', col='blue', xlab='Eigen-value index', ylab='Eigen-value')
grid()
```


Eigen−value
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7
8
Eigen−value index