Project 06

Jason Grahn

2/19/2019

knitr::opts\_chunk$set(  
 echo = TRUE,  
 message = FALSE,  
 warning = FALSE  
)  
library(magrittr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

probe <- read.table(here::here("assignment07/T3\_6\_PROBE.DAT")) %>%   
 rename(subject\_number = V1,  
 y1 = V2, # no explanation  
 y2 = V3, # no explanation  
 y3 = V4, # no explanation  
 y4 = V5, # no explanation  
 y5 = V6,) %>%   
 mutate(subject\_number = factor(subject\_number))# no explanation  
  
head(probe, 3)

## subject\_number y1 y2 y3 y4 y5  
## 1 1 51 36 50 35 42  
## 2 2 27 20 26 17 27  
## 3 3 37 22 41 37 30

# given the data has higher numbers, we should scale this

# First, variable selection  
probe.var <- probe %>% select(y1, y2, y3, y4)  
probe.subjects <- probe %>% select(subject\_number)

Do a principle component analysis of the data in Table 3.6 (page 79) You may use R to solve this part (NO built-in function).

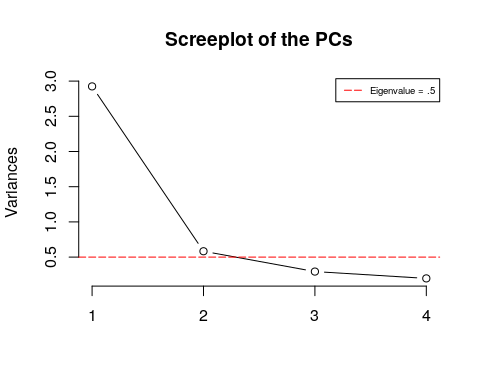
# scale, center, and apply PCA   
probe.pca <- prcomp(probe.var,  
 center = TRUE,  
 scale. = TRUE)   
# print method  
print(probe.pca)

## Standard deviations (1, .., p=4):  
## [1] 1.7101541 0.7636289 0.5427756 0.4445655  
##   
## Rotation (n x k) = (4 x 4):  
## PC1 PC2 PC3 PC4  
## y1 -0.5041678 0.4690392 -0.5015713 -0.5236824  
## y2 -0.4975351 -0.5183356 -0.4948926 0.4887416  
## y3 -0.4973780 0.5283284 0.4557798 0.5155085  
## y4 -0.5008879 -0.4818708 0.5438496 -0.4702547

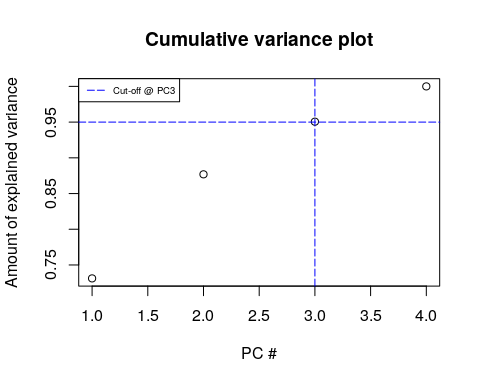
# summary method  
summary(probe.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4  
## Standard deviation 1.7102 0.7636 0.54278 0.44457  
## Proportion of Variance 0.7312 0.1458 0.07365 0.04941  
## Cumulative Proportion 0.7312 0.8769 0.95059 1.00000

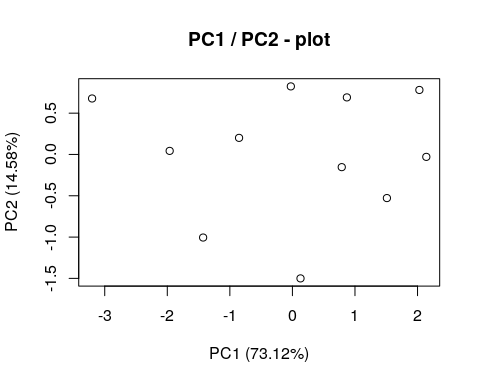
# plot method  
## borrowing some plotting code from   
## https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff   
screeplot(probe.pca, type = "l", main = "Screeplot of the PCs")  
abline(h = 0.5,   
 col="red",   
 lty=5)  
legend("topright",   
 legend=c("Eigenvalue = .5"),  
 col=c("red"),   
 lty=5,   
 cex=0.6)



cumpro <- round(cumsum(probe.pca$sdev^2 / sum(probe.pca$sdev^2)),4)  
plot(cumpro[0:4],   
 xlab = "PC #",   
 ylab = "Amount of explained variance",   
 main = "Cumulative variance plot")  
abline(v = 3, col="blue", lty=5)  
abline(h = 0.95, col="blue", lty=5)  
legend("topleft",   
 legend=c("Cut-off @ PC3"),  
 col=c("blue"), lty=5, cex=0.6)



plot(probe.pca$x[,1],  
 probe.pca$x[,2],   
 xlab="PC1 (73.12%)",   
 ylab = "PC2 (14.58%)",   
 main = "PC1 / PC2 - plot")



# Predict PCs  
predict(probe.pca,   
 newdata = tail(log(probe.var), 2))

## PC1 PC2 PC3 PC4  
## 10 7.134323 -0.8519285 0.2261655 -0.07706505  
## 11 7.186479 -0.8140774 0.2244972 -0.11762392