assignment07

Jason Grahn

2/19/2019

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).

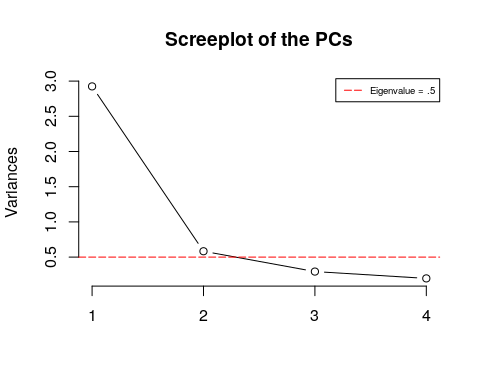
# scale, center, and apply PCA   
probe.pca <- prcomp(probe.var,  
 center = TRUE,  
 scale. = TRUE)   
# print method  
print(probe.pca)

## 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

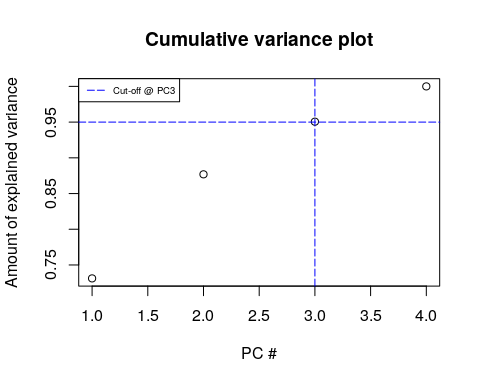
# summary method  
summary(probe.pca)

## 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

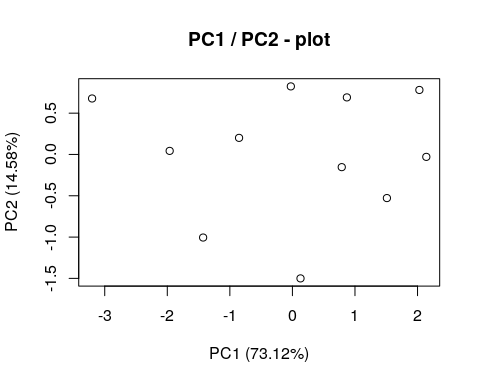
# plot method  
## borrowing some plotting code from   
## https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff   
screeplot(probe.pca, type = "l", main = "Screeplot of the PCs")  
abline(h = 0.5,   
 col="red",   
 lty=5)  
legend("topright",   
 legend=c("Eigenvalue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

# a) Use both S and R.

## Show your S and R matrix, and the corresponding eigenvalues and eigenvectors of S and R to get full credits

### Using R

# Center and scale  
probe.scaled <- scale(probe.var,   
 center = TRUE,   
 scale = TRUE)  
  
# 1. Correlation matrix  
res.cor <- cor(probe.scaled)  
(round(res.cor, 2))

## y1 y2 y3 y4  
## y1 1.00 0.61 0.76 0.58  
## y2 0.61 1.00 0.55 0.75  
## y3 0.76 0.55 1.00 0.61  
## y4 0.58 0.75 0.61 1.00

# 2. Calculate eigenvectors/eigenvalues  
(res.eig <- eigen(res.cor))

## eigen() decomposition  
## $values  
## [1] 2.9246270 0.5831291 0.2946053 0.1976385  
##   
## $vectors  
## [,1] [,2] [,3] [,4]  
## [1,] -0.5041678 0.4690392 -0.5015713 0.5236824  
## [2,] -0.4975351 -0.5183356 -0.4948926 -0.4887416  
## [3,] -0.4973780 0.5283284 0.4557798 -0.5155085  
## [4,] -0.5008879 -0.4818708 0.5438496 0.4702547

# 3. Calculate Propotion  
round(res.eig$values/(sum(res.eig$values)), 4)

## [1] 0.7312 0.1458 0.0737 0.0494

### Using S

# 1. Covariance matrix  
res.cov <- cov(probe.scaled)  
round(res.cov, 2)

## y1 y2 y3 y4  
## y1 1.00 0.61 0.76 0.58  
## y2 0.61 1.00 0.55 0.75  
## y3 0.76 0.55 1.00 0.61  
## y4 0.58 0.75 0.61 1.00

# 2. Calculate eigenvectors/eigenvalues  
(eigen(res.cov))

## eigen() decomposition  
## $values  
## [1] 2.9246270 0.5831291 0.2946053 0.1976385  
##   
## $vectors  
## [,1] [,2] [,3] [,4]  
## [1,] -0.5041678 -0.4690392 -0.5015713 0.5236824  
## [2,] -0.4975351 0.5183356 -0.4948926 -0.4887416  
## [3,] -0.4973780 -0.5283284 0.4557798 -0.5155085  
## [4,] -0.5008879 0.4818708 0.5438496 0.4702547

# 3. Calculate Propotion  
R.prop <- round(res.eig$values/(sum(res.eig$values)), 4)  
R.prop

## [1] 0.7312 0.1458 0.0737 0.0494

# b) Show the percent of variance explained.

round(sum(R.prop), 2)

## [1] 1

# alternatively, using the individual components  
round(R.prop[1] + R.prop[2] + R.prop[3] + R.prop[4], 2)

## [1] 1

# alternatively, using the model summary  
round(sum(summary(probe.pca)$importance[2,]), 2)

## [1] 1

Using all four PCA components appears to explain 100% of the variation of the data. This is a tiny dataset, so this isn’t completely out of the question; though it is highly unlikely in a real-world scenario.

# c) Decide how many components to retain. Show your reasons.

This is an assumptive decision made by the analyst. I would retain the first two components. The reasons I would retain only these two are:

* Makes for easy plotting.
* Accounts for enough variance within the data (87.7%) to be useful. Adding PC3 only provides 7% more explanation, which may not be necessary in context of data this size.
* Maintaining simplicity by sticking with two components versus 3 or more.
* We’ve reduced dimensions by 60% (from 5 down to 2), which for a larger dataset would be phenominal.

library("factoextra")  
fviz\_pca\_ind(probe.pca, geom.ind = "point", pointshape = 21,   
 pointsize = 2,   
 fill.ind = probe$subject\_number,   
 col.ind = "black",   
 palette = "jco",  
 label = "var",  
 col.var = "black",  
 repel = TRUE) +  
 ggtitle("2 Dimension PCA-plot from 5 feature dataset") +  
 theme\_linedraw() +  
 theme(plot.title = element\_text(hjust = 0.5))

