Appendix

**8.3**

$values

[1] 4 4 2

$vectors

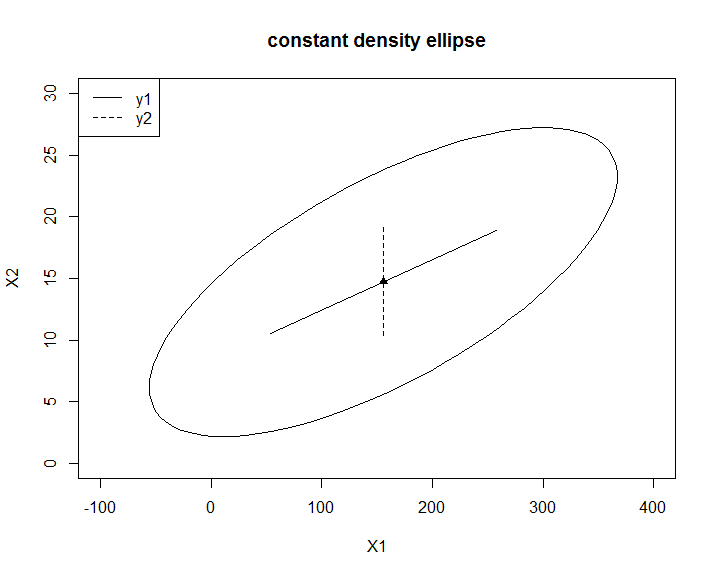
[,1] [,2] [,3]

[1,] 0 0 1

[2,] 0 1 0

[3,] 1 0 0

**8.6 c)**



**8.7 a)**

$values

[1] 1.6861434 0.3138566

$vectors

[,1] [,2]

[1,] 0.7071068 -0.7071068

[2,] 0.7071068 0.7071068

**8.18a)**

$values

[1] 5.80762446 0.62869342 0.27933457 0.12455472 0.09097174 0.05451882 0.01430226

$vectors

[,1] [,2] [,3] [,4] [,5] [,6] [,7]

[1,] -0.3777657 -0.4071756 -0.1405803 0.58706293 -0.16706891 0.53969730 0.08893934

[2,] -0.3832103 -0.4136291 -0.1007833 0.19407501 0.09350016 -0.74493139 -0.26565662

[3,] -0.3680361 -0.4593531 0.2370255 -0.64543118 0.32727328 0.24009405 0.12660435

[4,] -0.3947810 0.1612459 0.1475424 -0.29520804 -0.81905467 -0.01650651 -0.19521315

[5,] -0.3892610 0.3090877 -0.4219855 -0.06669044 0.02613100 -0.18898771 0.73076817

[6,] -0.3760945 0.4231899 -0.4060627 -0.08015699 0.35169796 0.24049968 -0.57150644

[7,] -0.3552031 0.3892153 0.7410610 0.32107640 0.24700821 -0.04826992 0.08208401

**d)**

> nation[(sort.int(new,index.return = TRUE))$ix,]

# A tibble: 54 × 1

country

<chr>

1 USA

2 GER

3 RUS

4 CHN

5 FRA

6 GBR

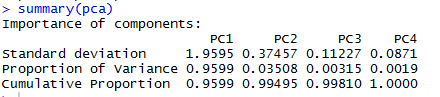
7 CZE

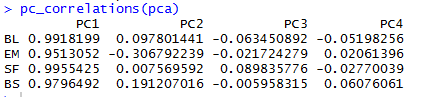
8 POL

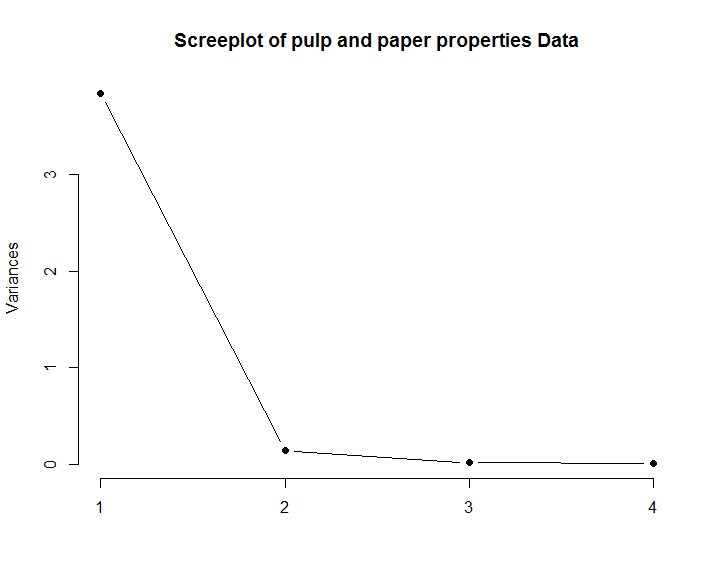
9 ROM

10 AUS

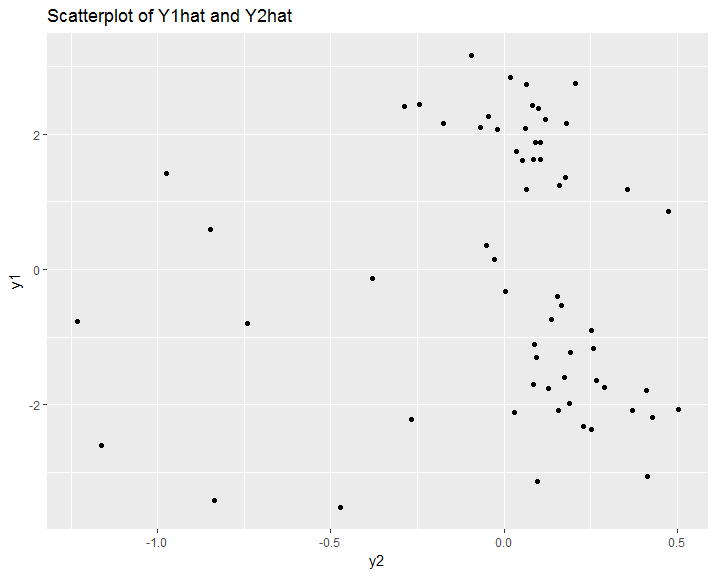
**8.27 a)**







**d)**



**Codes**

library(rstudioapi)

library(ellipse)

library(ggplot2)

this.dir <- dirname(rstudioapi::getActiveDocumentContext()$path)

setwd(this.dir)

##### 8.3 ######

eigen(as.matrix(rbind(c(2,0,0),c(0,4,0),c(0,0,4))))

##### 8.6 #####

s = as.matrix(cbind(c(7476.45,303.62),c(303.62,26.19)))

eigen = eigen(s)

xbar =c(155.6,14.7)

plot(ellipse(s,centre =xbar),type = 'l',xlim=c(-100,400),ylim=c(0,30),xlab = "X1",ylab = "X2",main = "constant density ellipse")

points(xbar[1],xbar[2],pch = 17)

c = sqrt(1.4)

ax1=c\*sqrt(eigen$values[1])\*eigen$vectors[,1]

ax2=c\*sqrt(eigen$values[2])\*eigen$vectors[,2]

lines(c(xbar[1]-ax1[1],xbar[1]+ax1[1]),c(xbar[2]-ax1[2],xbar[2]+ax1[2]),lty=1)

lines(c(xbar[1]-ax2[1], xbar[1]+ax2[1]),c(xbar[2]-ax2[2],xbar[2]+ax2[2]),lty=2)

legend(x = "topleft", c("y1","y2"), lty = c(1,2))

##### 8.7 #####

p = cov2cor(s)

eigen = eigen(p)

##### 8.18 #####

library(readr)

e8\_18 <- read\_delim("e8\_18.txt", " ", escape\_double = FALSE, trim\_ws = TRUE)

nation = e8\_18[,1]

X = as.matrix(e8\_18[,2:8])

r = cor(X)

eigen = eigen(r)

firstpc = -1\*eigen$vectors[,1]

stdX <- scale(X)

new = stdX%\*%firstpc

nation[(sort.int(new,index.return = TRUE))$ix,]

##### 8.27 #####

#assuming PCA output is based on correlation matrix (or variables of unit variance)

pc\_correlations <- function(pca\_output) {

p <- ncol(pca\_output$rotation)

pccor=matrix(NA, nrow=p, ncol=p)

for(i in 1:p){

for(k in 1:p){

pccor[k,i]=pca\_output$rotation[k,i]\*pca\_output$sdev[i]

}

}

colnames(pccor) <- paste0("PC", 1:p)

rownames(pccor) <- rownames(pca\_output$rotation)

pccor

}

e8\_27 <- read\_delim("e8\_27.txt",

" ", escape\_double = FALSE, trim\_ws = TRUE)

X = scale(as.matrix(e8\_27))

pca <- prcomp(x = X, retx = TRUE, center = TRUE, scale. = TRUE)

summary(pca)

pc\_correlations(pca)

screeplot(pca, type = "lines", pch = 19,

main = "Screeplot of pulp and paper properties Data")

e8\_27$y1 = X%\*%pca$rotation[,1]

e8\_27$y2 = X%\*%pca$rotation[,2]

ggplot(data = e8\_27, aes(x = y2, y = y1)) + geom\_point()+ggtitle("Scatterplot of Y1hat and Y2hat")