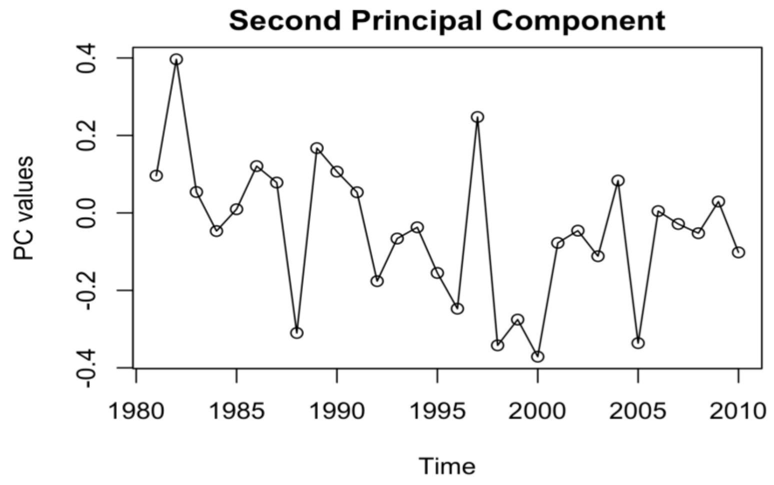
Solution of Math 524 Final
Fall 2023
By Sam Shen

#Problem 1.

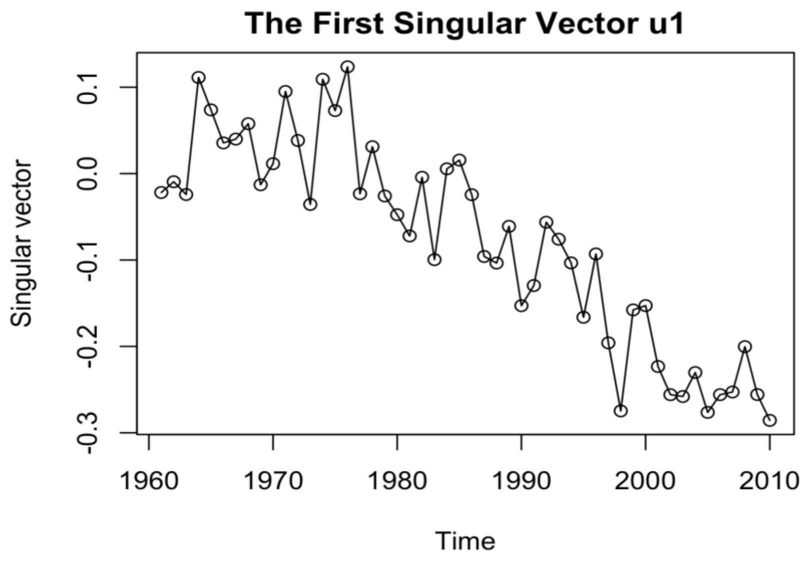
```
#(a) Extract data and form a 4x30 data matrix
setwd("/Users/sshen/Desktop/teach/524LinAlg2023Fall/Exams/Final")
dat = read.csv("NOAAGlobalT.csv", header = TRUE)
dim(dat)
#[1] 2592 1648

dat[1:2,1:5]
#  X   LAT LON X1880.1 X1880.2
#1 1 -87.5 2.5  -999.9  -999.9
#2 2 -87.5 7.5  -999.9  -999.9

#Extract Darwin data:
#12.46S, 130.84E, Dec 1981-2010
indexdarwin = which(dat$LAT > -15 & dat$LAT < -10 &
                    dat$LON > 130 & dat$LON < 135)
DarwinDat = dat[indexdarwin,]
DarwinDat[1:5]
#        X  LAT   LON X1880.1 X1880.2
#1107 1107 -12.5 132.5 -0.8558 -0.8591
DecIndex = seq(1227, 1575, by = 12)
DecIndex #Dec 1981- 2010, 30 years of Dec
#[1] 1227 1239 1251 1263 1275 1287
length(DecIndex)
#[1] 30
DarwinDat[DecIndex] #Verify the Dec of 1981 to 2010
#X1981.12 X1982.12 X1983.12 X1984.12 X1985.12 X1986.12 X1987.12
#0.0991   0.2114   0.2056   0.1335   0.6044   0.4491   -5e-04
DarwinDec =  DarwinDat[DecIndex]
plot(1981:2010, DarwinDec, type = 'o')
min(DarwinDec)
#[1] -0.6277 # not -999.9, i.e., no missing data

#Extract Tahiti data:
#17.65S, 149.43W, Dec 1981-2010
#149.43W is equivalent to 210.57E
indexTahiti = which(dat$LAT > -20 & dat$LAT < -15 &
                    dat$LON > 210 & dat$LON < 215)
TahitiDat = dat[indexTahiti,]
TahitiDat[1:5]
#        X   LAT   LON X1880.1 X1880.2
#1051 1051 -17.5 212.5  -999.9  -999.9
```

```
DecIndex = seq(1227, 1575, by = 12)
DecIndex #Dec 1981- 2010, 30 years of Dec
#[1] 1227 1239 1251 1263 1275 1287
length(DecIndex)
#[1] 30
TahitiDat[DecIndex] #Verify the Dec of 1981 to 2010
#X1981.12 X1982.12 X1983.12 X1984.12 X1985.12 X1986.12
#0.1363  -0.2046   0.3475  -0.2323   0.3948   0.0017
TahitiDec =  TahitiDat[DecIndex]
plot(1981:2010, TahitiDec, type = 'o')
min(TahitiDec)
#[1] -0.2491 # not -999.9, i.e., no missing data

#Extract Hanga Roa data:
#27.15S, 109.43W , Dec 1981-2010
#109.43W is equivalent to 250.57E
indexHanga = which(dat$LAT > -30 & dat$LAT < -25 &
                    dat$LON > 250 & dat$LON < 255)
HangaDat = dat[indexHanga,]
HangaDat[1:5]
#       X   LAT   LON X1880.1 X1880.2
#915 915 -27.5 252.5  -999.9  0.39991051 1051 -17.5 212.5
DecIndex = seq(1227, 1575, by = 12)
DecIndex #Dec 1981- 2010, 30 years of Dec
#[1] 1227 1239 1251 1263 1275 1287
length(DecIndex)
#[1] 30
HangaDat[DecIndex] #Verify the Dec of 1981 to 2010
#X1981.12 X1982.12 X1983.12 X1984.12 X1985.12 X1986.12
#-0.2055   -0.853  -0.3602   0.3465  -0.3267  -0.1755
HangaDec =  HangaDat[DecIndex]
plot(1981:2010, HangaDec, type = 'o')
min(HangaDec)
#[1] -0.853 # not -999.9, i.e., no missing data


#Extract Quito data:
#0.18S, 78.47W  , Dec 1981-2010
#78.47W is equivalent to 281.53E
indexQuito = which(dat$LAT > -5 & dat$LAT < 0 &
                   dat$LON > 280 & dat$LON < 285)
QuitoDat = dat[indexQuito,]
QuitoDat[1:5]
#       X   LAT   LON X1880.1 X1880.2
#1281 1281 -2.5 282.5  -999.9  -999.9
DecIndex = seq(1227, 1575, by = 12)
DecIndex #Dec 1981- 2010, 30 years of Dec
#[1] 1227 1239 1251 1263 1275 1287
length(DecIndex)
#[1] 30
QuitoDat[DecIndex] #Verify the Dec of 1981 to 2010
#X1981.12 X1982.12 X1983.12 X1984.12 X1985.12 X1986.12
#0.3853   0.6612  -0.1092   0.0737  -0.5695   0.2506
QuitoDec =  QuitoDat[DecIndex]
plot(1981:2010, QuitoDec, type = 'o')
min(QuitoDec)
#[1] -0.5856 # not -999.9, i.e., no missing data

mat = rbind(DarwinDec, TahitiDec, HangaDec, QuitoDec)
#mat is the 4-by-30 data matrix
dim(mat)
#[1]  4 30
A = mat[,1:5]
A #The first five columns of the 4-by-30 data matrix
#     X1981.12 X1982.12 X1983.12 X1984.12 X1985.12
#1107   0.0991   0.2114   0.2056   0.1335   0.6044
#1051   0.1363  -0.2046   0.3475  -0.2323   0.3948
#915   -0.2055  -0.8530  -0.3602   0.3465  -0.3267
#1281   0.3853   0.6612  -0.1092   0.0737  -0.5695

#(b)
svdmat = svd(mat)
svdmat$d #The four singular values
#[1] 3.014992 2.623243 1.814168 1.323533

#(c)
plot(1981:2010, svdmat$v[,2], type = 'o',
    xlab = "Time", ylab = "PC values",
    main = 'Second Principal Component')
```

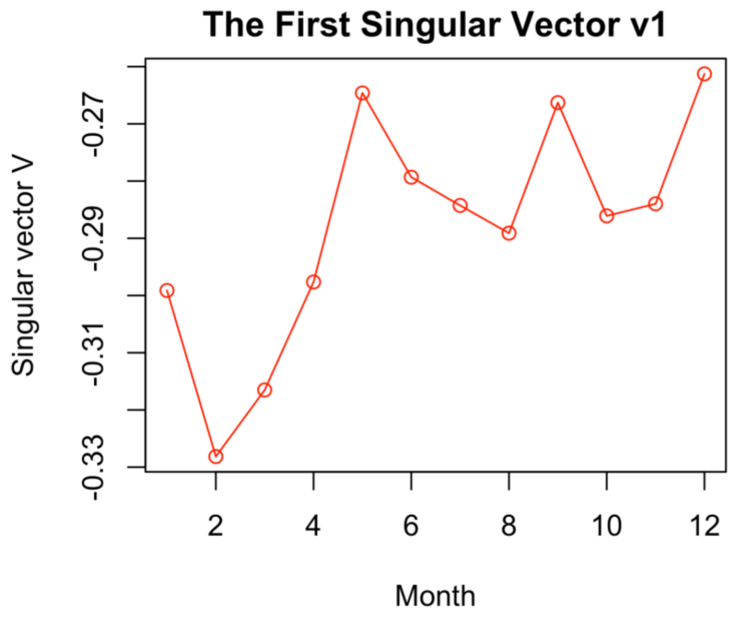


#Problem 2

```
#(a) Extract data and form a 4x30 data matrix
setwd("/Users/sshen/Desktop/teach/524LinAlg2023Fall/Exams/Final")
dat = read.csv("EarthTemperatureData.csv", header = TRUE)
dim(dat)
#[1] 166  14
dat[1:2, 1:5]
#  YEAR    JAN    FEB    MAR    APR
#1 1850 -0.702 -0.284 -0.732 -0.570
#2 1851 -0.303 -0.362 -0.485 -0.445
#data from 1850 to 2015 (166 rows/166 years)
#YEAR JAN to DEC and ANN (14 columns)
dat[112:161,1] #50 years from 1961 to 2010
#[1] 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973
#[14] 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986
#[27] 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999
#[40] 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010

mat = dat[112:161, 2:13]
dim(mat)
#[1] 50 12
#mat is the 50x12 data matrix
#The first a few rows and columns are shown below
mat[1:3, 1:5]
#      JAN   FEB    MAR    APR    MAY
#112  0.046 0.185  0.096  0.097  0.087
#113  0.055 0.139  0.027  0.025 -0.044
#114 -0.050 0.152 -0.142 -0.067 -0.021


#(b)
svdmat = svd(mat)
svdmat$d #12 singular values
#[1] 6.8082867 1.4728576 0.8167879 0.6921315 0.6476219 0.5834869
#[7] 0.4662021 0.3660210 0.3141819 0.2410995 0.2067049 0.1951964



#(c)
plot(1961:2010, svdmat$u[,1], type = 'o',
     xlab = "Time", ylab = "Singular vector",
     main = 'The First Singular Vector u1')
```

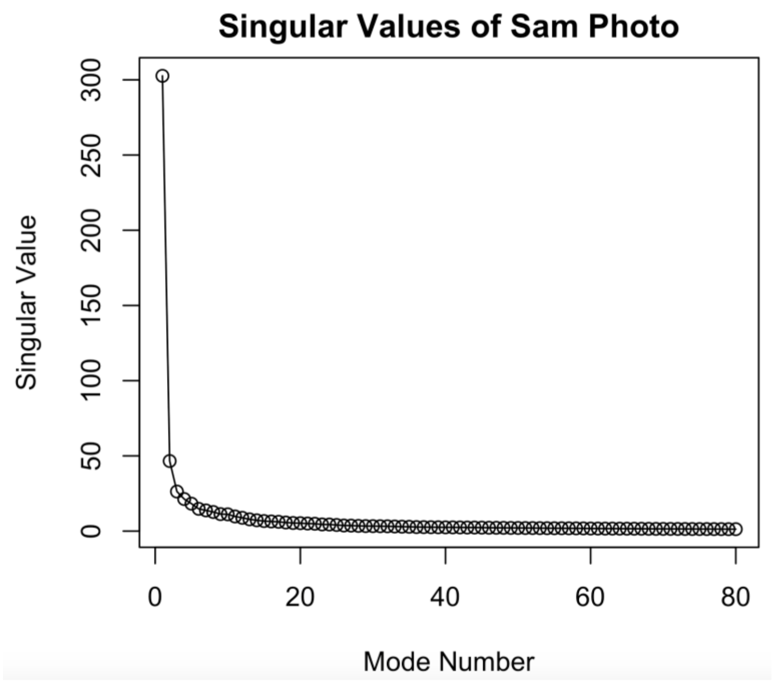


```
#(d)
plot(1:12, svdmat$v[,1], type = 'o',
     col='red',
     xlab = "Month", ylab = "Singular vector V",
     main = 'The First Singular Vector v1')
```

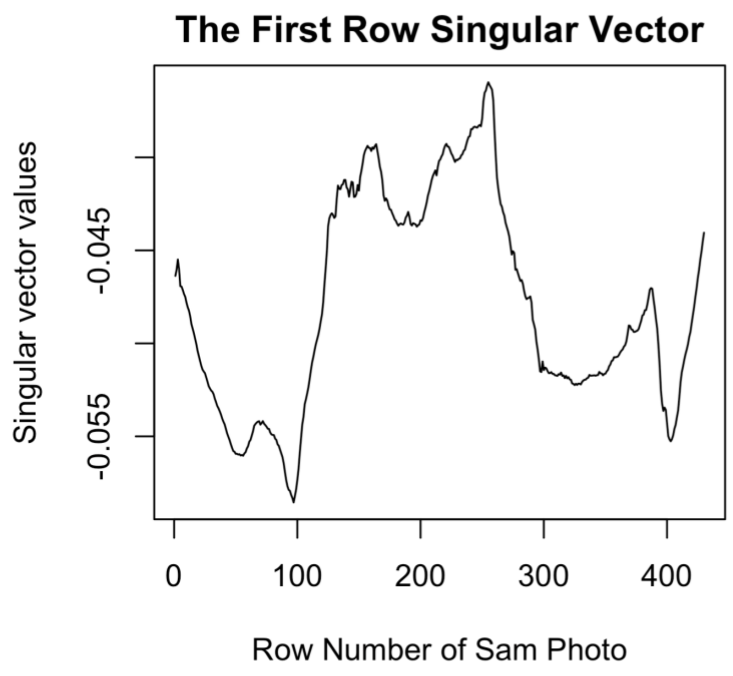


#Problem 3

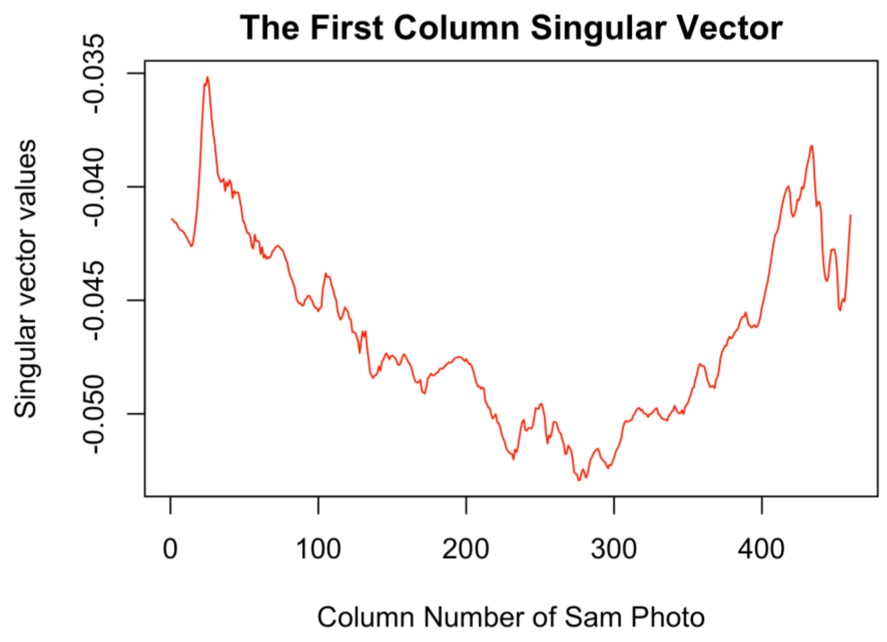
```
#(a) Input the image and produce data
setwd("/Users/sshen/Desktop/teach/524LinAlg2023Fall/Exams/Final")
#install.packages("imager")
library(imager)
dat <- load.image('sam.png')
graydat = grayscale(dat)[,,1,1]
dim(graydat)
#[1] 430 460
#graydat is the space-space data matrix

#(b)
mat = graydat
svdmat = svd(mat)
plot(1:80, svdmat$d[1:80], type = 'o',
     xlab = 'Mode Number', ylab = 'Singular Value',
     main = 'Singular Values of Sam Photo')
```



```
#(c)
plot(svdmat$u[,1], type = 'l',
     xlab = 'Row Number of Sam Photo',
     ylab = 'Singular vector values',
     main = 'The First Row Singular Vector')
```



```
#(d)
plot(svdmat$v[,1], type = 'l', col = 'red',
     xlab = 'Column Number of Sam Photo',
     ylab = 'Singular vector values',
     main = 'The First Column Singular Vector')
```



#Problem 4

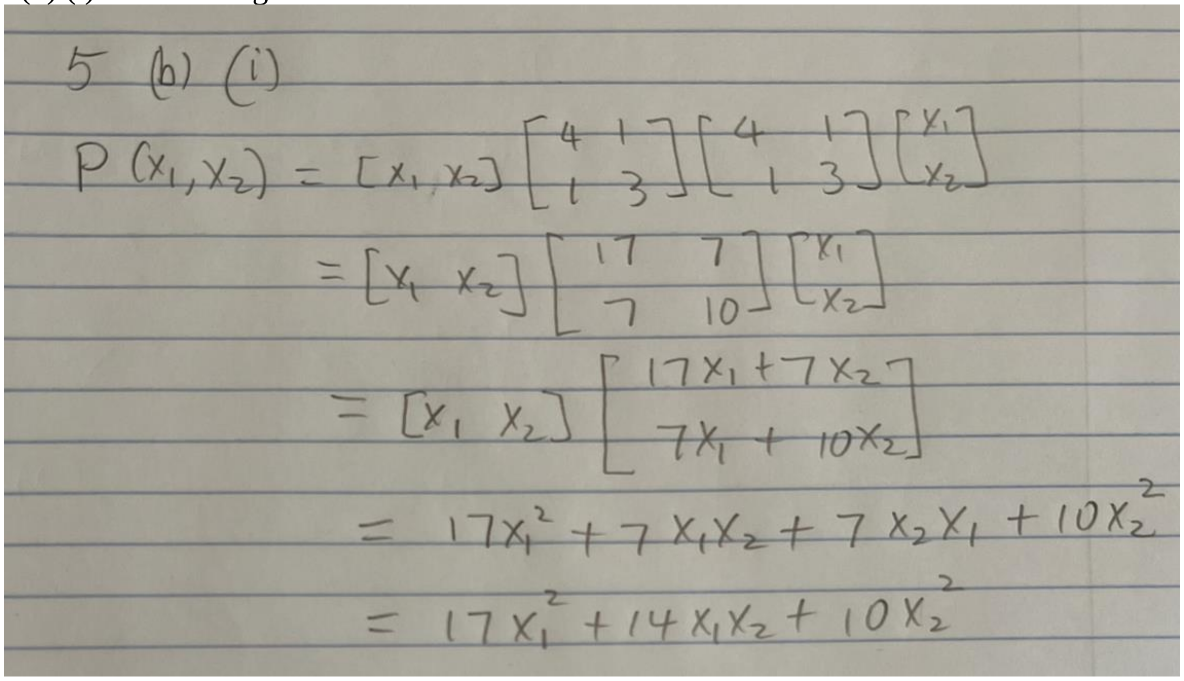
```
A = matrix(c(1, 4, 7, 2, 5, 0, 3, 6, 9),
           nrow = 3)
b = c(2, 5, 0)
x = solve(A, b)
x
#[1] 0 1 0
#x1 = 0, x2 = 1, x3 = 0 is the solution
```

#Problem 5
#(a)
#The answer is no.

#Proof:
#For the first three vectors v1, v2, v3 of any given five,
#there are only two cases:
#Case 1: v1, v2, v3 are not linearly independent.
#Then conclusion has reached, and there is no need to prove further.

#Case 2: v1, v2, v3 are linearly independent. Because v1, v2, v3 are
#in 3-dimensional space, three independent vectors can form
#a basis. The 4th vector, denoted by v4, must be able to
#represented at a linear combination of the basis vectors,
#i.e., there exist constants c1, c2, c3 such that
# v4 = c1 v1 + v2 v2 + c3 v3.
#Thus, v1, v2, v3 and v4 are not linearly independent.
#Thus, v1, v2, v3, v4, and v5 must not be linearly independent.
#The proof is complete.
#My example of a spatial case
#v1 = (1, 0, 0), v2 = (0, 1, 0), v3 = (0, 0, 1)
#are linearly independent and form a basis for 3D
#v4 = (2, 1, 0) = 2v1 + 1v2 + 0v3
#v5 = (-1, 5, 2)
#v1, v2, v3, v4 and v5 are not linearly independent.

#(b) (i) hand-writing



5 (b) (i)

$$P(x_1, x_2) = [x_1, x_2] \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= [x_1 \ x_2] \begin{bmatrix} 17 & 7 \\ 7 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= [x_1 \ x_2] \begin{bmatrix} 17x_1 + 7x_2 \\ 7x_1 + 10x_2 \end{bmatrix}$$

$$= 17x_1^2 + 7x_1x_2 + 7x_2x_1 + 10x_2^2$$

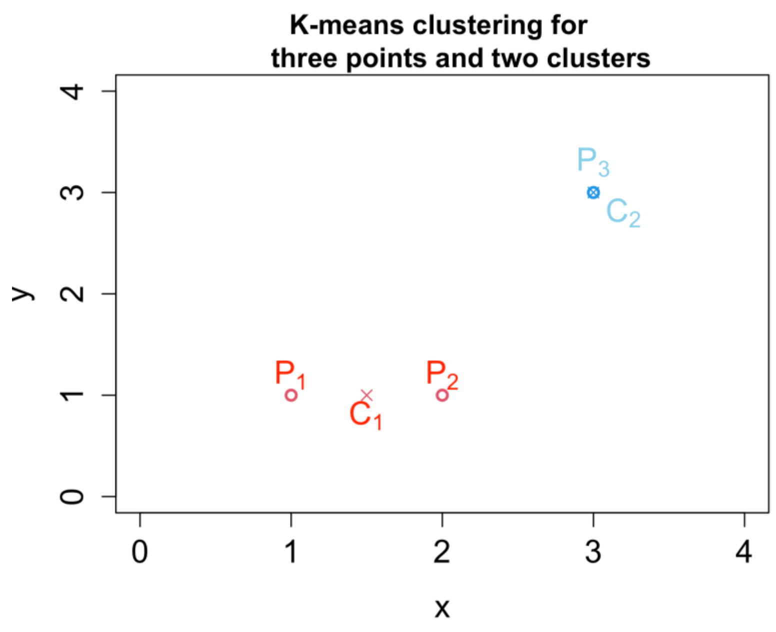
$$= 17x_1^2 + 14x_1x_2 + 10x_2^2$$

```
#(ii)
A = matrix(c(4, 1, 1, 3), nrow = 2)
evA = eigen(A)

#Eigenvectors
evA$vectors
#           [,1]       [,2]
#[1,] -0.8506508  0.5257311
#[2,] -0.5257311 -0.8506508

#Eigenvalues
evA$values
#[1] 4.618034 2.381966

#(iii)
x = evA$vectors[,1]
x
#[1] -0.8506508 -0.5257311
P = t(x)%*%A%*%t(A)%*%x
P
#         [,1]
#[1,] 21.32624
#i.e.,
#P(-0.8506508, -0.5257311) = 21.32624
```

#Problem 6

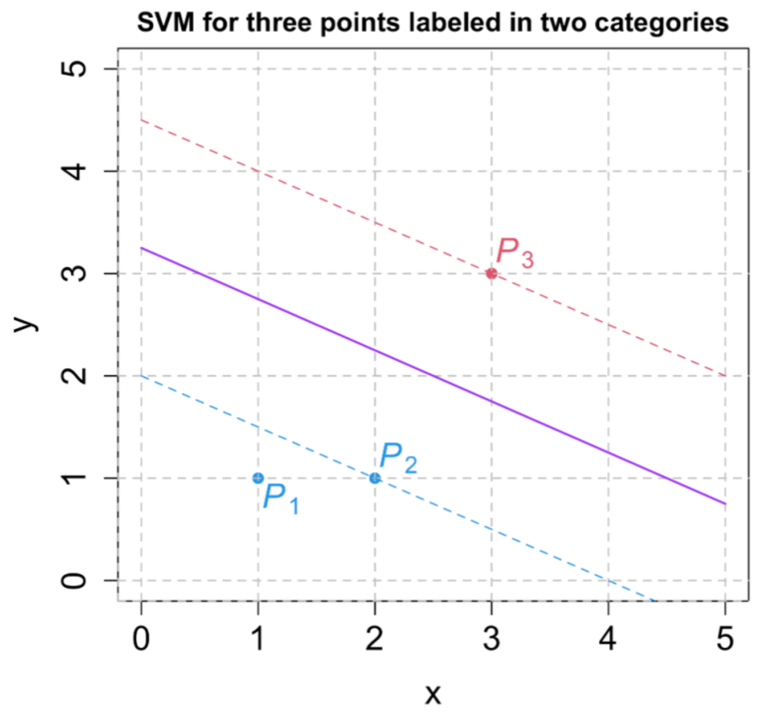
```
#(a)
#There are only 3 different cases
#Case 1: K1 = P1, P2; K2 = P3
#Case 2: K1 = P2, P3; K2 = P1
#Case 3: K1 = P3, P1; K2 = P2
#We compuete tWCSS for each case and
#see which case has the smallest tWCSS,
#which is the K-means solution.


#Coordinates data
N=3; K =2
mydata <- matrix(c(1, 1, 2, 1, 3, 3),
            nrow = N, byrow = TRUE)
mydata
x1 = mydata[1, ]
x1
x2 = mydata[2, ]
x2
x3 = mydata[3, ]
x3

#tWCSS calculation for N = 3 and K = 2
#Case 1: C1 = (P1 + P2)/2 = (1.5, 1)
#C2 = P3 = (3, 3)

#Case 1: K1 = (P1, P2)
c1 = (mydata[1, ] + mydata[2, ])/2
c1
c2 = mydata[3, ]
c2
tWCSS = norm(x1 - c1, type = '2')^2 +
  norm(x2 - c1, type = '2')^2 +
  norm(x3 - c2, type = '2')^2
tWCSS
#[1] 0.5

#Case 2: K1 = (P2, P3)
c1 = (mydata[2, ] + mydata[3, ])/2
c2 = mydata[1, ]
norm(x2 - c1, type = '2')^2 +
  norm(x3 - c1, type = '2')^2 +
  norm(x1 - c2, type = '2')^2
#[1] 2.5
```

```
#Case 3: K1 = (P1, P3)
c1 = (mydata[1, ] + mydata[3, ])/2
c2 = mydata[2, ]
norm(x1 - c1, type = '2')^2 +
  norm(x3 - c1, type = '2')^2 +
  norm(x2 - c2, type = '2')^2
#[1] 4

#Thus, Case 1: K1 = P1 and P2; K2 = P3
#has the smallest tWCSS = 0.5
#Case 1 is the K-means solution:
#Cluster 1 (P1, P2), Cluster 2 (P3).

#The case C1 = (P1, P2) can be quickly found by
kmeans(mydata, 2)
#Clustering vector:
#[1] 1 1 2 #points P1, P2 in C1
#Within cluster sum of squares by cluster:
#[1] 0.5 0.0 #tWCSS = 0.5 + 0 = 0.5



#(b)
par(mar = c(4,4,2.5,0.5))
plot(mydata[,1], mydata[,2], lwd = 2,
     xlim =c(0, 4), ylim = c(0, 4),
     xlab = 'x', ylab = 'y', col = c(2, 2, 4),
     main = 'K-means clustering for
     three points and two clusters',
     cex.lab = 1.4, cex.axis = 1.4)
points(Kclusters$centers[,1], Kclusters$centers[,2],
       col = c(2, 4), pch = 4)
text(1.5, 0.8, bquote(C[1]), col = 'red', cex = 1.4)
text(3.2, 2.8, bquote(C[2]), col = 'skyblue', cex = 1.4)
text(1, 1.2, bquote(P[1]), cex = 1.4, col = 'red')
text(2, 1.2, bquote(P[2]), cex = 1.4, col = 'red')
text(3, 3.3, bquote(P[3]), cex = 1.4, col = 'skyblue')
```



```
#(c)
#R plot Fig. 9.6: SVM for three points
#training data x
x = matrix(c(1, 1, 2, 1, 3, 3),
         ncol = 2, byrow = TRUE)
y = c(1, 1, -1) #two categories 1 and -1

plot(x, col = y + 3, pch = 19,
    xlab = 'x', ylab = 'y',
    xlim = c(-2, 8), ylim = c(-2, 8),
    main = 'SVM Points and Hyperplanes')
library(e1071)
dat = data.frame(x, y = as.factor(y))
svm3P = svm(y ~ ., data = dat,
          kernel = "linear", cost = 10,
          scale = FALSE,
          type = 'C-classification')
svm3P #This is the trained SVM

# Find hyperplane, normal vector, and SV (wx + b = 0)
w = t(svm3P$coefs) %*% svm3P$SV
w
#[1,] -0.4 -0.8
b = svm3P$rho
b
#[1] -2.6
```

```
x1 = seq(0, 5, len = 31)
x2 = (b - w[1]*x1)/w[2]
x2p = (1 + b - w[1]*x1)/w[2]
x2m = (-1 + b - w[1]*x1)/w[2]
x20 = (b - w[1]*x1)/w[2]
#plot the SVM results
#setEPS()
#postscript("fig0906.eps", height=7, width=7)
par(mar = c(4.5, 4.5, 2.0, 2.0))
plot(x, col = y + 3, pch = 19,
    xlim = c(0, 5), ylim = c(0, 5),
    xlab = 'x', ylab = 'y',
    cex.lab = 1.5, cex.axis = 1.5,
    main = 'SVM for three points labeled in two categories')
axis(2, at = (-2):8, tck = 1, lty = 2,
    col = "grey", labels = NA)
axis(1, at = (-2):8, tck = 1, lty = 2,
    col = "grey", labels = NA)
lines(x1, x2p, lty = 2, col = 4)
lines(x1, x2m, lty = 2, col = 2)
lines(x1, x20, lwd = 1.5, col = 'purple')
text(1.2, 0.8, bquote(italic(P[1])), cex = 1.5, col = 4)
text(2.2, 1.2, bquote(italic(P[2])), cex = 1.5, col = 4)
text(3.2, 3.2, bquote(italic(P[3])), cex = 1.5, col = 2)
```



```
#(d)
#According to the figure in (c),
#the support vectors are P2 and P3
#because they are on the boundaries
Dm = 2/norm(w, type ='2')
Dm #maximum margin of separation
#[1] 2.236068
```