537 Homework 1

Michael Pena

2024-07-13

Problem 1

(a).

```
ins <- c(25,-2,4,-2,4,1,4,1,9)
S = matrix(ins, nrow = 3, ncol =3, byrow = T)
#let's make a function where S goes in and v^(1/2) and rho goes out
f1 <- function(S){
  # render the v^(1/2)
  lam <- eigen(S)$values
  G <- eigen(S)$vectors
  L = diag(3)*lam^(1/2)
  v.5 = G%*%L%*%t(G)
  # render rho
  rho = matrix(0,3,3)
  for(i in 1:3){
    for(j in 1:3){
      rho[i,j] = S[i,j]/sqrt(S[i,i]*S[j,j])
    }
  }
  # outputs
  return(list("v^(1/2)" = v.5, rho = rho))
}

# run function
f1(S)
```

```
## $`v^(1/2)`
##            [,1]       [,2]      [,3]
## [1,]  4.9639854 -0.3062868 0.5148182
## [2,] -0.3062868  1.9622841 0.2358595
## [3,]  0.5148182  0.2358595 2.9460707
##
## $rho
##            [,1]       [,2]      [,3]
## [1,]  1.0000000 -0.2000000 0.2666667
## [2,] -0.2000000  1.0000000 0.1666667
## [3,]  0.2666667  0.1666667 1.0000000
```

(b).

$$\rho(x_1, \frac{1}{2}x_2 + \frac{1}{2}x_3) = \frac{cov(x_1, \frac{1}{2}x_2 + \frac{1}{2}x_3)}{\sqrt{var(x_1)var(\frac{1}{2}x_2 + \frac{1}{2}x_3)}}$$

$$var(x_1) = 25var(\frac{1}{2}x_2 + \frac{1}{2}x_3) = \frac{1}{4}(4) + \frac{1}{4}(9) - 2(1) = 13/4$$

$$cov(x_1, \frac{1}{2}x_2 + \frac{1}{2}x_3) = \frac{1}{2}cov(x_1, x_2) + \frac{1}{2}cov(x_1, x_3) = 0.5(-2) + 0.5(4) = 1$$

thus $\rho(x_1, \frac{1}{2}x_2 + \frac{1}{2}x_3) = \frac{2}{5\sqrt{5}}$

thus

$$\rho = \begin{bmatrix} 1 & \frac{2}{5\sqrt{13}} \\ \frac{2}{5\sqrt{13}} & 1 \end{bmatrix}$$

Problem 2

note that p=2 and

$\Sigma = I_2$ thus $|B|^{-1} = 1$ and $\Sigma^{-1} = I_2$

thus $(\vec{x} - \vec{\mu})^T \Sigma^{-1}(\vec{x} - \vec{\mu}) = (x_1 + 1)^2 + (x_2 - 1)^2$

$$ f_() =exp{- [(x_1 - _1)^2 + (x_2 - _2)^2]}

$$ and so

$$F_{\vec{X}}(0,0) = \int_{-\infty}^{0}\int_{-\infty}^{0} \frac{1}{2\pi} exp\{-\frac{1}{2}[(x_1+1)^2 + (x_2-1)^2]\} dx_1 dx_2 = \frac{1}{2\pi}\int_{-\infty}^{0} e^{-\frac{1}{2}(x_1+1)^2} dx_1 \int_{-\infty}^{0} e^{-\frac{1}{2}(x_2-1)^2} dx_2 = 0.133483764331$$

Problem 3

$$\nabla f = \begin{bmatrix} 2x_1 + 2x_2^2 + 2x_3 \\ 4x_1x_2 + 4x_2^3 + 4x_2x_3 \\ 2x_1 + 2x_2^2 + 2x_3 \end{bmatrix}$$

because $\frac{d}{dx_2}(4x_1x_2 + 4x_2^3 + 4x_2x_3) = 4x_1 + 12x_2^2 + 4x_3$

$$\nabla^2 f = \begin{bmatrix} 2 & 4x_2 & 2 \\ 4x_2 & 4x_1 + 12x_2^2 + 4x_3 & 4x_2 \\ 2 & 4x_2 & 2 \end{bmatrix}$$

Problem 4

$$|A - \lambda I| = \begin{vmatrix} 2-\lambda & 0 & 4 \\ 0 & 3-\lambda & -1 \\ 4 & -1 & 1-\lambda \end{vmatrix} = (2-\lambda)[(3-\lambda)(1-\lambda)-1] + 4[0-4(3-\lambda)] = -\lambda^3 + 6\lambda^2 + 6\lambda - 44$$

```
# define A
A = matrix(c(2,0,4,0,3,-1,4,-1,1), ncol=3, nrow = 3, byrow = T)
# get eigen pairs
Lambda = diag(3)*eigen(A)$values
as.matrix(eigen(A)$vectors) -> Gamma
# print values
Lambda;Gamma
```

```
##        [,1]     [,2]      [,3]
## [1,] 5.6977 0.000000  0.000000
## [2,] 0.0000 2.934181  0.000000
## [3,] 0.0000 0.000000 -2.631881
```

```
##            [,1]        [,2]       [,3]
## [1,]  0.7121118 -0.27071711  0.6477724
## [2,] -0.2440204 -0.96058049 -0.1331884
## [3,]  0.6582939 -0.06322468 -0.7501012
```

Problem 5

(a).

```
# render models
mod1 = lm(y ~ x1 + x2)
mod2 = lm(y ~ x1)
mod3 = lm(y ~ x2)
# getting r^2
sprintf("r^2 for y ~ x1 + x2 : %f",summary(mod1)$r.squared)
```

```
## [1] "r^2 for y ~ x1 + x2 : 0.415922"
```

```
sprintf("r^2 for y ~ x1 : %f",summary(mod2)$r.squared)
```

```
## [1] "r^2 for y ~ x1 : 0.253544"
```

```
sprintf("r^2 for y ~ x2 : %f",summary(mod3)$r.squared)
```

```
## [1] "r^2 for y ~ x2 : 0.344113"
```

(b)

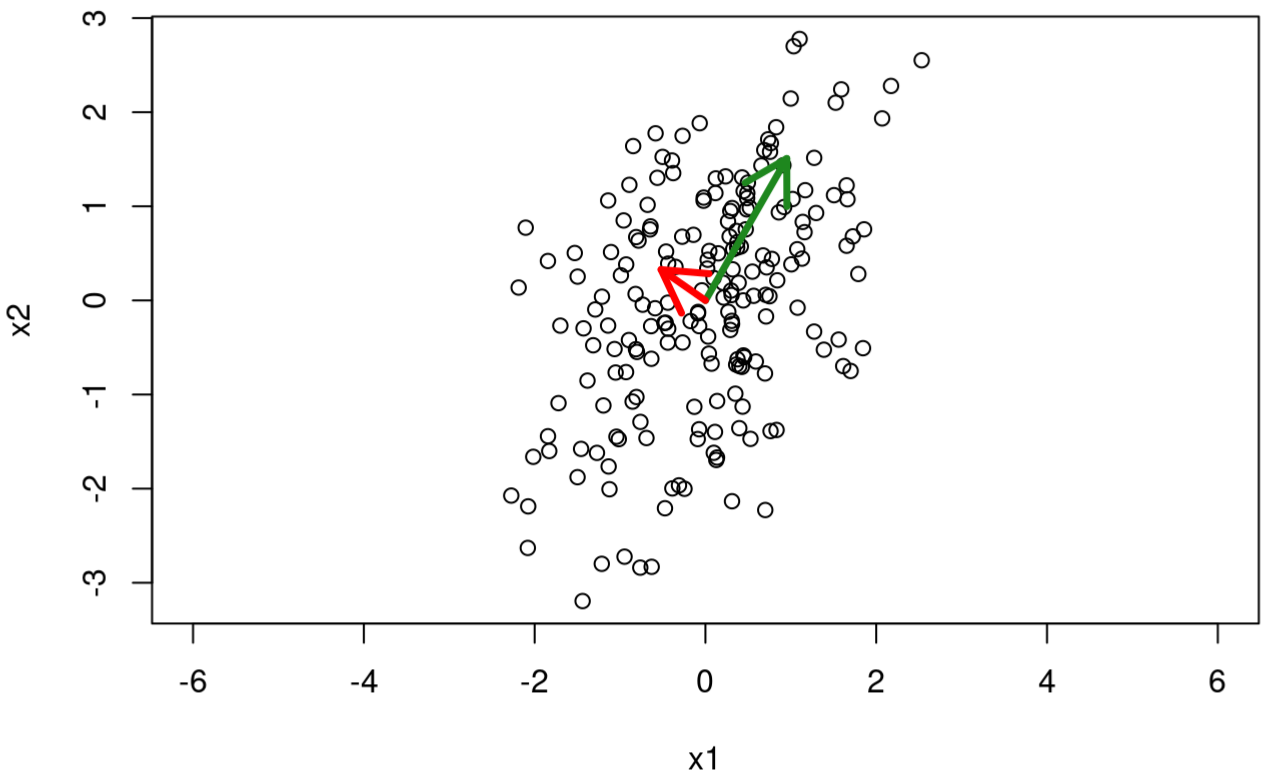
```
S <- cov(cbind(x1,x2))
Gamma = eigen(S)$vectors
lam = eigen(S)$values
Gamma;lam
```

```
##            [,1]       [,2]
## [1,] 0.5340500 -0.8454529
## [2,] 0.8454529  0.5340500
```

```
## [1] 1.7835906 0.6139426
```

(c)

```
plot(x1,x2,xlim = c(-6,6))
arrows(0,0,lam[1]*Gamma[1,1],lam[1]*Gamma[2,1],lwd=3.5,col="forestgreen")
arrows(0,0,lam[2]*Gamma[1,2],lam[2]*Gamma[2,2],lwd=3.5,col="red")
```



(d)

```
#
Cmat = cbind(x1,x2)%*%Gamma
Cmat[,1] -> c1
Cmat[,2] -> c2
# render models
mod4 = lm(y ~ c1 + c2)
mod5 = lm(y ~ c1)
mod6 = lm(y ~ c2)
# getting r^2
sprintf("r^2 for y ~ c1 + c2 : %f",summary(mod4)$r.squared)
```

```
## [1] "r^2 for y ~ c1 + c2 : 0.415922"
```

```
sprintf("r^2 for y ~ c1 : %f",summary(mod5)$r.squared)
```

```
## [1] "r^2 for y ~ c1 : 0.413674"
```

```
sprintf("r^2 for y ~ c2 : %f",summary(mod6)$r.squared)
```

```
## [1] "r^2 for y ~ c2 : 0.002249"
```

(e)

Model v. had the best r^2 value of the single variable models.

Both i. and iv. have the same r^2; I would want to pick either of the 2.