Multivariate Statistics EX3

203832050 and 312584204

Question 1

a) 1) We need to initial $h_{i,0}^2 = \sum_j \lambda_{ij}^2 \ for\ each\ i$. We choose to do it with $h_{i,0}^2 = \max_{j \neq i} |r_{ij}|$.
t represents the iteration number.
2) We estimate $\hat{\psi}_{i,t} = 1 - h_{i,t}^2\ i = (1,2,3)$.
3) We calculate the matrix $A_t = R - \hat{\psi}_t$ and find the matrix's eigen-values and eigen-vectors for estimating $\hat{\lambda}_{i,t} = \sqrt{\gamma_i} v_i\ for\ i = (1, \ldots, k)$
We notice that k (the number of factors) is 1 because the number of degrees of freedom: $p(k+1) \leq \frac{p(p+1)}{2} = 6 \rightarrow k \leq 1 \rightarrow k = 1$

Where γ_i is the eigen-value i and v_i is the corresponding eigen-vector, where the eigen-values arranged in descending order.
4) We update the values of $h_{i,t+1}^2$ and check if $|h_{i,t}^2 - h_{i,t+1}^2| < \varepsilon *\ \ \forall i$.
5) If the condition in step 4 exists, the algorithm stop, else return to step two with the updated $h_{i,t+1}^2$.
* We decide $\varepsilon = 10^{-20}$
The algorithm converged after 816 iterations and the results are:

```
> (psi)
          [,1]      [,2]      [,3]
[1,] 0.2633333 0.0000000 0.0000000
[2,] 0.0000000 0.8430769 0.0000000
[3,] 0.0000000 0.0000000 0.9082353
> (lambda)
          [,1]
[1,] 0.8582929
[2,] 0.3961352
[3,] 0.3029269
>
```

b) In the class we saw that:

$$l(\Sigma) = -\frac{n}{2}\log(|\Sigma|) - \frac{n}{2} tr(\Sigma^{-1} S)$$

We have R and not S and we assume $\Sigma = \Lambda\Lambda^T + \psi$, therefore:

$$l(\Lambda, \psi) = -\frac{n}{2}\log(|\Lambda\Lambda^T + \psi|) - \frac{n}{2} tr((\Lambda\Lambda^T + \psi)^{-1} R)$$

The algorithm start with initial value of $\psi \rightarrow \psi_0$, find $\Lambda_0 = (\lambda_1, \lambda_2, \lambda_3)^T$ that maximize $l(\Lambda, \psi_0)$.
Then find ψ_1, that maximize $l(\Lambda_0, \psi)$ and continue in this order until convergence:

$$|\psi_{i,t} - \psi_{i,t+1}| < \varepsilon = 10^{-20}, \forall i$$

After 501 iterations we got:

```
> (psi)
         [,1]      [,2]      [,3]
[1,] 0.267324 0.0000000 0.0000000
[2,] 0.000000 0.8422462 0.0000000
[3,] 0.000000 0.0000000 0.9076222
> (lambda_t)
[1] 0.8559446 0.3973118 0.3038759
```

As we can see, the results is very similar to the results in the pervious question.

R-code for this question:

```
#------Question 1------------
#-------task a---------
update_h <- function(hPrev){ #function for updating value in each iteration
  (psi<-1-hPrev)
  a<-R-diag(psi,p,p)
  eigenv<-eigen(a)
  (eigenvalues<-eigenv$values)
  (k<-1)
  (eigenvec<-eigenv$vectors)
  lambda<-matrix(nrow = p, ncol=k)
  for (i in c(1:k)) {
    (i_max<-which (eigenvalues==max(eigenvalues)))
    (lambda[,i]<-sqrt(eigenvalues[i_max])*as.matrix(eigenvec)[,i_max])
    (eigenvalues<-eigenvalues[-i_max])
    (eigenvec<-as.matrix(eigenvec)[,-i_max])
  }
  return (lambda)
}



R<-matrix(c(1,0.34,0.26,0.34,1,0.12,0.26,0.12,1), nrow = 3)
p<-3
n<-200
epsilon<-10^-20
hPrev<- c(0.34,0.34,0.26)
flag<-TRUE
count<-0
while(flag){
  count<-count+1
  lambda<-update_h(hPrev)
  hCurr<-rowSums(lambda^2)
  if(sum(abs(hPrev-hCurr)< epsilon)==3){
    flag<-FALSE
  }else{
    hPrev<-hCurr
  }
}
(count)
(psi<-diag(1-hCurr,p,p))
(lambda)

#-------task b---------
likelihoodFunc<- function(psi, lambda, r){
  psi<-diag(psi)
  A<-lambda%*%t(lambda)+psi
  return((log(det(A))+tr(solve(A)%*%R)))
}
```

```
R<-matrix(c(1,0.34,0.26,0.34,1,0.12,0.26,0.12,1), nrow = 3)
epsilon<-10^-15
psi0<-rep(1,p)
lambda_t<-c(0,1,1)
flag<-TRUE
count<-0
psi_t<-psi0
while (flag) {
  count<-count+1
  result<-optim(par = c(0,1,2), fn=likelihoodFunc, psi=psi0, r=R)
  (lambda_t<-result$par)
  result<-optim(par = c(0,1,2), fn=likelihoodFunc, lambda=lambda_t, r=R)
  (psi_t<-result$par)
  (x<-as.numeric(t(psi_t-psi0)%*%(psi_t-psi0)))
  if(x< epsilon | count>500){
    flag<-FALSE
  }else{
    psi0<-psi_t}
  (flag)
}
psi<-diag(psi_t,p)
(psi)
t(lambda_t)
```

Question 2

We start by dividing each point to a cluster. The measure was: -19.846

In each iteration of the algorithm we unify the closest two clusters to one by checking the distance between their centers.

After the union we check if we improved the given measure:

$aSSW - bSSB + ck$ where k is the numbers of clusters and,

$$SSW = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left(X_{ij} - X_i\right)^T \left(X_{ij} - X_i\right),$$

$$X_i - the\ average\ of\ cluster\ i, n_i - the\ numbers\ of\ points\ in\ cluster\ i$$

$$SSB = \sum_{i=1}^{k} (X_i - \bar{X})^T (X_i - \bar{X})$$

In our case the algorithm stops after two iterations:

The first iteration unified points 3 and 5 and the measure was: -22.83

The second iteration tried to unify point 1 to cluster with points 3 and 5 but the measure didn't improve (-17.378)

The final result:

4 clusters: 1) point 1, 2) point 2, 3) point 4, 4) points 3,5

R-code for this question:

```
#------Question 2--------
distance<-function(x,y){
  return(as.numeric(t(x-y)%*%(x-y)))
}

ss<-function(points, avg=c(0,0)){
  sumi<-0
  if(Norm(avg)==0)
    avg<-colMeans(points)
  for (i in 1:nrow(points)) {
    sumi<-sumi+as.numeric(t(points[i,]-avg)%*%(points[i,]-avg))
  }
  return(sumi)
}
set.seed(111)
p1<-c(1.9,0.64)
p2<-c(0.87,-1.2)
p3<-c(0.3,0.09)
p4<-c(-2.08,-0.45)
p5<-c(0.85,0.46)
a<-runif(1,0,1)
b<-runif(1,1,5)
c<-rnorm(1,5,1)
points<-rbind(p1,p2,p3,p4,p5)
avgPoi<-colMeans(points)
centers<-points
cluster<-list(c(1),c(2),c(3),c(4),c(5))
flag<-T
k<-5
(measure<--b*ss(centers,avgPoi)+c*k)
clusterBef<-cluster


while(flag && k>1){
   distMat<-matrix(NA, nrow = k,ncol = k)
   for (i in 1:k) {
```

```
    for (j in 1:k) {
      distMat[i,j]<-ifelse(i==j,10000 ,distance(centers[i,],centers[j,]))
    }
  }
  temp<- which(distMat==min(distMat), arr.ind = T)[,1]
  clusterBef<-cluster
  cluster[[temp[1]]]<-rbind(cluster[[temp[1]]],cluster[[temp[2]]])
  cluster<-cluster[-cluster[[temp[2]]]]
  centers<-matrix(NA, nrow=length(cluster), ncol = 2)
  ssw<-0
  for (i in 1:length(cluster)) {
    if(length(cluster[[i]])==1)
      centers[i,]<-points[cluster[[i]],]
    else{
      centers[i,]<-colMeans(points[cluster[[i]],])
      ssw<-ssw+ss(points[cluster[[i]],])
    }
  }
  ssb<-ss(centers,avgPoi)
  print(a*ssw-b*ssb+c*nrow(centers))
  if(measure<a*ssw-b*ssb+c*nrow(centers))
    flag<-F
  else{
    measure<-a*ssw-b*ssb+c*nrow(centers)
    k<-k-1
  }
}
clusterBef
```

Question 3

a) After sorting the data as said in the question we write the likelihood function as we saw in class:

$$L(\theta) = \prod_{i=1}^{k} f(y_i) \prod_{i=k+1}^{n} 1 - F(T_i) = \left(\frac{\theta^3}{2}\right)^k (\prod_{i=1}^{k} y_i^2) * e^{-\theta \sum_{i=1}^{k} y_i}$$
$$* \left(\prod_{i=k+1}^{n} \left(1 + \theta T_i + \frac{\theta^2 T_i^2}{2}\right)\right) e^{-\theta \sum_{i=k+1}^{n} T_i}$$
$$= e^{-\theta(\sum_{i=1}^{k} y_i + \sum_{i=k+1}^{n} T_i)} * \left(\frac{\theta^3}{2}\right)^k * \prod_{i=1}^{k} y_i^2 * \prod_{i=k+1}^{n} \left(1 + \theta T_i + \frac{\theta^2 T_i^2}{2}\right)$$

We take logarithm:

$$l(\theta) = -\theta\left(\sum_{i=1}^{k} y_i + \sum_{i=k+1}^{n} T_i\right) + 3k log(\theta) - k log(2) + \sum_{i=1}^{k} 2\log(y_i)$$
$$+ \sum_{i=k+1}^{n} log\left(1 + \theta T_i + \frac{\theta^2 T_i^2}{2}\right)$$

We derive by θ and equal to zero:

$$\frac{\partial l(\theta)}{\partial \theta} = -(\textstyle\sum_{i=1}^{k} y_i + \sum_{i=k+1}^{n} T_i) + \frac{3k}{\theta} + \sum_{i=k+1}^{n} \frac{T_i + \theta T_i^2}{\left(1 + \theta T_i + \frac{\theta^2 T_i^2}{2}\right)} = 0$$

There isn't a straight forward analytical solution that solve this equation, so we use a solver to find a numerical solution.

The solution for this equation (got by using optimize in R) is $\hat{\theta} = 1.123$.

b)

$$E(Y|Y > T) = \frac{\int_T^{\infty} y * f(y)dy}{1 - F(T)}$$

We start with the upper part:

$$\int_T^{\infty} x * f(x)dx = \int_T^{\infty} \frac{x^3 e^{-\theta x} \theta^3}{2} dx = -\frac{\theta^2}{2} \int_T^{\infty} -\theta e^{-\theta x} x^3 dx =$$

We do part integration where $u = e^{-\theta x}$ and $v = x^3$

$$= -\frac{\theta^2}{2} [x^3 e^{-\theta x}|_T^{\infty} - \int_T^{\infty} 3x^2 e^{-\theta x} dx] = \frac{\theta^2 T^3 e^{-\theta T}}{2} - \frac{3\theta}{2} \int_T^{\infty} -\theta x^2 e^{-\theta x} dx =$$

Again, we do part integration where $u = e^{-\theta x}$ and $v = x^2$

$$= \frac{\theta^2 T^3 e^{-\theta T}}{2} - \frac{3\theta}{2} [x^2 e^{-\theta x}|_T^{\infty} - \int_T^{\infty} 2x e^{-\theta x} dx] = \frac{\theta^2 T^3 e^{-\theta T}}{2} + \frac{3\theta T^2 e^{-\theta T}}{2}$$
$$- 3 \int_T^{\infty} -\theta x e^{-\theta x} dx] =$$

Again, we do part integration where $u = e^{-\theta x}$ and $v = x$

$$\frac{\theta^2 T^3 e^{-\theta T}}{2} + \frac{3\theta T^2 e^{-\theta T}}{2} - 3[xe^{-\theta x}|_T^\infty - \int_T^\infty e^{-\theta x} dx] =$$

$$\frac{\theta^2 T^3 e^{-\theta T}}{2} + \frac{3\theta T^2 e^{-\theta T}}{2} + 3Te^{-\theta T} + \frac{3e^{-\theta T}}{\theta} = \frac{e^{-\theta T}((\theta T)^3 + 3(\theta T)^2 + 6\theta T + 6)}{2\theta}$$

Now we do lower part: $1 - F(T) = \frac{2+2\theta T+(\theta T)^2}{2} e^{-\theta T}$

So we got:

$$E(Y|Y > T) = \frac{\int_T^\infty y * f(y)dy}{1 - F(T)} = \frac{1}{\theta} * \frac{((\theta T)^3 + 3(\theta T)^2 + 6\theta T + 6)}{2 + 2\theta T + (\theta T)^2}$$

c) The likelihood function is:

$$L(\theta, x) = \prod_{i=1}^{n} \frac{\theta^3 y_i^2 e^{-\theta y_i}}{2}$$

So the log likelihood is:

$$l(\theta, y) = 3nlog(\theta) - nlog(2) - \theta \sum_{i=1}^{n} y_i + \sum_{i=1}^{n} 2\log(y_i)$$

In iteration m, we need to find the expectation of log-likelihood under the parameter θ_m:

$E_{\theta_m}(l(\theta, y)|x) = 3nln(\theta) - nlog(2) + 2\sum_{i=1}^{k} \log(x_i) + 2\sum_{i=k+1}^{n} E_{\theta_m}(\log((y_i|y_i > T_i))) - \theta \left(\sum_{i=1}^{k} y_i + \sum_{i=k+1}^{n} E_{\theta_m}(y_i|\, y_i > T_i)\right) = *$

$= 3nln(\theta) - nlog(2) + 2\sum_{i=1}^{k} \log(x_i) + 2\sum_{i=k+1}^{n} E_{\theta_m}(\log((y_i|y_i > T_i))) - \theta \left(\sum_{i=1}^{k} y_i + \sum_{i=k+1}^{n} (\frac{1}{\theta_m} * \frac{((\theta_m T_i)^3+3(\theta_m T_i)^2+6\theta_m T_i+6)}{2+2\theta_m T_i+(\theta_m T_i)^2})\right)$

*From task b: $E_\theta(y_i|\, y_i > T_i) = \frac{1}{\theta} * \frac{((\theta T)^3+3(\theta T)^2+6\theta T+6)}{2+2\theta T+(\theta T)^2}$

We need to find the parameter that maximize the function above (Step M), so we need to derivative and equal it to zero:

$$\frac{\partial E_{\theta_m}(l(\theta, y)|x)}{\partial \theta} = \frac{3n}{\theta} - \left(\sum_{i=1}^{k} x_i + \sum_{i=k+1}^{n} \left(\frac{1}{\theta_m} * \frac{((\theta_m T_i)^3 + 3(\theta_m T_i)^2 + 6\theta_m T_i + 6)}{2 + 2\theta_m T_i + (\theta_m T_i)^2}\right)\right) = 0$$

$$\rightarrow \theta_{m+1} = \frac{3n}{\left(\sum_{i=1}^{k} x_i + \sum_{i=k+1}^{n} \left(\frac{1}{\theta_m} * \frac{((\theta_m T_i)^3 + 3(\theta_m T_i)^2 + 6\theta_m T_i + 6)}{2 + 2\theta_m T_i + (\theta_m T_i)^2}\right)\right)}$$

d) We run the equation above in R until convergence ($|\theta_{m+1} - \theta_m| < \varepsilon = 10^{-10}$
It took 34 iterations and the solution was:

$$\hat{\theta} = 1.123$$

The same as the result we got in task a.
*We start the algorithm with random value for theta.

R-code for this question:

```
#-----Question 3-------
#task a
data<-read.csv(choose.files(),header=F) #uploading the data

likelihood<-function(theta){
  -1*(theta*sum(data[,1]))+3*sum(data[,2])*log(theta)+sum(log(1+theta*data[38:100,1]+theta^2*data[38:100,1]/2))
}

optimise(likelihood, interval = c(-10^100,10^100), maximum = TRUE, tol = 0.00001)


f<-function(theta){
  -sum(data[,1])+3*37/theta+sum(data[38:100,1]*(1+theta)/(1+data[38:100,1]*(theta+theta^2/2)))
}

uniroot.all(f, c(-100000,100000) )


#-----task d--------
theta<-runif(1,0,10)
flag<-TRUE
epsilon<-10^-10
count<-0
while (flag) {
  count<-count+1
  temp<-3*nrow(data)/
    (sum(data[1:37,1])+
       (1/theta)*(sum(((theta*data[38:100,1])^3+3*((theta*data[38:100,1])^2)+6*theta*data[38:100,1]+6)/
                 (2+2*(theta*data[38:100,1])+(theta*data[38:100,1])^2))))
  if(abs(theta-temp)<epsilon)
    flag<-FALSE
  else
    theta<-temp
}
(theta)
```