Assignment4_3701

JinYao

2019/8/1

Problem 1

```
# I use the matrix of assignment 3
mat = read.csv('~/Desktop/3701/X.df.csv')
x.mat = as.matrix(mat[,2:4])
y.mat = as.matrix(mat[,1])
betahat = qr.coef(qr = qr(x=x.mat),y=y.mat)

x1.col = x.mat[,1:2]
betahat2 = qr.coef(qr = qr(x=x1.col),y=y.mat)

x2.col = x.mat[,1:2]
yi.hat = x2.col%*%betahat2
ei.hat = y.mat - yi.hat
x3.mat = x.mat[,c(1,3)]
betahat3 = qr.coef(qr = qr(x=x3.mat),y=ei.hat)

betahat
```

```
##                           [,1]
## Intercept           5.1073949
## Father.Height       0.7661411
## Grandfather.height 0.1632550
```

```
betahat2
```

```
##                     [,1]
## Intercept      6.6635357
## Father.Height 0.9001115
```

```
betahat3
```

```
##                            [,1]
## Intercept           -3.9054510
## Grandfather.height  0.0583725
```

We can find that the outcomes are different by using these two methods.

Problem 2

i.

```
set.seed(1729)
mat = read.csv('~/Desktop/3701/X.mat.3.csv')
x.mat = as.matrix(mat)
n = nrow(x.mat);p = ncol(x.mat);reps = 1e4
```

```
l = c(1/3,1/3,1/3,-1/2,-1/2)
t.perc = qt(0.975,n-p)
linxl = t(l)%*%qr.solve(crossprod(x.mat))%*%l
squalinxl = sqrt(linxl)
capture.list = numeric(reps)
y.list = rnorm(n = nrow(x.mat),mean = 0, sd = 1)
  beta.hat = qr.solve(t(x.mat)%*%x.mat)%*%t(x.mat)%*%(y.list)
  sE = sqrt(sum((y.list-x.mat%*%beta.hat)^2)/(n-p))
  betatimesl = t(l)%*%beta.hat
  moe = t.perc*sE*squalinxl
  left.pt = (betatimesl - moe)
  right.pt = (betatimesl + moe)
cat("confidence interval:",left.pt,right.pt)
```

```
## confidence interval: -0.0454454 1.174765
```

ii.

```
set.seed(1729)
mat = read.csv('~/Desktop/3701/X.mat.3.csv')
x.mat = as.matrix(mat)
n = nrow(x.mat);p = ncol(x.mat);reps = 1e4
l = c(1/3,1/3,1/3,-1/2,-1/2)
sqrtform = sqrt(t(l)%*%qr.solve(crossprod(x.mat))%*%l)
pval.list = numeric(reps)
y.list = rnorm(n = nrow(x.mat),mean = 0, sd = 1)
for(r in 1:reps){
  beta.hat = qr.solve(t(x.mat)%*%x.mat)%*%t(x.mat)%*%(y.list)
  sE = sqrt(sum((y.list-x.mat%*%beta.hat)^2)/(n-p))
  t.stat = (t(l)%*%beta.hat)/(sE*(sqrtform))
  pval.list[r] = 2*pt(-abs(t.stat),n-p)
}
mean(pval.list)
```

```
## [1] 0.06774305
```

The pval is higher than the significance level, so we fail to reject the null hypothesis.

iii.

We need to rewrite the hypothesis. The null hypothesis becomes to test: $H_0 : \frac{\beta_1+\beta_2+\beta_3}{3} - \frac{\beta_4-\beta_5}{2} = \beta_3 - \beta_4 = 0$
Then we can use the F test to do this

```
set.seed(1729)
mat = read.csv('~/Desktop/3701/X.mat.3.csv')
x.mat = as.matrix(mat)
n = nrow(x.mat);p = ncol(x.mat);d = 2;reps = 1e4;pval.list=numeric(reps)
C = rbind(c(1/3,1/3,1/3,-1/2,-1/2),c(0,0,1,-1,0))
XtXinv = qr.solve(crossprod(x.mat))
Cqfrom = t(C)%*%qr.solve(C%*%XtXinv%*%t(C))%*%C

for(r in 1:reps){
  y.list = rnorm(n = nrow(x.mat),mean = 0, sd = 1)
  beta.hat = XtXinv%*%crossprod(x.mat,y.list)
  rssf = sum((y.list-x.mat%*%beta.hat)^2)
  f.alt = t(beta.hat)%*%Cqfrom%*%beta.hat*(n-p)/(d*rssf)
  pval.list[r] = 1-pf(f.alt,d,n-p)
}
mean(pval.list)
```

```
## [1] 0.5020804
```

This is the pval we calculate, then we use this outcome to compare with the significance level, when the pval is less than the significance level, we reject the null hypothesis.

Problem 3

i.

$L(f(y)) = L(\lambda; y_1, y_2, ..., y_3) = \prod_{i=1}^{n} \lambda e^{-\lambda y_i}$ Then take log to both sides $logL(\lambda; y_1, y_2, ..., y_3) = log \prod_{i=1}^{n} \lambda e^{-\lambda y_i} = \sum_{i=1}^{n} log\lambda e^{-\lambda y_i} = \sum_{i=1}^{n}(log\lambda + loge^{-\lambda y_i}) = nlog\lambda - \lambda \sum_{i=1}^{n} y_i$ $\nabla logL(\lambda; y_1, y_2, ..., y_n) = \frac{1}{\lambda}n - \sum_{i=1}^{n} y_i = 0$ we get $n \times \frac{1}{\lambda} - \sum_{i=1}^{n} y_i = 0$ we get $\hat{\lambda} = \frac{1}{\bar{y}}$ $\nabla^2 logL(\lambda; y_1, ..., y_n) = \frac{-n}{\lambda^2} < 0$ so this is the maximum point, it is the MLE.

ii.

```
n = 40;reps = 5e4;lambda = 3
lam.list = matrix(NA, nrow=reps, ncol = 3)
for(r in 1:reps){
  y.list = rexp(n=n, rate=lambda)
  lam.list[r, 1] = n/sum(y.list)
  lam.list[r, 2] = (sqrt(var(y.list)))^-1
  lam.list[r, 3] = log(2)/median(y.list)
}
quan0025 = apply(lam.list, 2, quantile, 0.025)
quan0975 = apply(lam.list, 2, quantile, 0.975)
cat('lam.hat:',quan0025[1],quan0975[1],'lam.s:',quan0025[2],quan0975[2], 'lam.m:',quan0025[3],quan0975[
```

```
## lam.hat: 2.250275 4.208486 lam.s: 2.066017 4.770305 lam.m: 1.984703 4.77257
```

iii.

```
lam.hat.diff = quan0975[1]-quan0025[1]
lam.s.diff = quan0975[2]-quan0025[2]
lam.m.diff = quan0975[3]-quan0025[3]
cat("lam.hat.diff:",lam.hat.diff,"lam.s.diff:",lam.s.diff,"lam.m.diff:",lam.m.diff)
```

```
## lam.hat.diff: 1.958211 lam.s.diff: 2.704288 lam.m.diff: 2.787867
```

These three intervals all capture true lambda, while the first one(lambdahat) has a more accurate outcome with small interval.

Problem 4

First we need to write out the likelihood finction and the log likelihood:

$$g(x) = pf_1(x) + (1-p)f_2(x) = p\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x-\mu)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{-(x-\mu)^2}{2\sigma_2^2}}$$

$L(\mu; x_1, ..., x_n) = \prod_{i=1}^n [p\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x-\mu)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{-(x-\mu)^2}{2\sigma_2^2}}]$ Take log to both sides

$$l = logL(\mu; x_1, ..., x_n) = log[\prod_{i=1}^n [p\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x_i-\mu)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{-(x_i-\mu)^2}{2\sigma_2^2}}]] = \sum_{i=1}^n log[p\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x_i-\mu)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{-(x_i-\mu)^2}{2\sigma_2^2}}]$$

Take first derivative to the function and let the first derivative equal to zero:

$$\nabla l = \sum_{i=1}^n [p\frac{1}{\sqrt{2\pi}\sigma_1}e^{-\frac{(x-\mu)^2}{2\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2}e^{\frac{-(x-\mu)^2}{2\sigma_2^2}}]^{-1}[p\frac{1}{\sqrt{2\pi}2}\frac{x_i-\mu}{4}e^{\frac{-(x_i-\mu)^2}{8}} + (1-p)\frac{1}{\sqrt{2\pi}\sqrt{5}}\frac{x_i-\mu}{5}e^{\frac{-(x_i-\mu)^2}{10}}]$$

$\sigma_1 = 2, \sigma_2 = \sqrt{5}$

$$\nabla l = \sum_{i=1}^n [p\frac{1}{\sqrt{2\pi}2}e^{-\frac{(x-\mu)^2}{8}} + (1-p)\frac{1}{\sqrt{2\pi}\sqrt{5}}e^{\frac{-(x-\mu)^2}{10}}]^{-1}[p\frac{1}{\sqrt{2\pi}2}\frac{x_i-\mu}{4}e^{\frac{-(x_i-\mu)^2}{8}} + (1-p)\frac{1}{\sqrt{2\pi}\sqrt{5}}\frac{x_i-\mu}{5}e^{\frac{-(x_i-\mu)^2}{10}}] = 0$$

We then to simplify this euqation. We need to check the second derivative and let first derivative equal to 0, solve for the close form solution. If we can find the close form solution, we are done. Or we can use the bisection method because the function is differentiable.

```
# dsearch = function(f, a,b, L=1e-7, eps = (L/2.1), quiet = FALSE,...){
#   k = 0
#   while((b-a)>L){
#     k = k+1
#     m = (a+b)/2
#     m1 = m-eps
#     m2 = m+eps
#     f.at.m1 = f(m1,...)
#     f.at.m2 = f(m2,...)
#     if(f.at.m1 < f.at.m2){
#       b=m2
#     }else if(f.at.m1 > f.at.m2){
#       a = m1
#     }else{
#       a = m1
#       b = m2
#     }
#     if(!quiet){
#       cat("after iteration k =",k,
#           "the interval is", a, b, "\n")
#     }
#   }
#   return((a+b)/2)
# }
```