ps8

Baoyue Liang

11/27/2018

Problem 1

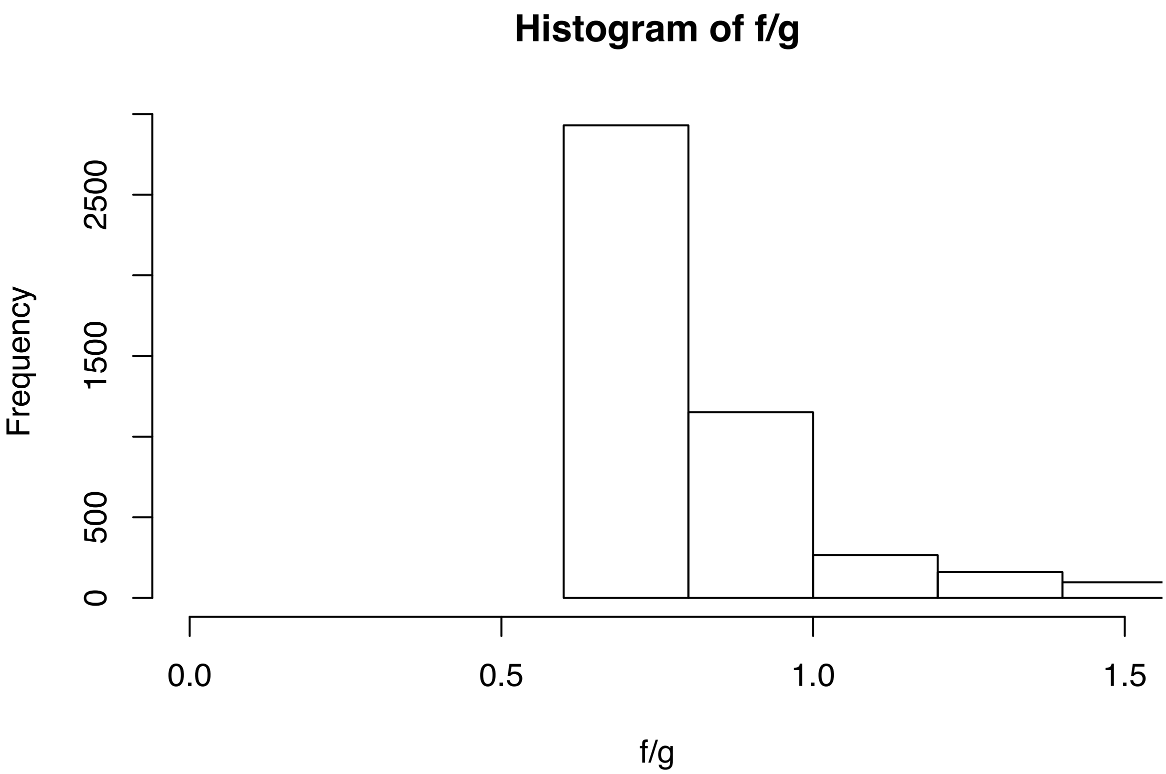
(a)

```
set.seed(1)
m <- 5000
sample <- rt(m, df = 3)
sample <- sample-4

# convert the samples values greater than -4
sample[sample>-4] <- -8-sample[sample>-4]

# estimate the mean
f <- dt(sample,3)/pt(-4,3)
g <- 2*dt(sample+4, df = 3)
E <- 1/m*sum(sample*f/g)

# create histograms of the weights
hist(f/g, breaks = 100, xlim = c(0,1.5))
```



```r
# estimate the variance
var <- var(sample*f/g)/m

# report the estimates of mean and variance
E
```

```
## [1] -6.18695
```

```r
var
```

```
## [1] 0.04357791
```

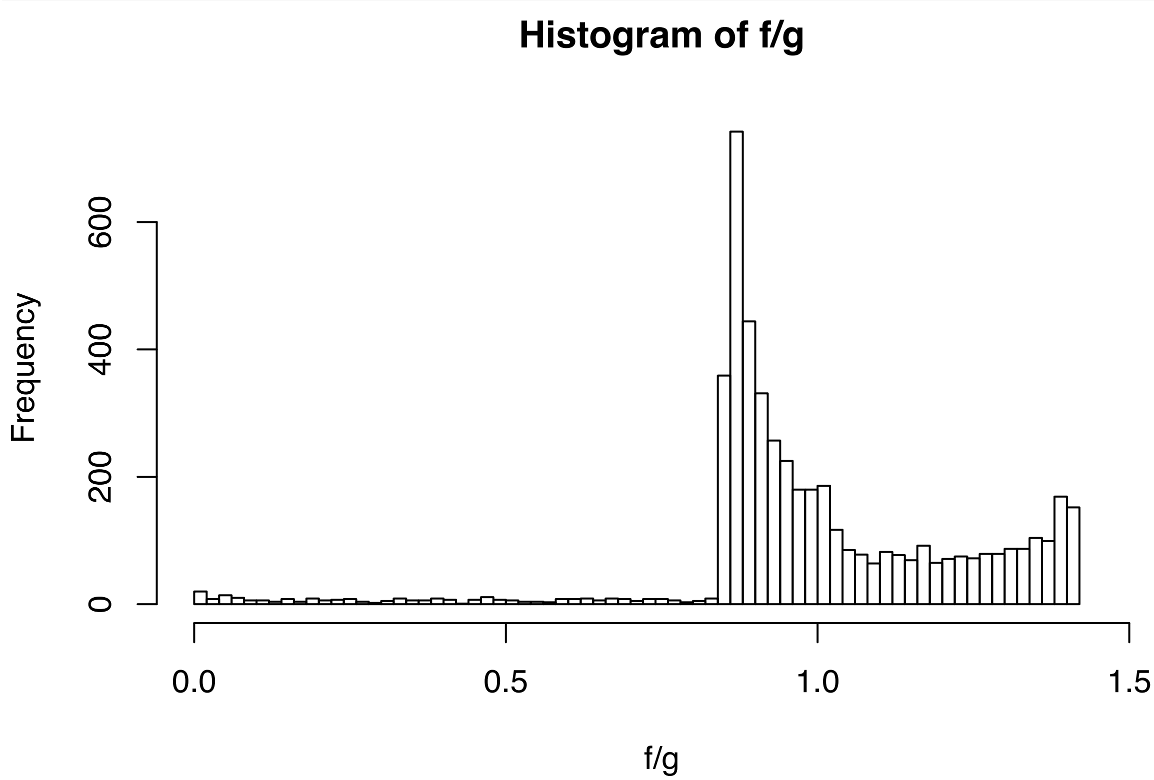
(b)

```r
set.seed(1)
m <- 5000
sample <- rt(m, df = 1)
sample <- sample-4

# convert the samples values greater than -4
sample[sample>-4] <- -8-sample[sample>-4]

# estimate the mean
f <- dt(sample,3)/pt(-4,3)
g <- 2*dt(sample+4, df = 1)
E <- 1/m*sum(sample*f/g)

# create histograms of the weights
hist(f/g, breaks = 100, xlim = c(0,1.5))
```



```
# estimate the variance
var <- var(sample*f/g)/m

# report the estimates of mean and variance
E
```

```
## [1] -6.208752
```

```
var
```

```
## [1] 0.00183077
```

Problem 2

```
library(fields)
```

```
## Loading required package: spam
```

```
## Loading required package: dotCall64
```

```
## Loading required package: grid
```

```
## Spam version 2.1-2 (2017-12-21) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

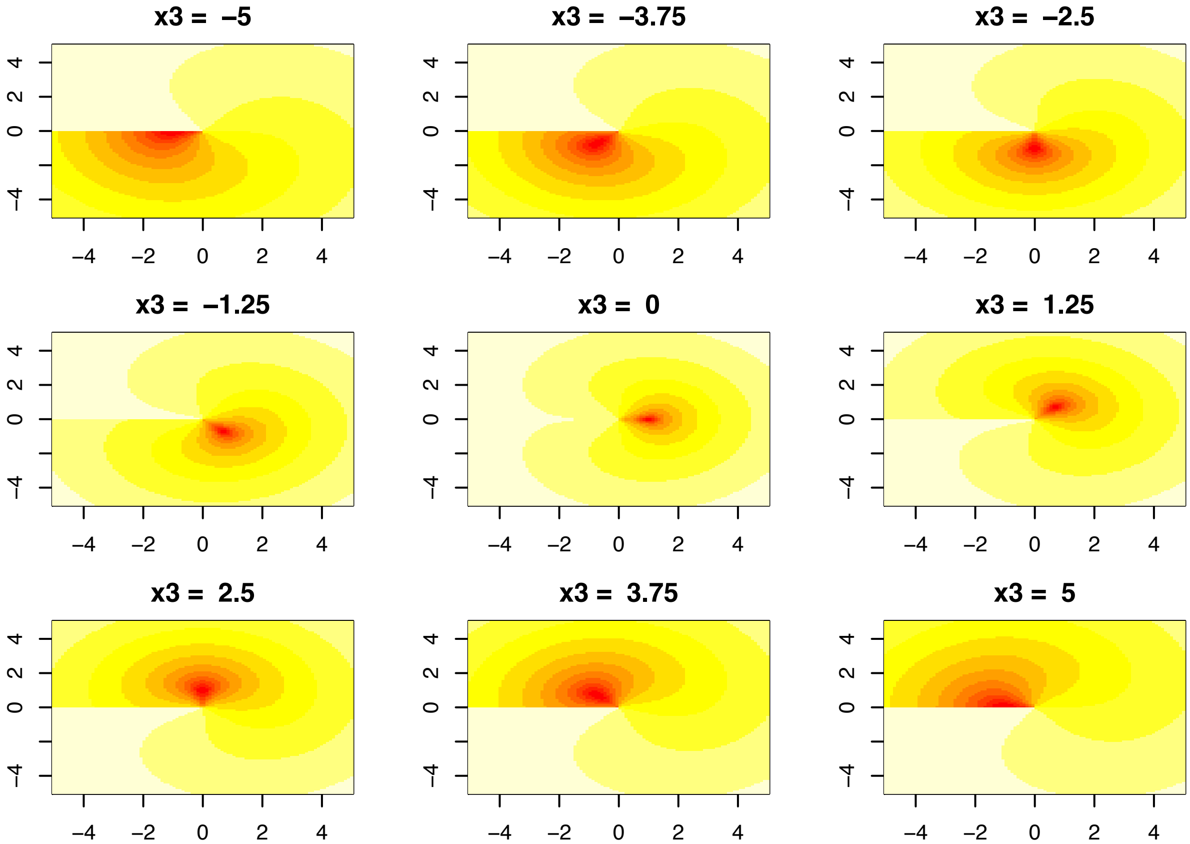
```
theta <- function(x1,x2) atan2(x2, x1)/(2*pi)

f <- function(x) {
  f1 <- 10*(x[3] - 10*theta(x[1],x[2]))
  f2 <- 10*(sqrt(x[1]^2 + x[2]^2) - 1)
  f3 <- x[3]
  return(f1^2 + f2^2 + f3^2)
}

## provide data
m <- 100
x1 <- x2 <-  seq(-5, 5, len = m)
x3 <- seq(-5, 5, length.out = 9)
xs <- as.matrix(expand.grid(x1, x2))

## apply function and plot slice
par(mfrow = c(3,3), mai = c(0.3,0.3,0.3,0.3))
for(i in x3){
  new = cbind(xs ,i)
  fx = apply(new, 1, f)
```

```
  title = paste('x3 = ', i)
  image(x1, x2, matrix(log(fx),length(x1),length(x1)),xlab = "x1",ylab = "x2",main = title)
}
```




```
set.seed(0)
start = runif(3,-5,5)
optim(start,f)
```

```
## $par
## [1]  1.000022621 -0.001650928 -0.002496620
##
## $value
## [1] 8.002852e-06
##
## $counts
## function gradient
##      140       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
set.seed(0)
start = runif(3,-5,5)
optim(start,f)
```

```
## $par
## [1]  1.000022621 -0.001650928 -0.002496620
##
## $value
## [1] 8.002852e-06
##
## $counts
## function gradient
##      140       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```r
set.seed(0)
start = runif(3,-5,5)
nlm(f,start)
```

```
## $minimum
## [1] 1.194957e-18
##
## $estimate
## [1]  1.000000e+00 -6.189898e-10 -9.370593e-10
##
## $gradient
## [1]  5.850742e-09 -1.530864e-08  7.744586e-09
##
## $code
## [1] 1
##
## $iterations
## [1] 20
```

```r
set.seed(0)
start = runif(3,-5,5)
nlm(f,start)
```

```
## $minimum
## [1] 1.194957e-18
##
## $estimate
## [1]  1.000000e+00 -6.189898e-10 -9.370593e-10
##
## $gradient
## [1]  5.850742e-09 -1.530864e-08  7.744586e-09
##
## $code
## [1] 1
##
## $iterations
## [1] 20
```

Problem 3

Question (a) and (b) are attached behind (c) and (d) and are written by hand

(c)

```
library(Rlab)
```

```
## Rlab 2.15.1 attached.
##
## Attaching package: 'Rlab'
## The following objects are masked from 'package:fields':
##
##     bplot, bplot.xy, cat.to.list, describe, set.panel, stats, US,
##     world, xline, yline
## The following object is masked from 'package:maps':
##
##     ozone
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
## The following object is masked from 'package:datasets':
##
##     precip
```

```
set.seed(2)
n = 100

## generate X
x1  = matrix(runif(n*3),n,3)
x0  = as.matrix(rep(1,n))
X = cbind(x1,x0)

## initialize beta
beta <- matrix(c(0.5,0.8,0,0))

## get pi and Y
P = pnorm(X %*% beta)
set.seed(1)
Y = rbern(n,P)

## glm test
Xdf = cbind.data.frame(X,Y)
names(Xdf) = c("intercept","X1","X2","X3","Y")
result = glm(Y ~ X1 + X2 + X3, family = binomial(link = "probit"),Xdf)
summary(result)
```

```
##
## Call:
## glm(formula = Y ~ X1 + X2 + X3, family = binomial(link = "probit"),
```

```
##     data = Xdf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9511  -1.3093   0.6863   0.8486   1.0606
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.1110     0.3291   0.337    0.736
## X1            0.6634     0.4486   1.479    0.139
## X2            0.3905     0.4585   0.852    0.394
## X3                NA         NA      NA       NA
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 116.65  on 99  degrees of freedom
## Residual deviance: 113.59  on 97  degrees of freedom
## AIC: 119.59
##
## Number of Fisher Scoring iterations: 4
```

```
## beta starting point
b0 = qnorm(mean(Y))
beta_sp = matrix(c(b0,0,0,0))

## EM algorithm
EM <- function(beta,X,Y,eps,max_iteration){
  mu = X %*% beta
  i = 0
  converge = FALSE
  while((!converge) & (i <max_iteration)){
    beta0 = beta
    z_updated = solve(t(X) %*% X) %*% z_updated
    converge = max(abs(beta-beta0)) <= eps
    mu = X %*% beta
    i = i + 1
  }
  return(list(beta = t(beta),iteration=i,epsilon = max(abs(beta-beta0)), convergence = converge))
}
```

(d)

```
MLE <- function(beta, X, Y){
  mu = X %*% beta
  ll = -sum(Y * pnorm(mu,log.p = T) + (1 - Y) * pnorm(-mu, log.p = T))
  return(ll)
}

MLE_result = optim(beta, MLE, X = X, Y = Y, method = "BFGS" )
MLE_result
```

```
## $par
##             [,1]
```

```
## [1,]  0.4366912
## [2,]  0.6692004
## [3,]  0.3995226
## [4,] -0.1073172
##
## $value
## [1] 56.33626
##
## $counts
## function gradient
##       27        9
##
## $convergence
## [1] 0
##
## $message
## NULL
```

Problem 3 a) Complete data: $\{Y, Z\}$, where $z_i \sim N(x_i^T\beta, 1)$ &

$y_i = I(z_i > 0) = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i \le 0 \end{cases}$ $\quad y_i \sim Ber(\Phi(x_i^T\beta))$

Observe y_i tells sign of z_i

$\boxed{\text{E-step}}$ $f(z_i, y_i \mid \beta) = f(z_i \mid y_i, \beta) f(y_i \mid \beta)$

$= \frac{1}{\sqrt{2\pi}} \exp\{-\frac{1}{2}(z_i - x_i^T\beta)^2$

$\log f(z_i, y_i \mid \beta) = \sum_{\text{all } z_i\, y_i} \log f(z_i, y_i \mid \beta)$

$= \sum -\frac{1}{2}\log(2\pi) - \frac{1}{2}(z_i - x_i^T\beta)^2 +$

$y_i \log \Phi(x_i^T\beta) + (1-y_i)\log \Phi(x^T\beta)$

$Q(\beta \mid \beta^t) = E(\log L(\beta \mid z_i y_i) \mid y, \beta^t)$

$= E(\log f(z_i, y_i \mid \beta) \mid y, \beta^t)$

$= \sum[-\frac{1}{2}\log(2\pi)] - E[\frac{1}{2}(Z - X\beta)^T(Z - X\beta) \mid y, \beta^t] + C$

remove constant $= -\frac{1}{2}E[(Z - X\beta)^T(Z - X\beta) \mid y, \beta^t]$

$= -\frac{1}{2}E[Z^TZ - Z^TX\beta - \beta^TX^TZ + \beta^TX^TX\beta \mid y, \beta^t]$

$= C - 2\beta^T E(X^TZ \mid y, \beta^t) + \beta^TX^TX\beta$

$\boxed{\text{M-step}}$ maximize Q by taking derivative wrt β and set to 0, $\beta^{t+1} = \arg\max Q(\beta \mid \beta^t)$

$\frac{dQ}{d\beta} = \frac{\partial}{\partial\beta}(2\beta^T E(X^TZ \mid y, \beta^t)) + \frac{\partial}{\partial\beta}(\beta^TX^TX\beta)$

$= -2[E(X^TZ \mid y, \beta^t)]^T + \beta^T(2X^TX) \stackrel{set}{=} 0$

$\Rightarrow E[X^TZ \mid y\, \beta^t]^T = \beta^TX^TX$

$\beta^{t+1} = (X^TX)^{-1}X^T E(Z \mid y, \beta^t)$

$= (X^TX)^{-1}X^T Z^{t+1}$ $\quad$ denote $Z^{t+1} = E(Z \mid y, \beta^t)$

$z_i \mid y_i = 0\ \beta^t \sim TN(x_i^T\beta^t, 1),\ z_i \le 0$

$z_i \mid y_i = 1\ \beta^t \sim TN(x_i^T\beta^t, 1),\ z_i > 0$

for $W \sim TN(\mu, \sigma^2; (\tau, +\infty))$

$$E(W \mid W > \tau) = \mu + \sigma \frac{\phi\left(\frac{\tau-\mu}{\sigma}\right)}{1-\Phi\left(\frac{\tau-\mu}{\sigma}\right)}$$

for $V \sim TN(\mu, \sigma^2, (-\infty, \tau))$

$$E(V \mid V > \tau) = \mu - \sigma \frac{\phi\left(\frac{\tau-\mu}{\sigma}\right)}{\Phi\left(\frac{\tau-\mu}{\sigma}\right)}$$

$$\Rightarrow z_i^{t+1} = \begin{cases} x_i^T\beta^t - \dfrac{\phi(x_i^T\beta^t)}{\Phi(-x_i^T\beta^t)} & \text{if } y_i = 0 \\ x_i^T\beta^t + \dfrac{\phi(x_i^T\beta^t)}{1-\Phi(-x_i^T\beta^t)} & \text{if } y_i = 1 \end{cases}$$

b) I think β should start from 0

Scanned with CamScanner