4.8

```
# R code is as follows
fun4.8<-function(th0){
  y<-c(14,0,1,5)
  a<-b<-1
  q<-rep(0,y[1]+1)
  for(k in 1:(y[1]+1)){
    zk<-k-1
    q[k]<-dbinom(zk,y[1],th0/(th0+2))/dbeta(th0,a+y[4]+zk,b+y[2]+y[3])
  }
  p<-q/sum(q)
  l<-data.frame(q,p)
  return(l)
}

fun4.8(0.8)
```

```
##                q            p
## 1  3.269477e-03 9.382900e-03
## 2  1.716475e-02 4.926022e-02
## 3  4.338868e-02 1.245189e-01
## 4  6.942189e-02 1.992302e-01
## 5  7.809963e-02 2.241340e-01
## 6  6.508303e-02 1.867783e-01
## 7  4.130269e-02 1.185324e-01
## 8  2.022989e-02 5.805669e-02
## 9  7.670499e-03 2.201316e-02
## 10 2.237229e-03 6.420506e-03
## 11 4.935064e-04 1.416288e-03
## 12 7.975861e-05 2.288950e-04
## 13 8.920371e-06 2.560010e-05
## 14 6.175641e-07 1.772315e-06
## 15 1.995530e-08 5.726867e-08
```

4.10(a)

```
# The R code is as follows:

fun4.10<-function(ind, th0, NumEM){
# input
# ind = 1: calculate the posterior mode in (b)
#     = 2: calculate the convergence rate of the 1-st EM algorithm in (b)
#     = 3: calculate the posterior mode in (d)
#     = 4: calculate the convergence rate of the 2-nd EM algorithm in (d)
# th0    = initial value of \theta, th0 = 0.5
# NumEM1 = the number of iterations in the 1-th & 2-nd EM

# Output
# TH = approximates of the posterior mode
# r1 = the convergence rate of the 1-st EM algorithm
# r2 = the convergence rate of the 2-nd EM algorithm
```

```
  y<-c(125,18,20,34)
  N<-sum(y)
  a0<-b0<-1
  if(ind==0){
    th<-th0
    TH<-matrix(0,NumEM,1)
    for(tt in 1:NumEM){
      Ez<-y[1]*th/(th+2)
      a<-a0+b0-2
      th<-(Ez + y[4]+a0-1)/(Ez+y[2]+y[3]+y[4]+a)
      TH[tt]<-th
    }
    return(TH)
  }
  if(ind==2){
    tth<-0.6268215
    b<-(N*tth+2*(N-y[1]))^2
    r1<-abs(2*y[1]*(y[2]+y[3])/b)
    return(r1)
  }
  if(ind==3){
    th<-th0
    TH<-matrix(0,NumEM,1)
    for(tt in 1:NumEM){
      Ez<-3*y[1*th/(th+2)]
      th<-(Ez+y[4]+a0-1)/(N=a0+b0-2)
      TH[tt]<-th
    }
    return(TH)
  }
  if(ind==4){
    tth<-0.6268215
    r2<-abs(6*y[1]/(N*(tth+2)^2))
    return(r2)
  }

}
```