EM-Algorithm: Examples

Patrick Mellady

Here are a couple examples of using the EM algorithm on statistical data. One example is a mixture of binomial distributions and the other is an exercise 29 out of Casella and Berger chapter 7.

Example 1: Mixture Distribution Problem

In this problem, we have a mixture of two binomial random variables with mixture parameter q. These binomials share a size, m, but they have success probabilities p_1 and p_2, respectively. The EM algorithm maximizes the posterior log likelihood given the previous iteration.

Distributional Problem Statement

We first state the problem by specifying the distribution from which we obtain our sample. After stating the distribution of the sample, we can maximize the incomplete log likelihood via an iterative method.

In addition to the mixture distribution as stated in the problem, we will assign latent variables $Z_1, Z_2, \cdots, Z_n$ that have the property that if $Z_i = 1$ then we know $X_i \sim bin(m, p_1)$. Thus, by the statement of our problem and the definition of our latent variables, we have:

$$X_1, X_2, X_2, \cdots, X_n \sim qbin(m, p_1) + (1-q)bin(m, p_2) X_i | Z_i = 1 \sim bin(m, p_1)$$

Finding the Necessary Qunatitites

We will first use this to find the posterior distribution of $Z_i|X_i$ using Bayes' rule.

$$P(Z_i = 1|X_i) = \frac{P(X_i|Z_i = 1)P(Z_i = 1)}{P(X_i)} = \frac{\binom{m}{x_i} p_1^{x_i}(1-p_1)^{m-x_i} \cdot q}{\binom{m}{x_i} p_1^{x_i}(1-p_1)^{m-x_i} \cdot q + \binom{m}{x_i} p_2^{x_i}(1-p_2)^{m-x_i} \cdot (1-q)} = \gamma_{1i}$$

We will write this formula to obtain the r^{th} iteration of γ_{1i}, denoted $\hat{\gamma}_{1i}^{(r)}$, in the following way

$$\hat{\gamma}_{1i}^{(r)} = \frac{\binom{m}{x_i} (\hat{p}_1^{(r-1)})^{x_i}(1-\hat{p}_1^{(r-1)})^{m-x_i} \cdot \hat{q}^{(r-1)}}{\binom{m}{x_i} (\hat{p}_1^{(r-1)})^{x_i}(1-\hat{p}_1^{(r-1)})^{m-x_i} \cdot \hat{q}^{(r-1)} + \binom{m}{x_i} (\hat{p}_2^{(r-1)})^{x_i}(1-\hat{p}_2^{(r-1)})^{m-x_i} \cdot (1-\hat{q}^{(r-1)})}$$

Now, to perform the EM, we will find the expected likelihood of our sample with respect to the latent z_i's. To simplify notation, we let

$$Q(\Theta^{(r)}, \Theta^{(r-1)}) = E_{Z|X,p_1,p_2,q}[L(X|p_1, p_2, q, Z)]$$
$$q(\Theta^{(r)}, \Theta^{(r-1)}) = E_{Z|X,p_1,p_2,q}[log(L(X|p_1, p_2, q, Z))]$$

where $\Theta = (q, p_1, p_2)$. So we have

$$Q(\Theta^{(r)}, \Theta^{(r-1)}) = E_{Z|X,p_1,p_2,q}[\Pi_{i=1}^{n}[\binom{m}{x_i} p_1^{x_i}(1-p_1)^{m-x_1} q]^{z_i} \cdot [\binom{m}{x_i} p_2^{x_i}(1-p_2)^{m-x_1}(1-q)]^{1-z_i}]$$

and

$$q(\Theta^{(r)}, \Theta^{(r-1)}) = E_{Z|X,p_1,p_2,q}[log(\Pi_{i=1}^{n}[\binom{m}{x_i} p_1^{x_i}(1-p_1)^{m-x_1}q]^{z_i} \cdot [\binom{m}{x_i} p_2^{x_i}(1-p_2)^{m-x_1}(1-q)]^{1-z_i}])$$
$$= E_{Z|X,p_1,p_2,q}[\Sigma_{i=1}^{n}[z_i[log\binom{m}{x_i} + x_i log(p_1) + (m-x_i)log(1-p_1) + log(q)] +$$
$$(1-z_i)[log\binom{m}{x_i} + x_i log(p_2) + (m-x_i)log(1-p_2) + log(1-q)]]]$$

Now that we have the expected log-likelihood, we can push the expectation through to obtain

$$q(\Theta^{(r)}, \Theta^{(r-1)}) = \Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}[log\binom{m}{x_i} + x_i log(p_1) +$$
$$(m-x_i)log(1-p_1) + log(q)] + (1-\hat{\gamma}_{1i}^{(r)})[log\binom{m}{x_i} + x_i log(p_2) + (m-x_i)log(1-p_2) + log(1-q)]]$$

Finding Derivatives to Maximize Iteratively

We will now take the derivative with respect to the $(r+1)^{th}$ iterations, and since $\hat{\gamma}_{1i}^{(r)}$ is based on the $(r-1)^{th}$ iteration, it will be treated as a constant. We now differentiate with respect to q and set this equal to 0 first:

$$\frac{dq(\Theta^{(r)}, \Theta^{(r-1)})}{dq} = \Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}\frac{1}{q} - (1-\hat{\gamma}_{1i}^{(r)})\frac{1}{1-q}] = 0$$
$$\implies \frac{1}{q(1-q)}\Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}(1-q) - (1-\hat{\gamma}_{1i}^{(r)})q] = 0$$
$$\implies \frac{1}{q(1-q)}\Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)} - q] = 0$$
$$\implies \Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)} = nq$$
$$\implies \frac{\Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)}}{n} = \hat{q}^{(r+1)}$$

Next, we will take the derivative with respect to p_1, which gives

$$\frac{dq(\Theta^{(r)}, \Theta^{(r-1)})}{dp_1} = \Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}[\frac{x_i}{p_1} - \frac{m-x_i}{1-p_1}]] = 0$$
$$\implies \frac{1}{p_1(1-p_1)}\Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}[x_i(1-p_1) - (m-x_i)p_1]] = 0$$
$$\implies \Sigma_{i=1}^{n}[\hat{\gamma}_{1i}^{(r)}[x_i - mp_1]] = 0$$
$$\implies \Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)}x_i = \Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)}mp_1$$
$$\implies \frac{\Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)}x_i}{m\Sigma_{i=1}^{n}\hat{\gamma}_{1i}^{(r)}} = \hat{p_1}^{(r+1)}$$

From the symmetry of the problem, we find that

$$\frac{\Sigma_{i=1}^{n}(1-\hat{\gamma}_{1i}^{(r)})x_i}{m\Sigma_{i=1}^{n}(1-\hat{\gamma}_{1i}^{(r)})} = \hat{p_2}^{(r+1)}$$

The Iterative Method

Now we have an iterative method:

- Initialize values for $\hat{q}, \hat{p_1}, \hat{p_2}$
- Use the values to find $\hat{\gamma}_{1i}$
- Use $\hat{\gamma}_{1i}$ to update $\hat{q}$, $\hat{p_1}$, $\hat{p_2}$
- Repeat steps 2-3 until convergence

Lastly, all we must do is implement this code in R. The R code is in this repository and covers data generation and running the algorithm. The algorithm and results are shown below.

```r
n<-1000
P.init<-c(.1, .6, .7)

# True Values
m<-20
q<-.4
p1<-.3
p2<-.9
P.true<-c(q, p1, p2)

# Generate Data
dat<-function(n){
  Z<-rbinom(n, 1, q)
  X<-c()
  for(i in 1:n){
    if(Z[i]==1){
      Xi<-rbinom(1,m,p1)
    }
    else{
      Xi<-rbinom(1,m,p2)
    }
    X<-c(X, Xi)
  }

  return(X)
}

# Define variable to calculate posterior probabilities
gamma1<-function(X, P){
  gam<-c()
  for(i in 1:n){
    num<-P[1]*dbinom(X[i], m, P[2])
    denom<-P[1]*dbinom(X[i], m, P[2])+(1-P[1])*dbinom(X[i], m, P[3])
    g<-num/denom
    gam<-c(gam,g)
  }
  return(gam)
}

# EM Algorithm Function
EM<-function(data, P, epsilon=.00001){
  q_vec<-c()
  p1_vec<-c()
  p2_vec<-c()
```

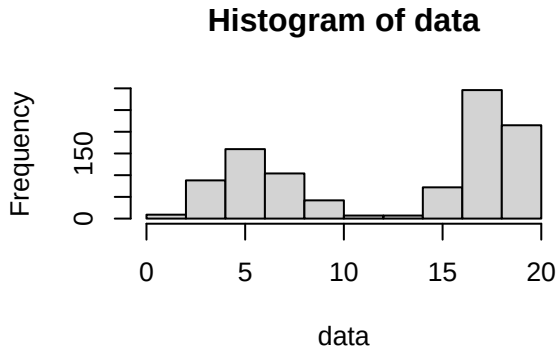
```
  P.old<-P

  gamma1_n<-gamma1(data, P.old)
  gamma0_n<-1-gamma1(data, P.old)

  q_n<-sum(gamma1_n)/sum(gamma0_n+gamma1_n)
  p1_n<-sum(gamma1_n*data)/sum(m*gamma1_n)
  p2_n<-sum(gamma0_n*data)/sum(m*gamma0_n)

  q_vec<-c(q_vec, q_n)
  p1_vec<-c(p1_vec, p1_n)
  p2_vec<-c(p2_vec, p2_n)

  P.new<-c(q_n, p1_n, p2_n)

  count<-0
  while(sum((P.new-P.old)^2)>=epsilon){
    P.old<-P.new

    gamma1_n<-gamma1(data, P.old)
    gamma0_n<-1-gamma1(data, P.old)

    q_n<-sum(gamma1_n)/sum(gamma0_n+gamma1_n)
    p1_n<-sum(gamma1_n*data)/sum(m*gamma1_n)
    p2_n<-sum(gamma0_n*data)/sum(m*gamma0_n)

    q_vec<-c(q_vec, q_n)
    p1_vec<-c(p1_vec, p1_n)
    p2_vec<-c(p2_vec, p2_n)

    P.new<-c(q_n, p1_n, p2_n)

    count<-count+1
    if(count%%100==0){
      print(count)
    }
    if(count>5000){
      break
    }

  }
  return(list(est=P.new, q_trace=q_vec, p1_trace=p1_vec, p2_trace=p2_vec, iterations=count))
}

data<-dat(n)
res<-EM(data, P.init)
```

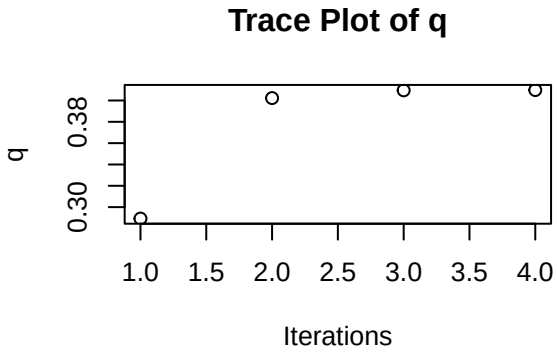
Results

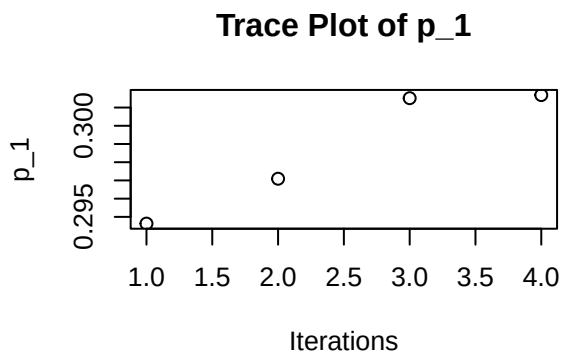
Plotting the data as well as a trace plot for each parameter gives the following

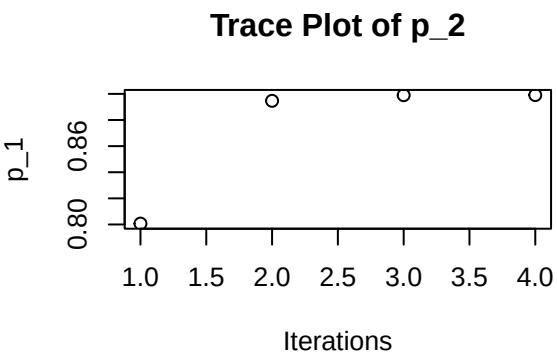
```
# PLotting the data as well as the trace of our parameters
par(mfrow=c(2,2))
hist(data)
```

```
plot(res$q_trace, ylab="q", xlab="Iterations", main="Trace Plot of q")
plot(res$p1_trace, ylab="p_1", xlab="Iterations", main="Trace Plot of p_1")
plot(res$p2_trace, ylab="p_1", xlab="Iterations", main="Trace Plot of p_2")
```









The above gives that the data is a mixture of two binomial distributions with parameters $p_1 = .29$, $p_2 = .89$ with mixture parameter $q = .39$, which is extremely close to the true values of the parameters which were $p_1 = .3$, $p_2 = .9$, $q = .4$. Thus, we have successfully distinguished between the two distributions that compose the data as well as finding the mixture parameter involved in the data generation process.

Example 2: Missing Data

The following problem is taken from Casella and Berger chapter 7 exercise 29. In this problem, we observe independent paired data (X_i, Y_i) for $i = 1, 2, \cdots, n$ where $Y_i \sim pois(m\beta\tau_i)$ and $(X_1, X_2, \cdots, X_n) \sim MN(m, \boldsymbol{\tau})$ with $\boldsymbol{\tau} = (\tau_1, \tau_2, \cdots, \tau_n)$ and $\sum_{i=1}^n \tau_i = 1$ and $\sum_{i=1}^n x_i = m$. We first find the joint mass function of (Y, X).

Distributional Statement

Since the multinomial mass function is $f(x) = \frac{m!}{x_1!x_2!\cdots x_n!}\tau_1^{x_1}\tau_2^{x_2}\cdots\tau_n^{x_n}$ and the poisson mass function is $f(y_i) = \frac{e^{-m\beta\tau_i}(m\beta\tau_i)^{y_i}}{y_i!}$, the likelihood of X and Y is given by:

$$f(\mathbf{y}, \mathbf{x}|\beta, \boldsymbol{\tau}) = m! \prod_{i=1}^n \frac{e^{-m\beta\tau_i}(m\beta\tau_i)^{y_i}}{y_i!} \frac{\tau_i^{x_i}}{x_i!}$$

This gives us the following log-likelihood:

$$l = \ln(m!) + \sum_{i=1}^{n}[-m\beta\tau_i + y_i \ln(m\beta\tau_i) + x_i \ln(\tau_i) - \ln(y_i!x_i!)]$$

Finding Estimates with Complete Data

First, we will find the maximum likelihood estimators by using the complete data. This amounts so simply differentiating the log-likelihood with respect to the parameters (β, τ_i) and setting this equal to zero to find estimates. We do this now:

$$\frac{dl}{d\beta} = \sum_{i=1}^{n}[-m\tau_i + \frac{y_i}{\beta}] = 0 \implies \hat{\beta} = \frac{\sum_{i=1}^{n} y_i}{m\sum_{i=1}^{n} \hat{\tau_i}}$$

However, recall that $\sum_{i=1}^{n} \hat{\tau}_i = 1$ and that $\sum_{i=1}^{n} x_i = m$, so the estimate above can be written as

$$\hat{\beta} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i}$$

Now, we find the estimates for τ_i

$$\frac{dl}{d\tau_i} = -m\beta + \frac{y_i + x_i}{\tau_i} \implies \hat{\tau_i} = \frac{y_i + x_i}{m\hat{\beta}}$$

Now, note that, since $1 = \sum_{i=1}^{n} \hat{\tau}_i = \frac{\sum_{i=1}^{n}[y_i + x_i]}{m\hat{\beta}}$, we have that $m\hat{\beta} = \sum_{i=1}^{n}[y_i + x_i]$, which gives us the following estimate for τ_i

$$\hat{\tau_i} = \frac{y_i + x_i}{\sum_{i=1}^{n}[y_i + x_i]}$$

Finding Estimates with Missing Data

While the above is useful is we have all the data, we will suppose we are missing x_1. We will use the above forms as a starting point for our analysis. See that we can write both forms above in terms of x_1

$$\hat{\beta} = \frac{\sum_{i=1}^{n} y_i}{x_1 + \sum_{i=1}^{n} x_i}$$
$$\hat{\tau_i} = \frac{y_i + x_i}{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} x_i + x_1}$$

We also know that, marginally, $X_1 \sim bin(m, \tau_1)$, so, heuristically, we can guess that the missing data estimates will have the form:

$$\hat{\beta}^{(r+1)} = \frac{\sum_{i=1}^{n} y_i}{m\hat{\tau_1}^{(r)} + \sum_{i=1}^{n} x_i}$$
$$\hat{\tau}_i^{(r+1)} = \frac{y_i + x_i}{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} x_i + m\hat{\tau_1}^{(r)}}, \quad \text{for } i \neq 1$$
$$\hat{\tau}_1^{(r+1)} = \frac{y_1 + m\hat{\tau_1}^{(r)}}{\sum_{i=1}^{n} y_i + \sum_{i=1}^{n} x_i + m\hat{\tau_1}^{(r)}}$$

We can make this process rigorous by reintroducing the conditional expected log-likelihood that we saw in the first example. Let

$$q(\Theta^{(r)}, \Theta^{(r-1)}) = E_{X_1|X,Y}(l)$$

We will differentiate q with respect to our parameters. See that

$$\begin{aligned}
\frac{dq}{d\beta} = \frac{d}{d\beta}E_{X_1|X,Y}(l) = E_{X_1|X,Y}(\frac{dl}{d\beta}) &= E_{X_1|X,Y}(\sum_{i=1}^{n}[-m\tau_i + \frac{y_i}{\beta}]) \\
&= E_{X_1|X,Y}(-m\sum_{i=1}^{n}\tau_i + \frac{1}{\beta}\sum_{i=1}^{n}y_i) \\
&= E_{X_1|X,Y}(-m + \frac{1}{\beta}\sum_{i=1}^{n}y_i) \\
&= E_{X_1|X,Y}(-\sum_{i=1}^{n}x_i + \frac{1}{\beta}\sum_{i=1}^{n}y_i) \\
&= E_{X_1|X,Y}(-x_1 - \sum_{i=2}^{n}x_i + \frac{1}{\beta}\sum_{i=1}^{n}y_i) \\
&= -E_{X_1|X,Y}(x_1) - \sum_{i=2}^{n}x_i + \frac{1}{\beta}\sum_{i=1}^{n}y_i \\
&= -m\tau_1^{(r)} - \sum_{i=2}^{n}x_i + \frac{1}{\beta^{(r+1)}}\sum_{i=1}^{n}y_i
\end{aligned}$$

Setting this equal to zero and solving gives the following estimate

$$\hat{\beta}^{(r+1)} = \frac{\sum_{i=1}^{n}y_i}{m\hat{\tau_1}^{(r)} + \sum_{i=2}^{n}x_i}$$

Similarly, for τ_j

$$\begin{aligned}
\frac{dq}{d\tau_j} = \frac{d}{d\tau_j}E_{X_1|X,Y}(l) = E_{X_1|X,Y}(\frac{dl}{d\tau_j}) &= E_{X_1|X,Y}(-m\beta + \frac{y_j + x_j}{\tau_j}) \\
&= -m\beta^{(r)} + E_{X_1|X,Y}(\frac{y_j + x_j}{\tau_j^{(r+1)}}) \\
&= -m\beta^{(r)} + \frac{y_j}{\tau_j^{(r+1)}} + E_{X_1|X,Y}(\frac{x_j}{\tau_j^{(r+1)}}) \\
&= -m\beta^{(r)} + \frac{y_j}{\tau_j^{(r+1)}} + E_{X_1|X,Y}(\frac{x_j}{\tau_j^{(r+1)}}) \\
&= -m\beta^{(r)} + \frac{y_j}{\tau_j^{(r+1)}} + \frac{E_{X_1|X,Y}(x_j)}{\tau_j^{(r+1)}}
\end{aligned}$$

Now, if $j = 1$, we get the following when we set the above expression equal to zero

$$\hat{\tau_1}^{(r+1)} = \frac{y_1 + m\hat{\tau_1}^{(r)}}{m\hat{\beta}^{(r)}}$$

and if $j \neq 1$

$$\hat{\tau_j}^{(r+1)} = \frac{y_j + x_j}{m\hat{\beta}^{(r)}}$$

The Iterative Method

The above lends itself nicely to an iterative method:

- Initialize $\hat{\beta}^{(0)}$ and $\hat{\tau_j}^{(0)}$
- Calculate $\hat{\beta}^{(r+1)}$
- Calculate $\hat{\tau_j}^{(r+1)}$
- Iterate to convergence

To illustrate this, we implement the above algorithm in R.

```
# Initialize the true values
n<-100
m<-200
beta<-5
tau<-rep(1/n, n)
vrai<-c(tau, beta)

# Generate the Data
Y<-c()
for(i in 1:n){
  y<-rpois(1, m*beta*tau[i])
  Y<-c(Y,y)
}

X<-rmultinom(1, m, tau)

# Initialize Starting Guesses for Parameters
t<-c(rep(1/(2*n), n/2 ), rep(3/(2*n), (n/2)))
b<-3
init<-c(t,b)

# EM algorithm Function
EM<-function(X, Y, init, epsilon){
  vec<-init
  t<-vec[1:n]
  b<-vec[n+1]
  k<-0
  while(sum((vrai-vec)^2)>=epsilon){
    b0 <- b
    t0 <- t

    b<-sum(Y)/(m*t0[1]+sum(X[2:n]))

    t[1]<-(Y[1]+m*t0[1])/(m*b)

    for(i in 2:n){
      t[i]<-(Y[i]+X[i])/(m*b)
    }


    vec<-c(t,b)


    k<-k+1
    if(k>750000){
```

```
      break
    }

  }
  return(list(sol=vec, iter=k, error_tau=sum((vrai[1:n]-vec[1:n])^2),
              error_beta=sum((vrai[n+1]-vec[n+1])^2)))
}


res<-EM(X, Y, init, .00001)
```

The above algorithm gives the following estimates for $\boldsymbol{\tau}$ and β:

```
## Estimate for tau
```

```
##   [1] 0.008771479 0.014387883 0.009891669 0.008093184 0.011690155 0.008093184
##   [7] 0.009891669 0.014387883 0.010790912 0.009891669 0.003596971 0.010790912
##  [13] 0.016186368 0.006294699 0.012589397 0.009891669 0.017984853 0.007193941
##  [19] 0.012589397 0.012589397 0.006294699 0.008992427 0.015287125 0.011690155
##  [25] 0.009891669 0.014387883 0.014387883 0.012589397 0.015287125 0.015287125
##  [31] 0.011690155 0.013488640 0.009891669 0.015287125 0.009891669 0.008992427
##  [37] 0.013488640 0.014387883 0.008992427 0.017984853 0.012589397 0.009891669
##  [43] 0.015287125 0.014387883 0.015287125 0.009891669 0.006294699 0.009891669
##  [49] 0.008992427 0.015287125 0.010790912 0.007193941 0.013488640 0.010790912
##  [55] 0.016186368 0.009891669 0.016186368 0.007193941 0.014387883 0.013488640
##  [61] 0.010790912 0.008992427 0.008992427 0.011690155 0.016186368 0.010790912
##  [67] 0.017984853 0.009891669 0.013488640 0.012589397 0.008093184 0.007193941
##  [73] 0.016186368 0.007193941 0.015287125 0.011690155 0.011690155 0.007193941
##  [79] 0.012589397 0.007193941 0.013488640 0.011690155 0.016186368 0.008992427
##  [85] 0.012589397 0.006294699 0.011690155 0.008992427 0.010790912 0.015287125
##  [91] 0.013488640 0.009891669 0.009891669 0.010790912 0.011690155 0.013488640
##  [97] 0.008093184 0.013488640 0.017984853 0.008992427
```

```
## Estimate for beta  5.560234
```

Where the true value of $\boldsymbol{\tau}$ and β are:

```
## True tau
```

```
##   [1] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [16] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [31] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [46] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [61] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [76] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
##  [91] 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
```

```
## True beta  5
```