

HW04

Zach White

March 4, 2017

Problem 5.20

```
iter.n = 1
x1 = c(1,1,-1,-1,2,2,-2,-2,NA,NA,NA,NA)
x2 = c(1,-1,1,-1,NA,NA,NA,NA,2,2,-2,-2)
full.data = cbind(x1,x2)
cc.data = full.data[complete.cases(full.data),]
cc.n = nrow(cc.data)

r1 = as.numeric(!is.na(x1))
r1.not = which(is.na(x1))
r2 = as.numeric(!is.na(x2))
r2.not = which(is.na(x2))
r = cbind(r1,r2)
# Initial Values
sigma1.2 = 5
sigma2.2 = 2
sigma12 = 3
Sigmat = matrix(c(sigma1.2,sigma12,sigma12,sigma2.2),nrow = 2,ncol = 2)

full.data[is.na(full.data)] = 0
max.diff = 1
n = nrow(full.data)
itermat = NULL
# Second try
while(max.diff > .001){
  ex1 <- (r[,1]==1)*full.data[,1] + (r[,1]==0)*( Sigmat[1,2]*(full.data[,2])/Sigmat[2,2])
  ex2 <- (r[,2]==1)*full.data[,2] + (r[,2]==0)*(Sigmat[1,2]*(full.data[,1])/Sigmat[1,1])
  ex1.2 <- (r[,1]==1)*full.data[,1]^2 + (r[,1]==0)*(Sigmat[1,1]-Sigmat[1,2]^2/Sigmat[2,2]+ ex1^2)
  ex2.2 <- (r[,2]==1)*full.data[,2]^2 + (r[,2]==0)*(Sigmat[2,2]-Sigmat[1,2]^2/Sigmat[1,1]+ ex2^2)
  ex1x2 <- (r[,1]*r[,2]*full.data[,1]*full.data[,2] + r[,1]*(1-r[,2])*full.data[,1]*ex2
    + (1-r[,1])*r[,2]*ex1*full.data[,2])

  mu.new = c(mean(ex1),mean(ex2))
  s11 = mean(ex1.2) - mu.new[1]^2
  s22 = mean(ex2.2) - mu.new[2]^2
  s12 = mean(ex1x2) - mu.new[1]*mu.new[2]
  # Define these as the previous parameters
  sigma.mat = c(s11,s22,s12)

  sigma1.2.diff = abs(s11 - sigma1.2)
  sigma2.2.diff = abs(s22 - sigma2.2)
  sigma12.diff = abs(s12 - sigma12)
  max.diff = max(sigma12.diff,sigma2.2.diff,sigma1.2.diff)
  itermat = rbind(itermat,c(iter.n,sigma.mat))
}
```

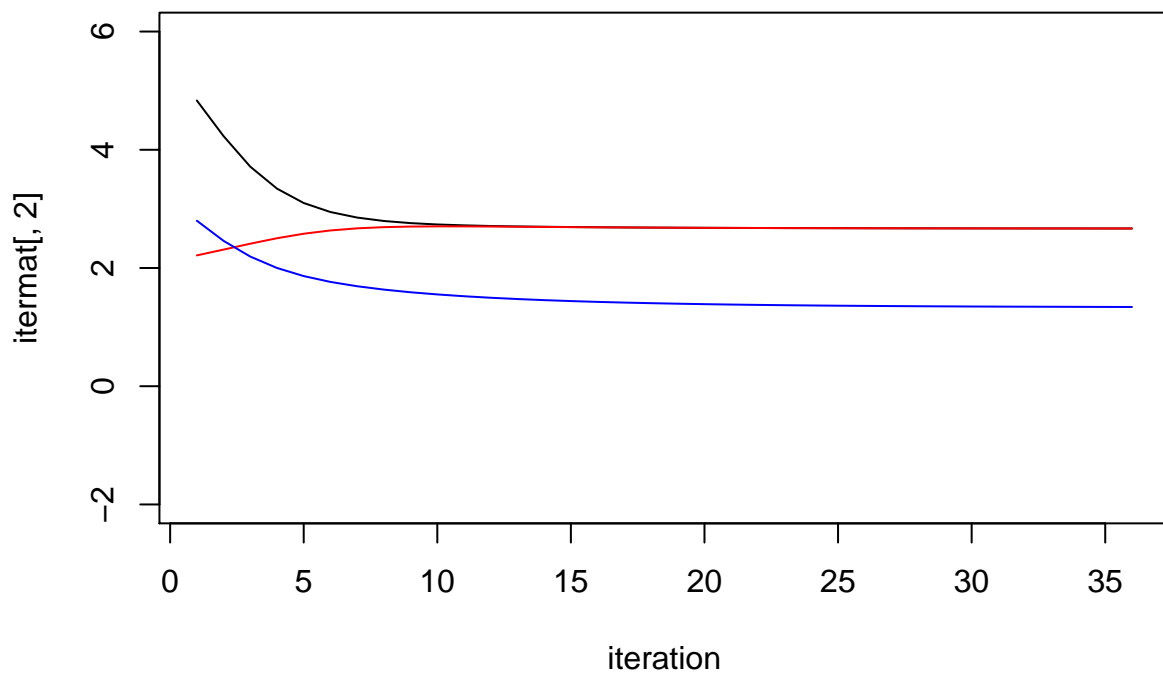
```

sigma1.2 = sigma.mat[1]
sigma2.2 = sigma.mat[2]
sigma12 = sigma.mat[3]

Sigmat = matrix(c(sigma1.2,sigma12,sigma12,sigma2.2),nrow = 2,ncol = 2)

iter.n = iter.n + 1
}
plot(itermat[,1],itermat[,2], type = "l", xlab = "iteration",ylim = c(-2,6))
lines(itermat[,1],itermat[,3], type = "l" , col = "red")
lines(itermat[,1],itermat[,4], type = "l" , col = "blue")

```



It's clear that these all converge very quickly. With the starting values of $\sigma_{11}^2 = 1$, $\sigma_{22}^2 = 2$, $\sigma_{21}^2 = 3$, they converge to 2.6683567, 2.6683567, and 1.3405923

Problem 26

Part A

```

x = c(125,18,20,34)

start.val = chain = current = diff = .1
iter.count = 2

```

```

while(diff > .0001){
  new.val = ((current * x[1] / (2+ current)) + x[4]) / ((current*x[1] / (2 + current)) + x[2] + x[3] + 1)
  chain = c(chain,new.val)
  diff = abs(current - chain[iter.count])
  current = chain[iter.count]
  iter.count = iter.count + 1
}

```

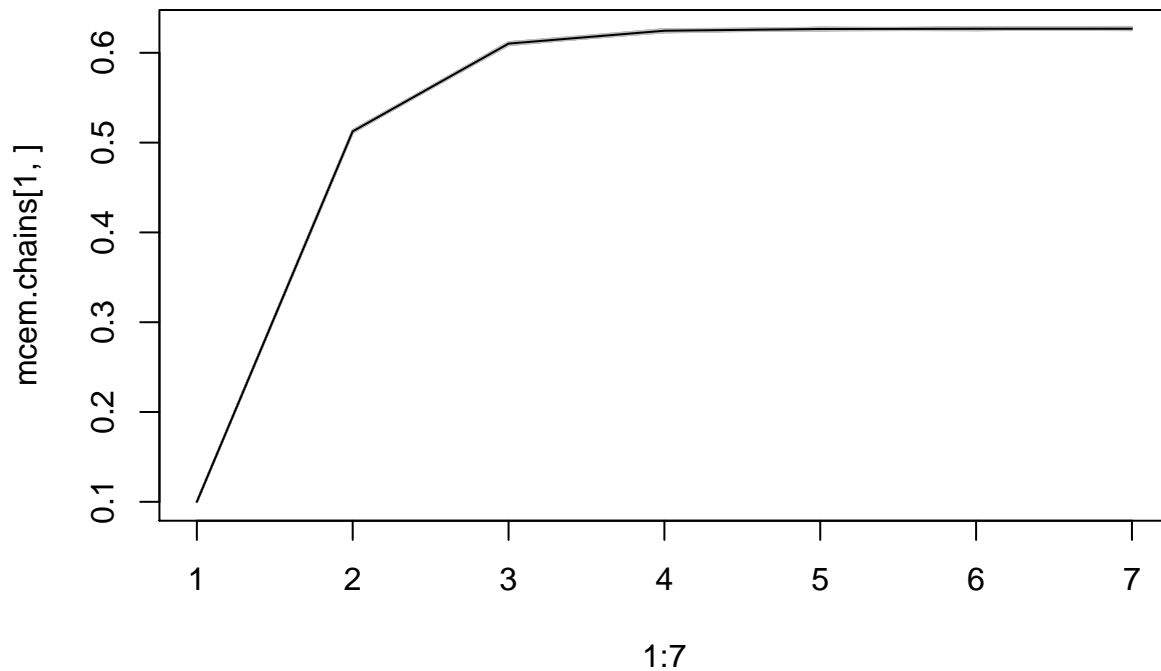
Part B

I now use Monte Carlo EM algorithm. This means we will simulate some binomial draws and then insert an approximated value into a part of the equation

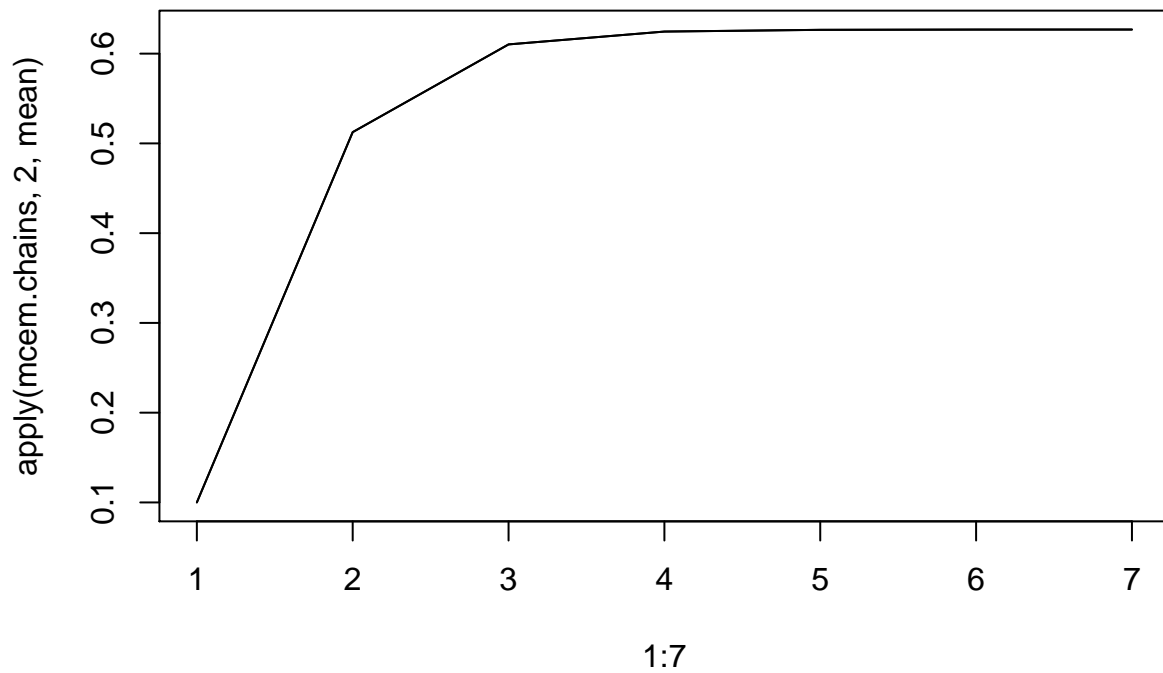
```

M = 1000
mcm.chains = matrix(start.val, length(chain), nrow = 500)
for(i in 2:length(chain)){
  mcm.chains[,i] = 1/(1+(x[2]+x[3])/(x[4]+rbinom(500,M*x[1],
1/(1+2/mcm.chains[,i-1]))/M))
}
plot(1:7,mcm.chains[1,], type = "l")
for(i in 2:500){
  lines(1:7, mcm.chains[i,],type = "l",col = "grey")
}
lines(1:7, chain)

```



```
plot(1:7, apply(mcem.chains,2,mean), type = "l")  
lines(1:7,chain)
```



Both of these methods converge very quickly. Even under the Monte Carlo method, our converges very quickly. I assume it is partly because of the number of iterations we choose. If that were smaller, the variance would be larger, and convergence would be slower.