Bayesian Methods - Assignment 5

Ryan Durfey

May 6, 2016

```
library(rjags)
library(runjags)
source("//Users/rdurfey/R_misc/BayesianMethods/DBDA2Eprograms/DBDA2E-utilities.R")
```

```
##
## *********************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *********************************************************************
```

```
# read in data
dat<-read.csv("//Users/rdurfey/R_misc/BayesianMethods/CourseProject/MScA_32014_BayesianMethods_CoursePr
head(dat)
```

```
##      sex    race   age      education state y
## 1 1.Male 1.White 18-24     1.NoCollege    GA 0
## 2 1.Male 1.White 25-34     1.NoCollege    AZ 0
## 3 1.Male 1.White 25-34   2.SomeCollege    SD 0
## 4 1.Male 1.White 18-24 3.CollegeOrMore    SC 0
## 5 1.Male 1.White 18-24 3.CollegeOrMore    SC 0
## 6 1.Male 1.White 18-24 3.CollegeOrMore    SC 0
```

Hierarchical Model

```
# data prep
y=dat$y
s<-as.numeric(dat$sex)
(Ntotal<-length(y))
```

```
## [1] 23223
```

```
(Nsubj<-length(unique(s)))
```

```
## [1] 2
```

```
dataList<-list(y=y,s=s,Ntotal=Ntotal,Nsubj=Nsubj)

# model prep
modelString <- "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dbern( theta[s[i]] )
  }
  for ( sIdx in 1:Nsubj ) {
    theta[sIdx] ~ dbeta( 100 , 100 ) # N.B.: 2,2 prior; change as appropriate.
  }
}
"
writeLines( modelString , con="TEMPmodel.txt" )

# initialize chains
initsList = function() {
  thetaInit = rep(0,Nsubj)
  for ( sIdx in 1:Nsubj ) { # for each subject
    includeRows = ( s == sIdx ) # identify rows of this group
    yThisSubj = y[includeRows]  # extract data of this group
    resampledY = sample( yThisSubj , replace=TRUE ) # resample
    thetaInit[sIdx] = sum(resampledY)/length(resampledY)
  }
  thetaInit = 0.001+0.998*thetaInit # keep away from 0,1
  return( list( theta=thetaInit ) )
}

# send the model to JAGS
parameters = c( "theta")     # The parameters to be monitored
adaptSteps = 500             # Number of steps to adapt the samplers
burnInSteps = 500            # Number of steps to burn-in the chains
nChains = 4                  # nChains should be 2 or more for diagnostics
numSavedSteps<-50000
nIter = ceiling(numSavedSteps / nChains )

# create, initialize, and adapt the model
jagsModel = jags.model( "TEMPmodel.txt" , data=dataList , inits=initsList ,
                        n.chains=nChains , n.adapt=adaptSteps )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 23223
##    Unobserved stochastic nodes: 2
##    Total graph size: 46454
##
## Initializing model
```

```
# run burn-in
update( jagsModel , n.iter=burnInSteps )

# the main run
codaSamples = coda.samples( jagsModel , variable.names=parameters , n.iter=nIter)
head(codaSamples)
```

```
## [[1]]
## Markov Chain Monte Carlo (MCMC) output:
## Start = 501
## End = 507
## Thinning interval = 1
##       theta[1]  theta[2]
## [1,] 0.3370995 0.3756738
## [2,] 0.3519597 0.3848680
## [3,] 0.3463097 0.3775964
## [4,] 0.3317873 0.3859674
## [5,] 0.3451339 0.3756923
## [6,] 0.3387219 0.3768030
## [7,] 0.3506739 0.3885059
##
## [[2]]
## Markov Chain Monte Carlo (MCMC) output:
## Start = 501
## End = 507
## Thinning interval = 1
##       theta[1]  theta[2]
## [1,] 0.3433247 0.3839714
## [2,] 0.3457631 0.3865902
## [3,] 0.3458708 0.3834056
## [4,] 0.3465827 0.3837366
## [5,] 0.3429097 0.3897394
## [6,] 0.3474132 0.3912559
## [7,] 0.3431602 0.3855227
##
## [[3]]
## Markov Chain Monte Carlo (MCMC) output:
## Start = 501
## End = 507
## Thinning interval = 1
##       theta[1]  theta[2]
## [1,] 0.3442615 0.3830243
## [2,] 0.3480587 0.3831883
## [3,] 0.3513742 0.3849250
## [4,] 0.3326955 0.3798393
## [5,] 0.3435827 0.3903325
## [6,] 0.3444698 0.3886022
## [7,] 0.3399544 0.3850590
##
## [[4]]
## Markov Chain Monte Carlo (MCMC) output:
## Start = 501
## End = 507
## Thinning interval = 1
##       theta[1]  theta[2]
## [1,] 0.3417682 0.3902091
## [2,] 0.3462326 0.3890719
## [3,] 0.3456627 0.3873913
## [4,] 0.3435215 0.3801477
## [5,] 0.3442133 0.3811207
## [6,] 0.3501859 0.3908499
## [7,] 0.3438432 0.3871884
```

```
##
## attr(,"class")
## [1] "mcmc.list"
```

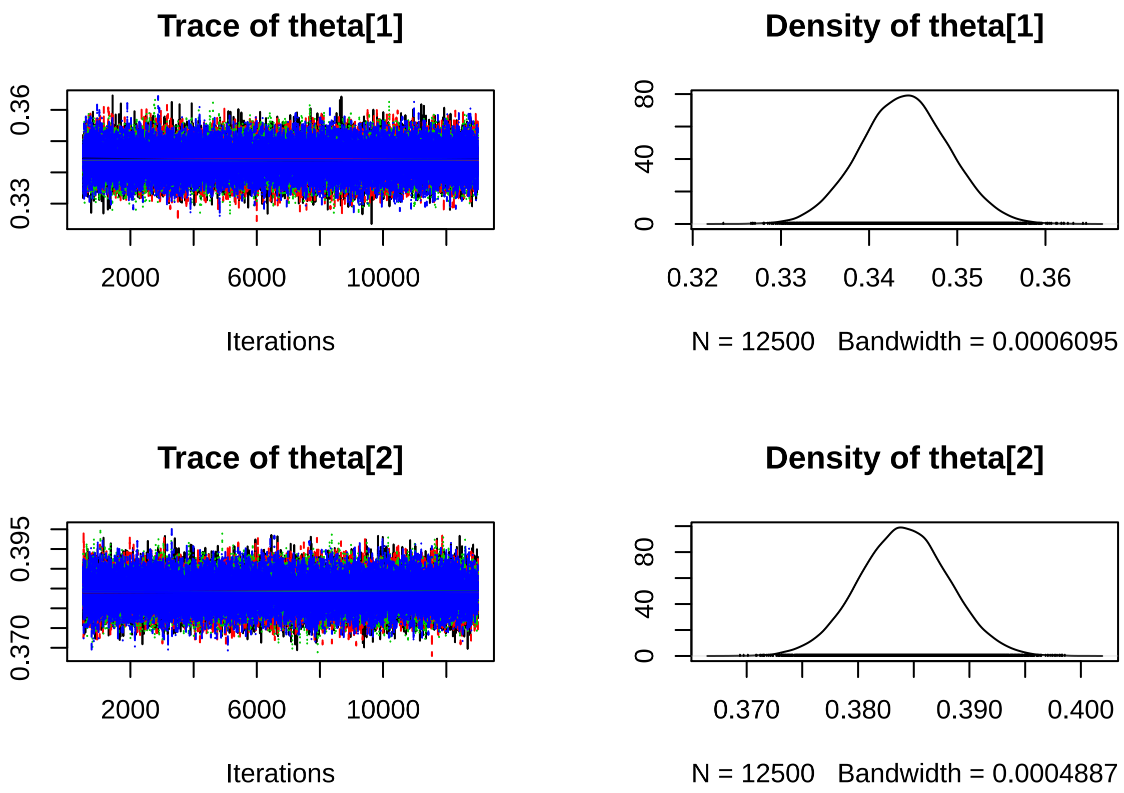
```r
list.samplers(jagsModel)
```

```
## $`bugs::ConjugateBeta`
## [1] "theta[2]"
##
## $`bugs::ConjugateBeta`
## [1] "theta[1]"
```

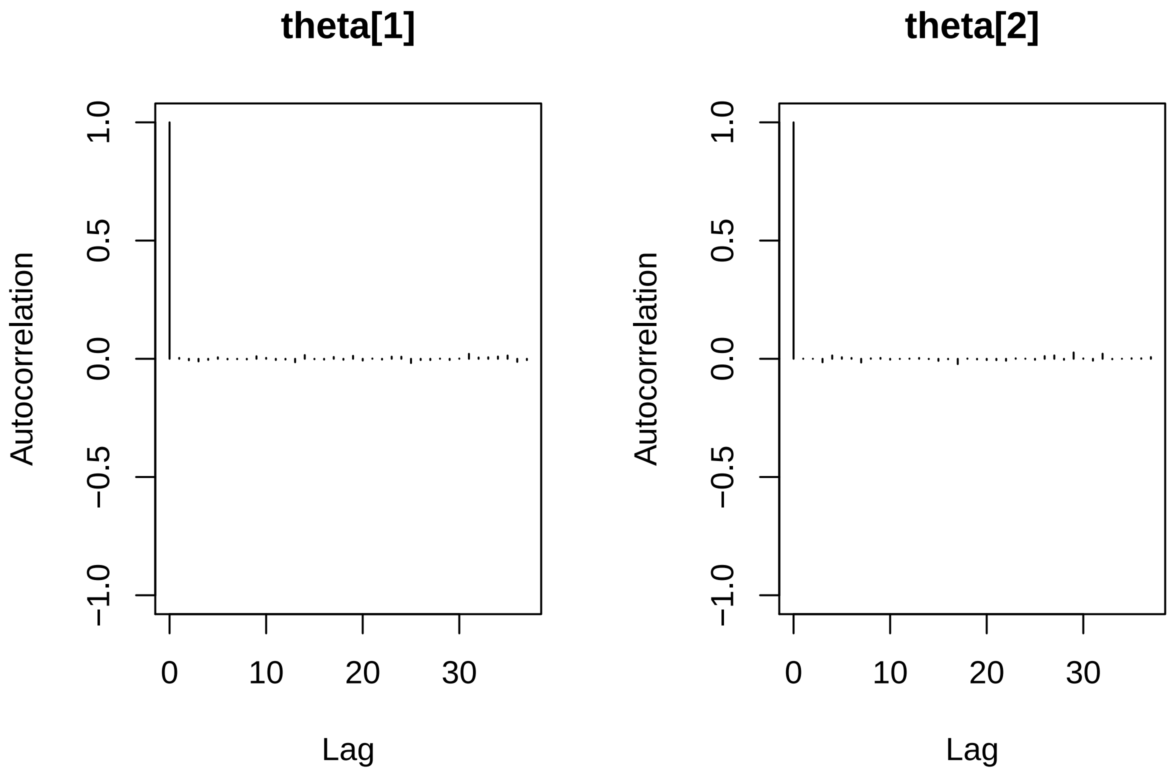
```r
# analysis
summary(codaSamples)
```

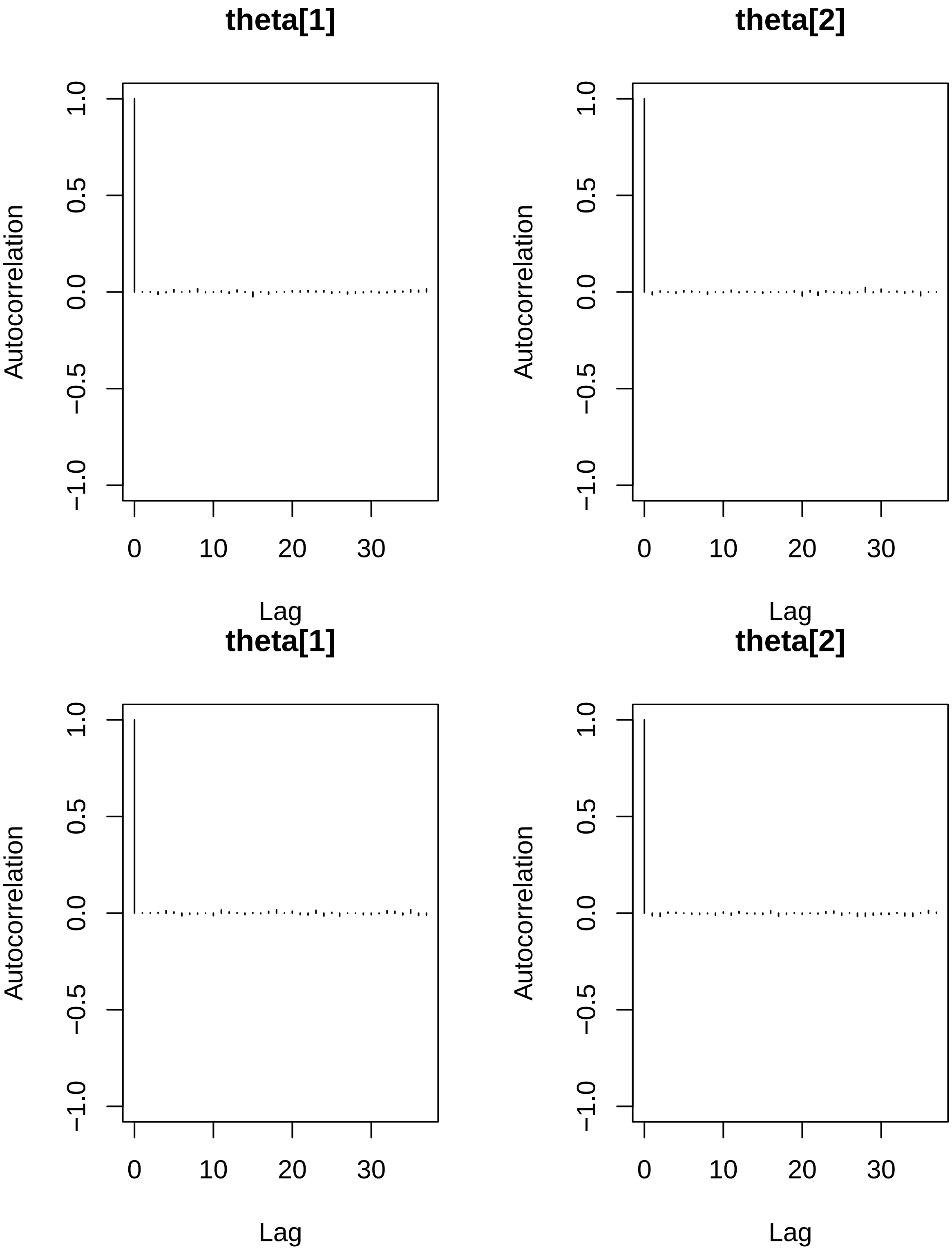
```
##
## Iterations = 501:13000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 12500
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean       SD  Naive SE Time-series SE
## theta[1] 0.3440 0.005006 2.239e-05      2.239e-05
## theta[2] 0.3841 0.004014 1.795e-05      1.775e-05
##
## 2. Quantiles for each variable:
##
##            2.5%    25%    50%    75%  97.5%
## theta[1] 0.3342 0.3407 0.3440 0.3474 0.3539
## theta[2] 0.3763 0.3814 0.3841 0.3868 0.3920
```

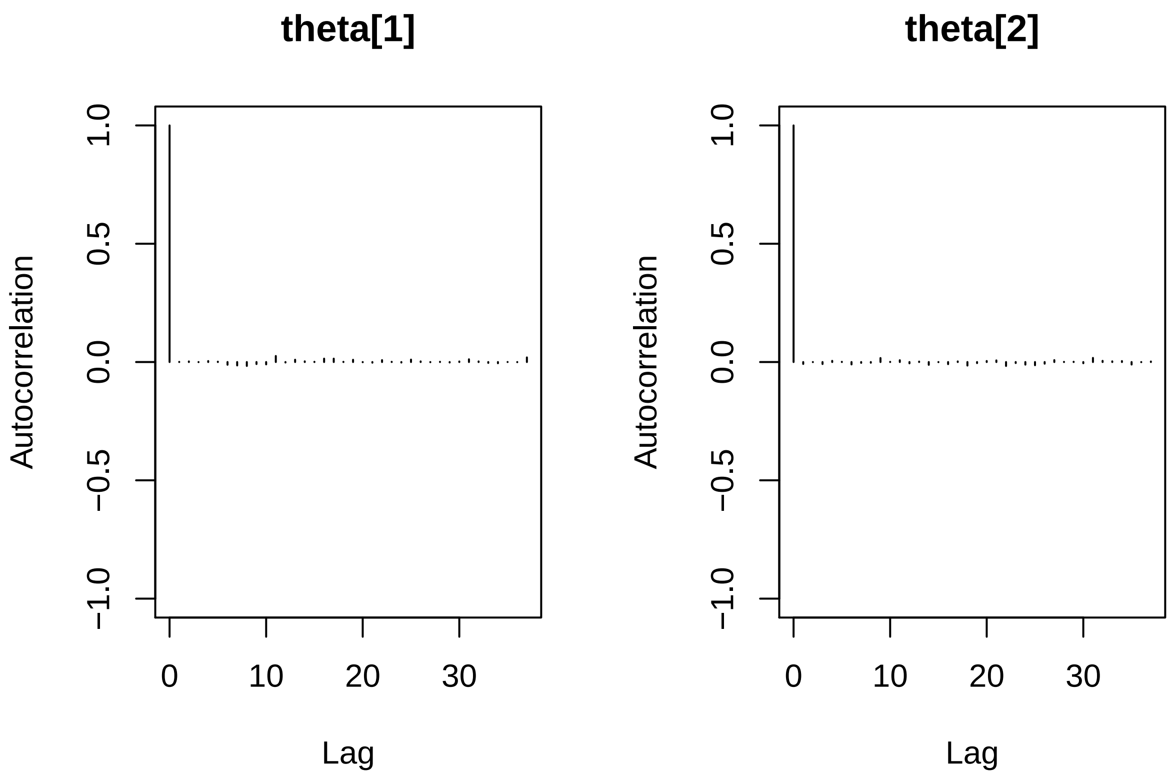
```r
plot(codaSamples)
```



```
autocorr.plot(codaSamples,ask=F)
```




theta[1]
theta[2]
theta[1]
theta[2]
Autocorrelation
−1.0
−0.5
0.0
0.5
1.0
Lag
0
10
20
30



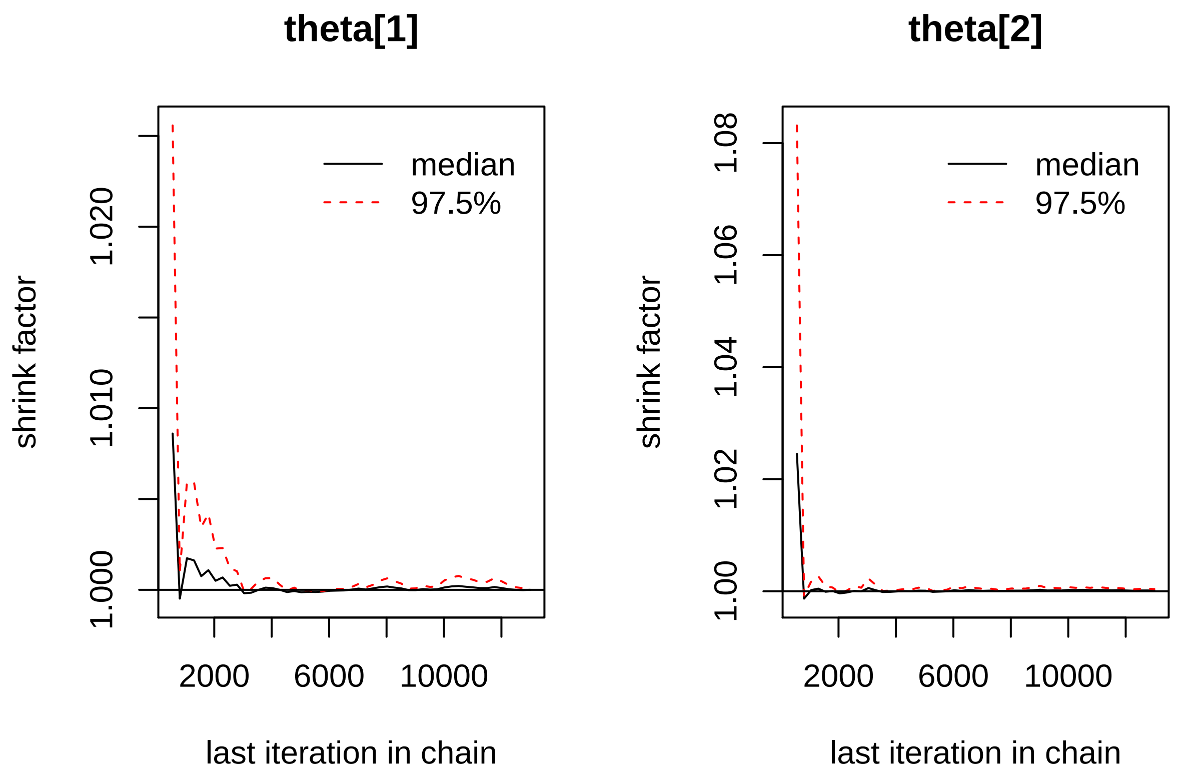
```r
effectiveSize(codaSamples)
```

```
## theta[1] theta[2]
## 50000.00 51155.39
```

```r
gelman.diag(codaSamples)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## theta[1]          1          1
## theta[2]          1          1
##
## Multivariate psrf
##
## 1
```

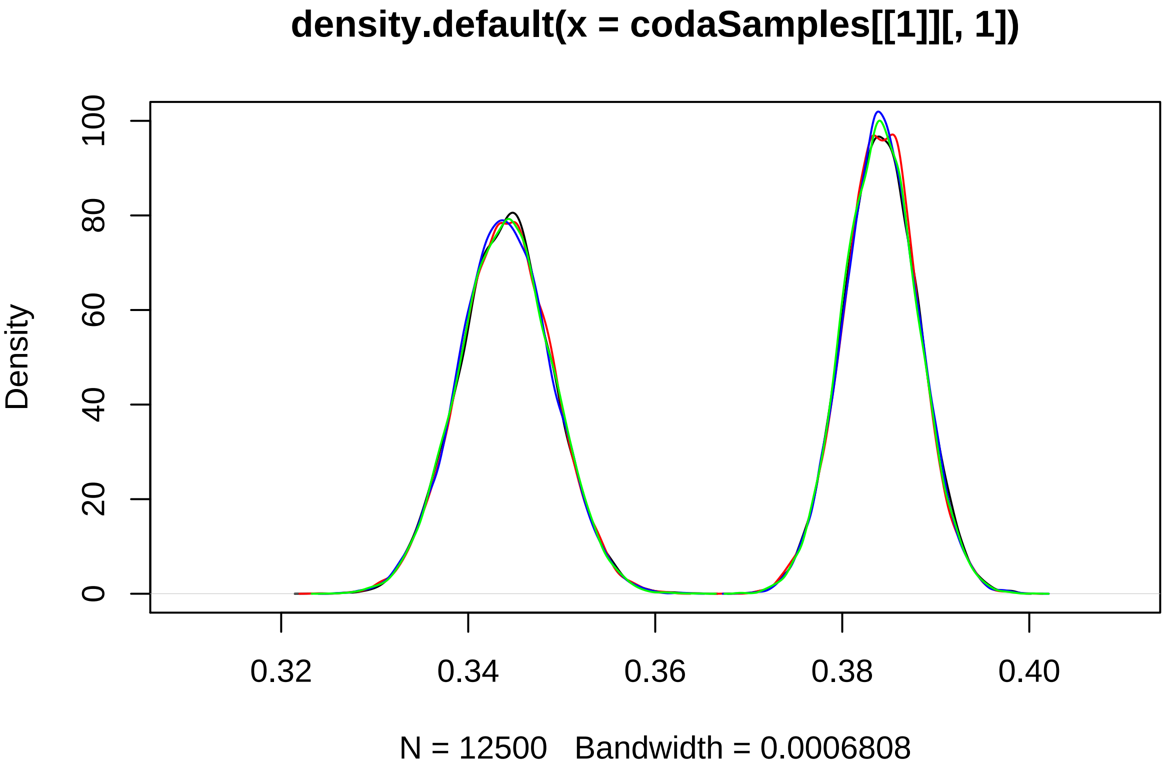
```r
gelman.plot(codaSamples)
```



```
#estimated means
matrix(unlist(lapply(codaSamples,function(z) apply(z,2,mean))),ncol=2,byrow = T)
```

```
##           [,1]      [,2]
## [1,] 0.3440627 0.3841760
## [2,] 0.3440798 0.3841247
## [3,] 0.3439887 0.3841710
## [4,] 0.3440306 0.3840953
```

```
# plot of posterior densities
plot(density(codaSamples[[1]][,1]),xlim=c(0.31,0.41),ylim=c(0,100))
lines(density(codaSamples[[1]][,2]))
lines(density(codaSamples[[2]][,1]),col="red")
lines(density(codaSamples[[2]][,2]),col="red")
lines(density(codaSamples[[3]][,1]),col="blue")
lines(density(codaSamples[[3]][,2]),col="blue")
lines(density(codaSamples[[4]][,1]),col="green")
lines(density(codaSamples[[4]][,2]),col="green")
```



```
# HDIs of each chain
(HDIofChains<-lapply(codaSamples,function(z) cbind(Theta1=HDIofMCMC(z[,1]),Theta2=HDIofMCMC(z[,2]))))
```

```
## [[1]]
##         Theta1   Theta2
## [1,] 0.3343431 0.376293
## [2,] 0.3539668 0.392170
##
## [[2]]
##         Theta1    Theta2
## [1,] 0.3344075 0.3762063
## [2,] 0.3539985 0.3920200
##
## [[3]]
##         Theta1    Theta2
## [1,] 0.3340272 0.3763631
## [2,] 0.3536700 0.3920694
##
## [[4]]
##         Theta1    Theta2
## [1,] 0.3340504 0.3765031
## [2,] 0.3535443 0.3921986
```

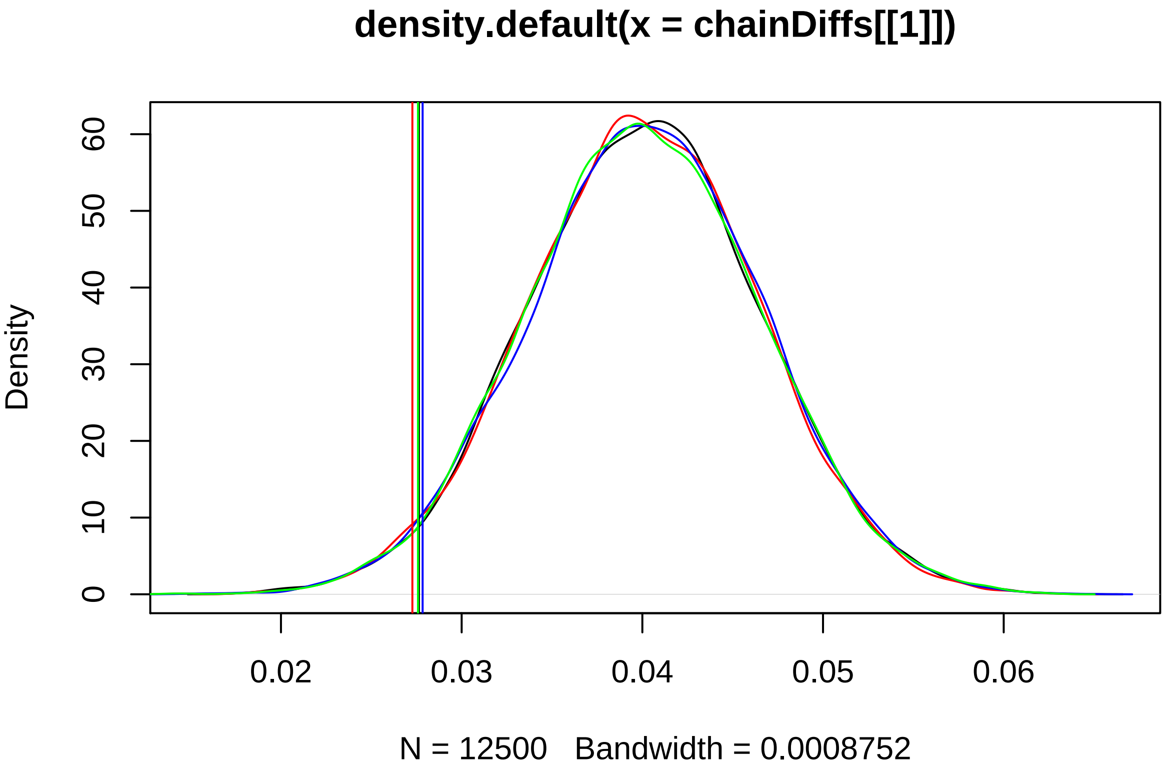
```
# differences between theta1 and theta2
chainDiffs<-lapply(codaSamples,function(z) z[,2]-z[,1])

(leftBounds<-unlist(lapply(chainDiffs,function(z) HDIofMCMC(z,.95)[1])))
```

```
## [1] 0.02763485 0.02727219 0.02783894 0.02758252
```

```
head(chainDiffs[[1]])
```

```
## Markov Chain Monte Carlo (MCMC) output:
## Start = 501
## End = 507
## Thinning interval = 1
## [1] 0.03857433 0.03290829 0.03128669 0.05418010 0.03055844 0.03808108
## [7] 0.03783206
```

```
# plot of left bound of 95% HDI
plot(density(chainDiffs[[1]]))#,xlim=c(-.5,1),ylim=c(0,3),col="black")
lines(density(chainDiffs[[2]]),col="red")
lines(density(chainDiffs[[3]]),col="blue")
lines(density(chainDiffs[[4]]),col="green")
abline(v=leftBounds,col=c("black","red","blue","green"))
```



From the MCMC output and the plots above, we see clear distinction between the two groups. The Male group's distribution is centered around 0.344 while the Female group is centered around 0.384. There is also little to no overlap between the group distributions. This is also evidenced in the differences between the Theta1 and Theta2 not including zero. This means that we can be confident that there is indeed a difference between the groups. Specifically, males seem to have a lower probability of having a postive opinion about Barack Obama than females. However, both genders probabilities are well below fifty percent.

Separate Models for Each Group

Even though the hierarchical model gave us a pretty solid distinction between the Male and Female groups, we can also try this same example as two separate models where we treat each group's data as separate datasets.

```
# try it again, but with separate samples


# data prep
dat.Male<-dat[s==1,]
dat.Female<-dat[s==2,]
y.Male<-dat.Male$y
y.Female<-dat.Female$y

(Ntotal.Male<-length(y.Male))
```

```
## [1] 8796
```

```
(Ntotal.Female<-length(y.Female))
```

```
## [1] 14427
```

```
# MLE means
(MLE.meanMale<-sum(y.Male)/Ntotal.Male)
```

```
## [1] 0.3404957
```

```
(MLE.meanFemale<-sum(y.Female)/Ntotal.Female)
```

```
## [1] 0.3825466
```

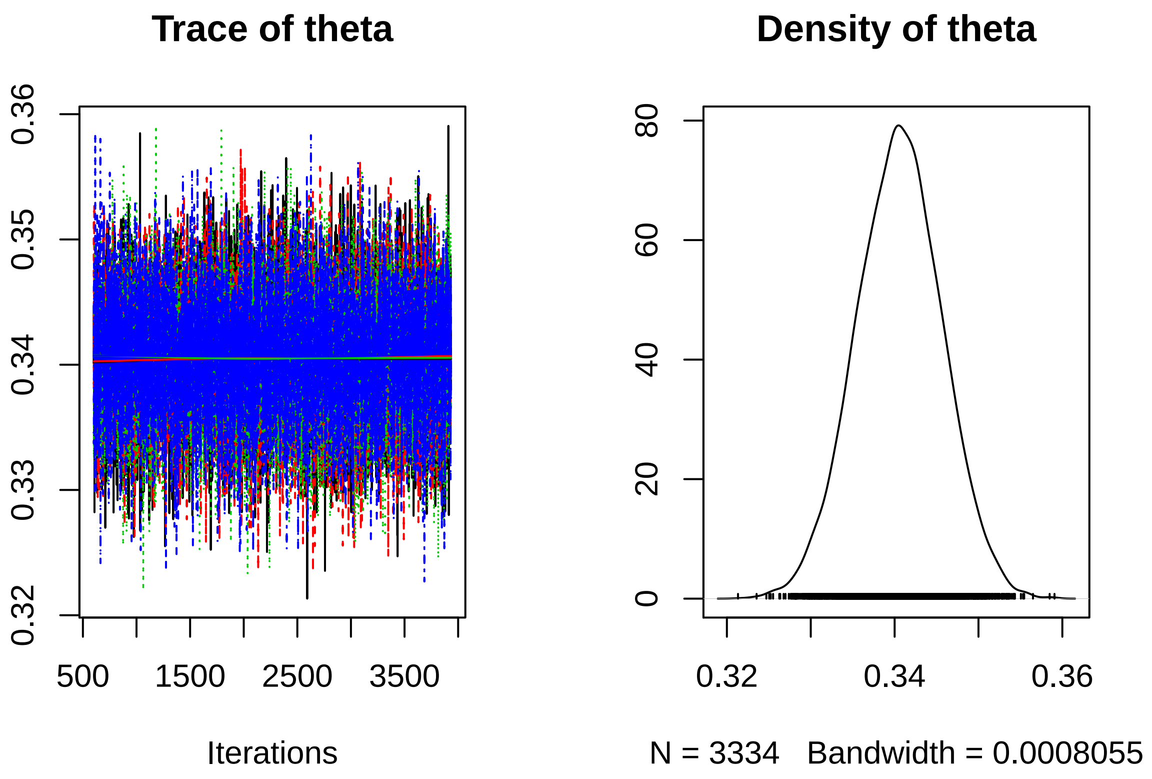
```
# create data lists
dataList.Male<-list(y=y.Male,Ntotal=Ntotal.Male)
dataList.Female<-list(y=y.Female,Ntotal=Ntotal.Female)



# create model strings

# modelString.Male
modelString.Male="
model {
  for (i in 1:Ntotal) {
    y[i]~dbern(theta)
  }
  theta~dbeta(1,1) #Consider different parameterizations of prior
}
"
writeLines( modelString.Male , con="TempModelMale.txt" )

#ModelString.Contr
modelString.Female="
model {
  for (i in 1:Ntotal) {
    y[i]~dbern(theta)
  }
  theta~dbeta(1,1) #Consider different parameterizations of prior
}
"
writeLines( modelString.Female , con="TempModelFemale.txt" )

#Initialize function
initsList = function() {
  thetaInit = rep(0,Nsubj)
  for ( sIdx in 1:Nsubj ) { # for each subject
    includeRows = ( s == sIdx ) # identify rows of this group
    yThisSubj = y[includeRows]  # extract data of this group
    resampledY = sample( yThisSubj , replace=TRUE ) # resample
    thetaInit[sIdx] = sum(resampledY)/length(resampledY)
  }
  thetaInit = 0.001+0.998*thetaInit # keep away from 0,1
  return( list( theta=thetaInit ) )
}



### Run MCMC for the Male group

#Create model Male
jagsModel.Male<-jags.model(file="TempModelMale.txt",data=dataList.Male,n.chains=4,n.adapt=500)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 8796
##    Unobserved stochastic nodes: 1
##    Total graph size: 8800
##
## Initializing model
```

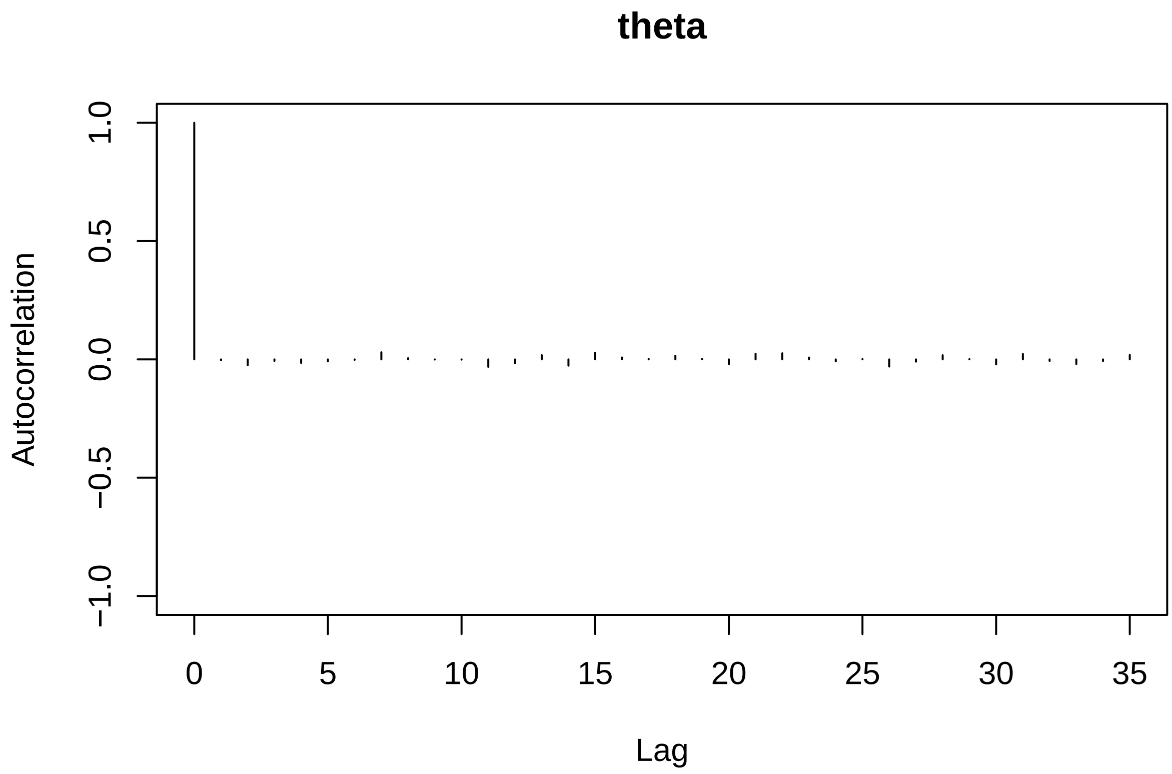
```
# Update Male
update(jagsModel.Male,n.iter=600)

# Run Male
codaSamples.Male<-coda.samples(jagsModel.Male,variable.names=c("theta"),n.iter=3334)
list.samplers(jagsModel.Male)
```

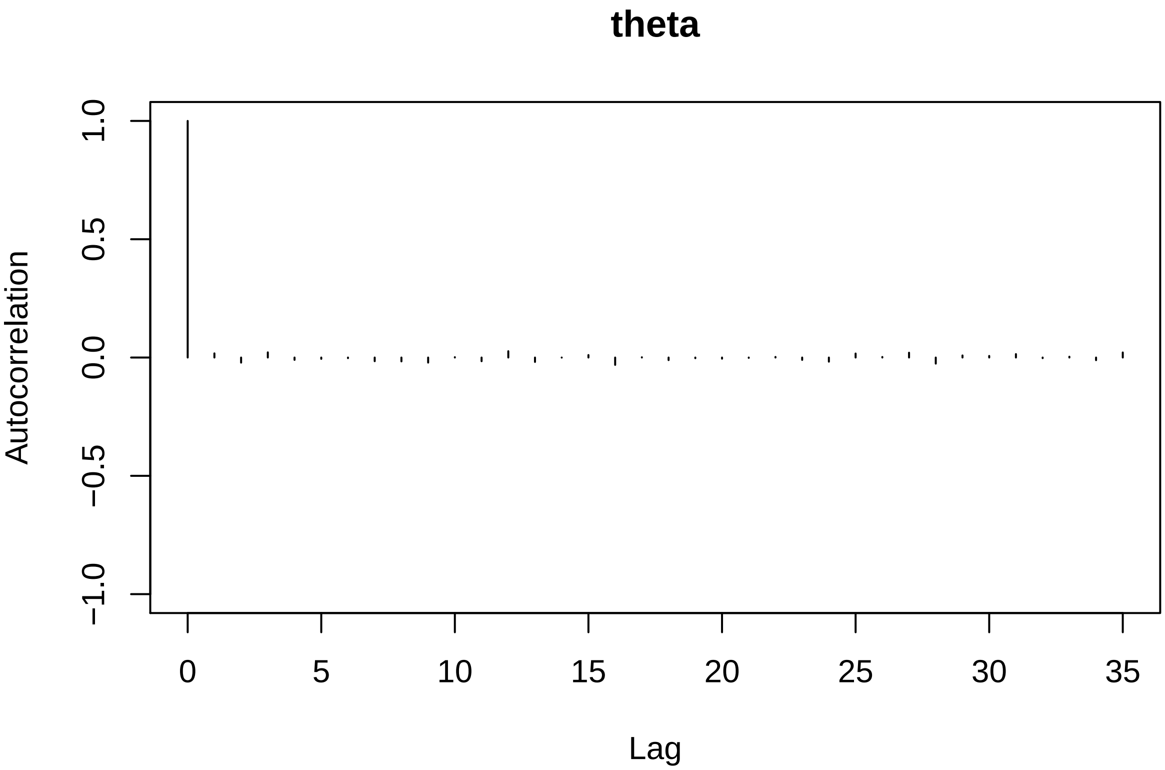
```
## $`bugs::ConjugateBeta`
## [1] "theta"
```

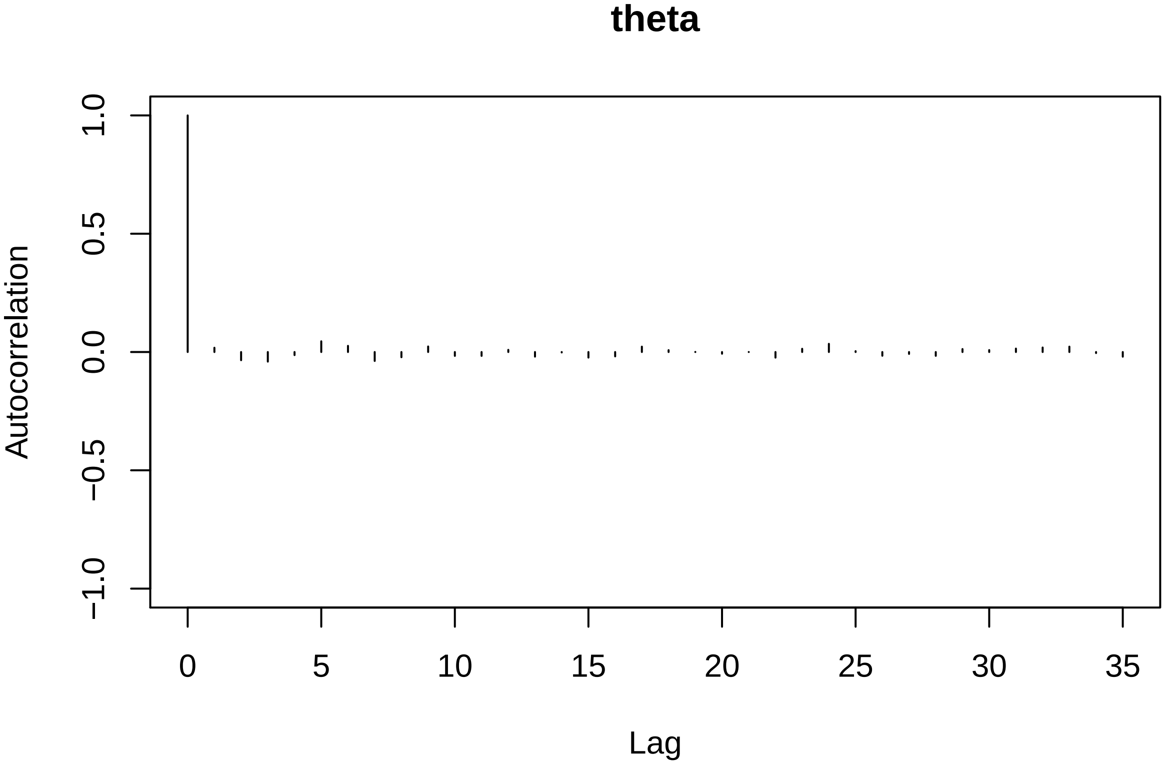
```
# analyze Male
plot(codaSamples.Male)
```

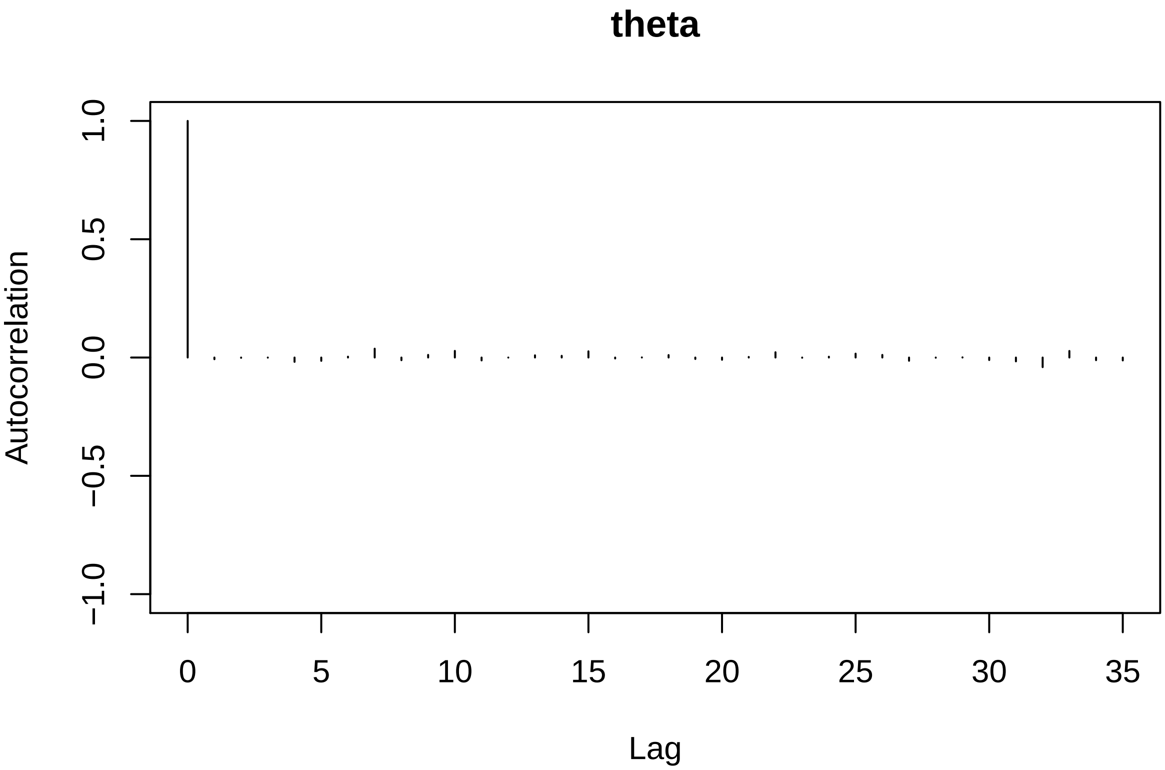


```r
autocorr.plot(codaSamples.Male,ask=F)
```




theta
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
Lag


theta
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
Lag



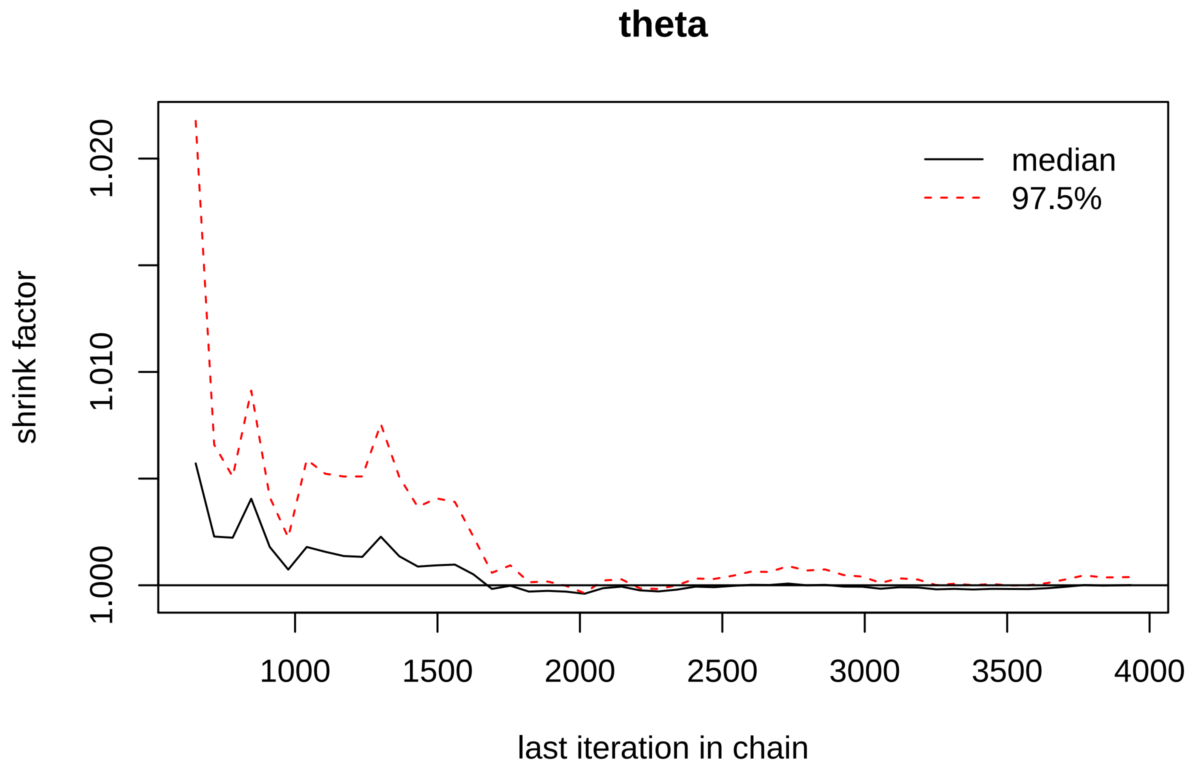
```
effectiveSize(codaSamples.Male)
```

```
##    theta
## 13611.02
```

```
gelman.diag(codaSamples.Male)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## theta          1          1
```

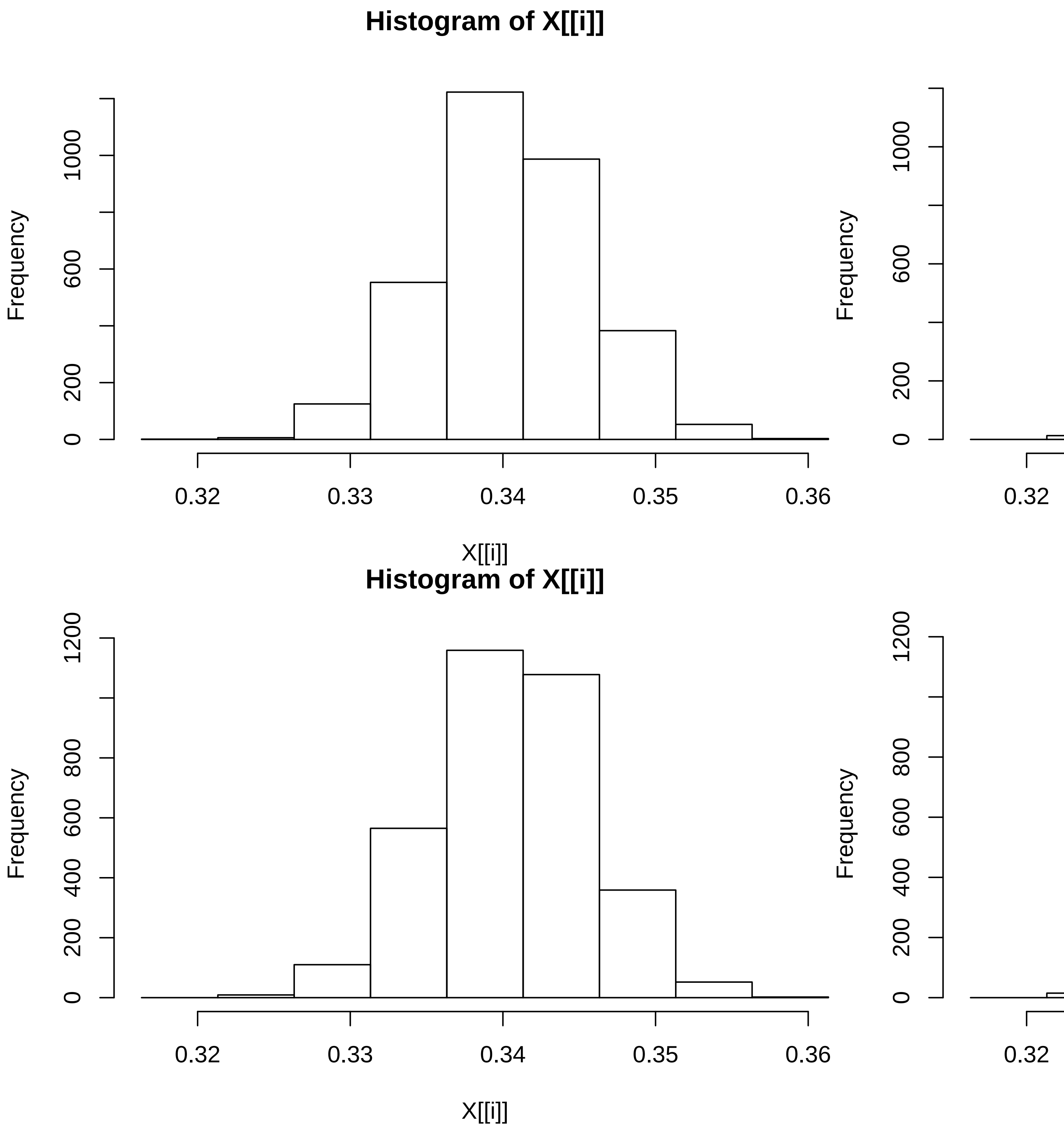
```
gelman.plot(codaSamples.Male)
```



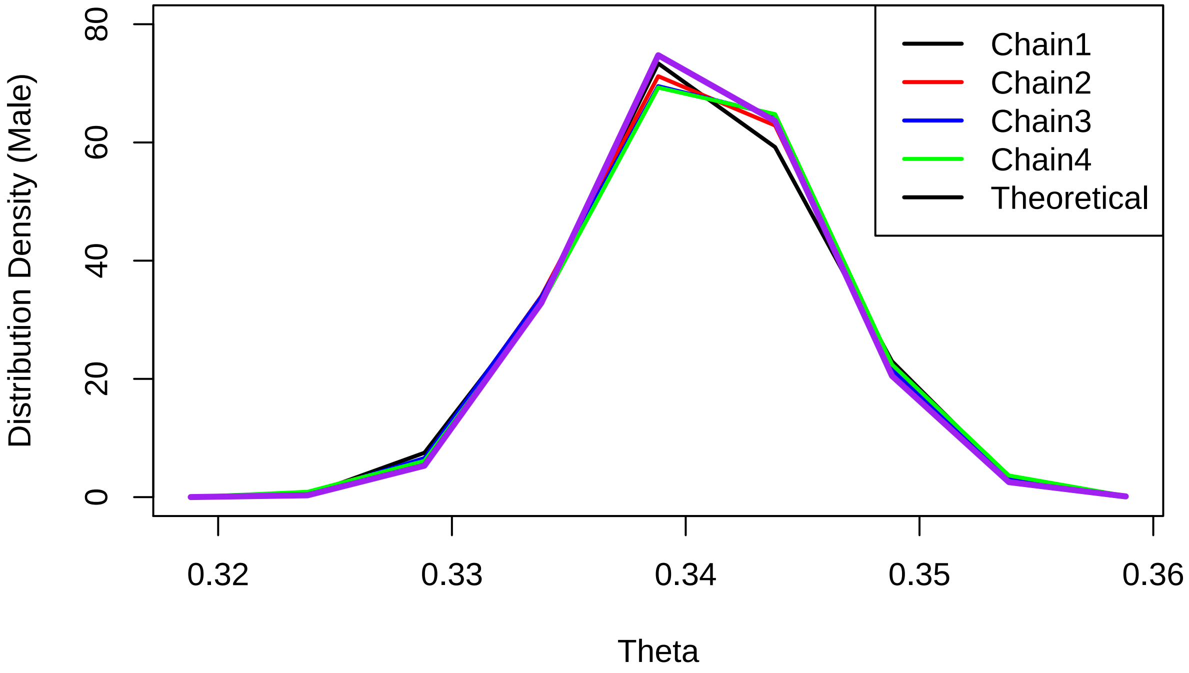
```r
lapply(codaSamples.Male,mean)
```

```
## [[1]]
## [1] 0.3405173
##
## [[2]]
## [1] 0.3405159
##
## [[3]]
## [1] 0.3405245
##
## [[4]]
## [1] 0.3406673
```

```r
l<-min(unlist(codaSamples.Male))-.005
h<-max(unlist(codaSamples.Male))+.005
histBreaks<-seq(l,h,by=.005)
postHist.Male<-lapply(codaSamples.Male,hist,breaks=histBreaks)
```



```
plot(postHist.Male[[1]]$mids,postHist.Male[[1]]$density,type="l",col="black",lwd=2,ylab="Distribution De
lines(postHist.Male[[2]]$mids,postHist.Male[[2]]$density,type="l",col="red",lwd=2)
lines(postHist.Male[[3]]$mids,postHist.Male[[3]]$density,type="l",col="blue",lwd=2)
lines(postHist.Male[[4]]$mids,postHist.Male[[4]]$density,type="l",col="green",lwd=2)
lines(postHist.Male[[3]]$mids,dbeta(postHist.Male[[3]]$mids,1+sum(y.Male),Ntotal.Male-sum(y.Male)+1),ty
legend("topright",legend=c("Chain1","Chain2","Chain3","Chain4","Theoretical"),col=c("black","red","blue
```



```
(HDIofChains.Contr<-lapply(codaSamples.Male,function(z)
  Theta.Contr=HDIofMCMC(z[,1])))
```

```
## [[1]]
## [1] 0.3307718 0.3508809
##
## [[2]]
## [1] 0.3307916 0.3506921
##
## [[3]]
## [1] 0.3302398 0.3496254
##
## [[4]]
## [1] 0.3306799 0.3506580
```

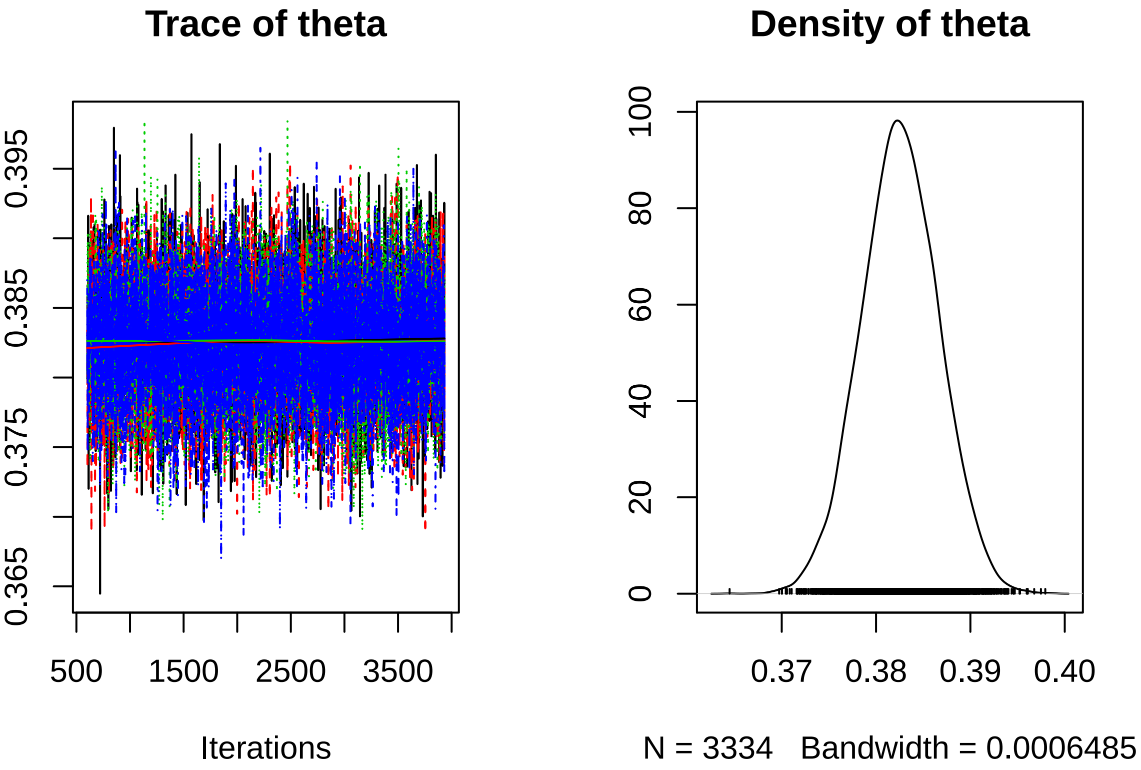
```
### Run MCMC for the Female group

#Create model FeFemale
jagsModel.Female<-jags.model(file="TempModelFemale.txt",data=dataList.Female,n.chains=4,n.adapt=500)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 14427
##    Unobserved stochastic nodes: 1
##    Total graph size: 14431
##
## Initializing model
```

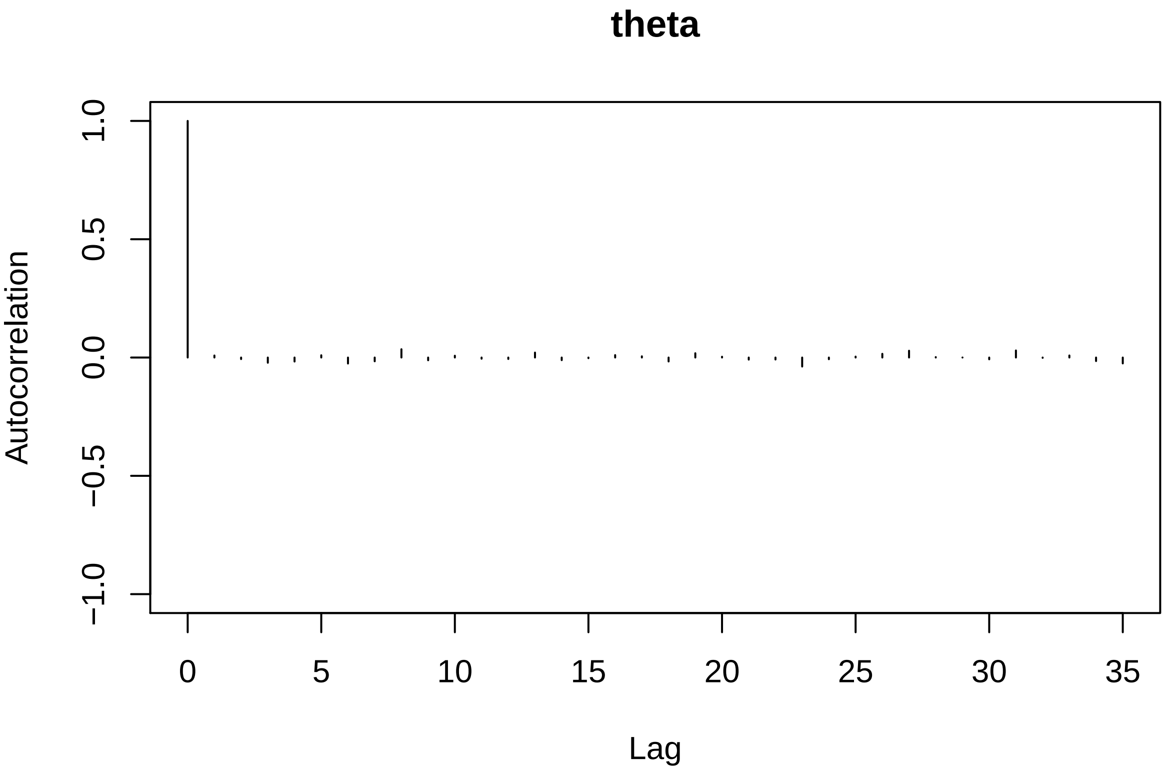
```
# Update Female
update(jagsModel.Female,n.iter=600)

# Run Female
codaSamples.Female<-coda.samples(jagsModel.Female,variable.names=c("theta"),n.iter=3334)
list.samplers(jagsModel.Female)
```

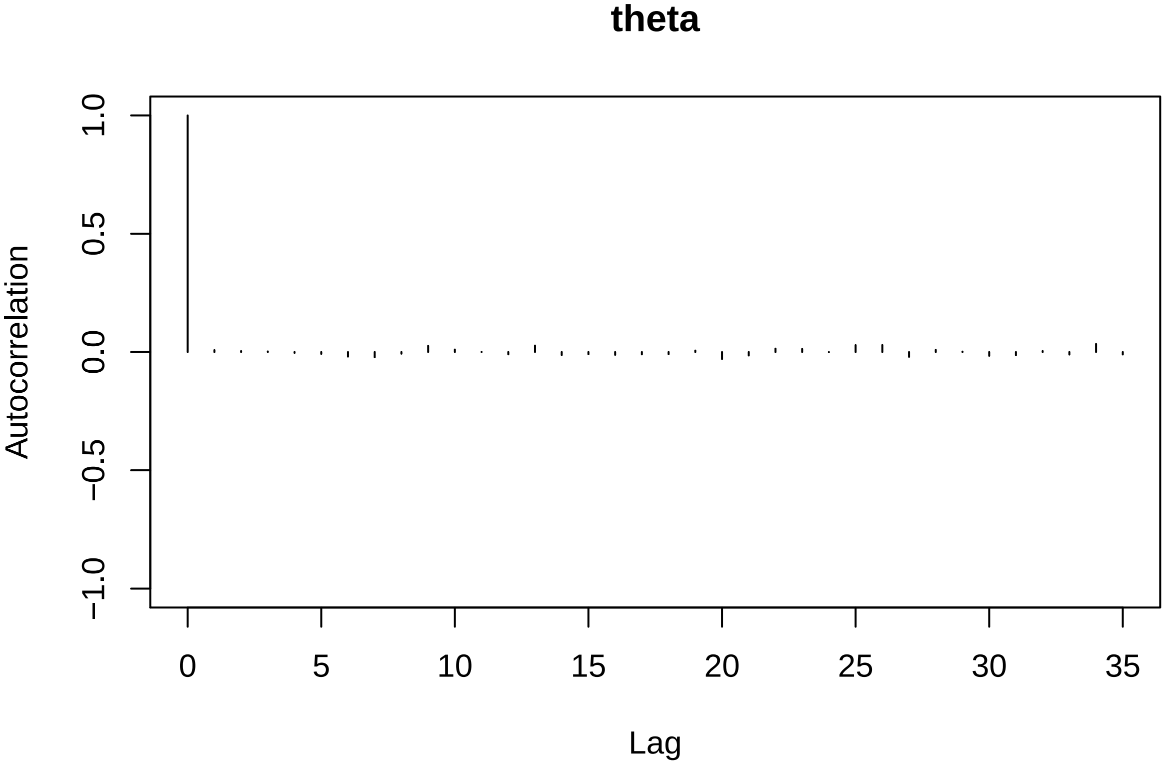
```
## $`bugs::ConjugateBeta`
## [1] "theta"
```

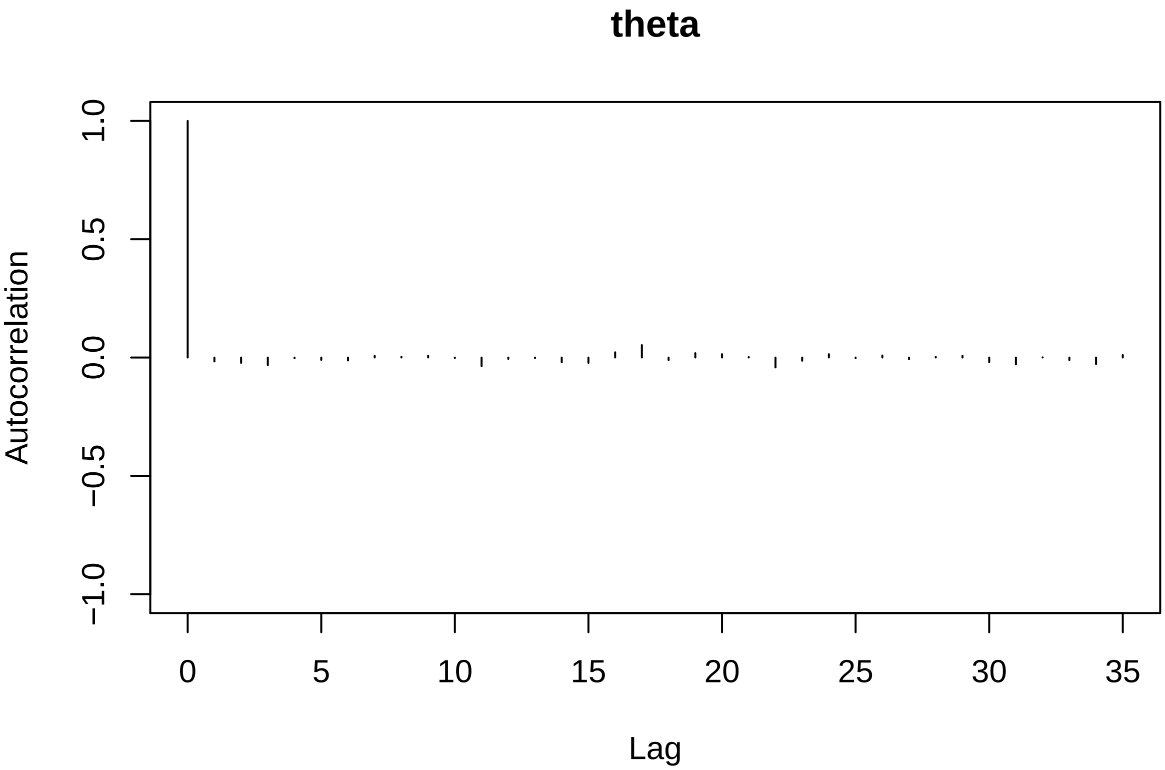
```
# analyze Female
plot(codaSamples.Female)
```

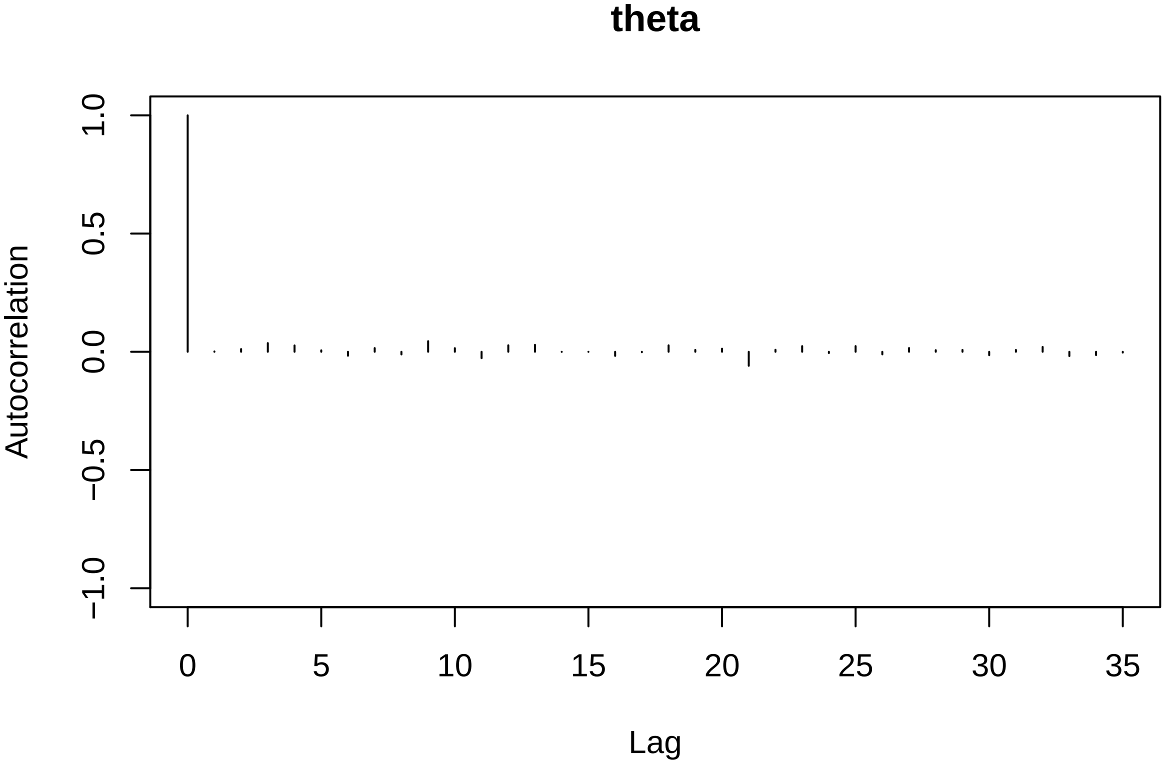


```
autocorr.plot(codaSamples.Female,ask=F)
```


theta
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
Lag


theta
Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
30
35
Lag





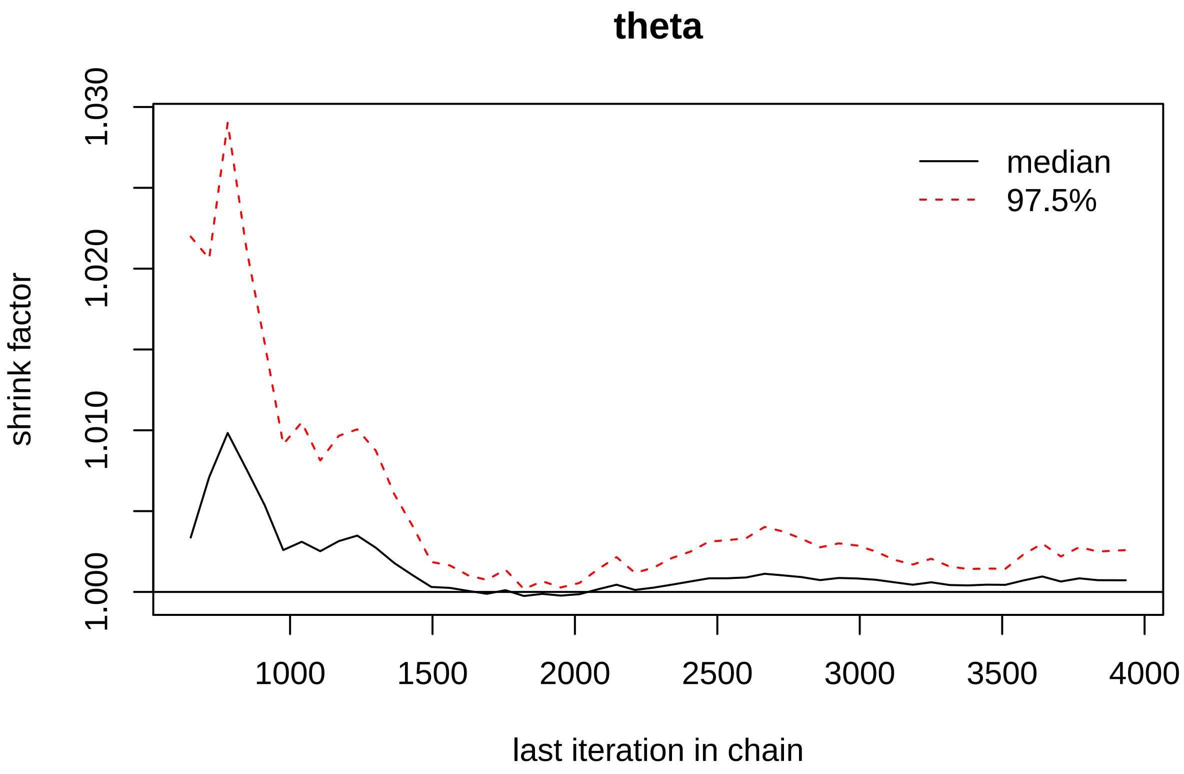
```
effectiveSize(codaSamples.Female)
```

```
##    theta
## 13164.51
```

```
gelman.diag(codaSamples.Female)
```

```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## theta          1          1
```

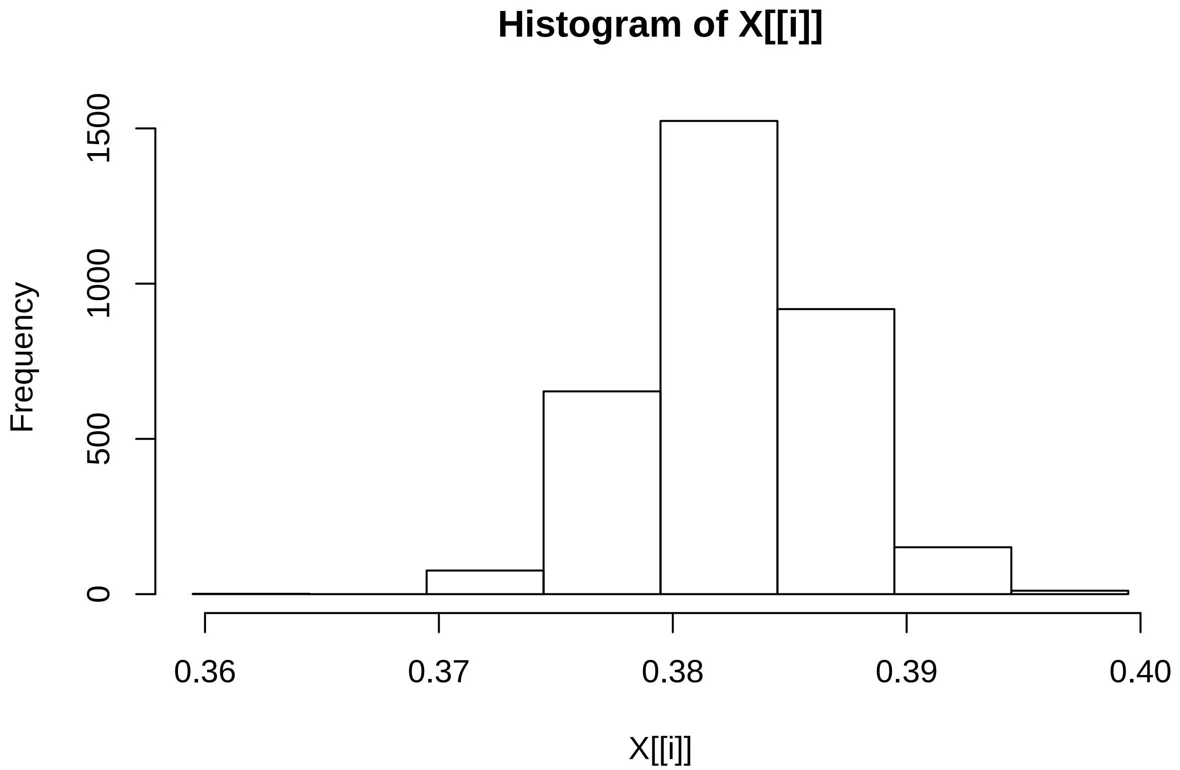
```
gelman.plot(codaSamples.Female)
```

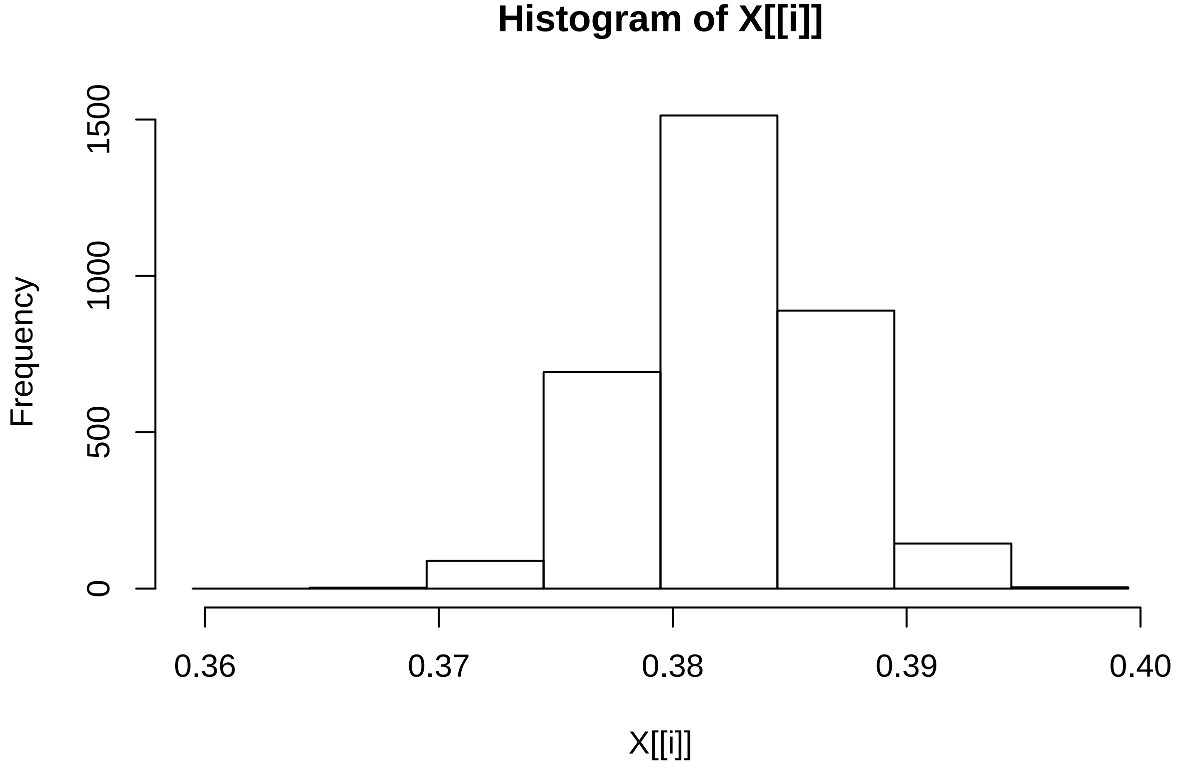


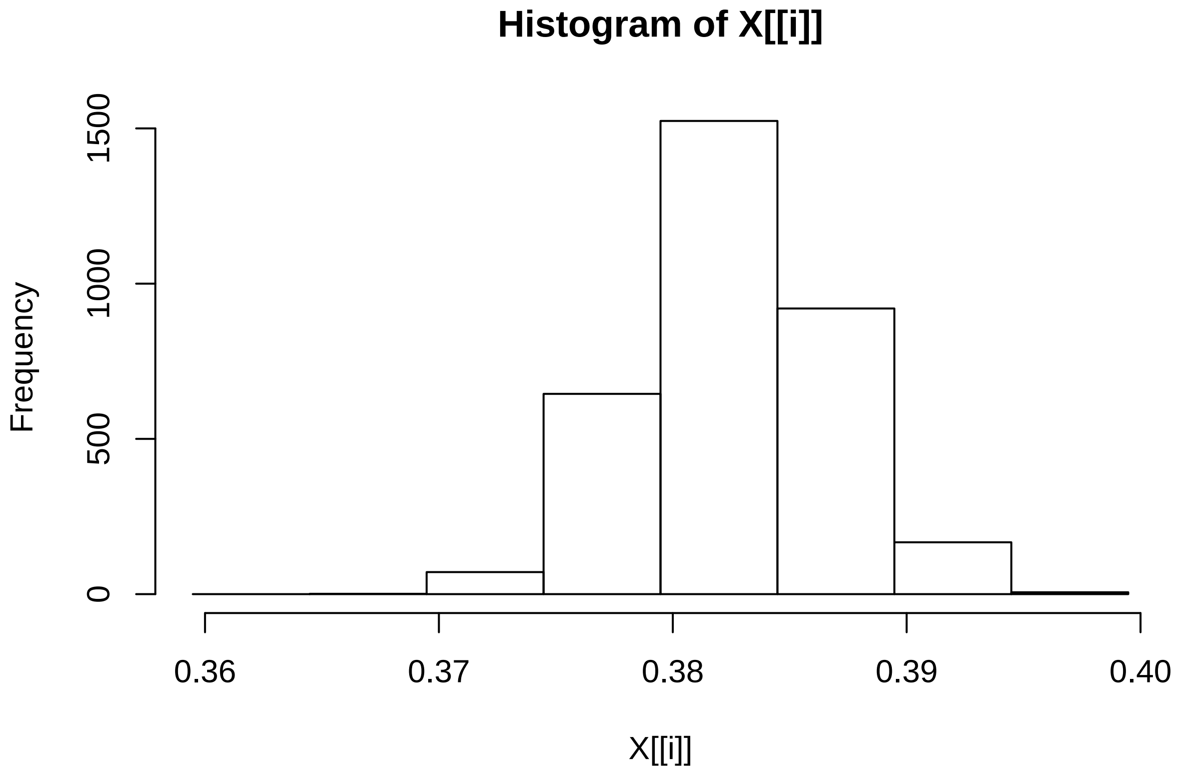
```
lapply(codaSamples.Female,mean)
```

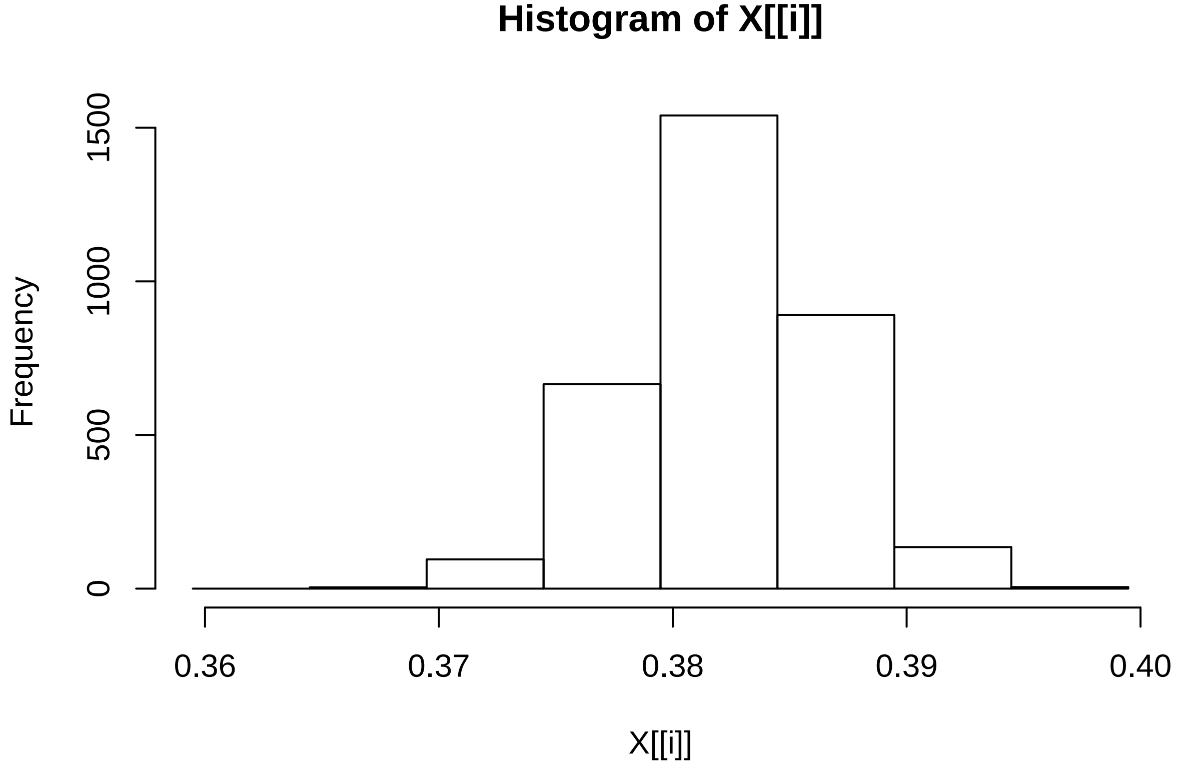
```
## [[1]]
## [1] 0.382648
##
## [[2]]
## [1] 0.3824893
##
## [[3]]
## [1] 0.3826521
##
## [[4]]
## [1] 0.3824415
```

```
l<-min(unlist(codaSamples.Female))-.005
h<-max(unlist(codaSamples.Female))+.005
histBreaks<-seq(l,h,by=.005)
postHist.Female<-lapply(codaSamples.Female,hist,breaks=histBreaks)
```

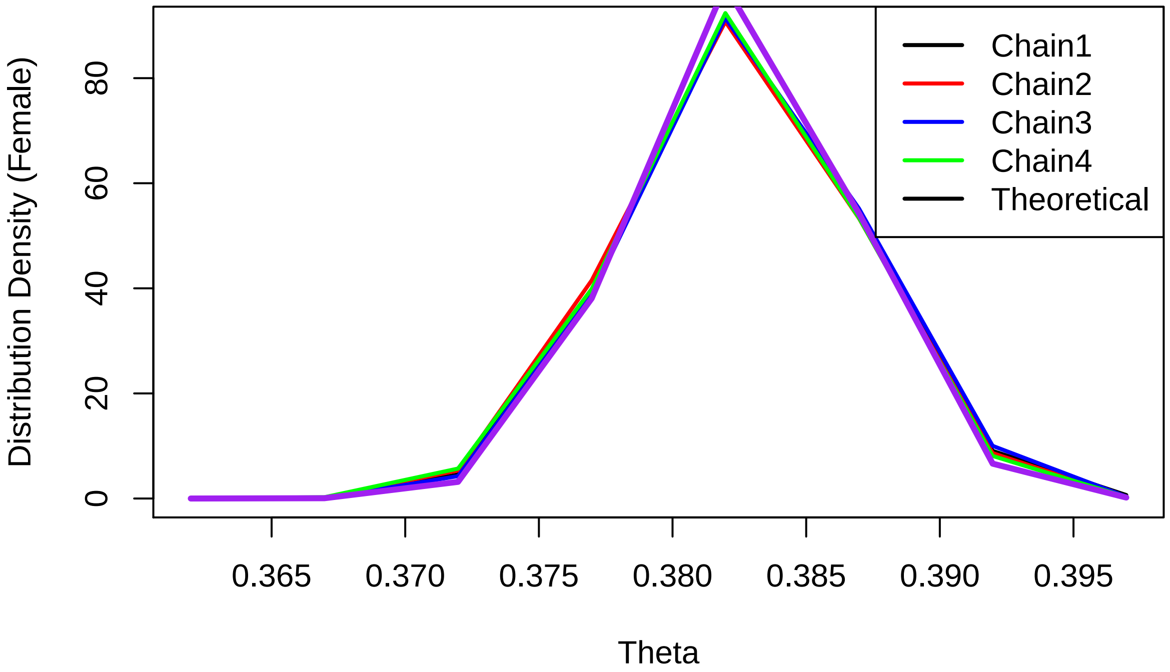

Histogram of X[[i]]
Frequency
0
500
1000
1500
0.36
0.37
0.38
0.39
0.40
X[[i]]


Histogram of X[[i]]
Frequency
0
500
1000
1500
0.36
0.37
0.38
0.39
0.40
X[[i]]





```
plot(postHist.Female[[1]]$mids,postHist.Female[[1]]$density,type="l",col="black",lwd=2,ylab="Distributi
lines(postHist.Female[[2]]$mids,postHist.Female[[2]]$density,type="l",col="red",lwd=2)
lines(postHist.Female[[3]]$mids,postHist.Female[[3]]$density,type="l",col="blue",lwd=2)
lines(postHist.Female[[4]]$mids,postHist.Female[[4]]$density,type="l",col="green",lwd=2)
lines(postHist.Female[[4]]$mids,dbeta(postHist.Female[[4]]$mids,1+sum(y.Female),Ntotal.Female-sum(y.Fem
legend("topright",legend=c("Chain1","Chain2","Chain3","Chain4","Theoretical"),col=c("black","red","blue
```



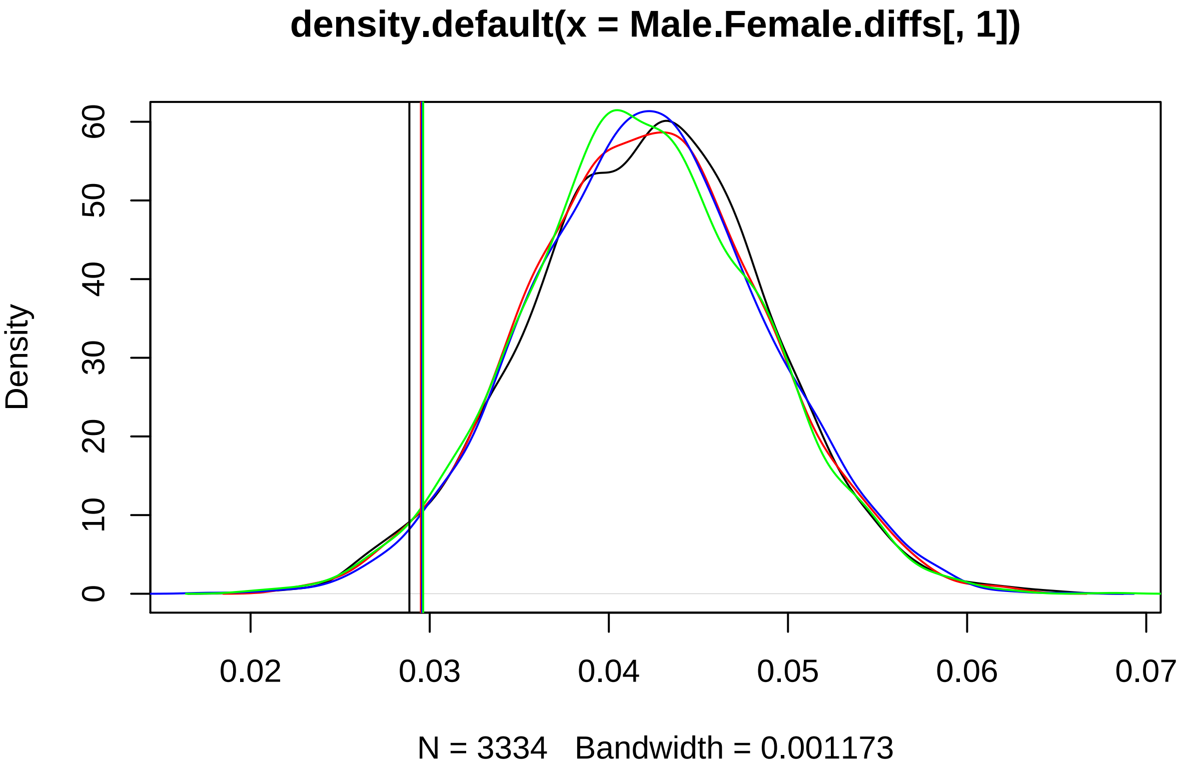
```
(HDIofChains.Contr<-lapply(codaSamples.Female,function(z)
  Theta.Contr=HDIofMCMC(z[,1])))
```

```
## [[1]]
## [1] 0.3743895 0.3906134
##
## [[2]]
## [1] 0.3747732 0.3906804
##
## [[3]]
## [1] 0.3752612 0.3912258
##
## [[4]]
## [1] 0.3747985 0.3908319
```

```
#Find differences between Male and Female groups and their HDIs
Male.Female.diffs<-sapply(1:4,function(z) codaSamples.Female[[z]]-codaSamples.Male[[z]])
(HDI.diffs<-apply(Male.Female.diffs,2,function(z) HDIofMCMC(z,.95)))
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] 0.02886465 0.02950897 0.02957676 0.02963599
## [2,] 0.05480324 0.05529563 0.05514638 0.05524510
```

```
#Plot densities and the 95% HDI limits
plot(density(Male.Female.diffs[,1]))#,xlim=c(-.5,1),ylim=c(0,3),col="black")
lines(density(Male.Female.diffs[,2]),col="red")
lines(density(Male.Female.diffs[,3]),col="blue")
lines(density(Male.Female.diffs[,4]),col="green")
abline(v=HDI.diffs[1,],col=c("black","red","blue","green"))
```



```r
#Compare left bounds of hierarchical model and separate models
rbind(Hierarchical=leftBounds,Separate=HDI.diffs[1,])
```

```
##                    [,1]       [,2]       [,3]       [,4]
## Hierarchical 0.02763485 0.02727219 0.02783894 0.02758252
## Separate     0.02886465 0.02950897 0.02957676 0.02963599
```

Just like in the first hierarchical model, we see the same distinction between groups. Males have a lower probability by a few percentage points than females to have a positive opinion about Barack Obama.