Homework 7

[redacted]

```
setwd("/Users/[redacted]Bayesian Data Analysis/HW")
source("/Users/[redacted]Bayesian Data Analysis/DBDA2Eprograms/BernGrid.R")
source("/Users/[redacted]Bayesian Data Analysis/DBDA2Eprograms/DBDA2E-utilities.R")
```

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

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
[redacted]/Bayesian Data Analysis/DBDA2Eprograms/BernBeta.R")
source("/Users/[redacted]Bayesian Data Analysis/DBDA2Eprograms/HDIofICDF.R")
```

Ex8.1

```
dat = data.frame(y =  c(rep(1,45),rep(0,15) ,  rep(1,20),rep(0,10) , rep(1,3),rep(0,12)  ),
                 s = c( rep("A",60)         ,  rep("B",30)       ,rep("C",15)        ))
head(dat)
```

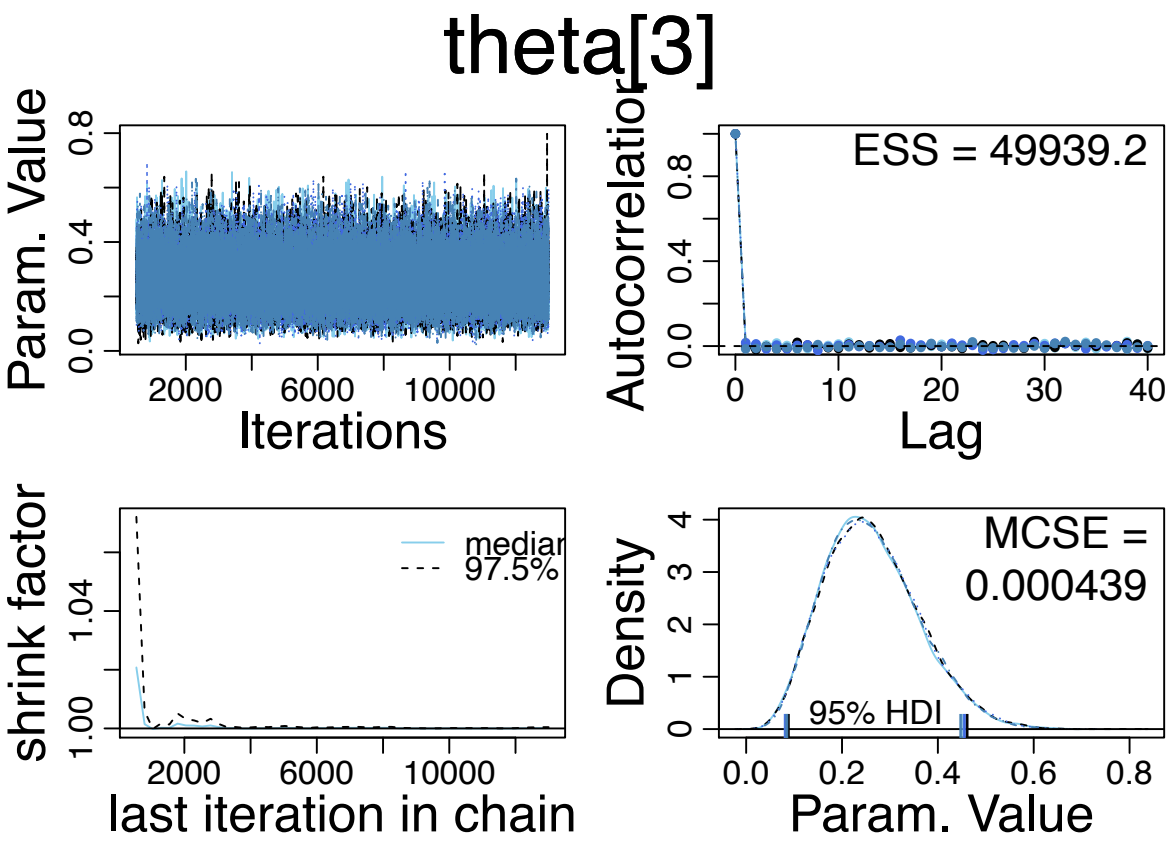
```
##   y s
## 1 1 A
## 2 1 A
## 3 1 A
## 4 1 A
## 5 1 A
## 6 1 A
```

```
write.csv(dat,"HW7.ex1.csv")
#
graphics.off()
# Load The data
myData = read.csv("HW7.ex1.csv")
# N.B.: The functions below expect the data to be a data frame,
# with one component named y being a vector of integer 0,1 values,
# and one component named s being a factor of subject identifiers.
#------------------------------------------------------------------------------
# Load the relevant model into R's working memory:
source("Jags-Ydich-XnomSsubj-MbernBeta.R")
#------------------------------------------------------------------------------
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments
# out of function calls.
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "pdf"
#------------------------------------------------------------------------------
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
```

```
## Graph information:
##    Observed stochastic nodes: 105
##    Unobserved stochastic nodes: 3
##    Total graph size: 216
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

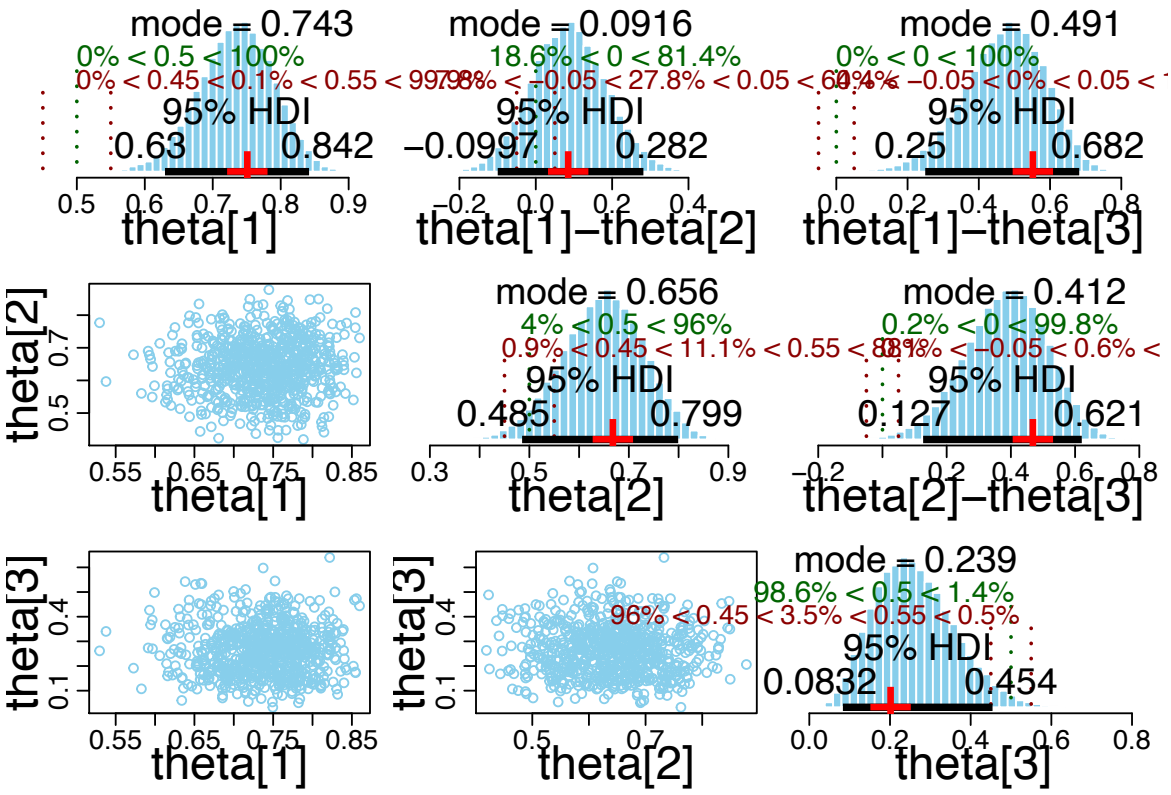
```
#-------------------------------------------------------------------------------
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
                saveName=fileNameRoot , saveType=graphFileType )
}
#-------------------------------------------------------------------------------
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=.5 , rope=c(0.45,0.55) ,
                        compValDiff=0.0 , ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```




```
##                          Mean     Median       Mode     ESS HDImass
## theta[1]           0.73419342 0.73643508 0.74329022 50000.0    0.95
## theta[2]           0.64730335 0.65038545 0.65647226 50000.0    0.95
## theta[3]           0.26250689 0.25417704 0.23906564 50000.0    0.95
## theta[1]-theta[2]  0.08689007 0.08610675 0.09162443 50782.9    0.95
## theta[1]-theta[3]  0.47168653 0.47875641 0.49118000 50000.0    0.95
## theta[2]-theta[3]  0.38479646 0.39052786 0.41181094 50000.0    0.95
##                        HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]           0.62972350 0.8417499     0.5        99.992    0.45
## theta[2]           0.48536412 0.7991540     0.5        95.984    0.45
## theta[3]           0.08322585 0.4541796     0.5         1.438    0.45
## theta[1]-theta[2] -0.09966428 0.2819209     0.0        81.356   -0.05
```

```
## theta[1]-theta[3]  0.24955078 0.6819047     0.0        99.994   -0.05
## theta[2]-theta[3]  0.12672140 0.6208635     0.0        99.758   -0.05
##                    ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]               0.55      0.000      0.134     99.866
## theta[2]               0.55      0.944     11.100     87.956
## theta[3]               0.55     96.032      3.514      0.454
## theta[1]-theta[2]      0.05      7.810     27.806     64.384
## theta[1]-theta[3]      0.05      0.002      0.026     99.972
## theta[2]-theta[3]      0.05      0.076      0.642     99.282
```

```
# Display posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=.5 , rope=c(0.45,0.55) ,
          compValDiff=0.0 , ropeDiff = c(-0.05,0.05) ,
          saveName=fileNameRoot , saveType=graphFileType )
#-------------------------------------------------------------------------------
```



As shown in this posterior distribution graph, the HDI for theta 1 is narrower than that for theta 2 and the mode for theta 1 is larger than that for theta 2; this is consistent with the predefined data (more data points for theta 1 and more heads, i.e.,1s). the HDI for theta 3 is much wider and the mode is towards 0 consistent with the data (less data poitns for theta 3 and lots tails 0s)

Ex8.2

The output is shown below, containing the results for the parameters and the paremeter differences (which are in each row). The first three columns, mean, median, mode, are the mean, median, mode of the parameter in the MCMC chain. ESS = effective sample size (chain length/autocorrelation). HDImass = probability mass of HDI. HDIlow and HDIhigh are the lower and upper bounds of the HDI. CompVal = comparison value (predifined as .5). PcntGtCompVal = percentage of posterior larger than .5 comparison value. ROPElow and ROPEhigh are the lower and upper bounds of the ROPE, which are predefined. PcntLtROPE, PcntInROPE, PcntGtROPE are percentages of the posterior less than ROPE lower bound, within the ROPE, and greater than the ROPE upper bound

```
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=.5 , rope=c(0.45,0.55) ,
                        compValDiff=0.0 , ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

```
##                        Mean      Median        Mode     ESS HDImass
## theta[1]          0.73419342 0.73643508 0.74329022 50000.0     0.95
```

```
## theta[2]          0.64730335 0.65038545 0.65647226 50000.0    0.95
## theta[3]          0.26250689 0.25417704 0.23906564 50000.0    0.95
## theta[1]-theta[2] 0.08689007 0.08610675 0.09162443 50782.9    0.95
## theta[1]-theta[3] 0.47168653 0.47875641 0.49118000 50000.0    0.95
## theta[2]-theta[3] 0.38479646 0.39052786 0.41181094 50000.0    0.95
##                        HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]           0.62972350 0.8417499     0.5        99.992    0.45
## theta[2]           0.48536412 0.7991540     0.5        95.984    0.45
## theta[3]           0.08322585 0.4541796     0.5         1.438    0.45
## theta[1]-theta[2] -0.09966428 0.2819209     0.0        81.356   -0.05
## theta[1]-theta[3]  0.24955078 0.6819047     0.0        99.994   -0.05
## theta[2]-theta[3]  0.12672140 0.6208635     0.0        99.758   -0.05
##                   ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]              0.55      0.000      0.134     99.866
## theta[2]              0.55      0.944     11.100     87.956
## theta[3]              0.55     96.032      3.514      0.454
## theta[1]-theta[2]     0.05      7.810     27.806     64.384
## theta[1]-theta[3]     0.05      0.002      0.026     99.972
## theta[2]-theta[3]     0.05      0.076      0.642     99.282
```

Ex8.3

the following line is responsible for saving the graphs. The first line defines the beginning of the name of the graph (since there will be a few graphs saved). The second line defines the format of the graph. Here I used pdf.

```
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments
# out of function calls.
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "pdf"
```

The summary statistics of chain is also saved. the name is prespecified. We want this saved because we want to maybe present our results to other people.

```
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=.5 , rope=c(0.45,0.55) ,
                        compValDiff=0.0 , ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

In the file Jags-Ydich-XnomSsubj-Mbernbeta.R, a few other files are saved. Here, the MCMC chain is saved in the working directory and the file name has the "Jags-Ydich-XnomSsubj-MbernBeta-" prefix. We want to save this in case we want to reproduce the results later.

```
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )
```

Diagnostics graphs are also saved with the following lines.

```
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
                saveName=fileNameRoot , saveType=graphFileType )
}
```

Ex8.4

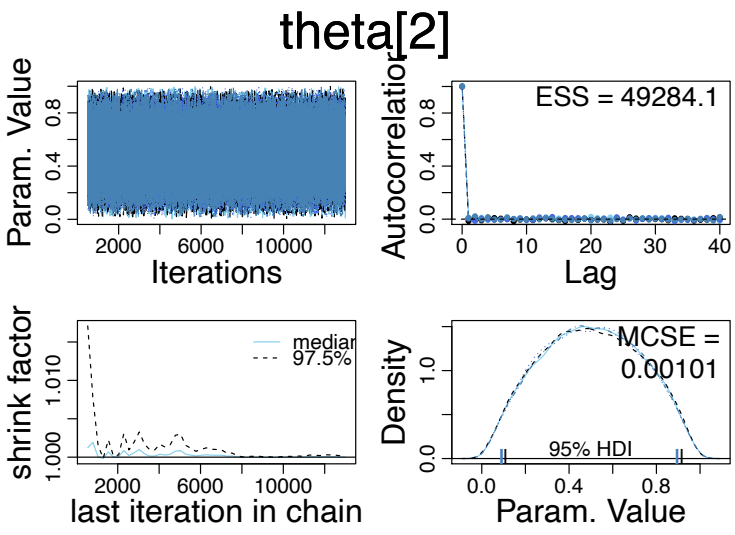
A)

I redefied the dataList without y = y in Jags-Ydich-XnomSsubj-MbernBeta.R and saved the file and reran the example script.

```r
# Load The data
myData = read.csv("/Users/yuejiateng/Dropbox/USF Grad/2018 Fall/Bayesian Data Analysis/DBDA2Eprograms/z6N8z2N7.
# N.B.: The functions below expect the data to be a data frame,
# with one component named y being a vector of integer 0,1 values,
# and one component named s being a factor of subject identifiers.
#------------------------------------------------------------------------------
# Load the relevant model into R's working memory:
source("Jags-Ydich-XnomSsubj-MbernBeta81.R")
#------------------------------------------------------------------------------
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments
# out of function calls.
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
#------------------------------------------------------------------------------
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 17
##    Total graph size: 35
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

```r
#------------------------------------------------------------------------------
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
            saveName=fileNameRoot , saveType=graphFileType )
}
#------------------------------------------------------------------------------
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=NULL , #rope=c(0.45,0.55) ,
                        compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

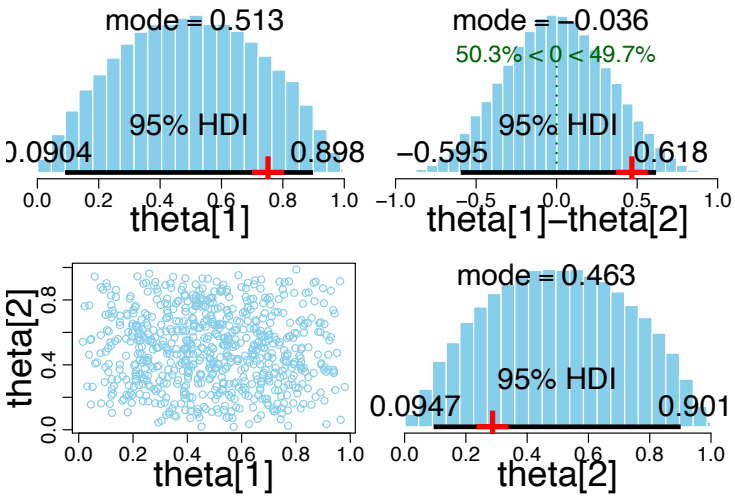


```
##                            Mean       Median        Mode     ESS HDImass
## theta[1]           0.5005786416  0.500466163  0.51308203 50000.0    0.95
## theta[2]           0.5014090339  0.500440308  0.46312089 50000.0    0.95
## theta[1]-theta[2] -0.0008303922 -0.002042092 -0.03603996 50997.7    0.95
##                         HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]           0.09041324 0.8982945      NA            NA      NA
## theta[2]           0.09470099 0.9009671      NA            NA      NA
## theta[1]-theta[2] -0.59541681 0.6177066       0        49.732      NA
##                   ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                NA         NA         NA         NA
## theta[2]                NA         NA         NA         NA
## theta[1]-theta[2]       NA         NA         NA         NA
```

```
# Display posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=NULL , #rope=c(0.45,0.55) ,
          compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
          saveName=fileNameRoot , saveType=graphFileType )
```

```
## pdf
##   2
```

```
#------------------------------------------------------------------------------
```



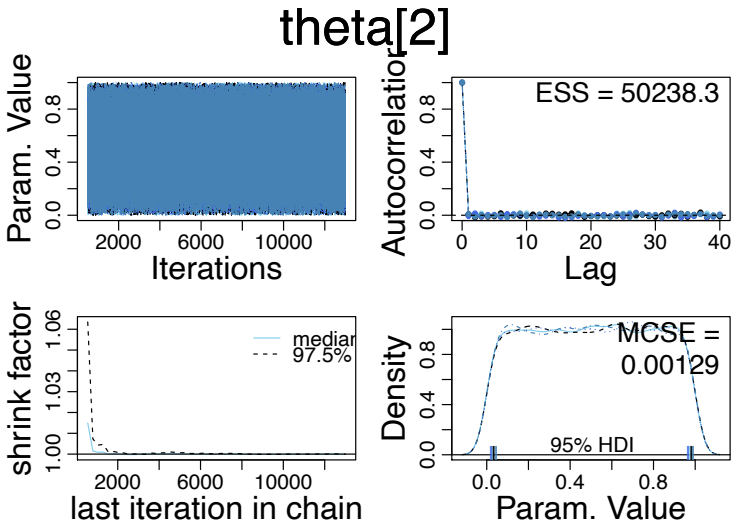
B)

I redefied the modelString and theta follows dbeta(1,1) distribution in Jags-Ydich-XnomSsubj-MbernBeta.R and saved the file and reran the example script. Theta1 and theta2 are uniform as I specified. The difference distribution is triagular as the theta1 by theta2 plot is evenly distributed.

```
# Load The data
myData = read.csv("/Users/yuejiateng/Dropbox/USF Grad/2018 Fall/Bayesian Data Analysis/DBDA2Eprograms/z6N8z2N7.
# N.B.: The functions below expect the data to be a data frame,
# with one component named y being a vector of integer 0,1 values,
# and one component named s being a factor of subject identifiers.
#------------------------------------------------------------------------------
# Load the relevant model into R's working memory:
source("Jags-Ydich-XnomSsubj-MbernBeta82.R")
#------------------------------------------------------------------------------
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments
# out of function calls.
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
#------------------------------------------------------------------------------
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 17
##    Total graph size: 35
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

```
#------------------------------------------------------------------------------
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
            saveName=fileNameRoot , saveType=graphFileType )
}
#------------------------------------------------------------------------------
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=NULL , #rope=c(0.45,0.55) ,
                        compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

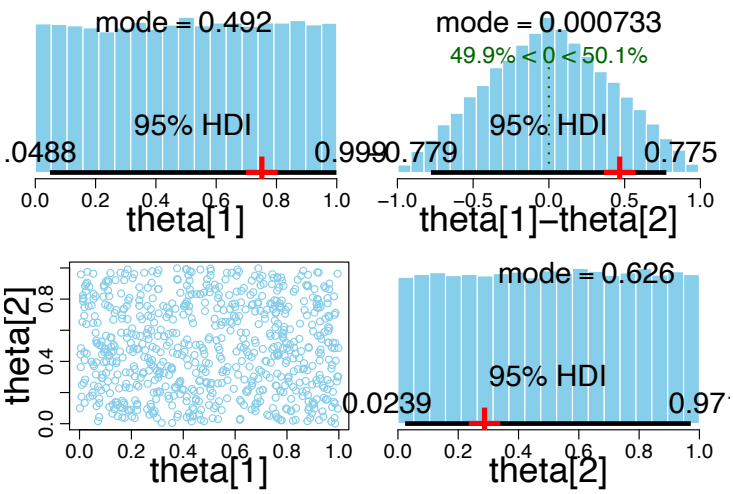



```
##                          Mean      Median          Mode     ESS HDImass
## theta[1]           0.5025046073 0.504196925 0.4919151328 50000.0    0.95
## theta[2]           0.5017812928 0.503514644 0.6257007538 50000.0    0.95
## theta[1]-theta[2] 0.0007233145 0.001423465 0.0007331368 49262.8    0.95
##                        HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]            0.04881875 0.9989146      NA            NA      NA
## theta[2]            0.02386608 0.9712860      NA            NA      NA
## theta[1]-theta[2] -0.77861614 0.7752005       0        50.134      NA
##                   ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                NA         NA         NA         NA
## theta[2]                NA         NA         NA         NA
## theta[1]-theta[2]       NA         NA         NA         NA
```

```
# Display posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=NULL , #rope=c(0.45,0.55) ,
          compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
          saveName=fileNameRoot , saveType=graphFileType )
```

```
## pdf
##   2
```

```
#------------------------------------------------------------------------------
```



C)

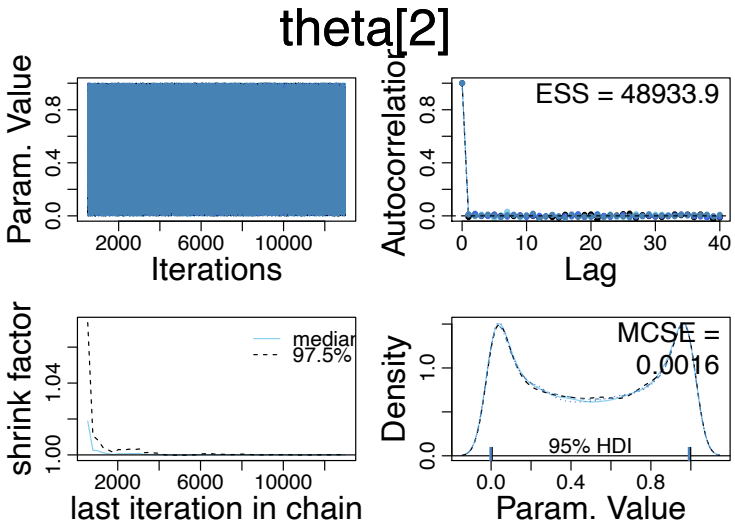
I redefied the modelString and theta follows dbeta(.5,.5) distribution in Jags-Ydich-XnomSsubj-MbernBeta.R and saved the file and reran the example script. Theta1 and theta2 have similar shapes and higher probability mass towards the extremes. The difference distribution has a extreme distribution as the theta1 by theta2 plot is distributed towards the cornors.

```
# Load The data
myData = read.csv("/Users/yuejiateng/Dropbox/USF Grad/2018 Fall/Bayesian Data Analysis/DBDA2Eprograms/z6N8z2N7.
# N.B.: The functions below expect the data to be a data frame,
# with one component named y being a vector of integer 0,1 values,
# and one component named s being a factor of subject identifiers.
#------------------------------------------------------------------------------
# Load the relevant model into R's working memory:
source("Jags-Ydich-XnomSsubj-MbernBeta83.R")
#------------------------------------------------------------------------------
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments
# out of function calls.
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
#------------------------------------------------------------------------------
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 0
##    Unobserved stochastic nodes: 17
##    Total graph size: 35
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...
```

```
#------------------------------------------------------------------------------
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
```

```
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
                saveName=fileNameRoot , saveType=graphFileType )
}
#------------------------------------------------------------------------------
# Get summary statistics of chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=NULL , #rope=c(0.45,0.55) ,
                        compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

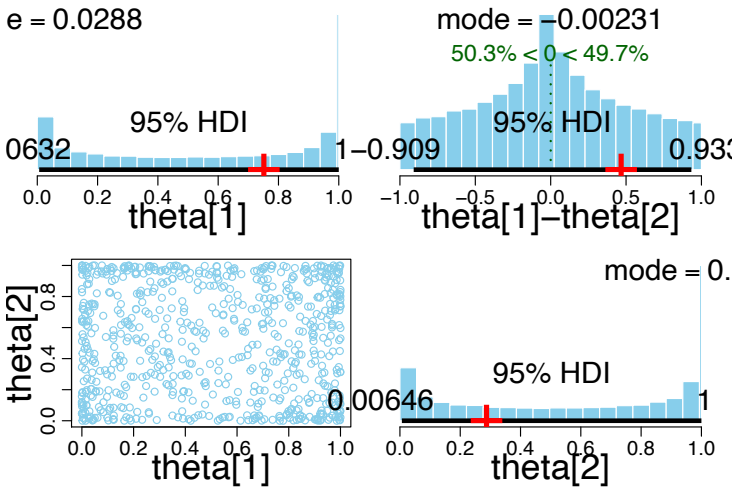



```
##                             Mean       Median         Mode   ESS HDImass
## theta[1]            0.5007727932  0.504378962  0.028794657 50000    0.95
## theta[2]            0.5016246459  0.504144840  0.971175228 50000    0.95
## theta[1]-theta[2] -0.0008518527 -0.001388936 -0.002311483 50000    0.95
##                         HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]            0.006316647 1.0000000      NA            NA      NA
## theta[2]            0.006463751 1.0000000      NA            NA      NA
## theta[1]-theta[2] -0.908684785 0.9333204       0         49.72      NA
##                   ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                NA         NA         NA         NA
## theta[2]                NA         NA         NA         NA
## theta[1]-theta[2]       NA         NA         NA         NA
```

```
# Display posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=NULL , #rope=c(0.45,0.55) ,
          compValDiff=0.0 , #ropeDiff = c(-0.05,0.05) ,
          saveName=fileNameRoot , saveType=graphFileType )
```

```
## pdf
##   2
```

```
#------------------------------------------------------------------------------
```


e = 0.0288
95% HDI
0632
1
0.0
0.2
0.4
0.6
0.8
1.0
theta[1]
mode = −0.00231
50.3% < 0 < 49.7%
95% HDI
−0.909
0.93
−1.0
−0.5
0.0
0.5
1.0
theta[1]−theta[2]

theta[2]
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
theta[1]
mode = 0.
95% HDI
0.00646
1
0.0
0.2
0.4
0.6
0.8
1.0
theta[2]