

677-HW

Jingning Yang

4/6/2020

Exercise 8.1

```
# install.packages("rjags")
# install.packages("runjags")
require(rjags)

## Loading required package: rjags
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
source("Jags-Ydich-XnomSsubj-MbernBeta.R")

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

#3 subjects
myData <- read.csv("file.csv")
y = myData$y
s = as.numeric(myData$s)
Ntotal = length(y)
Nsubj = length(unique(s))
dataList = list(y=y,s=s,Ntotal = Ntotal , Nsubj = Nsubj)

modelString = "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dbern( theta )
  }
  theta ~ dbeta( 2 , 2 )
}
"
writeLines( modelString , con="TEMPmodel.txt" )

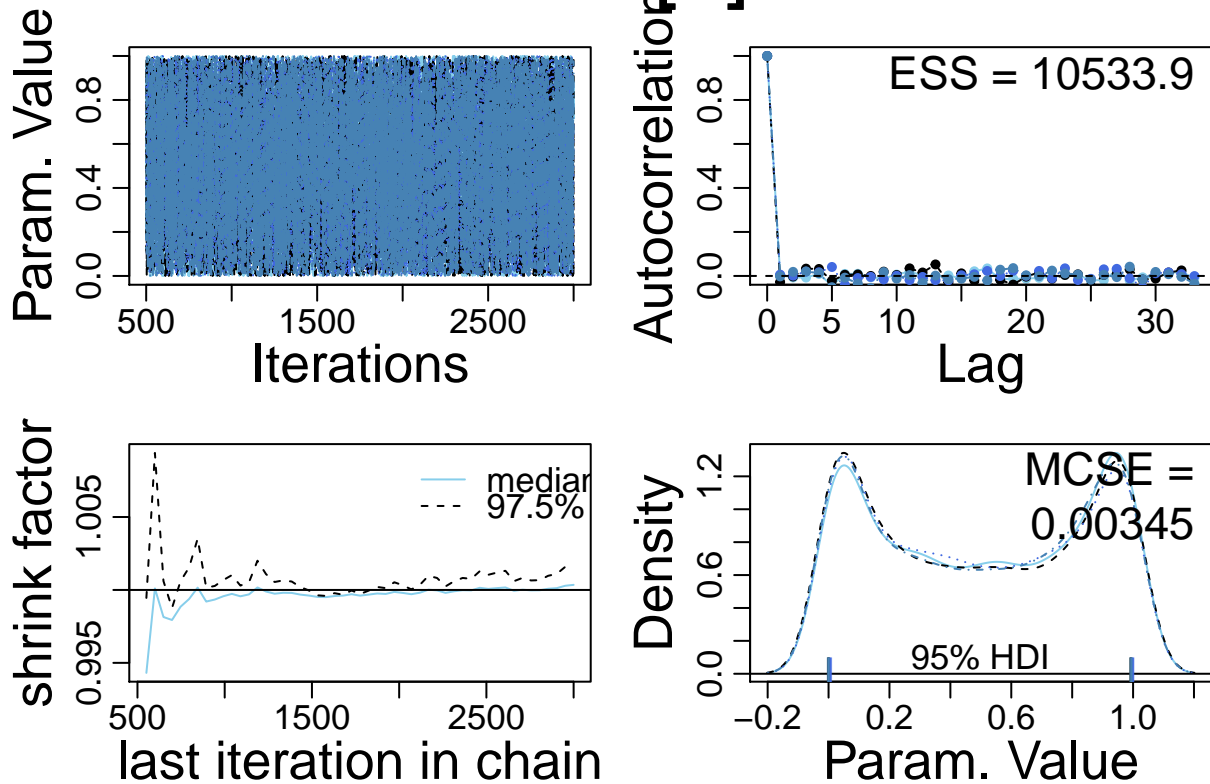
# Generate the MCMC chain:
mcmcCoda = genMCMC(data=myData , numSavedSteps=10000)

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

```
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 0
## Unobserved stochastic nodes: 23
## Total graph size: 46
##
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...

# Display diagnostics of chain, for specified parameter:
diagMCMC( mcmcCoda , parName="theta[1]" )
# Display numerical summary statistics of chain:
smryMCMC( mcmcCoda , compVal=NULL , compValDiff=0.0 )
```

theta[1]



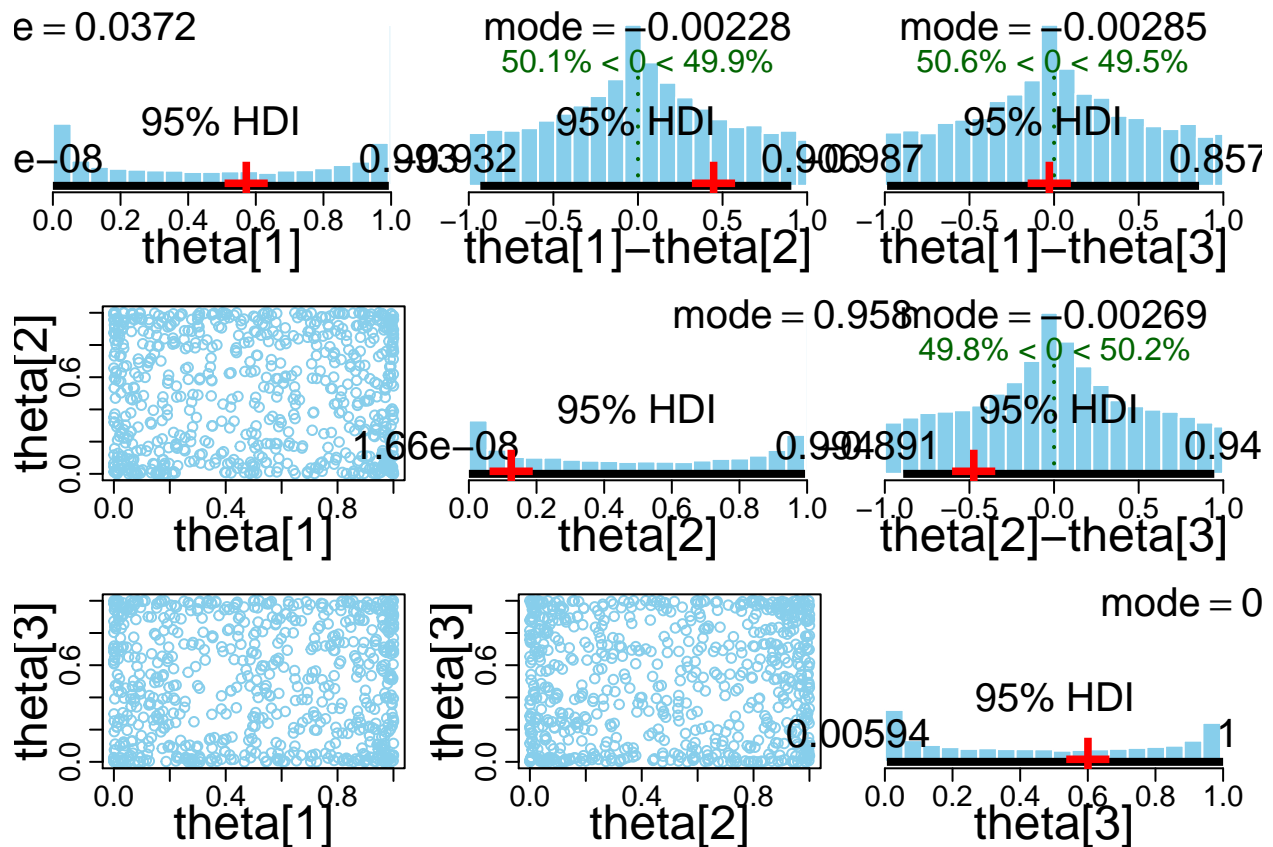
	Mean	Median	Mode	ESS	HDI _{mass}
## theta[1]	0.497945690	0.4975349564	0.037222395	9714.5	0.95
## theta[2]	0.500356361	0.4972467501	0.957779778	10000.0	0.95
## theta[3]	0.502362533	0.5002809898	0.960449521	10229.3	0.95
## theta[1]-theta[2]	-0.002410672	-0.0007587892	-0.002275014	10000.0	0.95
## theta[1]-theta[3]	-0.004416843	-0.0038477128	-0.002846761	10000.0	0.95
## theta[2]-theta[3]	-0.002006172	0.0009587373	-0.002691743	10000.0	0.95
	HDI _{low}	HDI _{high}	CompVal	PcntGtCompVal	ROPE _{low}
## theta[1]	2.080790e-08	0.9932242	NA	NA	NA
## theta[2]	1.660165e-08	0.9938847	NA	NA	NA
## theta[3]	5.935573e-03	1.0000000	NA	NA	NA
## theta[1]-theta[2]	-9.319097e-01	0.9064369	0	49.87	NA

```
## theta[1]-theta[3] -9.866071e-01 0.8567216      0      49.45      NA
## theta[2]-theta[3] -8.913535e-01 0.9470040      0      50.18      NA
##                ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                NA                NA                NA                NA
## theta[2]                NA                NA                NA                NA
## theta[3]                NA                NA                NA                NA
## theta[1]-theta[2]        NA                NA                NA                NA
## theta[1]-theta[3]        NA                NA                NA                NA
## theta[2]-theta[3]        NA                NA                NA                NA

##                Mean                Median                Mode                ESS HDImass
## theta[1]                0.497945690 0.4975349564 0.037222395 9714.5      0.95
## theta[2]                0.500356361 0.4972467501 0.957779778 10000.0     0.95
## theta[3]                0.502362533 0.5002809898 0.960449521 10229.3     0.95
## theta[1]-theta[2]       -0.002410672 -0.0007587892 -0.002275014 10000.0     0.95
## theta[1]-theta[3]       -0.004416843 -0.0038477128 -0.002846761 10000.0     0.95
## theta[2]-theta[3]       -0.002006172 0.0009587373 -0.002691743 10000.0     0.95
##                HDIlow  HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]                2.080790e-08 0.9932242      NA      NA      NA
## theta[2]                1.660165e-08 0.9938847      NA      NA      NA
## theta[3]                5.935573e-03 1.0000000      NA      NA      NA
## theta[1]-theta[2]       -9.319097e-01 0.9064369      0      49.87      NA
## theta[1]-theta[3]       -9.866071e-01 0.8567216      0      49.45      NA
## theta[2]-theta[3]       -8.913535e-01 0.9470040      0      50.18      NA
##                ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                NA                NA                NA                NA
## theta[2]                NA                NA                NA                NA
## theta[3]                NA                NA                NA                NA
## theta[1]-theta[2]        NA                NA                NA                NA
## theta[1]-theta[3]        NA                NA                NA                NA
## theta[2]-theta[3]        NA                NA                NA                NA

# Display graphical posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=NULL , compValDiff=0.0 )

#2 subjects
mydata <- read.csv("file1.csv")
```



```

y = mydata$y
s = as.numeric(mydata$s)
Ntotal = length(y)
Nsubj = length(unique(s))
dataList = list(y=y,s=s,Ntotal = Ntotal, Nsubj = Nsubj)

modelString = "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dbern( theta )
  }
  theta ~ dbeta( 2 , 2 )
}
"
writeLines(modelString , con="TEMPmodel.txt")

# Generate the MCMC chain:
mcmcCoda = genMCMC(data=mydata, numSavedSteps=10000)

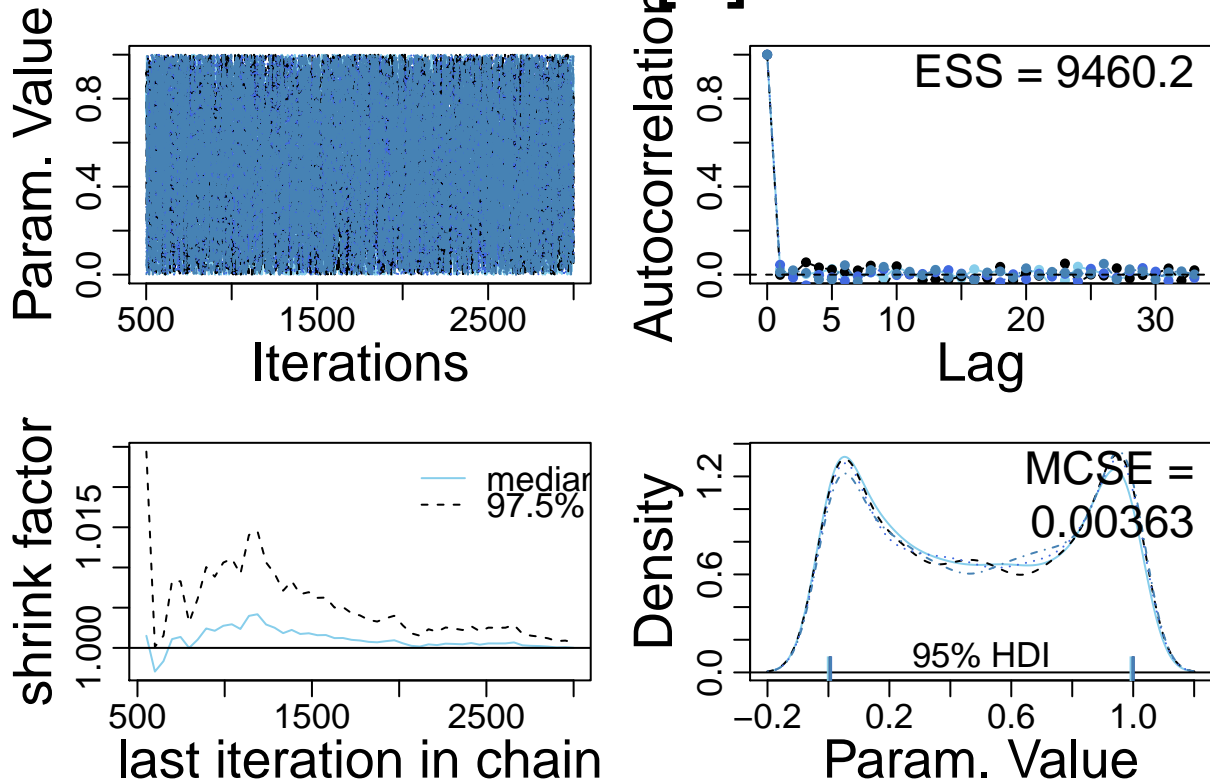
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 0
##   Unobserved stochastic nodes: 17
##   Total graph size: 35
##

```

```
## Initializing model
##
## Burning in the MCMC chain...
## Sampling final MCMC chain...

# Display diagnostics of chain, for specified parameter:
diagMCMC( mcmcCoda , parName="theta[1]" )
# Display numerical summary statistics of chain:
smryMCMC( mcmcCoda , compVal=NULL , compValDiff=0.0 )
```

theta[1]



```
##               Mean      Median      Mode      ESS HDImass
## theta[1]      0.5000300937 0.495545522 0.960057364 9600.3   0.95
## theta[2]      0.5003989276 0.499331265 0.037204433 10000.0  0.95
## theta[1]-theta[2] -0.0003688339 0.003489718 0.007207807 10000.0  0.95
##               HDIlow   HDIhigh CompVal PcmtGtCompVal ROPElow
## theta[1]      6.529347e-03 1.0000000    NA             NA      NA
## theta[2]      1.494091e-09 0.9940705    NA             NA      NA
## theta[1]-theta[2] -9.075822e-01 0.9410849    0             50.57   NA
##               ROPEhigh PcmtLtROPE PcmtInROPE PcmtGtROPE
## theta[1]      NA         NA         NA         NA
## theta[2]      NA         NA         NA         NA
## theta[1]-theta[2] NA         NA         NA         NA

##               Mean      Median      Mode      ESS HDImass
## theta[1]      0.5000300937 0.495545522 0.960057364 9600.3   0.95
## theta[2]      0.5003989276 0.499331265 0.037204433 10000.0  0.95
## theta[1]-theta[2] -0.0003688339 0.003489718 0.007207807 10000.0  0.95
##               HDIlow   HDIhigh CompVal PcmtGtCompVal ROPElow
```

```
## theta[1]          6.529347e-03 1.0000000      NA      NA      NA
## theta[2]          1.494091e-09 0.9940705      NA      NA      NA
## theta[1]-theta[2] -9.075822e-01 0.9410849      0      50.57     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 graphical posterior information:
plotMCMC( mcmcCoda , data=myData , compVal=NULL , compValDiff=0.0 )
```

The estimate are reasonable. Based on 2 different graphs, 2 subjects with 4 plots usually has wider HDI than 3 subjects with 9 plots in theta[1], theta[1]-thera[2] and theta[2].

Exercise 8.2

```
data <- read.csv("Jags-Ydich-XnomSsubj-MbernBeta-Example.R")
y = data$y
s = as.numeric(data$s)
Ntotal = length(y)
Nsubj = length(unique(s))
dataList = list(y=y,s=s,Ntotal = Ntotal, Nsubj = Nsubj)

modelString = "
model {
  for ( i in 1:Ntotal ) {
    y[i] ~ dbern( theta )
  }
  theta ~ dbeta( 2 , 2 )
}
"
writeLines(modelString , con="TEMPmodel.txt")

# Generate the MCMC chain:
summaryInfo = smryMCMC( mcmcCoda , compVal=0.5 , rope=c(0.45,0.55) ,
                        compValDiff=0.0 , ropeDiff = c(-0.05,0.05) )
```

```
##                Mean      Median      Mode      ESS HDImass
## theta[1]          0.5000300937 0.495545522 0.960057364 9600.3    0.95
## theta[2]          0.5003989276 0.499331265 0.037204433 10000.0   0.95
## theta[1]-theta[2] -0.0003688339 0.003489718 0.007207807 10000.0   0.95
##                HDIlow   HDIhigh CompVal PcntGtCompVal ROPElow
## theta[1]          6.529347e-03 1.0000000    0.5      49.64    0.45
## theta[2]          1.494091e-09 0.9940705    0.5      49.96    0.45
## theta[1]-theta[2] -9.075822e-01 0.9410849    0.0      50.57   -0.05
##                ROPEhigh PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]          0.55      47.11      6.28      46.61
## theta[2]          0.55      46.84      6.50      46.66
## theta[1]-theta[2] 0.05      44.30     11.01      44.69
```

The output will differ in the numerical details because of randomness in the MCMC chain. Although the output includes many decimal places, most are not significant because of the sampling randomness in the MCMC chain; only the first few digits are stable, depending on the ESS.

Each row corresponds to the parameter or parameter difference indicated in the left-most column. The columns labelled Mean, Median, and Mode are the corresponding values of the MCMC chain for that

parameter. The ESS is the effective sample size, which is the chain length divided by the autocorrelation, as defined in the book. The next three columns indicate the probability mass of the HDI (which here is the default of 95%), the lower limit of the HDI, and the upper limit. Next is the comparison value (CompVal) for single-parameter decisions, which was specified in the argument as 0.5. The next column indicates the percentage of the posterior that is greater than the comparison value (PcntGtCompVal). Next are the columns for the ROPE, which repeat the specifications in the arguments. The final three columns indicate the percentage of the posterior distribution that is less than the ROPE lower limit, within the ROPE limits, and greater than the ROPE upper limit.

Exercise 8.3

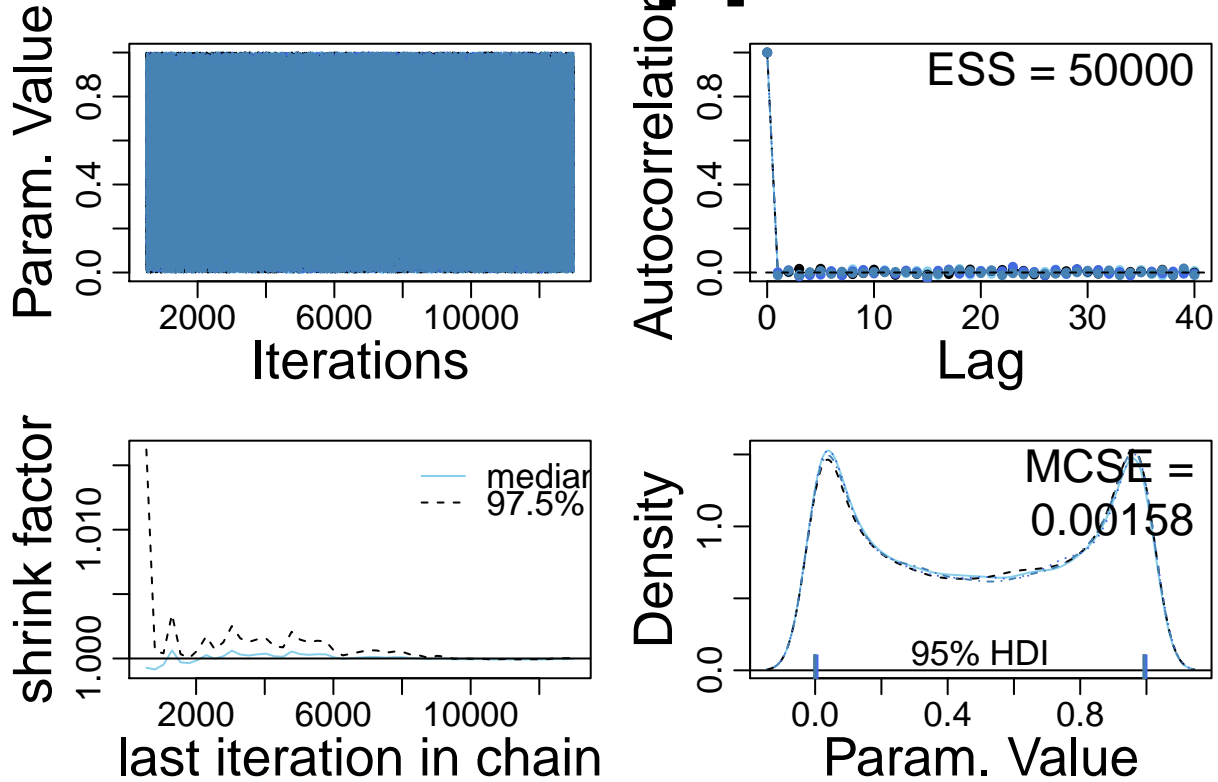
```
fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
# Generate the MCMC chain:
mcmcCoda = genMCMC( data=myData , numSavedSteps=50000 , saveName=fileNameRoot )

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 0
##   Unobserved stochastic nodes: 23
##   Total graph size: 46
##
## 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=0.5 , rope=c(0.45,0.55) ,
                        compValDiff=0.0 , ropeDiff = c(-0.05,0.05) ,
                        saveName=fileNameRoot )
```

theta[3]



##	Mean	Median	Mode	ESS	HDI _{mass}
## theta[1]	0.5011418100	0.4989109262	0.971283186	50000.0	0.95
## theta[2]	0.5007252434	0.4995887390	0.971191395	50000.0	0.95
## theta[3]	0.5001538698	0.5001249071	0.971131114	50000.0	0.95
## theta[1]-theta[2]	0.0004165666	0.0026226163	0.002301611	50000.0	0.95
## theta[1]-theta[3]	0.0009879402	-0.0005998594	-0.002294967	51822.8	0.95
## theta[2]-theta[3]	0.0005713736	-0.0002083852	-0.002198306	49160.7	0.95
##	HDI _{low}	HDI _{high}	CompVal	PcntGtCompVal	ROPE _{low}
## theta[1]	6.247300e-03	1.0000000	0.5	49.942	0.45
## theta[2]	5.878657e-10	0.9938046	0.5	49.976	0.45
## theta[3]	7.196564e-11	0.9935764	0.5	50.006	0.45
## theta[1]-theta[2]	-8.972241e-01	0.9460848	0.0	50.438	-0.05
## theta[1]-theta[3]	-8.791725e-01	0.9630820	0.0	49.898	-0.05
## theta[2]-theta[3]	-9.507599e-01	0.8904114	0.0	49.952	-0.05
##	ROPE _{high}	PcntLtROPE	PcntInROPE	PcntGtROPE	
## theta[1]	0.55	46.668	6.490	46.842	
## theta[2]	0.55	46.852	6.322	46.826	
## theta[3]	0.55	46.778	6.414	46.808	
## theta[1]-theta[2]	0.05	44.382	10.732	44.886	
## theta[1]-theta[3]	0.05	44.526	10.992	44.482	
## theta[2]-theta[3]	0.05	44.630	10.734	44.636	

```
# 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
```

Files “Jags-Ydich-XnomSsubj-MbernBeta-Mcmc.Rdata”, “Jags-Ydich-XnomSsubj-MbernBeta-Diagtheta[1].eps”, “Jags-Ydich-XnomSsubj-MbernBeta-Diagtheta[2].eps”, “Jags-Ydich-XnomSsubj-MbernBeta-Diagtheta[3].eps”, “Jags-Ydich-XnomSsubj-MbernBeta-Summaryinfo.csv” and “Jags-Ydich-XnomSsubj-MbernBeta-Post.eps” are created in my working directory.

The first line above specifies the beginning of the filenames for saved information, and the second line above specifies the graphics format for saved graphs.

The MCMC chain is saved in a file named Jags-Ydich-XnomSsubj-MbernBeta-Mcmc.Rdata. Notice the name is the fileNameRoot with Mcmc appended. It is in compressed Rdata format.

The diagnostic graphs are saved in files named Jags-Ydich-XnomSsubj-MbernBeta-Diagtheta[1].eps and Jags-Ydich-XnomSsubj-MbernBeta-Diagtheta[2].eps Notice the names are the fileNameRoot with Diag appended.

The summary information is saved in a file named Jags-Ydich-XnomSsubj-MbernBeta-SummaryInfo.csv which is in comma-separated-value format. Notice the name is the fileNameRoot with SummaryInfo appended.

The graph of the posterior distribution is saved in a file named Jags-Ydich-XnomSsubj-MbernBeta-Post.eps Notice the name is the fileNameRoot with Post appended.

Exercise 8.4

(A)

```
# Optional generic preliminaries:
graphics.off() # This closes all of R's graphics windows.
rm(list=ls()) # Careful! This clears all of R's memory!
myData = read.csv("file1.csv")
source("Jags-Ydich-XnomSsubj-MbernBeta.R")

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

fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
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...

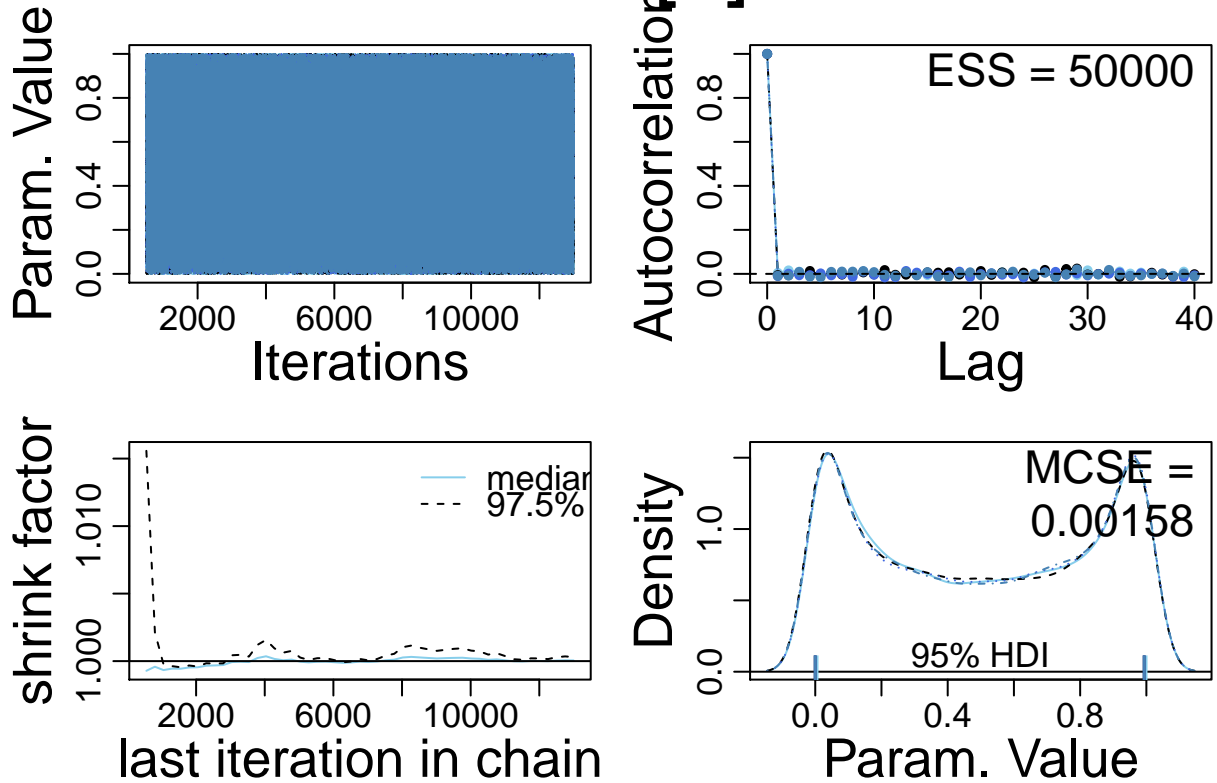
parameterNames = varnames(mcmcCoda) # get all parameter names
for ( parName in parameterNames ) {
  diagMCMC( codaObject=mcmcCoda , parName=parName ,
            saveName=fileNameRoot , saveType=graphFileType )
}
```

```

}
# 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 )

```

theta[2]



```

##               Mean      Median      Mode      ESS HDImass
## theta[1]      0.497428120 0.4958505752 0.028573419 50000.0  0.95
## theta[2]      0.496320732 0.4937229316 0.028622293 50000.0  0.95
## theta[1]-theta[2] 0.001107388 0.0007874868 0.002221212 50681.4  0.95
##
##               HDIlow  HDIhigh  CompVal  PcntGtCompVal  ROPElow
## theta[1]      7.587340e-09 0.9937124      NA           NA      NA
## theta[2]      1.261394e-11 0.9937920      NA           NA      NA
## theta[1]-theta[2] -9.310465e-01 0.9128707      0       50.158      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 )

```

```

## X11cairo
##      2

```

Basically, we are just run the file “Jags-Ydich-XnomSsubj-MbernBeta-Example.R” in our directory, and in

the file “Jags-Ydich-XnomSsubj-MbernBeta.R”, comment out the line that specifies y in the dataList in line 23. After we run, the graph of the posterior looks like Figure 8.7 shows.

(B)

```
# Optional generic preliminaries:
graphics.off() # This closes all of R's graphics windows.
rm(list=ls()) # Careful! This clears all of R's memory!
myData = read.csv("file1.csv")
source("Jags-Ydich-XnomSsubj-MbernBeta.R")

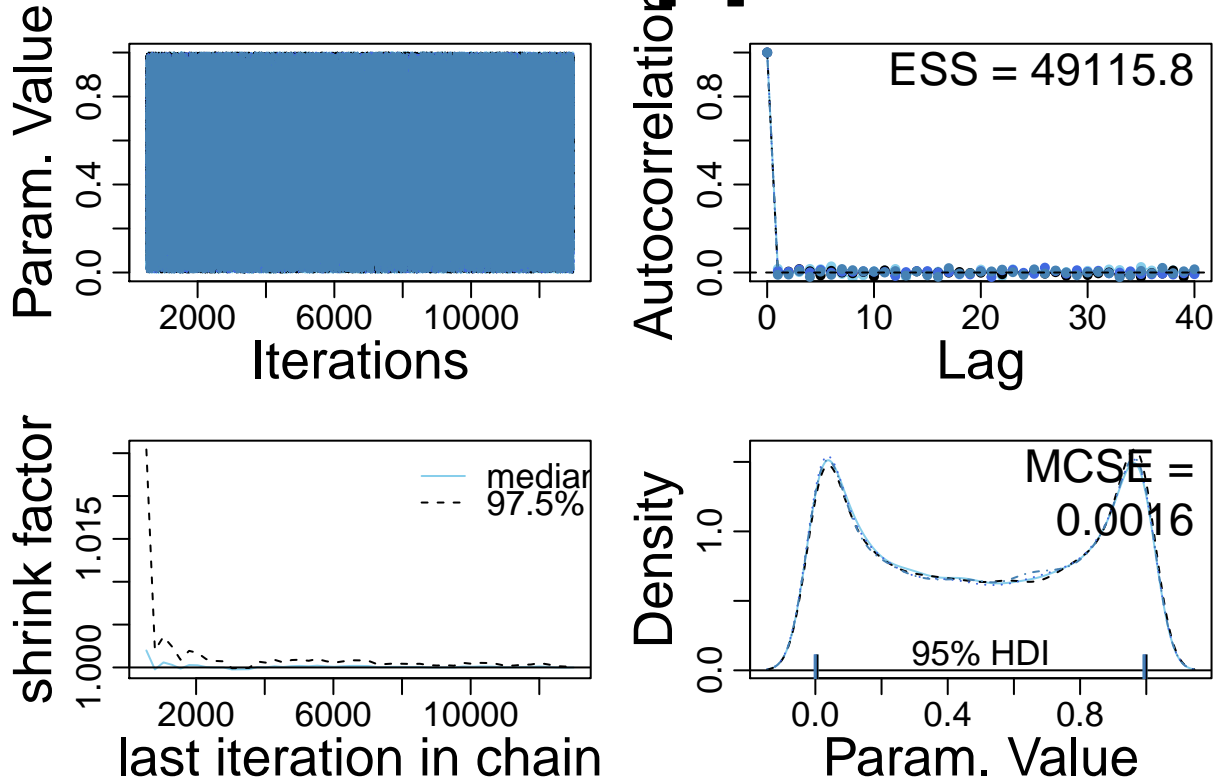
##
## *****
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *****

fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
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...

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 )
```

theta[2]



```
##               Mean      Median      Mode    ESS HDImass
## theta[1]      0.502522444 0.5065191197 0.971512679 50000    0.95
## theta[2]      0.501115429 0.4989982525 0.971545589 50000    0.95
## theta[1]-theta[2] 0.001407015 0.0007157592 -0.002181833 50000    0.95
##               HDIlow  HDIhigh CompVal PcmtGtCompVal ROPElow ROPEhigh
## theta[1]      0.006246965 1.00000000    NA           NA        NA      NA
## theta[2]      0.006082106 1.00000000    NA           NA        NA      NA
## theta[1]-theta[2] -0.881660063 0.9614546    0         50.156      NA      NA
##               PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]      NA        NA        NA
## theta[2]      NA        NA        NA
## theta[1]-theta[2] 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 )
```

```
## X11cairo
##      2
```

In the file Jags-Ydich-XnomSsubj-MbernBeta.R, change the specification of the prior to `dbeta(1,1)` in line 36, then we run the script Jags-Ydich-XnomSsubj-MbernBeta-Example.R.

Notice that the distributions on $\theta[1]$ and $\theta[2]$ look uniform, as they should, because that is a `dbeta(1,1)` distribution.

Interestingly, the prior distribution on $\theta[1]-\theta[2]$ is not uniform, but is triangular. This makes sense if you consider the lower left panel, which shows the uniform distribution on $\theta[1] \times \theta[2]$ space. If you

collapse that square space along the $\theta[1]=\theta[2]$ diagonal, you see that there are a lot of points along the diagonal, but the number of points drops off linearly toward the corners. In any case, the moral is that uniform priors on $\theta[1]$ and $\theta[2]$ do not imply a uniform prior on the difference of parameters.

(C)

```
# Optional generic preliminaries:
graphics.off() # This closes all of R's graphics windows.
rm(list=ls()) # Careful! This clears all of R's memory!
myData = read.csv("file1.csv")
source("Jags-Ydich-XnomSsubj-MbernBeta.R")

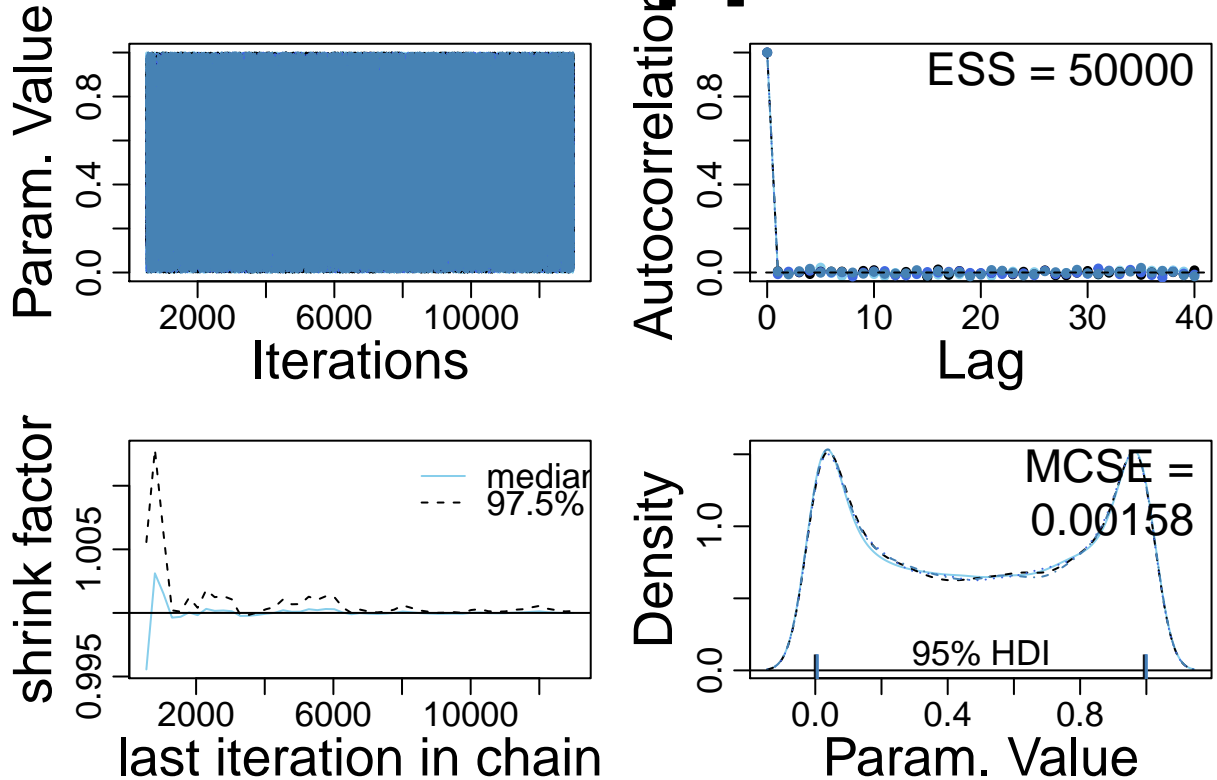
##
## *****
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## *****

fileNameRoot = "Jags-Ydich-XnomSsubj-MbernBeta-"
graphFileType = "eps"
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...

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 )
```

theta[2]



```
##                               Mean      Median      Mode    ESS HDImass
## theta[1]                    0.4992993040  0.4992926256  0.028904805 50000    0.95
## theta[2]                    0.5000323209  0.5012473493  0.028566343 50000    0.95
## theta[1]-theta[2] -0.0007330169 -0.0002887781 -0.002283213 50000    0.95
##                               HDIlow  HDIhigh CompVal PcmtGtCompVal ROPElow ROPEhigh
## theta[1]                    0.006486845  1.00000000    NA             NA        NA        NA
## theta[2]                    0.006134300  1.00000000    NA             NA        NA        NA
## theta[1]-theta[2] -0.865276845  0.9824366    0          49.938    NA        NA
##                               PcntLtROPE PcntInROPE PcntGtROPE
## theta[1]                    NA           NA           NA
## theta[2]                    NA           NA           NA
## theta[1]-theta[2]          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 )
```

```
## X11cairo
##      2
```

In the file Jags-Ydich-XnomSsubj-MbernBeta.R, change the specification of the prior to $\text{dbeta}(1,1)$, then we run the script Jags-Ydich-XnomSsubj-MbernBeta-Example.R.

The individual parameters can be seen to have $\text{dbeta}(0.5,0.5)$ marginal distributions, but the resulting distribution on $\theta[1]-\theta[2]$ is curious. By looking at the joint distribution in the lower left panel, some insight can be gleaned; the joint distribution has higher density toward the corners. Again, the main point is that a prior on individual parameters may have unforeseen implications for the prior on the difference of parameters. When you are done with this exercise, you might want to return the program Jags-Ydich-

XnomSsubj-MbernBeta.R to its original condition!