

STAT 8700 Homework 9

Brian Detweiler

Friday, November 11th, 2016

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

1. Repeat Assignment 8, using Stan instead of JAGS.

8.1. Suppose we have a population described by a Normal Distribution with known variance $\sigma^2 = 1600$ and unknown mean μ .

4 observations are collected from the population and the corresponding values were:

940, 1040, 910, and 990.

```
fileName <- "Assignment_9_1_1"  
data <- c(940, 1040, 910, 990)  
N <- length(data)  
y.bar <- mean(data)  
  
# Write the Stan model file  
modelString =  
"  
data {  
    int<lower=0> N;  
    real<lower=0> y[N];  
}  
parameters {  
    real<lower=0> theta;  
}  
model {  
    theta ~ normal(1000, 200); // Prior  
    y ~ normal(theta, 40);    // Likelihood  
}  
"  
  
stan.file <- paste0(fileName, ".stan")  
writeLines(modelString, con=stan.file)  
  
model.data <- list(y = data, N = N)  
fit <- stan(file = stan.file,
```

```

    data = model.data,
    iter = 120000,
    warmup = 2000,
    chains = 4,
    cores = 4,
    open_progress = F)

## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/base.hpp:8:
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/prng.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/model.hpp:8:
##     from file1c307c05b4c.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In static member function <anonymous>::operator<<:
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:42:43:
##       typedef typename Array::index_range index_range;
##   ^
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:43:37:
##       typedef typename Array::index index;
##   ^
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In static member function <anonymous>::operator<<:
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:53:43:
##       typedef typename Array::index_range index_range;
##   ^
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:54:37:
##       typedef typename Array::index index;
##   ^
## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/prng.hpp:8:
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/prng.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/prng.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/prng.hpp
##   from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/model.hpp
##   from file1c307c05b4c.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad.hpp: In static member function <anonymous>::set_zero_all_ad():
##   C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad.hpp:8:
##       static void set_zero_all_adjoints() {
##   ^
print(fit)

## Inference for Stan model: Assignment_9_1_1.
## 4 chains, each with iter=120000; warmup=2000; thin=1;
## post-warmup draws per chain=118000, total post-warmup draws=472000.
##
##          mean se_mean    sd   2.5%    25%    50%    75%   97.5% n_eff Rhat
## theta  970.22     0.03 19.39  931.98  957.29  970.23  983.09 1008.54 341641     1
## lp__    3.33     0.00  0.69    1.38    3.18    3.59    3.76    3.80 201207     1
##
```

```

## Samples were drawn using NUTS(diag_e) at Fri Nov 11 16:32:19 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

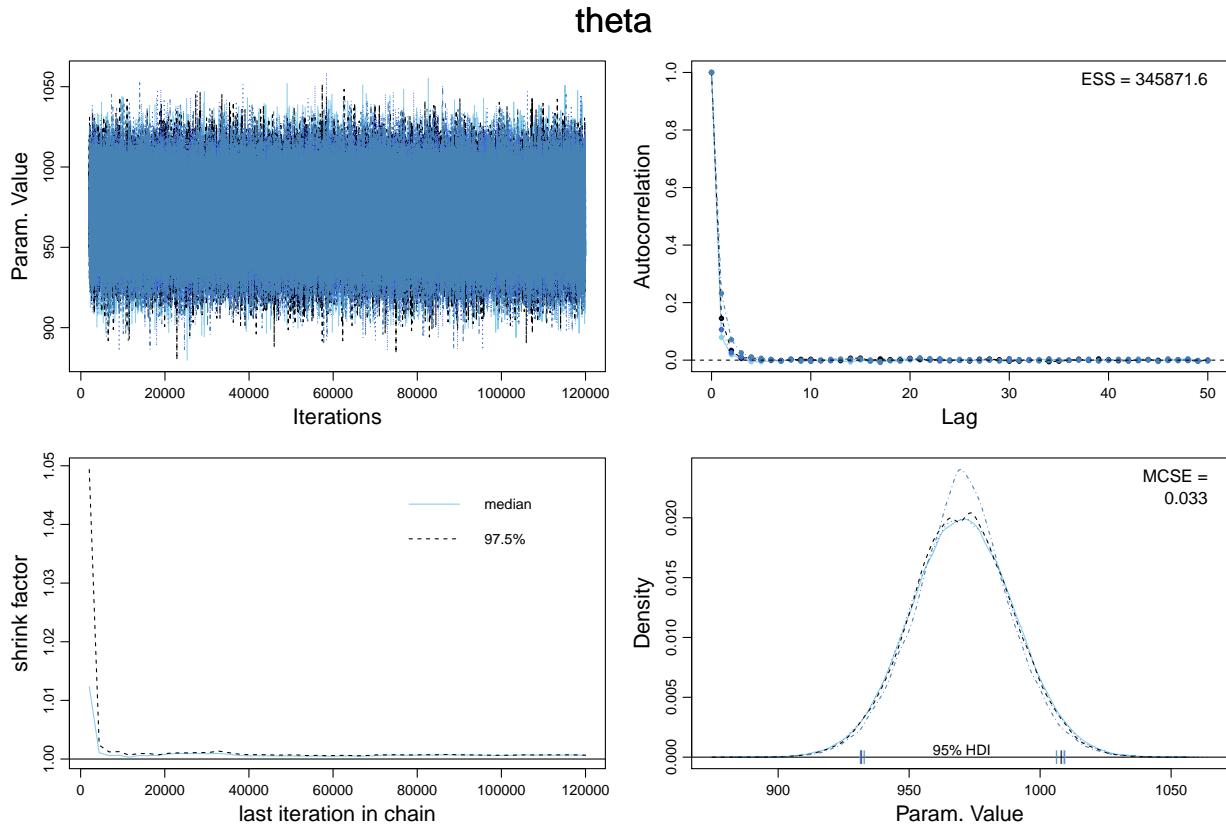
```

```

fit.mcmc <- As.mcmc.list(fit)
diagMCMC(fit.mcmc)

fit.mcmc.M <- as.matrix(fit)

```



```

hist(fit.mcmc.M[, "theta"], breaks=80, freq=F)
x <- seq(0, 1200, length = 1000)
y <- dnorm(x, mean = y.bar, sd = sqrt(1600 / N))
lines(x, y, type="l", lwd=1, col="red")

CI <- quantile(fit.mcmc.M[, "theta"], c(0.025, 0.975))
CI[[1]]

```

```

## [1] 931.9821

```

```

CI[[2]]

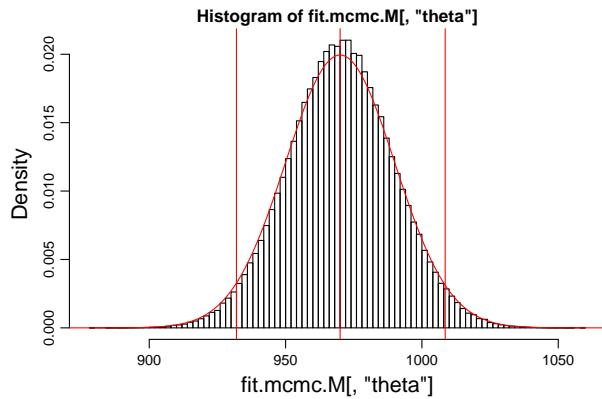
```

```

## [1] 1008.537

```

```
abline(v=CI[[1]], col="red")
abline(v=CI[[2]], col="red")
abline(v=y.bar, col="red")
```



■

9.1.2. Suppose we consider a Normal population with a variance of 16, and we collect 15

observations from this population with values:

26.8, 26.3, 28.3, 28.5, 26.3, 31.9, 28.5, 27.2, 20.9, 27.5, 28.0, 18.6, 22.3, 25.0, 31.5.

(a) If we choose a $\text{Normal}(20, 25)$ prior,

Use R to find the posterior distribution for the population mean.

```
fileName <- "Assignment_9_1_2"

var <- 16
data <- c(26.8, 26.3, 28.3, 28.5, 26.3, 31.9, 28.5, 27.2, 20.9, 27.5, 28.0, 18.6, 22.3, 25.0, 31.5)
y.bar <- mean(data)
N <- length(data)
sigma_2 <- var

# Write the Stan model
modelString =
"
data {
  int<lower=0> N;
  real<lower=0> y[N];
}
parameters {
  real<lower=0, upper=100> theta;
}
model {
  theta ~ normal(20, 5);    // Prior
  y ~ normal(theta, 4);    // Likelihood
}

"

stan.file <- paste0(fileName, ".stan")
writeLines(modelString, con=stan.file)

model.data <- list(y = data, N = N)
fit <- stan(file = stan.file,
            data = model.data,
            iter = 120000,
            warmup = 2000,
            chains = 4,
            cores = 4,
            open_progress = F)

## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/base.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint/
```

```

##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint.hpp
##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/pr
##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/mod
##           from file1c3025947ecd.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In stan
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:42:43:
##     typedef typename Array::index_range index_range;
##                                         ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:43:37:
##     typedef typename Array::index index;
##                                         ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In stan
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:53:43:
##     typedef typename Array::index_range index_range;
##                                         ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:54:37:
##     typedef typename Array::index index;
##                                         ^
## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev
##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev
##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math.hpp
##           from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/mod
##           from file1c3025947ecd.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad
##     static void set_zero_all_adjoint() {
##                                         ^

print(fit)

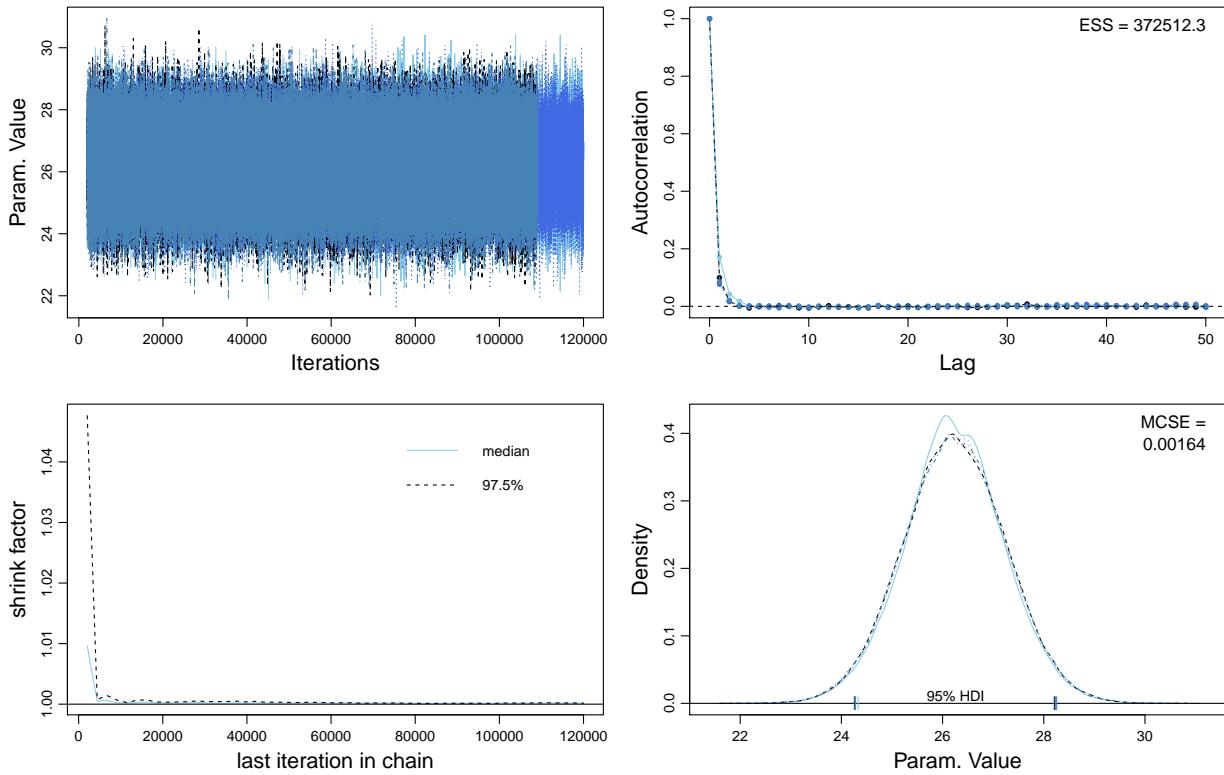
## Inference for Stan model: Assignment_9_1_2.
## 4 chains, each with iter=120000; warmup=2000; thin=1;
## post-warmup draws per chain=118000, total post-warmup draws=472000.
##
##           mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## theta    26.24      0 1.00 24.26 25.57 26.23 26.91 28.21 371196    1
## lp__   -4.06      0 0.71 -6.07 -4.22 -3.79 -3.62 -3.57 226989    1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov 11 16:33:04 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

fit.mcmc <- As.mcmc.list(fit)
diagMCMC(fit.mcmc)

fit.mcmc.M <- as.matrix(fit)

```

theta



```
hist(fit.mcmc.M[, "theta"], breaks=80, freq=F)
x <- seq(20, 40, length = 1000)
y <- dnorm(x, mean = y.bar, sd = sqrt(var / N))
lines(x, y, type="l", lwd=1, col="red")
```

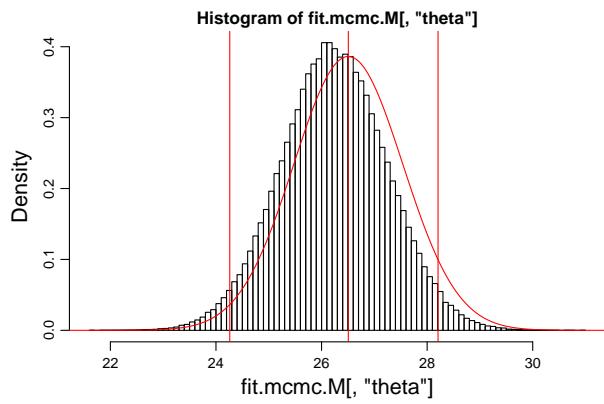
```
CI <- quantile(fit.mcmc.M[, "theta"], c(0.025, 0.975))
CI[[1]]
```

```
## [1] 24.26123
```

```
CI[[2]]
```

```
## [1] 28.20641
```

```
abline(v=CI[[1]], col="red")
abline(v=CI[[2]], col="red")
abline(v=y.bar, col="red")
```



9.1.3

```
fileName <- "Assignment_9_1_3"
years <- c(1976:1985)

fatal.accidents <- c(24, 25, 31, 31, 22, 21, 26, 20, 16, 22)
passenger.deaths <- c(734, 516, 754, 877, 814, 362, 764, 809, 223, 1066)
death.rate <- c(0.19, 0.12, 0.15, 0.16, 0.14, 0.06, 0.13, 0.13, 0.03, 0.15)

airline.deaths <- as.data.frame(cbind(years, fatal.accidents, passenger.deaths, death.rate))

data <- fatal.accidents
N <- length(data)

modelString =
"
data {
  int<lower=0> N;
  int<lower=0> y[N];
}
parameters {
  real<lower=0> theta;
}
model {
  theta ~ gamma(1, 1);    // Prior
  y ~ poisson(theta);    // Likelihood
}
"

stan.file <- paste0(fileName, ".stan")
writeLines(modelString, con=stan.file)

model.data <- list(y = data, N = N)
fit <- stan(file = stan.file,
            data = model.data,
            iter = 120000,
            warmup = 2000,
            chains = 4,
            cores = 4,
            open_progress = F)

## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/base.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint/algorithm.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/primitive.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/primitive.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/primitive.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/primitive.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/reverse.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math.hpp
##                                         from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/model.hpp
##                                         from file1c301cff36e1.cpp:8:
```

```

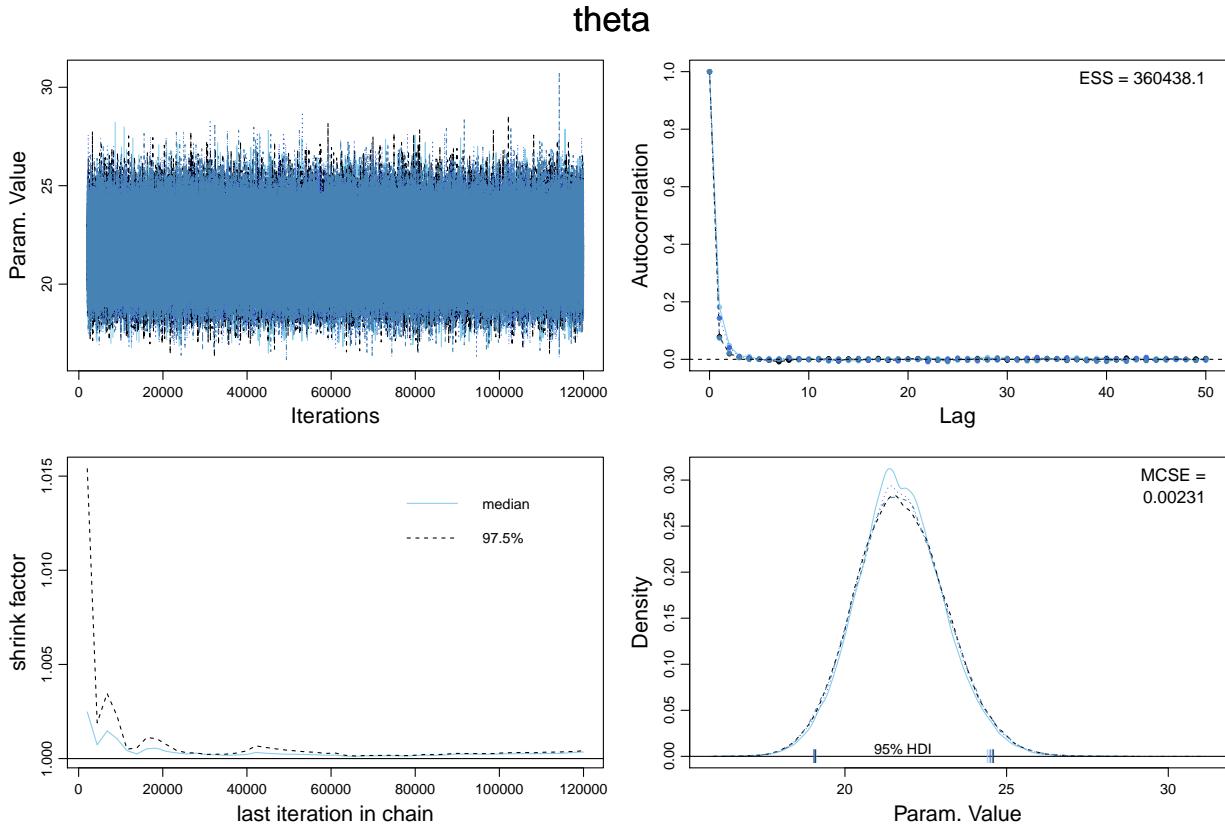
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In st
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:42:43:
##     typedef typename Array::index_range index_range;
##                                     ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:43:37:
##     typedef typename Array::index index;
##                                     ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In st
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:53:43:
##     typedef typename Array::index_range index_range;
##                                     ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:54:37:
##     typedef typename Array::index index;
##                                     ^
## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/re
##                 from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/re
##                 from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math.hpp
##                 from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/mod
##                 from file1c301cff36e1.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad
##     static void set_zero_all_adjoint() {
##                                     ^
##
print(fit)

## Inference for Stan model: Assignment_9_1_3.
## 4 chains, each with iter=120000; warmup=2000; thin=1;
## post-warmup draws per chain=118000, total post-warmup draws=472000.
##
##          mean se_mean    sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## theta  21.72      0 1.39 19.09 20.78 21.68 22.63 24.54 356182    1
## lp__  496.29      0 0.70 494.30 496.14 496.56 496.73 496.78 206056    1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov 11 16:33:49 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

fit.mcmc <- As.mcmc.list(fit)
diagMCMC(fit.mcmc)

fit.mcmc.M <- as.matrix(fit)

```



```

hist(fit.mcmc.M[, "theta"], breaks=80, freq=F)
x <- seq(10, 30, length = 1000)
test <- data + 1
y <- dgamma(x, shape = sum(data) + 1, rate = N + 1)
lines(x, y, type="l", lwd=1, col="red")

CI <- quantile(fit.mcmc.M[, "theta"], c(0.025, 0.975))
CI[[1]]

```

```
## [1] 19.08636
```

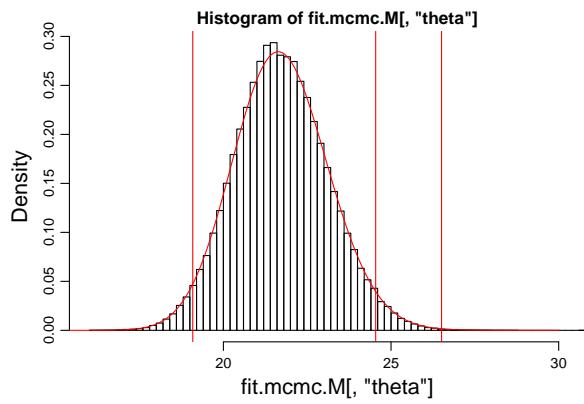
```
CI[[2]]
```

```
## [1] 24.54203
```

```

abline(v=CI[[1]], col="red")
abline(v=CI[[2]], col="red")
abline(v=y.bar, col="red")

```



■

9.1.4

```
fileName <- "Assignment_9_1_4"

years <- c(1976:1985)

fatal.accidents <- c(24, 25, 31, 31, 22, 21, 26, 20, 16, 22)
death.rate <- c(0.19, 0.12, 0.15, 0.16, 0.14, 0.06, 0.13, 0.13, 0.03, 0.15)

passenger.deaths.data <- c(734, 516, 754, 877, 814, 362, 764, 809, 223, 1066)
N <- length(passenger.deaths.data)

modelString =
"
data {
  int<lower=0> N;
  int<lower=0> y[N];
}
parameters {
  real<lower=0> theta;
}
model {
  theta ~ gamma(1, 1);    // Prior
  y ~ poisson(theta);    // Likelihood
}
"

stan.file <- paste0(fileName, ".stan")
writeLines(modelString, con=stan.file)

model.passenger.deaths.data <- list(y = passenger.deaths.data, N = N)

fit <- stan(file = stan.file,
            data = model.passenger.deaths.data,
            iter = 120000,
            warmup = 2000,
            chains = 4,
            cores = 4,
            open_progress = F)

## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/base.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/numeric/odeint/algorithm.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/primitive.hpp
##                                     from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/math/primitive.hpp
##                                     from file1c302e502b9d.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In st
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:42:43:
```

```

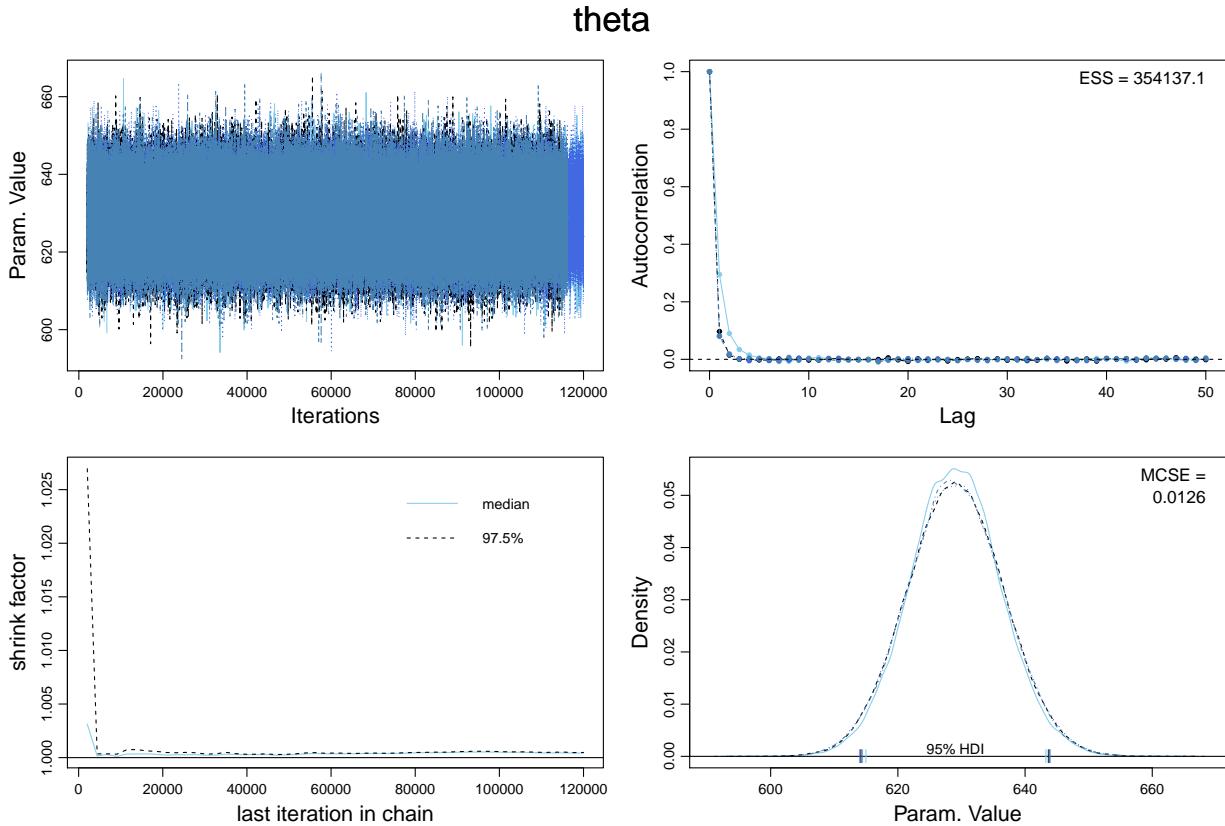
##      typedef typename Array::index_range index_range;
##      ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:43:37:
##      typedef typename Array::index index;
##      ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp: In st
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:53:43:
##      typedef typename Array::index_range index_range;
##      ^
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/BH/include/boost/multi_array/concept_checks.hpp:54:37:
##      typedef typename Array::index index;
##      ^
## In file included from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/re
##                  from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/re
##                  from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math.hpp
##                  from C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/src/stan/mod
##                  from file1c302e502b9d.cpp:8:
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad_
## C:/Users/icom942/Desktop/apps/R-3.3.1/library/StanHeaders/include/stan/math/rev/core/set_zero_all_ad_
##     static void set_zero_all_adoints() {
##             ^
##
print(fit)

## Inference for Stan model: Assignment_9_1_4.
## 4 chains, each with iter=120000; warmup=2000; thin=1;
## post-warmup draws per chain=118000, total post-warmup draws=472000.
##
##          mean se_mean    sd      2.5%      25%      50%      75%     97.5%
## theta    629.08    0.01 7.48   614.51   624.05   629.04   634.09   643.86
## lp__   37673.90    0.00 0.70 37671.92 37673.74 37674.17 37674.34 37674.39
##          n_eff Rhat
## theta  346885     1
## lp__   215154     1
##
## Samples were drawn using NUTS(diag_e) at Fri Nov 11 16:34:33 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

fit.mcmc <- As.mcmc.list(fit)
diagMCMC(fit.mcmc)

fit.mcmc.M <- as.matrix(fit)

```



```

hist(fit.mcmc.M[, "theta"], breaks=80, freq=F)
x <- seq(200, 1200, length = 1000)

y <- dgamma(x, shape = sum(data) + 1, rate = N + 1)
lines(x, y, type="l", lwd=1, col="red")

CI <- quantile(fit.mcmc.M[, "theta"], c(0.025, 0.975))
CI[[1]]

```

```
## [1] 614.5107
```

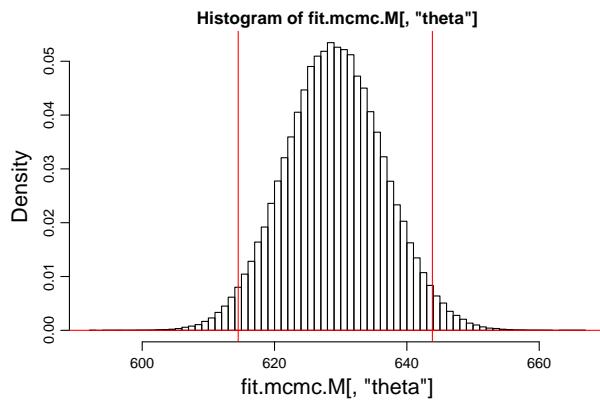
```
CI[[2]]
```

```
## [1] 643.857
```

```

abline(v=CI[[1]], col="red")
abline(v=CI[[2]], col="red")
abline(v=y.bar, col="red")

```



■

2. In Section 3.7 (starting on page 74) and in class, we discussed the analysis of a bioassay

experiment. We simulated from the posterior distribution using a uniform prior via

the grid method (this code is in Chapter3.R).

$$\begin{aligned}\alpha &\sim Uniform(-5, 10) \\ \beta &\sim Uniform(-10, 40)\end{aligned}$$

Use JAGS or stan to fit this model to the data, using the following prior distribution distributions (which effectively match the one used in the grid method): Don't forget to check the model diagnostics and re-simulate if necessary. Obtain at least 10,000 simulations per chain of α and β , summarize each posterior distribution (mean, variance/sd, 95% credible interval) and overlay the simulated values on the contour plot from Chapter3.R to verify that your simulated values seem to be from the 'correct' posterior distribution. Add a variable to your model that takes the value 1 if $\beta > 0$ and 0 otherwise. What fraction of your simulated values of β are greater than 0?

(c) Add the variable LD50 to your model, where LD50 is defined on page 77. Summarize the posterior distribution of LD50. Draw a histogram of the simulated posterior values of LD50 and compare it to the histogram in page 77 (they should be very similar).

(d) Suppose we now wish to give 20 animals a dose level of -0.2 Simulate the posterior predictive distribution of the number deaths in the group of 20 animals. Draw a histogram of this distribution and give a 95% credible interval.

```
fileName <- "Assignment_9_2_0"

x <- c(-0.86, -0.3, -0.05, 0.73)
n <- c(5, 5, 5, 5)
y <- c(0, 1, 3, 5)
y.mod <- c(0.5, 1, 3, 4.5)

data <- cbind(y, n - y)

bivariate_normal <- function(u, v) {
  mu_U <- 0
  mu_V <- 10

  sigma_2_U <- 4
  sigma_2_V <- 100
```

```

sigma_U <- sqrt(sigma_2_U)
sigma_V <- sqrt(sigma_2_V)

rho <- 0.5

first <- 1 / (2 * pi * sigma_U * sigma_V * sqrt(1 - rho^2))

second <- exp(-(1 / (2 * (1 - rho^2))))
  * ( ((u - mu_U)^2 / sigma_2_U)
    + ((v - mu_V)^2 / sigma_2_V)
    - (2 * rho * (u - mu_U) * (v - mu_V)) / (sqrt(sigma_2_U) * sqrt(sigma_2_V)))
rval <- first * second
return(rval)
}

bioassay <- data.frame(cbind(x, n, y, y.mod))

inv_logit <- function(x){
  exp(x) / (1 + exp(x))
}

posterior <- function(a, b) {
  temp <- 1
  x <- bioassay$x
  y <- bioassay$y
  n <- bioassay$n
  for (i in 1: length(x)) {
    temp <- (temp * (inv_logit(a + (b * x[i]))^y[i])
              * ((1 - inv_logit(a + (b * x[i])))^(n[i] - y[i])))
  }
  bivariate_normal(a, b) * temp
}

N <- length(x)
C <- 10000 # this just has to be large enough to ensure all phi[i]'s < 1
# Note that JAGS will NOT WARN you if C is too small, but will sample from a truncated di

ones <- rep(1, N)
model.data <- list(y=y, x=x, n=n, N=N)

modelString =
"
model {

  for (i in 1:N) {
    line[i] <- alpha + (beta * x[i])
    p[i] <- ilogit(line[i])
    y[i] ~ dbin(p[i], n[i])
  }

  # 9.2 d.) - Could make the dosage and number of animals part of the data, but we'll keep it simple for
}

```

```

px <- alpha + (beta * -0.2)
y_pred ~ dbin(ilogit(px), 20)

alpha ~ dunif(-5, 10)    # Prior
beta ~ dunif(-10, 40)    # Prior

# Question 9.2 b.)
# This will contain the percentage of alphas and betas > 0
beta0 <- step(beta)
alpha0 <- step(alpha)

ld50 <- -alpha / beta
}

"

jags.file <- paste0(fileName, ".jags")
writeLines(modelString, con=jags.file)

# There are 3 steps to getting JAGS to generate samples from the posterior distribution.

# Step 1 is to get all the model information into JAGS, so that it can figure out which algorithm to use

bioassayModel = jags.model(file = jags.file,
                           data = model.data,
                           n.chains = 4)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 4
##   Unobserved stochastic nodes: 3
##   Total graph size: 45
##
## Initializing model

#Step 2 is to run the chains for the burn-in/warm-up period

update(bioassayModel, n.iter=2000)

#Step 3 is to runs and records the MCMC samples that we will use.

bioassaySamples <- coda.samples(bioassayModel, n.iter=10000, variable.names=c("alpha", "beta", "beta0"))

summary(bioassaySamples)

##
## Iterations = 3001:13000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,

```

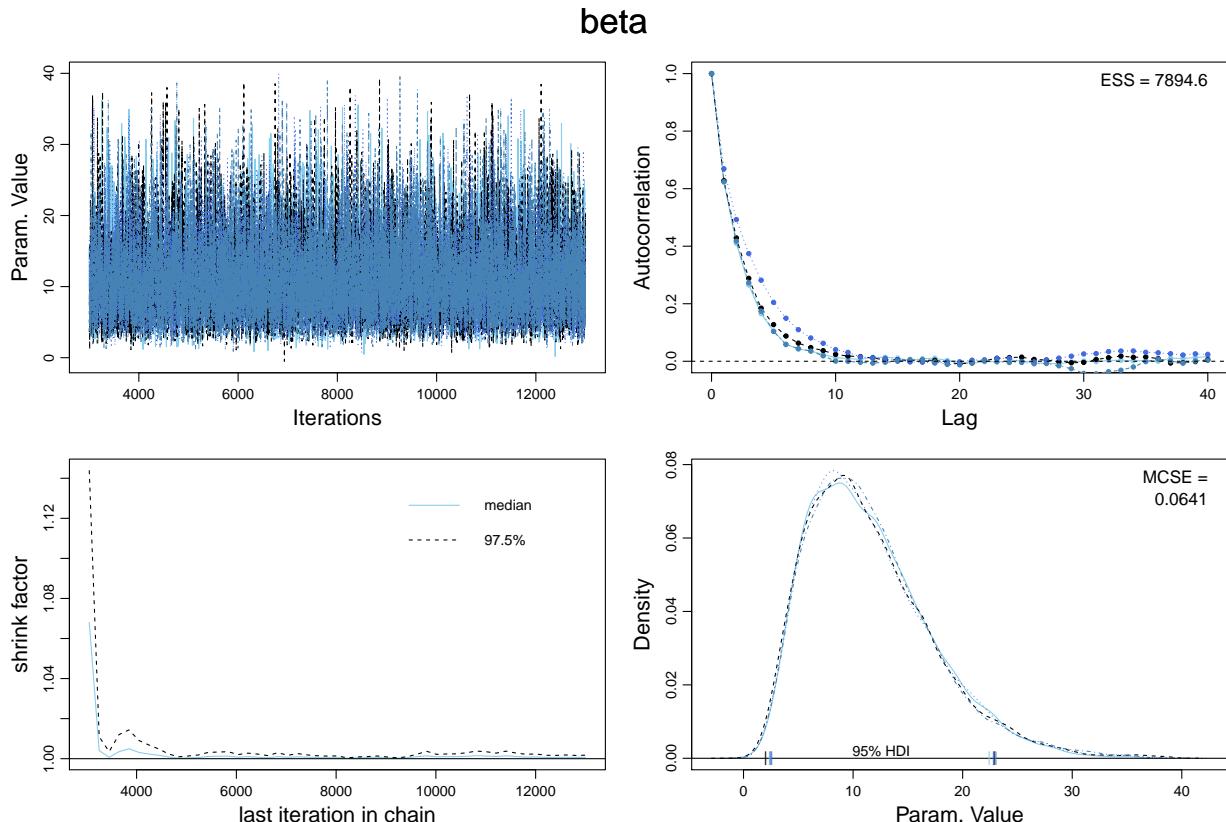
```

##      plus standard error of the mean:
##
##          Mean        SD  Naive SE Time-series SE
## alpha    1.2912 1.09972 0.0054986      0.0115643
## alpha0   0.8921 0.31032 0.0015516      0.0022888
## beta     11.5447 5.69124 0.0284562      0.0645870
## beta0    1.0000 0.00500 0.0000250      0.0000250
## ld50    -0.1050 0.09769 0.0004885      0.0006839
## y_pred   5.9556 3.75756 0.0187878      0.0198155
##
## 2. Quantiles for each variable:
##
##          2.5%       25%       50%       75%     97.5%
## alpha   -0.6176  0.5306  1.2111  1.9665  3.704
## alpha0   0.0000  1.0000  1.0000  1.0000  1.000
## beta    3.3734  7.2900 10.5926 14.7846 25.083
## beta0   1.0000  1.0000  1.0000  1.0000  1.000
## ld50   -0.2774 -0.1613 -0.1105 -0.0581  0.111
## y_pred  0.0000  3.0000  6.0000  8.0000 14.000

# Plot some diagnostics and save the plots to disk
diagMCMC(codaObject = bioassaySamples, parName=c("alpha"))
diagMCMC(codaObject = bioassaySamples, parName=c("beta"))

# To draw a histogram, we need to combine all the chains into 1 matrix
alpha.M <- as.matrix(bioassaySamples[, "alpha"])

```



```

beta.M <- as.matrix(bioassaySamples[, "beta"])
alpha0.M <- as.matrix(bioassaySamples[, "alpha0"])
beta0.M <- as.matrix(bioassaySamples[, "beta0"])
ld50.M <- as.matrix(bioassaySamples[, "ld50"])
y_pred.M <- as.matrix(bioassaySamples[, "y_pred"])

sims <- data.frame(alpha.M, beta.M)

posterior_contour <- function(alpha_min,
                                alpha_max,
                                grid_size_alpha,
                                beta_min,
                                beta_max,
                                grid_size_beta,
                                drawlabels = TRUE) {

  # Generate a list of alpha values
  alpha <- seq(alpha_min, alpha_max, length = grid_size_alpha)

  # Generate a list of beta values
  beta <- seq(beta_min, beta_max, length = grid_size_beta)

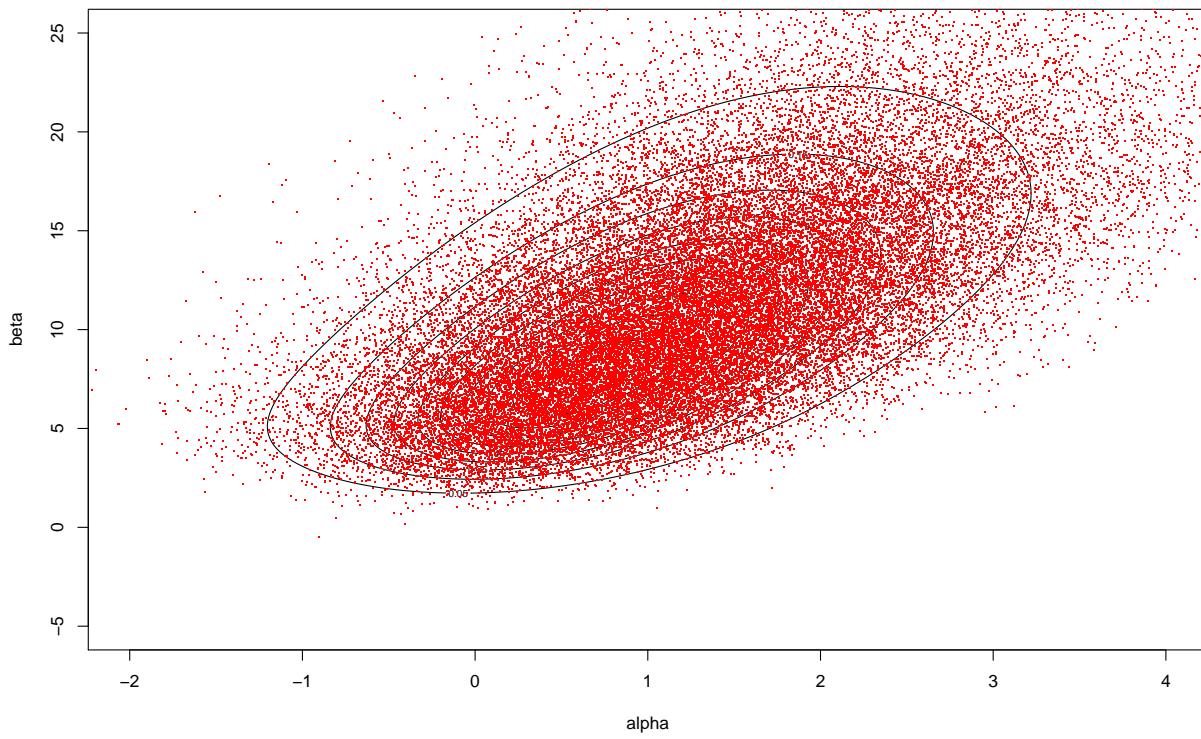
  # Evaluate the posterior density and all possible combinations of alpha and beta values.
  post.dens <- outer(alpha, beta, 'posterior')

  # Rescale the posterior so values are now relative to height of posterior mode.
  scaled.dens <- post.dens / max(post.dens)

  # Draw contour plot.
  contour(alpha,
          beta,
          scaled.dens,
          levels = c(0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95),
          xlab="alpha",
          ylab="beta",
          drawlabels = drawlabels)
}

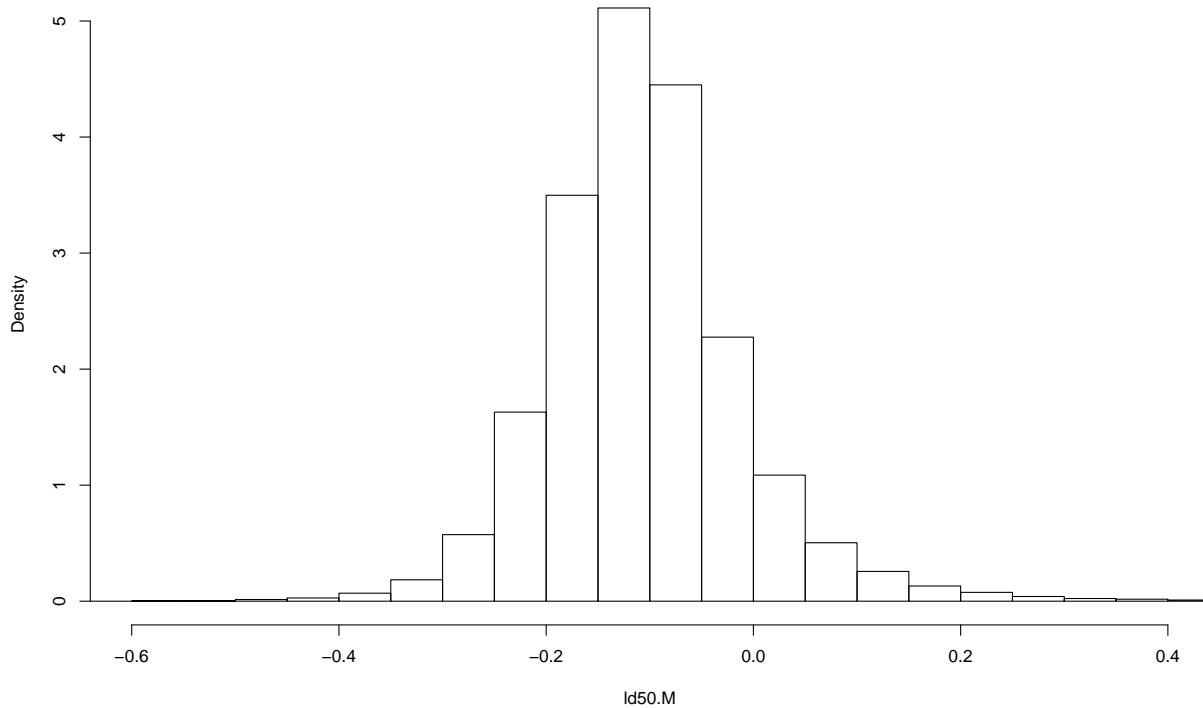
posterior_contour(-2, 4, 2001, -5, 25, 2001)
points(sims, xlim = c(-2, 4), ylim = c(-5, 25), pch='.', col="red")

```



```
# Here's the LD50 plot  
hist(ld50.M, breaks=100, freq=F, xlim=c(-0.6, 0.4))
```

Histogram of Id50.M

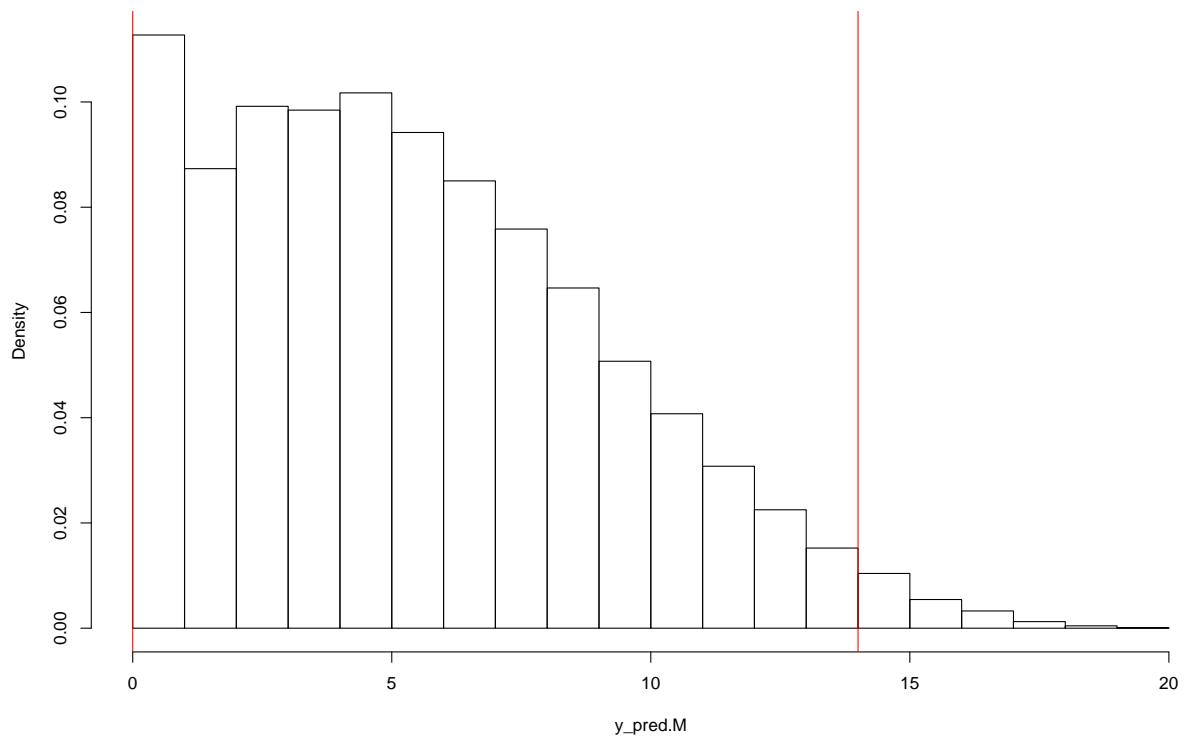


```
# Here's the predictive results for y given 20 animals and an x_i of -0.2
hist(y_pred.M, breaks=20, freq=F)
ci <- quantile(y_pred.M, c(0.025, 0.975))
ci
```

```
## 2.5% 97.5%
##      0     14
```

```
abline(v=ci, col="red")
```

Histogram of y_pred.M



■

(b)

```
fileName <- "Assignment_9_3_0"

# Read in the rat data - You may need to specify the full path to the file.
ratdata <- read.table("rats.txt", header=TRUE)

# Log Posterior (u, v space)
log.post2 <- function(u, v) {

  deaths <- ratdata$y
  rats <- ratdata$N
  alpha <- exp(u + v) / (1 + exp(u))
  beta <- exp(v) / (1 + exp(u))
  ldens <- 0

  # Loop over each of the 71 experiments
  for(i in 1:length(rats)) {

    # There is no gamma function in R - have to use Log Gamma, so the density is logged
    # This is why it's additive rather than multiplicative
    # deaths[i] is the same as y_i
    # rats[i] is the same as n_i
    ldens <- (ldens
      + (lgamma(alpha + beta) + lgamma(alpha + deaths[i]) + lgamma(beta + rats[i] - deaths[i]))
      - (lgamma(alpha) + lgamma(beta) + lgamma(alpha + beta + rats[i])))
  }

  # Return the final posterior density, which is still in logged form
  ldens - 5 / 2 * log(alpha + beta) + log(alpha) + log(beta)
}

# Just defines the size of each contour: 0.05, 0.15, 0.25, ..., 0.95
contours <- seq(0.05, 0.95, 0.1)

# Do the same steps as above, but refine the grid space
# Also, I changed the length to 200, because 2001 was slowing my computer to a crawl
u2 <- seq(-2.3, -1.3, length = 200)
v2 <- seq(1, 5, length = 200)
logdens2 <- outer(u2, v2, log.post2)
# dens2 is a 200x200 matrix of probabilities for the u2, v2 values of alpha and beta
# For instance, u2[1] = -2.3, and v2[1] = 1. dens2[1, 1] = 1.978023e-14
# This is equivalent to saying p(alpha = -2.3, beta = 1) = 1.978023e-14
dens2 <- exp(logdens2 - max(logdens2))

modelString ="
model {

  for (j in 1:count) {
    y[j] ~ dbin(theta[j], N[j])
    theta[j] ~ dbeta(alpha, beta)
}
```

```

}

y72 ~ dbin(theta72, 30)
theta72 ~ dbeta(alpha, beta)

lnx <- log(alpha / beta)
lny <- log(alpha + beta)

alpha <- u / pow(v, 2)
beta <- (1 - u) / pow(v, 2)

u ~ dunif(0.09, 0.22)
v ~ dunif(0.08, 0.61)
}

"

writeLines(modelString, con="rats.jags")

#ratdata$theta.bar <- ratdata$y / ratdata$N

ratsModel = jags.model(file="rats.jags",
                        data=list(y=ratdata$y,
                                  N=ratdata$N,
                                  count=length(ratdata$N)),
                        n.chains=4)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 71
##   Unobserved stochastic nodes: 75
##   Total graph size: 234
##
## Initializing model

update(ratsModel, n.iter=2000)

ratsSamples <- coda.samples(ratsModel, n.iter=50000, variable.names=c("alpha", "beta", "theta", "y", "lnx"))

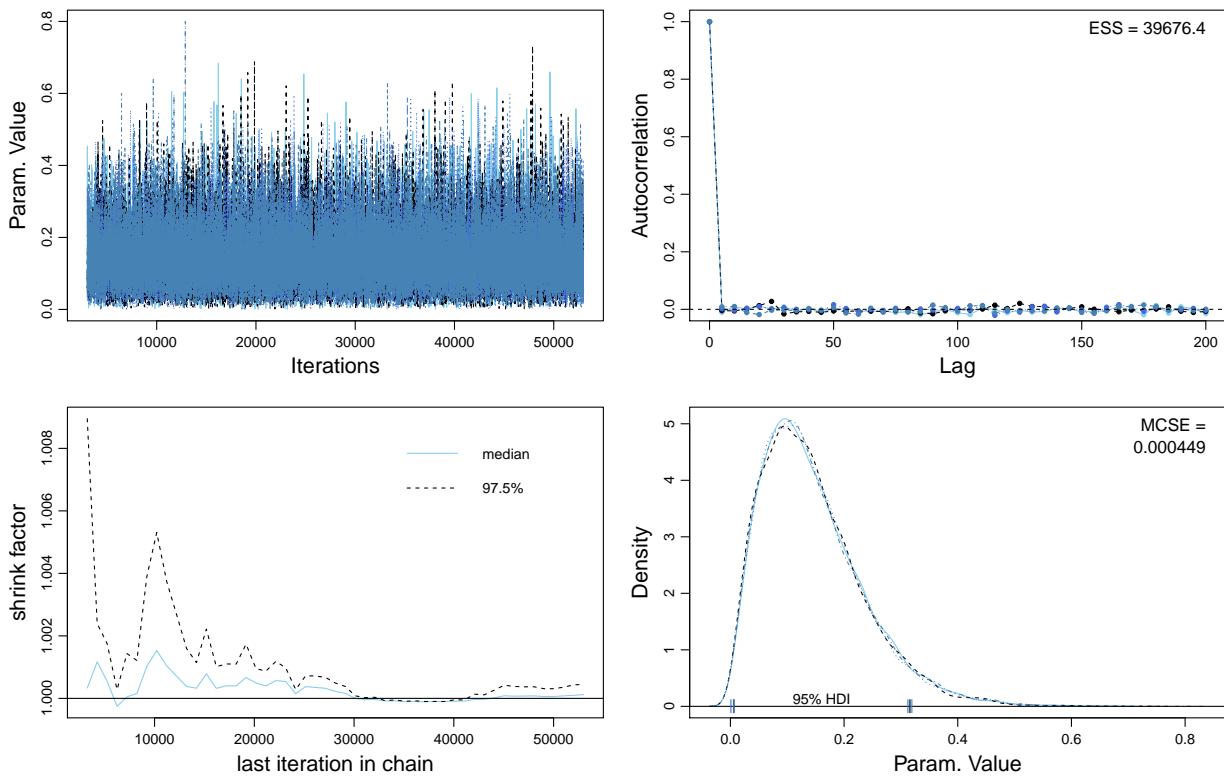
ratsSamples.M <- as.matrix(ratsSamples)

diagMCMC(codaObject = ratsSamples, parName="alpha")
diagMCMC(codaObject = ratsSamples, parName="beta")
diagMCMC(codaObject = ratsSamples, parName="y72")
diagMCMC(codaObject = ratsSamples, parName="theta72")

hist(ratsSamples.M[, "lnx"], breaks=50, freq=F)

```

theta72



```

hist(ratsSamples.M[, "beta"], breaks=50, freq=F)

contour(u2, v2, dens2, levels = contours, drawlabels = FALSE, xlim=c(-2.2, -1), ylim=c(1, 5))
points(ratsSamples.M[, "lnx"], ratsSamples.M[, "lny"], col="red", pch=".")

# 9.3 b.) Simulating a 72nd experiment:
# 9.3 c.) Experiment will contain 30 rats

summary(ratsSamples.M[, "theta72"])

##      Min.   1st Qu.    Median     Mean   3rd Qu.   Max.
## 0.0002496 0.0781500 0.1283000 0.1443000 0.1936000 0.7994000

summary(ratsSamples.M[, "y72"])

##      Min.   1st Qu.    Median     Mean   3rd Qu.   Max.
## 0.000   2.000   4.000   4.323   6.000  23.000

CI <- quantile(ratsSamples.M[, "y72"], probs = c(0.025, 0.975))
CI

## 2.5% 97.5%
## 0     12

```

```
hist(ratsSamples.M[, "y72"], breaks=30, freq=F)
abline(v=CI, col="red")
```

