

# drugTrial3

*Dr. J*

*September 13, 2016*

## JAGS run: drug trial stuff

first load the libraries, and then on with the analysis

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#setwd("U:\\Courses\\Workshops\\DOD 2016\\Examples for Workshop\\SBP Example")
```

```
#####  
#### Create Initial Values as a funtion #####  
#####
```

```
jags.inits<-list(list("mu"=50,"sd"=20),list("mu"=100,"sd"=3))
```

```
#####  
#### Specify the parameters to be monitored #####  
#####
```

```
jags.params<-c("mu","sd","tau","sbp19")
```

```
#####  
#### Another way to define the data #####  
#####
```

```
dat<-list(N=19,sbp=c(121,94,119,122,142,168,116,172,155,107,180,119,157,  
101,145,148,120,147,NA))
```

Put the BUGS model right in the code:

```
#####  
#### put the BUGS model in the code #####  
#####
```

```
cat(`  
  model  
  {  
  
    #likelihood
```

```

    for(i in 1:N)
    {
        sbp[i]~dnorm(mu,tau) #note this uses precision not variance
    }

#priors

    mu~dnorm(120,0.0001)      #diffuse prior on mean
    sd~dunif(0,500)          #diffuse prior on sd
    tau<-1/(sd*sd)           #compute precision from sd
#vari<-1/tau

#missing data value

    sbp19~dnorm(mu,tau)

}',
fill =TRUE,
file = "spb.bugs.missing.txt")

```

Run the JAGS code, print the summary results:

```

#####
#### Save and display MCMC results #####
#####
jagsfit<-jags(data=dat,
  inits = jags.inits,
  parameters.to.save=jags.params,
  model.file="spb.bugs.missing.txt",
  n.chains=2,
  n.iter=10000,
  n.burnin=1000,
  n.thin=1)

## module glm loaded

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 18
##   Unobserved stochastic nodes: 4
##   Total graph size: 31
##
## Initializing model

jagsfit

## Inference for Bugs model at "spb.bugs.missing.txt", fit using jags,
## 2 chains, each with 10000 iterations (first 1000 discarded)
## n.sims = 18000 iterations saved
##          mu.vect sd.vect   2.5%    25%    50%    75%   97.5%  Rhat
## mu       135.054   6.548 122.071 130.839 135.031 139.221 148.110 1.001
## sbp19     134.939  28.776  77.544 116.517 134.985 153.380 192.121 1.001

```

```
## sd      27.437  5.201  19.544  23.739  26.699  30.299  39.920  1.001
## tau     0.001  0.001  0.001  0.001  0.001  0.002  0.003  1.001
## deviance 168.659  2.257 166.469 167.047 167.952 169.534 174.811 1.001
##          n.eff
## mu      18000
## sbp19   18000
## sd      13000
## tau     13000
## deviance 7600
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 2.5 and DIC = 171.2
## DIC is an estimate of expected predictive error (lower deviance is better).
```

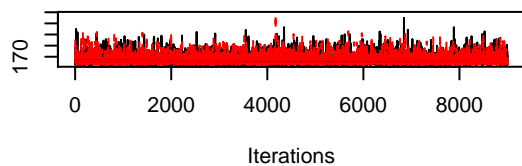
## Diagnostics!

```
## save output as an mcmc object

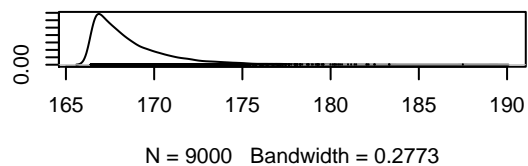
jagsfit.mcmc<-as.mcmc(jagsfit)
#####
#### Plots of the MCMC Results #####
#####

plot(jagsfit.mcmc)
```

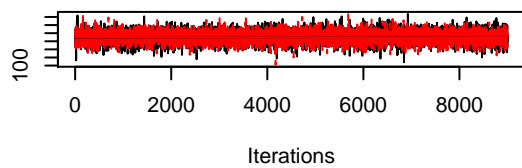
**Trace of deviance**



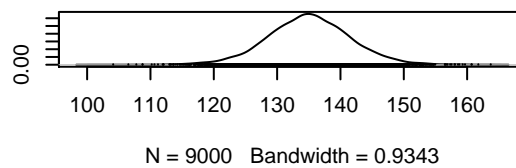
**Density of deviance**



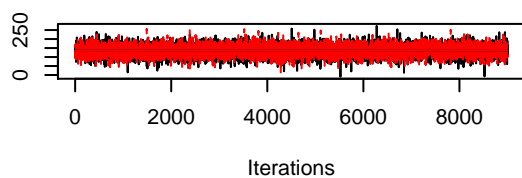
**Trace of mu**



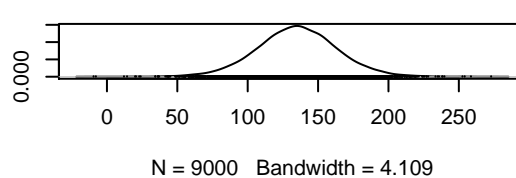
**Density of mu**

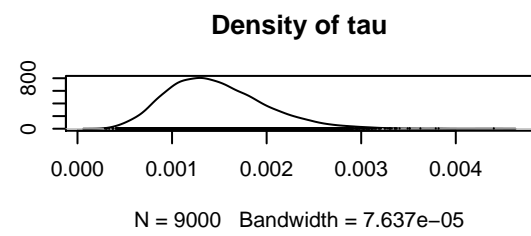
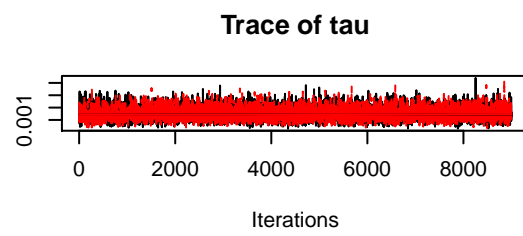
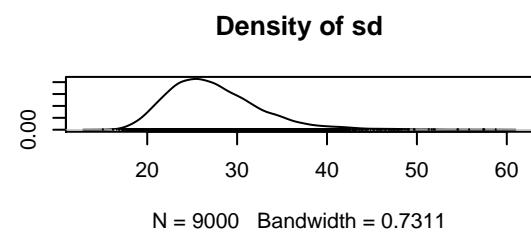
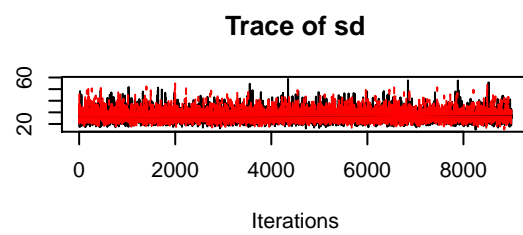


**Trace of sbp19**

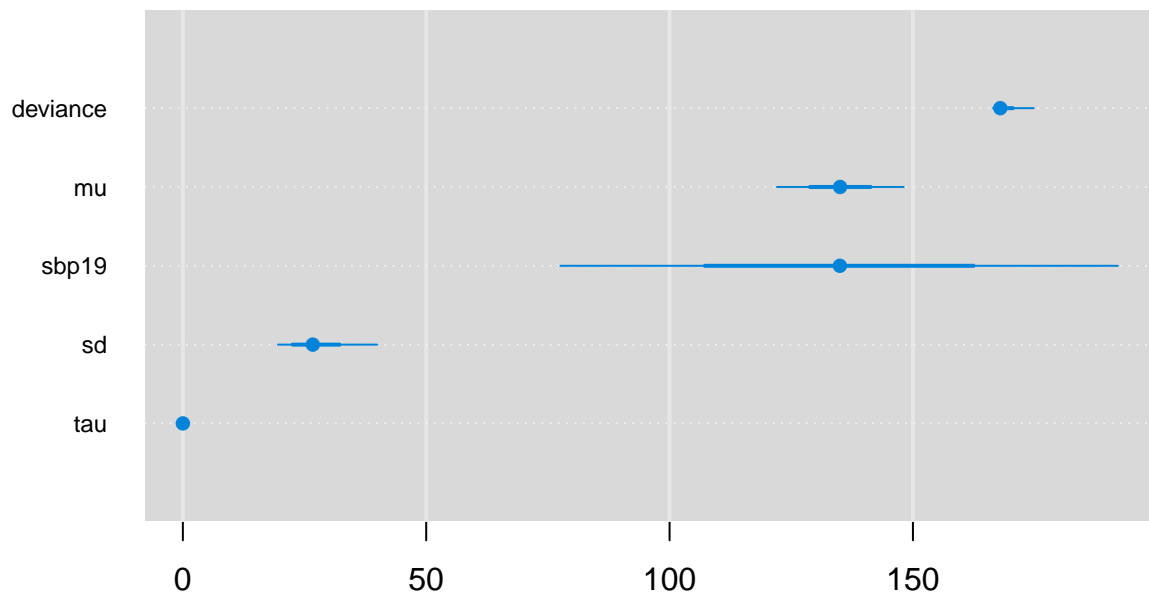


**Density of sbp19**



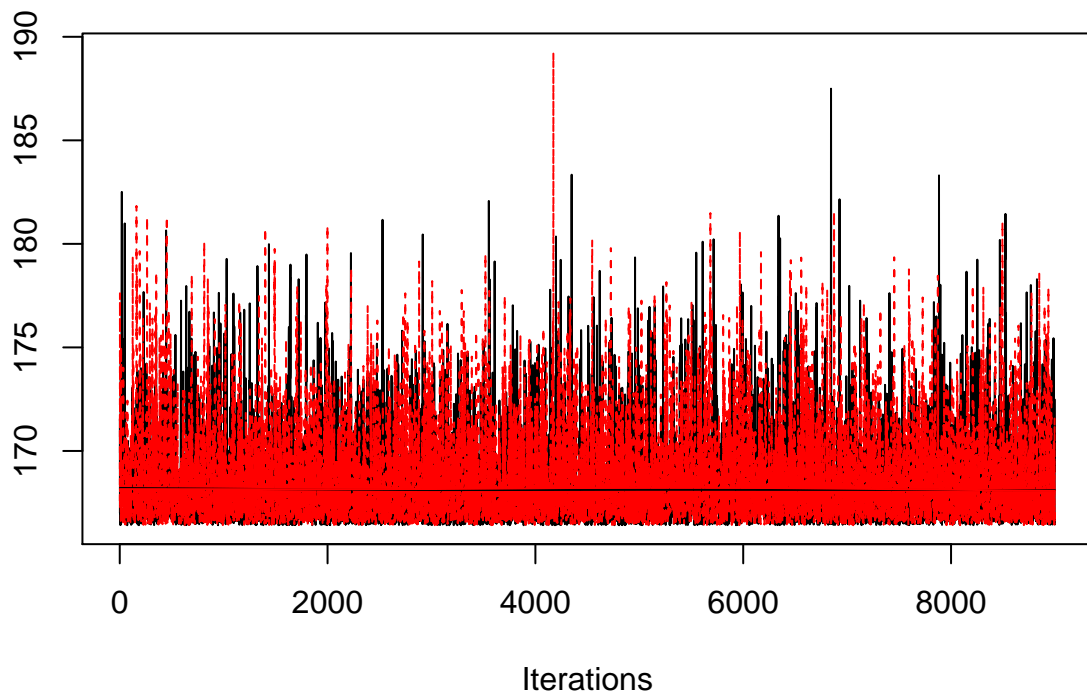


```
caterplot(jagsfit.mcmc)
```

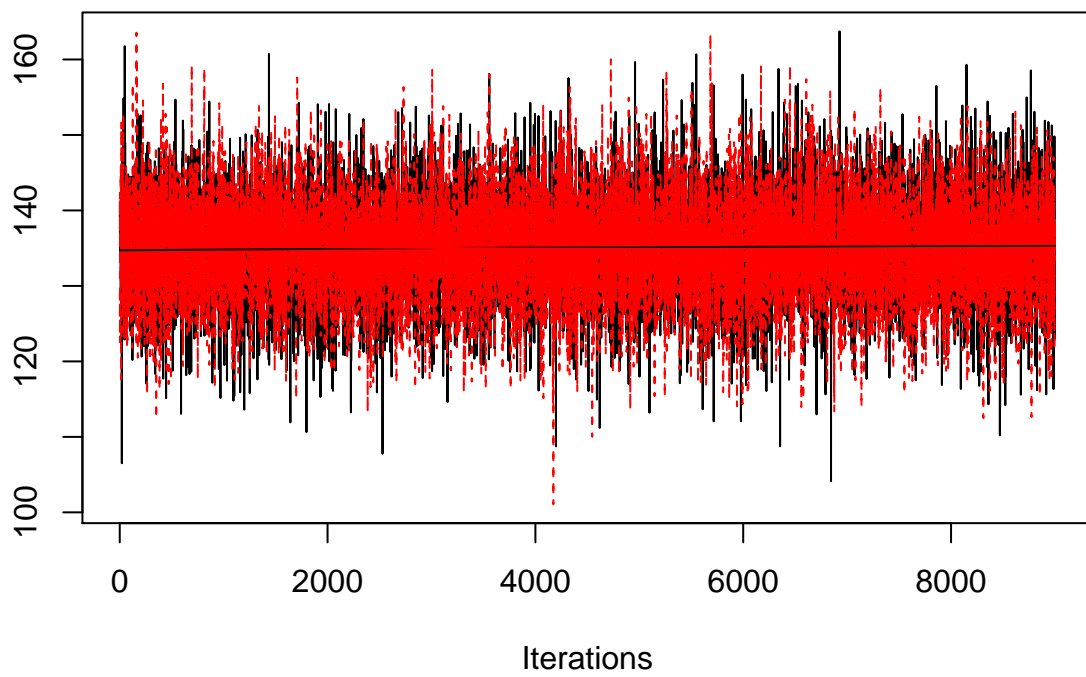


```
traceplot(jagsfit.mcmc)    #need to click on the output window to cycle through the plots
```

**Trace of deviance**

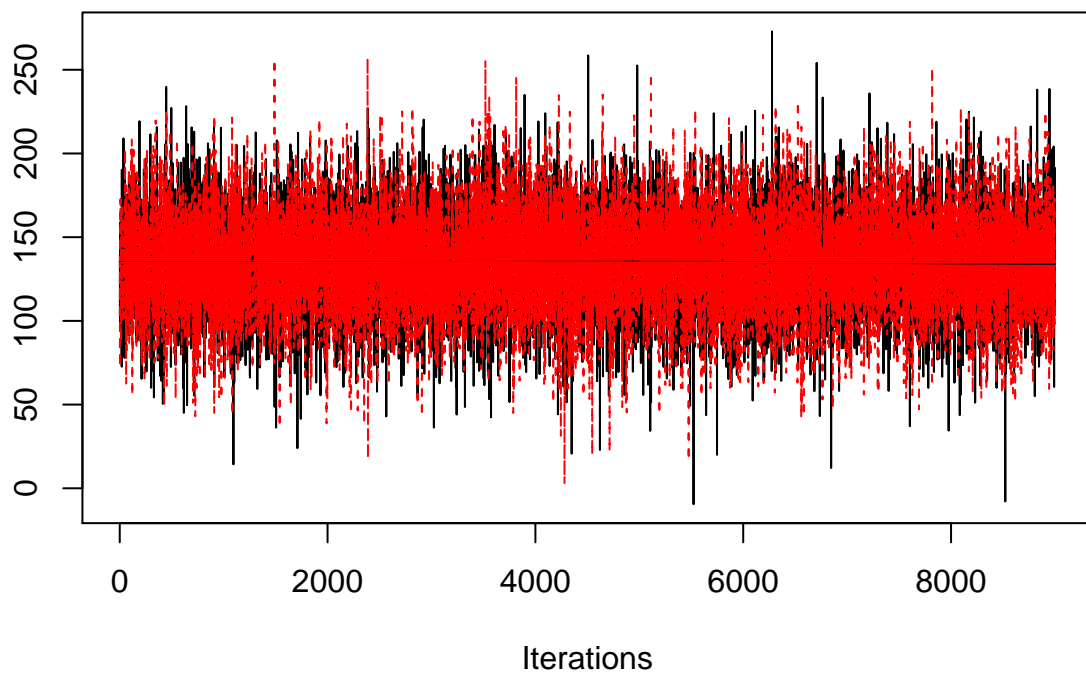


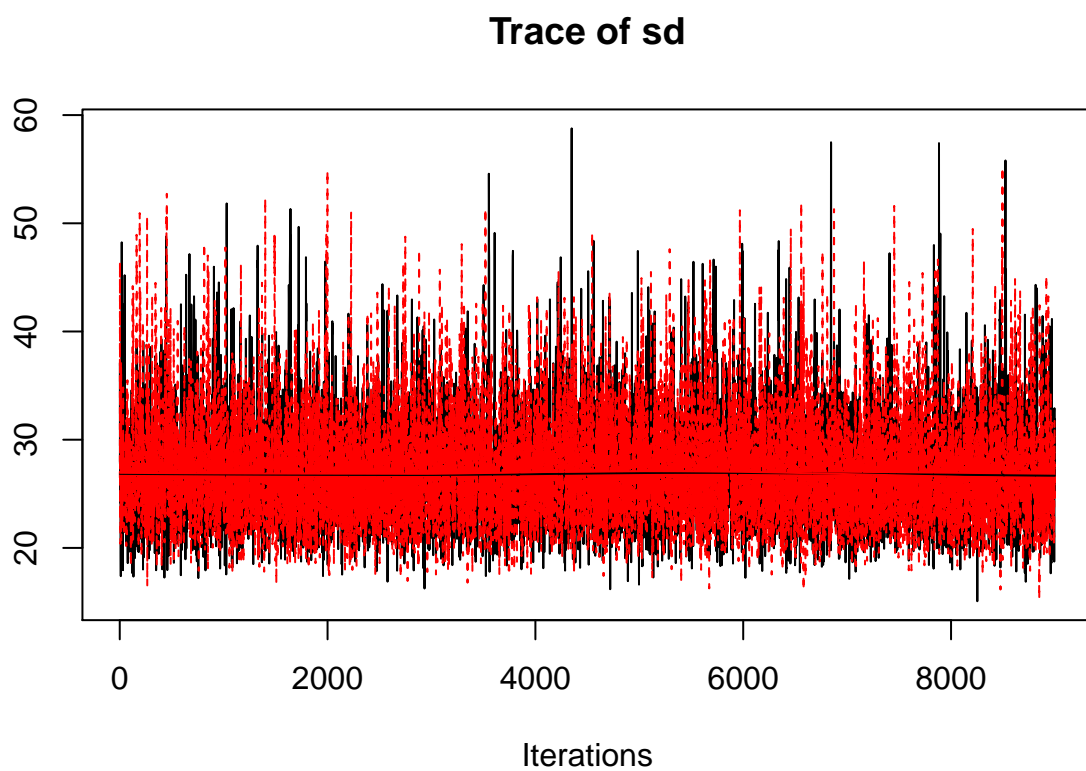
**Trace of mu**

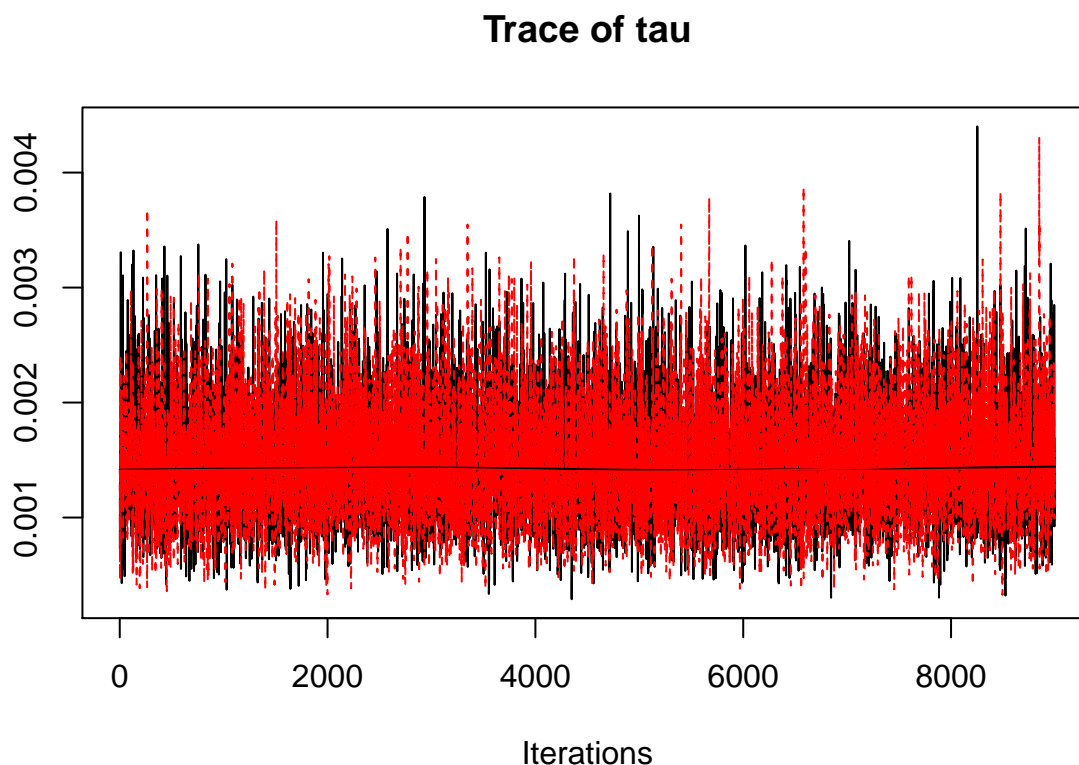




**Trace of sbp19**







And a whole lot of diagnostics!

```
raftery.diag(jagsfit.mcmc)
```

```
## [[1]]
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
##      Burn-in  Total Lower bound  Dependence
##      (M)      (N)   (Nmin)      factor (I)
## deviance  2      3797   3746      1.01
## mu        3      4197   3746      1.12
## sbp19     2      3973   3746      1.06
## sd        4      5142   3746      1.37
## tau       9      9411   3746      2.51
##
##
## [[2]]
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
```

```
##          Burn-in  Total Lower bound  Dependence
##          (M)      (N)   (Nmin)      factor (I)
## deviance 2        3865  3746         1.030
## mu       2        3973  3746         1.060
## sbp19    2        3660  3746         0.977
## sd       4        4818  3746         1.290
## tau      7        8085  3746         2.160
```

```
geweke.diag(jagsfit.mcmc)
```

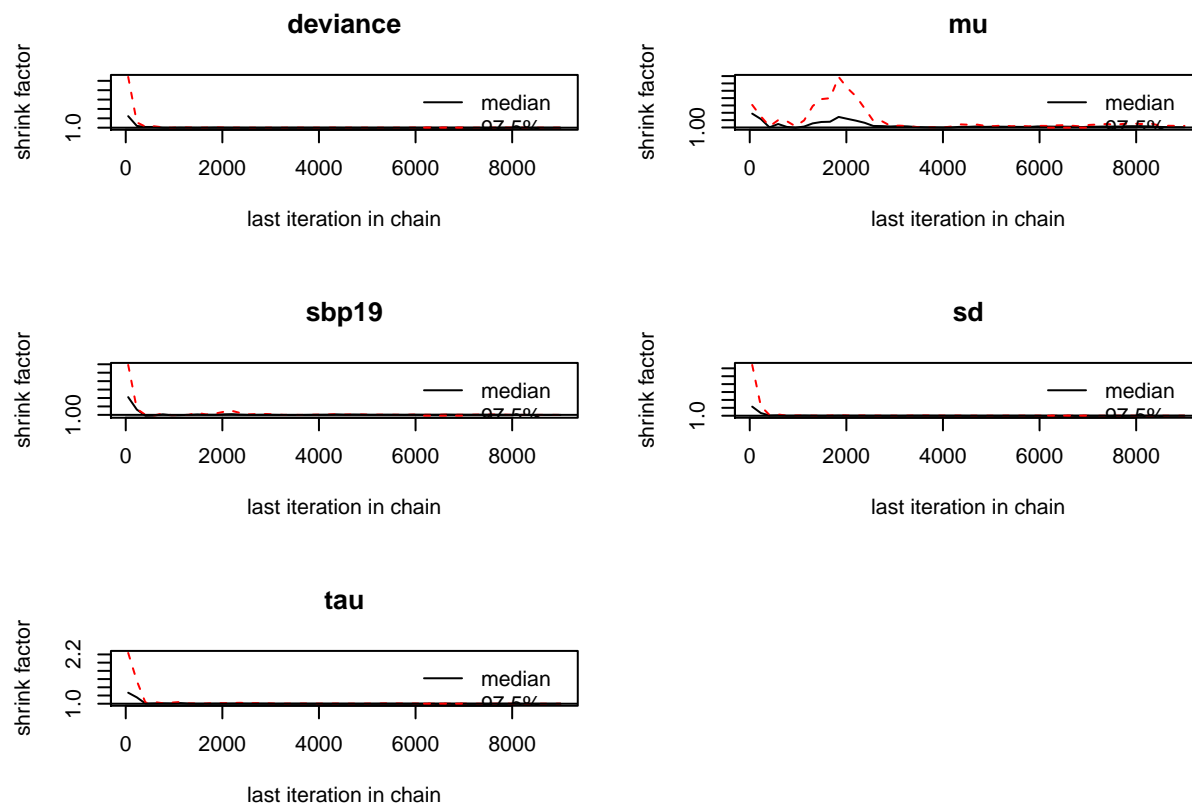
```
## [[1]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## deviance      mu    sbp19      sd      tau
##  1.1677 -1.0992  0.8243  0.1776  0.4964
##
##
```

```
## [[2]]
##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## deviance      mu    sbp19      sd      tau
##  1.0850  0.8718 -0.2282  1.9952 -2.5473
```

```
gelman.diag(jagsfit.mcmc)
```

```
## Potential scale reduction factors:
##
##          Point est. Upper C.I.
## deviance      1          1
## mu            1          1
## sbp19         1          1
## sd            1          1
## tau           1          1
##
## Multivariate psrf
##
## 1
```

```
gelman.plot(jagsfit.mcmc)
```

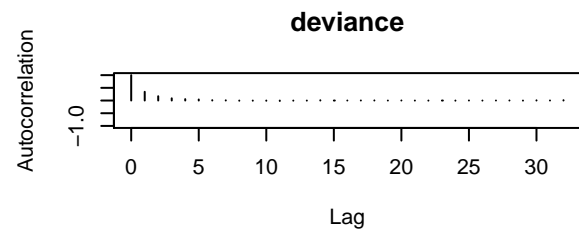
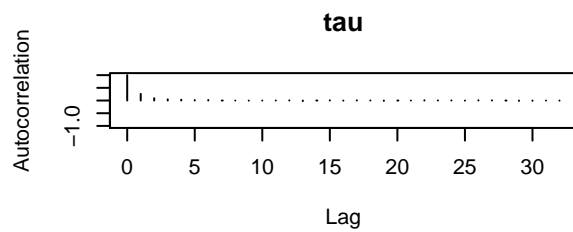
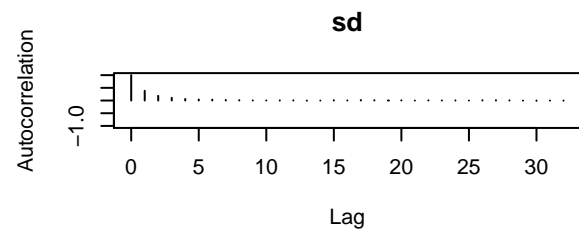
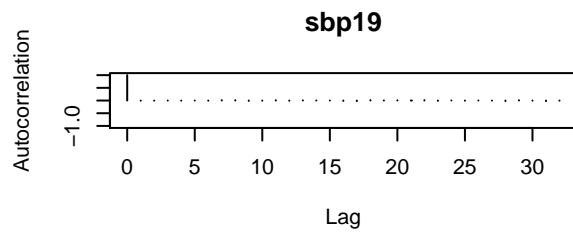
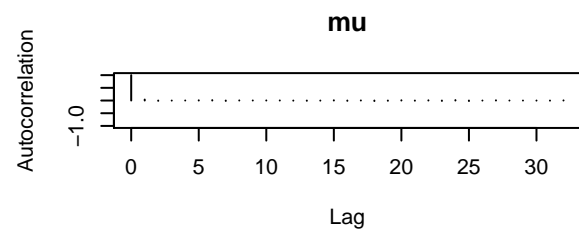
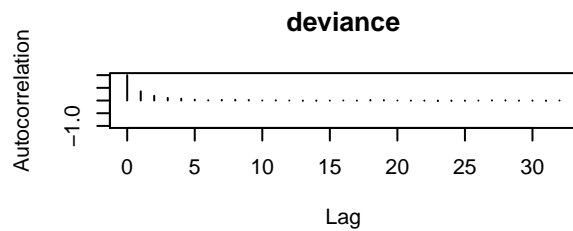


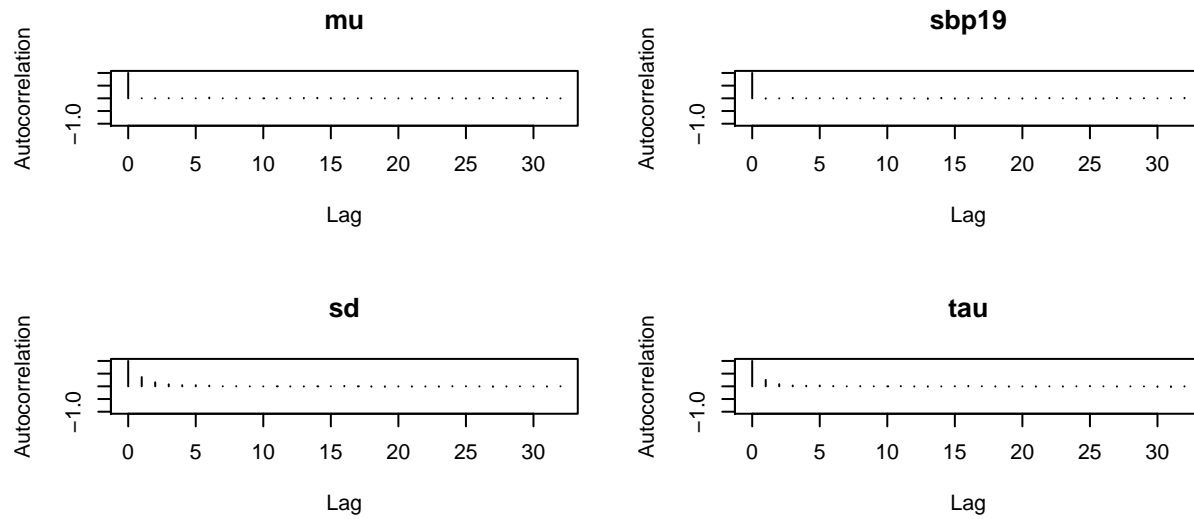
```
heidel.diag(jagsfit.mcmc)
```

```
## [[1]]
##
##      Stationarity start    p-value
##      test      iteration
## deviance passed         1    0.372
## mu      passed      1801    0.260
## sbp19   passed         1    0.805
## sd      passed         1    0.477
## tau     passed         1    0.533
##
##      Halfwidth Mean      Halfwidth
##      test
## deviance passed    1.69e+02 7.55e-02
## mu      passed    1.35e+02 1.53e-01
## sbp19   passed    1.35e+02 5.90e-01
## sd      passed    2.74e+01 1.71e-01
## tau     passed    1.47e-03 1.44e-05
##
## [[2]]
##
##      Stationarity start    p-value
##      test      iteration
## deviance passed         1    0.751
## mu      passed         1    0.270
```

```
## sbp19    passed      1      0.916
## sd       passed      1      0.243
## tau      passed      1      0.124
##
##          Halfwidth Mean      Halfwidth
##          test
## deviance passed  1.69e+02 7.24e-02
## mu        passed  1.35e+02 1.36e-01
## sbp19     passed  1.35e+02 5.99e-01
## sd        passed  2.75e+01 1.72e-01
## tau       passed  1.46e-03 1.47e-05
```

```
autocorr.plot(jagsfit.mcmc)
```





```
effectiveSize(jagsfit.mcmc)
```

```
## deviance      mu      sbp19      sd      tau
## 7172.268 17334.570 18000.000 7068.630 9546.257
```

```
jagsfit<-update(jagsfit,10000)
jagsfit
```

```
## Inference for Bugs model at "spb.bugs.missing.txt", fit using jags,
## 2 chains, each with 10000 iterations (first 0 discarded)
## n.sims = 20000 iterations saved
##      mu.vect sd.vect  2.5%   25%   50%   75%  97.5% Rhat
## mu      135.175   6.564 122.289 130.927 135.109 139.493 148.093 1.001
## sbp19    135.010  28.636  78.455 116.651 135.004 153.600 191.307 1.001
## sd       27.436   5.197  19.500  23.728  26.724  30.319  39.588 1.001
## tau       0.001   0.001   0.001   0.001   0.001   0.002   0.003 1.001
## deviance 168.675   2.224 166.474 167.078 168.006 169.575 174.583 1.001
##      n.eff
## mu      20000
## sbp19    20000
## sd       20000
## tau      20000
## deviance 3800
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
```

```
## DIC info (using the rule,  $pD = \text{var}(\text{deviance})/2$ )  
##  $pD = 2.5$  and  $DIC = 171.1$   
## DIC is an estimate of expected predictive error (lower deviance is better).
```