HW3 Tardella - final project

Valerio Rossini

Rats: a normal hierarchical model

This example is taken from section 6 of Gelfand et al (1990), and concerns 30 young rats whose weights were measured weekly for five weeks. Part of the data is shown below, where Y_{ij} is the weight of the ith rat measured at age x_j.

```
rats.data  <- list(x = c(8.0, 15.0, 22.0, 29.0, 36.0),
                   N = 30,
                   T = 5,
                   xbar=22,
                   Y = matrix(c(151, 199, 246, 283, 320,
                                145, 199, 249, 293, 354,
                                147, 214, 263, 312, 328,
                                155, 200, 237, 272, 297,
                                135, 188, 230, 280, 323,
                                159, 210, 252, 298, 331,
                                141, 189, 231, 275, 305,
                                159, 201, 248, 297, 338,
                                177, 236, 285, 350, 376,
                                134, 182, 220, 260, 296,
                                160, 208, 261, 313, 352,
                                143, 188, 220, 273, 314,
                                154, 200, 244, 289, 325,
                                171, 221, 270, 326, 358,
                                163, 216, 242, 281, 312,
                                160, 207, 248, 288, 324,
                                142, 187, 234, 280, 316,
                                156, 203, 243, 283, 317,
                                157, 212, 259, 307, 336,
                                152, 203, 246, 286, 321,
                                154, 205, 253, 298, 334,
                                139, 190, 225, 267, 302,
                                146, 191, 229, 272, 302,
                                157, 211, 250, 285, 323,
                                132, 185, 237, 286, 331,
                                160, 207, 257, 303, 345,
                                169, 216, 261, 295, 333,
                                157, 205, 248, 289, 316,
                                137, 180, 219, 258, 291,
                                153, 200, 244, 286, 324),
                              nrow=30, ncol=5, byrow=T))

Y <- rats.data$Y
Y
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]  151  199  246  283  320
##  [2,]  145  199  249  293  354
##  [3,]  147  214  263  312  328
##  [4,]  155  200  237  272  297
##  [5,]  135  188  230  280  323
##  [6,]  159  210  252  298  331
##  [7,]  141  189  231  275  305
##  [8,]  159  201  248  297  338
##  [9,]  177  236  285  350  376
```

```
## [10,]  134  182  220  260  296
## [11,]  160  208  261  313  352
## [12,]  143  188  220  273  314
## [13,]  154  200  244  289  325
## [14,]  171  221  270  326  358
## [15,]  163  216  242  281  312
## [16,]  160  207  248  288  324
## [17,]  142  187  234  280  316
## [18,]  156  203  243  283  317
## [19,]  157  212  259  307  336
## [20,]  152  203  246  286  321
## [21,]  154  205  253  298  334
## [22,]  139  190  225  267  302
## [23,]  146  191  229  272  302
## [24,]  157  211  250  285  323
## [25,]  132  185  237  286  331
## [26,]  160  207  257  303  345
## [27,]  169  216  261  295  333
## [28,]  157  205  248  289  316
## [29,]  137  180  219  258  291
## [30,]  153  200  244  286  324
```

```
T <- rats.data$T
T
```

```
## [1] 5
```

```
x <-  rats.data$x
x
```

```
## [1]  8 15 22 29 36
```

```
xbar <- rats.data$xbar
xbar
```

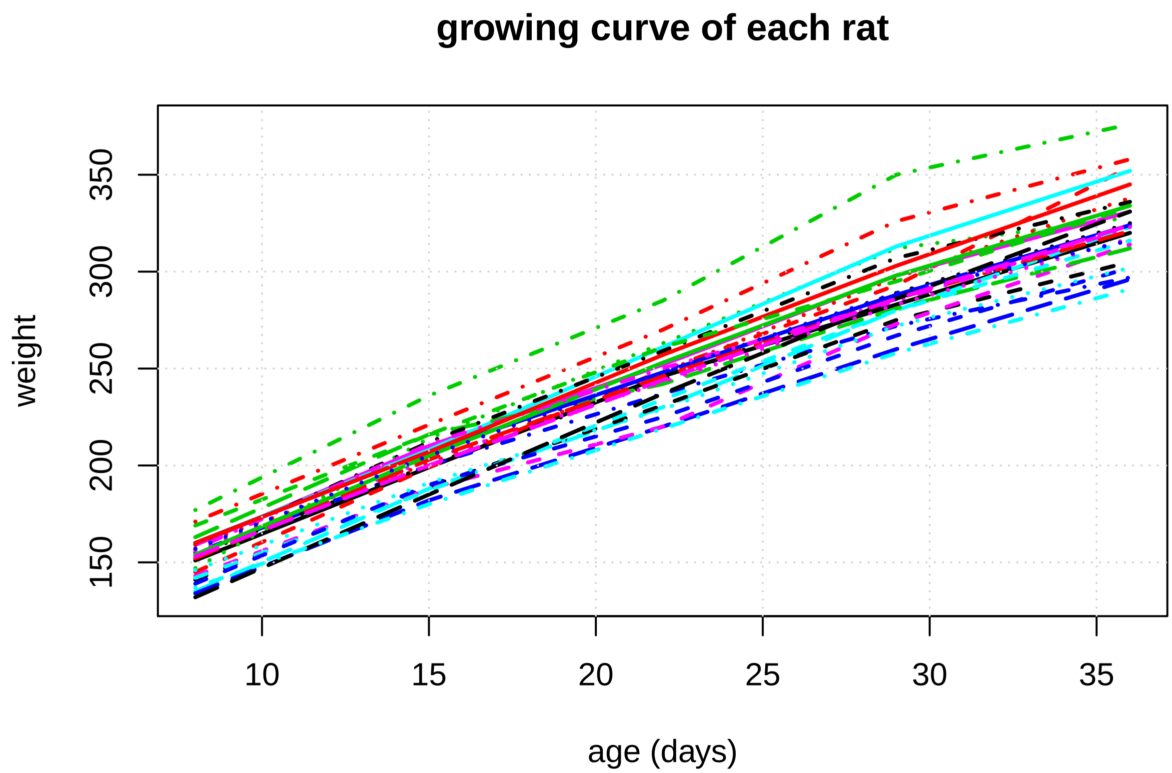
```
## [1] 22
```

```
N <- rats.data$N
N
```

```
## [1] 30
```

We can represent graphically the growing curve of each rat in a single plot:

```
matplot(x, t(Y), type = "l", lwd = 2,
        xlab = "age (days)", ylab = "weight")
title(main= "growing curve of each rat")
grid()
```



Possible analysis:

1. each rat has its own line: intercept=b_{i0}, slope=b_{i1}
2. all rats follow the same line: $b_{i0} = \beta_0$, $b_{i1} = \beta_1$
3. a compromise between these two: each rat has its own line, BUT... the lines come from a common assumed distribution (a slope and intercept are estimated for each rat)

The model is essentially a random effects linear growth curve

$$Y_{ij} \sim Normal(\alpha_i + \beta_i x_j, \tau_c)$$

$$\alpha_i \sim Normal(\alpha_c, \tau_\alpha)$$

$$\beta_i \sim Normal(\beta_c, \tau_\beta)$$

where τ represents the precision (1/variance) of a normal distribution

$\alpha_c, \tau_\alpha, \beta_c, \tau_\beta, \tau_c$ are given independent "noninformative" priors. Interest particularly focuses on the intercept at zero time (birth), denoted $\alpha_0 = \alpha_c - \beta_c x_{bar}$

$$\alpha_c \sim Normal(0, 1.0E-6)$$

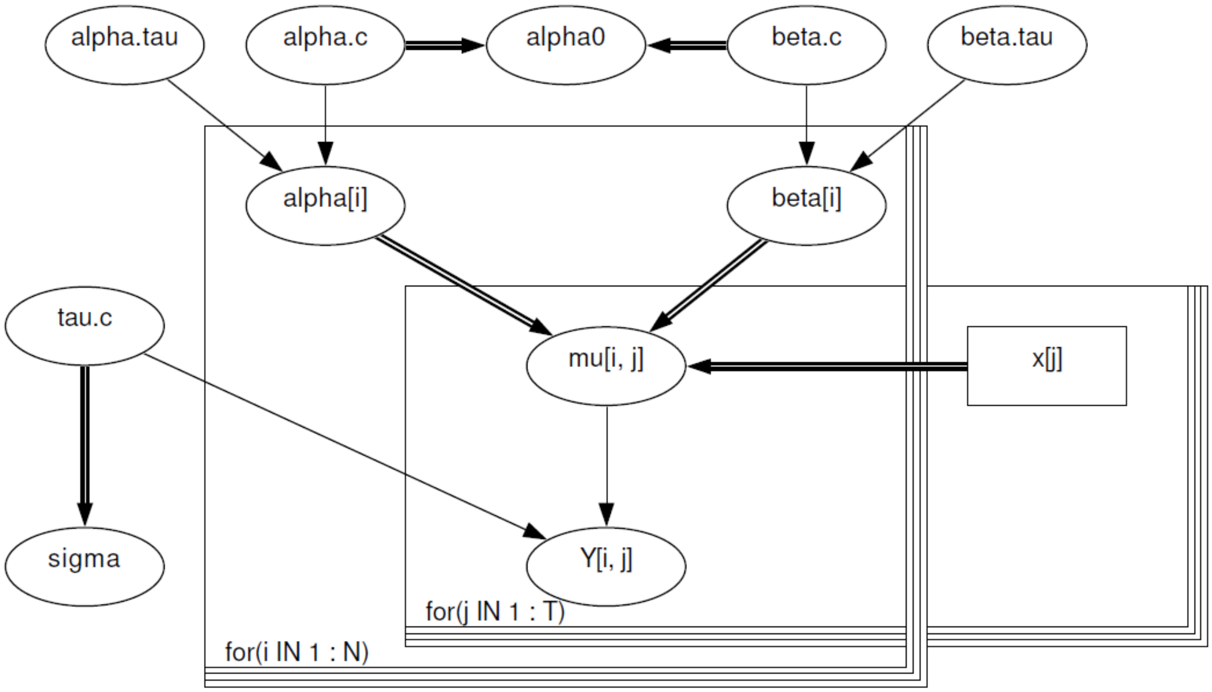
$$\tau_\alpha \sim Gamma(1.0E-3, 1.0E-3)$$

$$\beta_c \sim Normal(0, 1.0E-6)$$

$$\tau_\beta \sim Gamma(1.0E-3, 1.0E-3)$$

$$\tau_c \sim Gamma(1.0E-3, 1.0E-3)$$

The graphical model for rats example is given below:



We can also fit a linear model for each rat predicting weight Y from time x_j

```
# frequentistic approach

intercept_vec <- rep(NA,N)
slope_vec <- rep(NA,N)
for(i in 1:N)
{
  lmfit <- lm(rats.data$Y[i,] ~ rats.data$x)
  intercept_vec[i] <- lmfit$coefficients[[1]]
  slope_vec[i] <- lmfit$coefficients[[2]]
}

intercept_vec
```

```
##  [1] 107.17143  87.08571 108.22857 120.31429  84.11429 114.22857  98.08571
##  [8] 105.91429 123.88571  92.05714 105.11429  93.40000 106.94286 118.65714
## [15] 128.71429 116.85714  93.20000 114.05714 111.82857 109.28571 106.42857
## [22]  97.94286 104.48571 117.60000  77.37143 107.94286 126.88571 116.65714
## [29]  95.68571 106.88571
```

```
slope_vec
```

```
##  [1] 6.028571 7.314286 6.571429 5.085714 6.685714 6.171429 5.914286
##  [8] 6.485714 7.314286 5.742857 6.985714 6.100000 6.157143 6.842857
## [15] 5.185714 5.842857 6.300000 5.742857 6.471429 6.014286 6.471429
## [22] 5.757143 5.614286 5.800000 7.128571 6.657143 5.814286 5.742857
## [29] 5.514286 6.114286
```

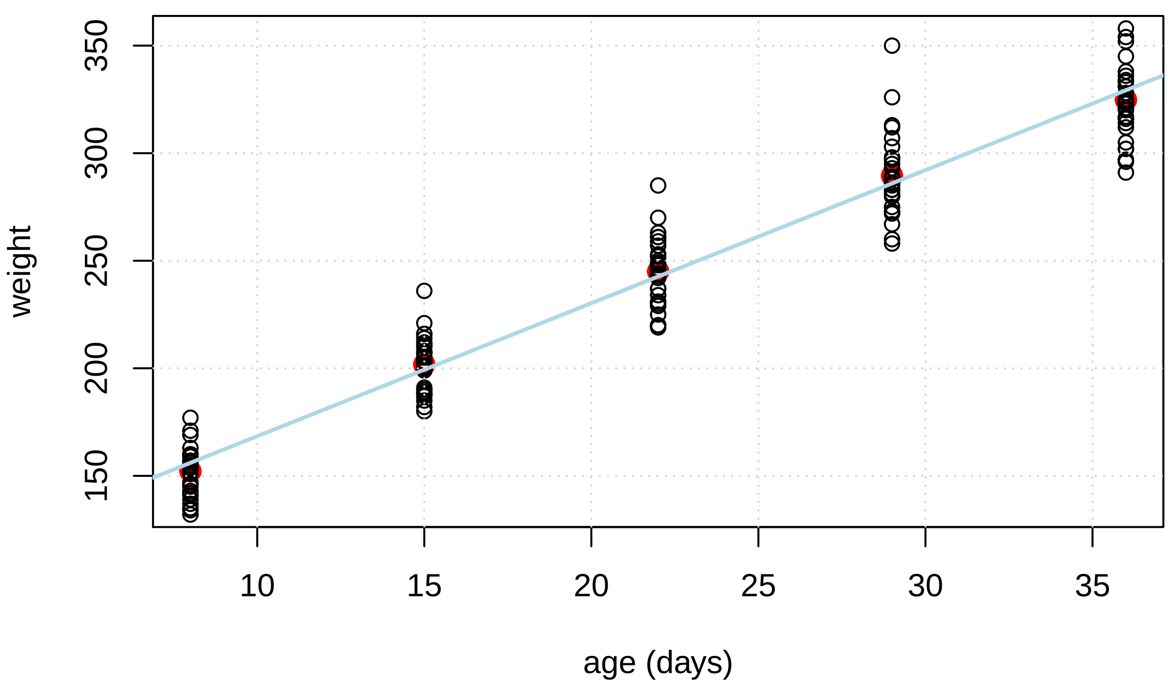
```r
# mean of intercept
mean_alpha <- mean(intercept_vec)
mean_alpha
```

```
## [1] 106.5676
```

```r
# mean of slope
mean_beta <- mean(slope_vec)
mean_beta
```

```
## [1] 6.185714
```

```r
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mean_alpha, mean_beta, col="lightblue", lwd=2)
grid()
```



The likelihood function can be derived in this way:

$$L(y_{ij}|\alpha_i,\beta_i,\tau_c,x_j) = \prod_{i=1}^{N}\prod_{j=1}^{T}\frac{1}{\sqrt{2\pi\tau_c^2}}\exp\left\{\frac{-(y_{ij}-\mu_{ij})^2}{2\tau_c^2}\right\} \propto \prod_{i=1}^{N}\prod_{j=1}^{T}\exp\left\{\frac{-(y_{ij}-\mu_{ij})^2}{2\tau_c^2}\right\}$$

$$\propto \exp\left\{-\frac{1}{2\tau_c^2}\sum_{i=1}^{N}\sum_{j=1}^{T}(y_{ij}-\mu_{ij})^2\right\} \propto \exp\left\{-\frac{1}{2\tau_c^2}\sum_{i=1}^{N}\sum_{j=1}^{T}(y_{ij}-(\alpha_i+\beta_i x_j))^2\right\}$$

Before writing the expression of the joint prior distribution of the parameters, we need to compute separately the prior/hyperprior of each parameter:

$$\pi(\alpha_i|\alpha_c,\tau_\alpha) = \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\tau_\alpha^2}}exp\left\{-\frac{(\alpha_i-\alpha_c)^2}{2\tau_\alpha^2}\right\} \propto \prod_{i=1}^{N}exp\left\{-\frac{(\alpha_i-\alpha_c)^2}{2\tau_\alpha^2}\right\} \propto \exp\left\{-\frac{1}{2\tau_\alpha^2}\sum_{i=1}^{N}(\alpha_i-\alpha_c)^2\right\}$$

$$\pi(\alpha_c) = \frac{1}{\sqrt{2\pi(1.0E-6)^2}}exp\left\{-\frac{(\alpha_c-0)^2}{2(1.0E-6)^2}\right\} \propto exp\left\{-\frac{(\alpha_c-0)^2}{2(1.0E-6)^2}\right\} \propto exp\left\{-\frac{\alpha_c^2}{2(1.0E-6)^2}\right\}$$

$$\pi(\tau_\alpha) = \frac{(1.0E-3)^{1.0E-3}\tau_\alpha^{(1.0E-3-1)}e^{-1.0E-3\tau_\alpha}}{\Gamma(1.0E-3)}$$

$$\pi(\beta_i|\beta_c,\tau_\beta) = \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\tau_\beta^2}}exp\left\{-\frac{(\beta_i-\beta_c)^2}{2\tau_\beta^2}\right\} \propto \prod_{i=1}^{N}exp\left\{-\frac{(\beta_i-\beta_c)^2}{2\tau_\beta^2}\right\} \propto \exp\left\{-\frac{1}{2\tau_\beta^2}\sum_{i=1}^{N}(\beta_i-\beta_c)^2\right\}$$

$$\pi(\beta_c) = \frac{1}{\sqrt{2\pi(1.0E-6)^2}}exp\left\{-\frac{(\beta_c-0)^2}{2(1.0E-6)^2}\right\} \propto exp\left\{-\frac{(\beta_c-0)^2}{2(1.0E-6)^2}\right\} \propto exp\left\{-\frac{\beta_c^2}{2(1.0E-6)^2}\right\}$$

$$\pi(\tau_\beta) = \frac{(1.0E-3)^{1.0E-3}\tau_\beta^{(1.0E-3-1)}e^{-1.0E-3\tau_\beta}}{\Gamma(1.0E-3)}$$

$$\pi(\tau_c) = \frac{(1.0E-3)^{1.0E-3}\tau_c^{(1.0E-3-1)}e^{-1.0E-3\tau_c}}{\Gamma(1.0E-3)}$$

```r
# load the library

library(R2jags, quietly = T)
```

```
## Warning: package 'R2jags' was built under R version 3.3.3

## Warning: package 'rjags' was built under R version 3.3.3

## Warning: package 'coda' was built under R version 3.3.3

## Linked to JAGS 4.2.0

## Loaded modules: basemod,bugs

##
## Attaching package: 'R2jags'

## The following object is masked from 'package:coda':
##
##     traceplot
```

```r
library(mcmcplots, quietly = T)
```

```
## Warning: package 'mcmcplots' was built under R version 3.3.3
```

```r
library(ggmcmc, quietly = T)
```

```
## Warning: package 'ggmcmc' was built under R version 3.3.3

## Warning: package 'dplyr' was built under R version 3.3.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Warning: package 'tidyr' was built under R version 3.3.3

## Warning: package 'ggplot2' was built under R version 3.3.3
```

```r
library(corrplot, quietly = T)
```

```
## Warning: package 'corrplot' was built under R version 3.3.3
```

First model

In **Examples Volume 1 - Rats a normal hierarchical model**, we read: *"for now, we standardise the x_j's around their mean to reduce dependence between α_i and β_i in their likelihood: in fact for the full balanced data, complete independence is achieved. (Note that, in general, prior independence does not force the posterior distributions to be independent)."*

In this first model (and also in the other models) we don't standardise the x_j's around their mean and we use the priors that we have introduced before, i.e.:

$$\alpha_i \sim Normal(\alpha_c, \tau_\alpha)$$

$$\beta_i \sim Normal(\beta_c, \tau_\beta)$$

$$\alpha_c \sim Normal(0, 1.0E-6)$$

$$\tau_\alpha \sim Gamma(1.0E-3, 1.0E-3)$$

$$\beta_c \sim Normal(0, 1.0E-6)$$

$$\tau_\beta \sim Gamma(1.0E-3, 1.0E-3)$$

$$\tau_c \sim Gamma(1.0E-3, 1.0E-3)$$

```r
# first model with prior 2 (tau.alpha and tau.beta are distributed as inverse gamma)
# and not centered variables

model <- function()
{
  for (i in 1:N)
  {
    for (j in 1:T)
    {
      Y[i,j] ~ dnorm(mu[i,j], tau.c)
      mu[i, j] <- alpha[i] + beta[i] * (x[j])
    }
    alpha[i] ~ dnorm(alpha.c, tau.alpha)
    beta[i] ~ dnorm(beta.c, tau.beta)
  }
    alpha.c ~ dnorm(0, 1.0E-6)
    beta.c ~ dnorm(0, 1.0E-6)
    tau.c ~ dgamma(1.0E-3, 1.0E-3)
    tau.alpha ~ dgamma(1.0E-3, 1.0E-3)
    tau.beta ~ dgamma(1.0E-3, 1.0E-3)
    sigma.c <- 1.0/sqrt(tau.c)
    xbar <- mean(x[])
    alpha0 <- alpha.c - beta.c*xbar
}

## Read in the rats data for JAGS
rats.data.list  <- list("Y", "x", "T", "N")

## Name the JAGS parameters
rats.params <- c("tau.c", "alpha.c", "beta.c", "tau.alpha", "tau.beta")

## Define the starting values for JAGS
rats.inits <- function(){
```

```r
  list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250,
                 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250),
       beta  = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
                 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
       alpha.c = 150, beta.c = 10,
       tau.c = 1, tau.alpha = 1, tau.beta = 1)
}

mod1 <- jags(data=rats.data.list, inits=rats.inits, rats.params, n.chains=2, n.iter=10000,
             n.burnin=1000, n.thin = 1, model.file=model, DIC=TRUE)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 150
##    Unobserved stochastic nodes: 65
##    Total graph size: 540
##
## Initializing model
```

```r
mod1
```

```
## Inference for Bugs model at "C:/Users/valer/AppData/Local/Temp/RtmpERaEBF/model17282cc91afb.txt", fi
##  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
## alpha.c   106.610   2.340 102.098 105.066 106.601 108.123  111.191 1.001
## beta.c      6.183   0.108   5.971   6.111   6.183   6.255    6.392 1.001
## tau.alpha   0.010   0.004   0.005   0.007   0.009   0.012    0.020 1.001
## tau.beta    4.323   1.489   2.068   3.269   4.103   5.138    7.803 1.001
## tau.c       0.027   0.004   0.020   0.024   0.027   0.030    0.036 1.001
## deviance  969.593  14.924 943.752 959.524 968.774 978.669 1000.740 1.001
##           n.eff
## alpha.c   18000
## beta.c    18000
## tau.alpha 18000
## tau.beta  15000
## tau.c     18000
## deviance   7100
##
## 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 = 111.4 and DIC = 1080.9
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```r
# DIC
mod1$BUGSoutput$DIC
```

```
## [1] 1080.948
```

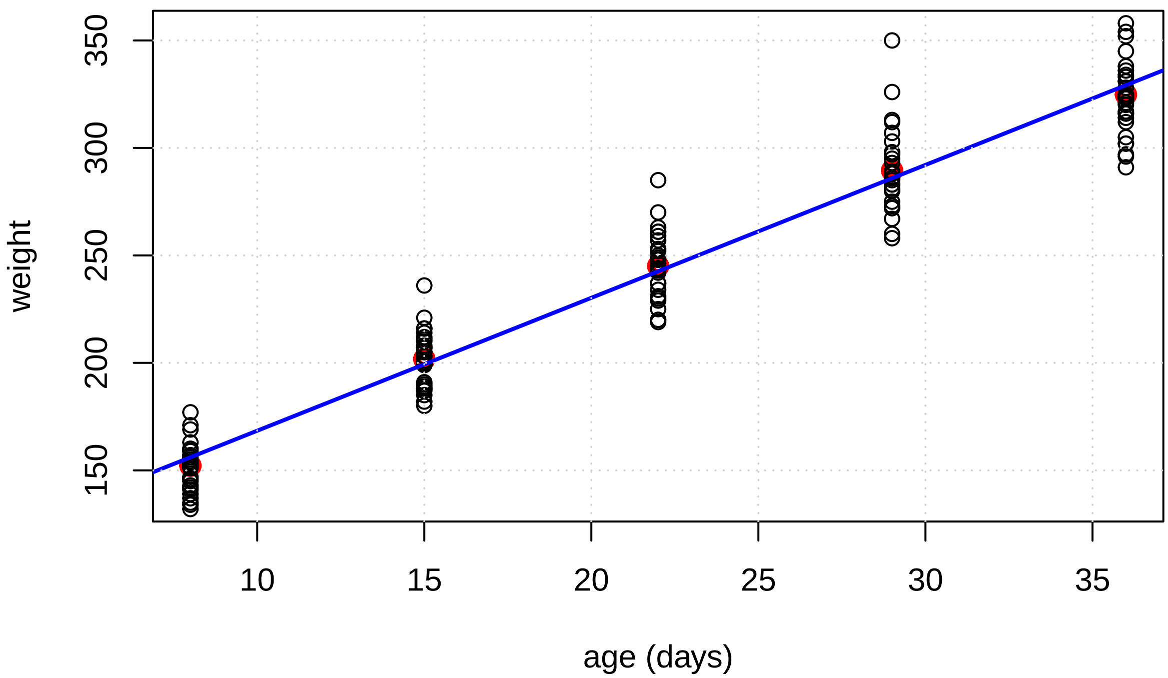
```r
# mean of the intercept
mod1$BUGSoutput$summary[,"mean"]["alpha.c"]
```

```
##  alpha.c
## 106.6098
```

```r
# mean of the slope
mod1$BUGSoutput$summary[,"mean"]["beta.c"]
```

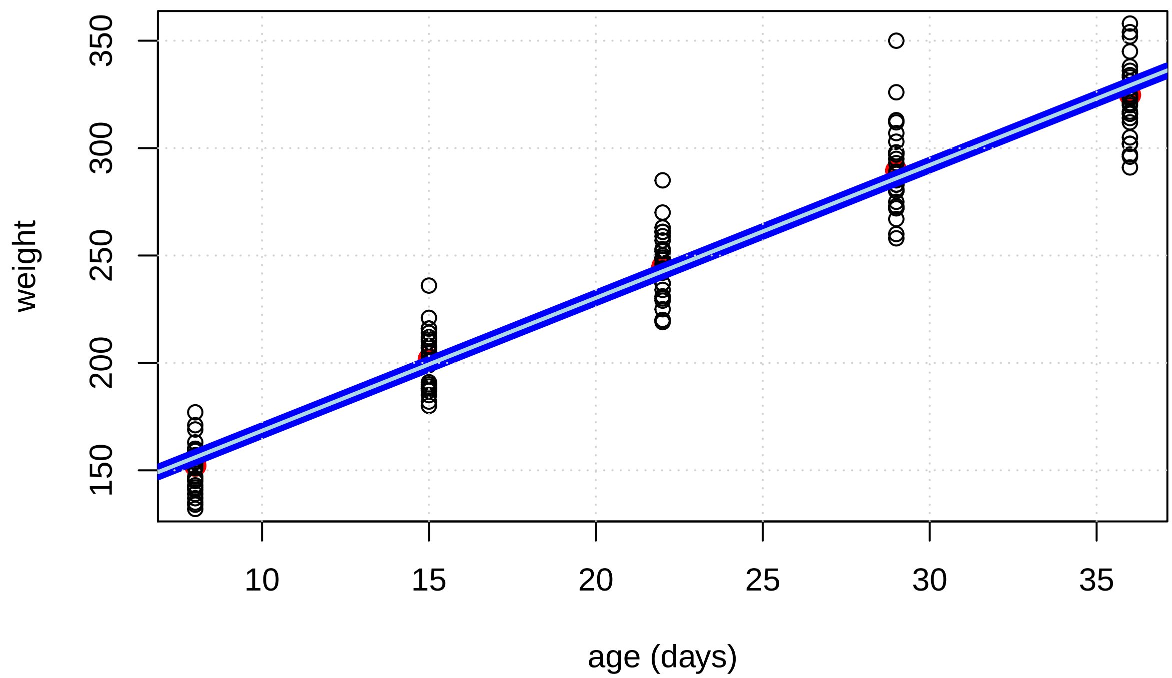
```
##   beta.c
## 6.182628
```

```r
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod1$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod1$BUGSoutput$summary[,"mean"]["beta.c"], col="blue", lwd=2)
grid()
```




```r
# comparison with frequentistic approach
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
```

```
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod1$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod1$BUGSoutput$summary[,"mean"]["beta.c"], col="blue", lwd=8)
abline(mean_alpha, mean_beta, col="lightblue", lwd=2)
grid()
```



Second model

In this second model we use other different priors for τ_α and τ_β that are suggested in **Examples Volume 1 - Rats a normal hierarchical model**. So our priors will be:

$$\alpha_i \sim Normal(\alpha_c, \tau_\alpha)$$
$$\beta_i \sim Normal(\beta_c, \tau_\beta)$$
$$\alpha_c \sim Normal(0, 0.1E-6)$$
$$\tau_\alpha \sim Unif(0, 100)$$
$$\beta_c \sim Normal(0, 1.0E-6)$$
$$\tau_\beta \sim Unif(0, 100)$$
$$\tau_c \sim Gamma(1.0E-3, 1.0E-3)$$

```r
# second model with prior 1 (sigma.alpha and sigma.beta are distributed as uniform)
# and not centered variables

model2 <- function()
{
  for (i in 1:N)
  {
    for (j in 1:T)
    {
      Y[i,j] ~ dnorm(mu[i,j], tau.c)
      mu[i, j] <- alpha[i] + beta[i] * (x[j])
    }
    alpha[i] ~ dnorm(alpha.c, tau.alpha)
    beta[i] ~ dnorm(beta.c, tau.beta)
  }
  alpha.c ~ dnorm(0, 1.0E-6)
  beta.c ~ dnorm(0, 1.0E-6)
  tau.c ~ dgamma(1.0E-3, 1.0E-3)
  sigma.alpha ~ dunif(0,100)
  sigma.beta ~ dunif(0,100)
  tau.alpha <- 1/(sigma.alpha*sigma.alpha)
  tau.beta <- 1/(sigma.beta*sigma.beta)
  sigma.c <- 1.0/sqrt(tau.c)
  xbar <- mean(x[])
  alpha0 <- alpha.c - beta.c*xbar
}


## Read in the rats data for JAGS
rats.data.list  <- list("Y", "x", "T", "N")

## Name the JAGS parameters
rats.params <- c("tau.c", "alpha.c", "beta.c", "tau.alpha", "tau.beta")

## Define the starting values for JAGS
rats.inits.2 <- function(){
  list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250,
                 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250),
```

```r
      beta  = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
                6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
      alpha.c = 150, beta.c = 10,
      tau.c = 1, sigma.alpha = 1, sigma.beta = 1)
}

mod2 <- jags(data=rats.data.list, inits=rats.inits.2, rats.params, n.chains=2, n.iter=10000,
             n.burnin=1000, n.thin = 1, model.file=model2, DIC=TRUE)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 150
##    Unobserved stochastic nodes: 65
##    Total graph size: 548
##
## Initializing model
```

```r
mod2
```

```
## Inference for Bugs model at "C:/Users/valer/AppData/Local/Temp/RtmpERaEBF/model172825393022.txt", fi
##  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
## alpha.c   106.585   2.364 101.933 105.005 106.582 108.143 111.297 1.001
## beta.c      6.184   0.109   5.968   6.112   6.185   6.255   6.397 1.001
## tau.alpha   0.009   0.004   0.004   0.007   0.009   0.011   0.018 1.001
## tau.beta    4.098   1.430   1.966   3.088   3.873   4.862   7.515 1.002
## tau.c       0.027   0.004   0.020   0.024   0.027   0.030   0.036 1.002
## deviance  968.961  14.257 943.375 958.914 968.127 978.193 998.554 1.003
##           n.eff
## alpha.c   18000
## beta.c    18000
## tau.alpha  9800
## tau.beta   1900
## tau.c      1300
## deviance    830
##
## 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 = 101.5 and DIC = 1070.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```r
mod2$BUGSoutput$DIC
```

```
## [1] 1070.475
```

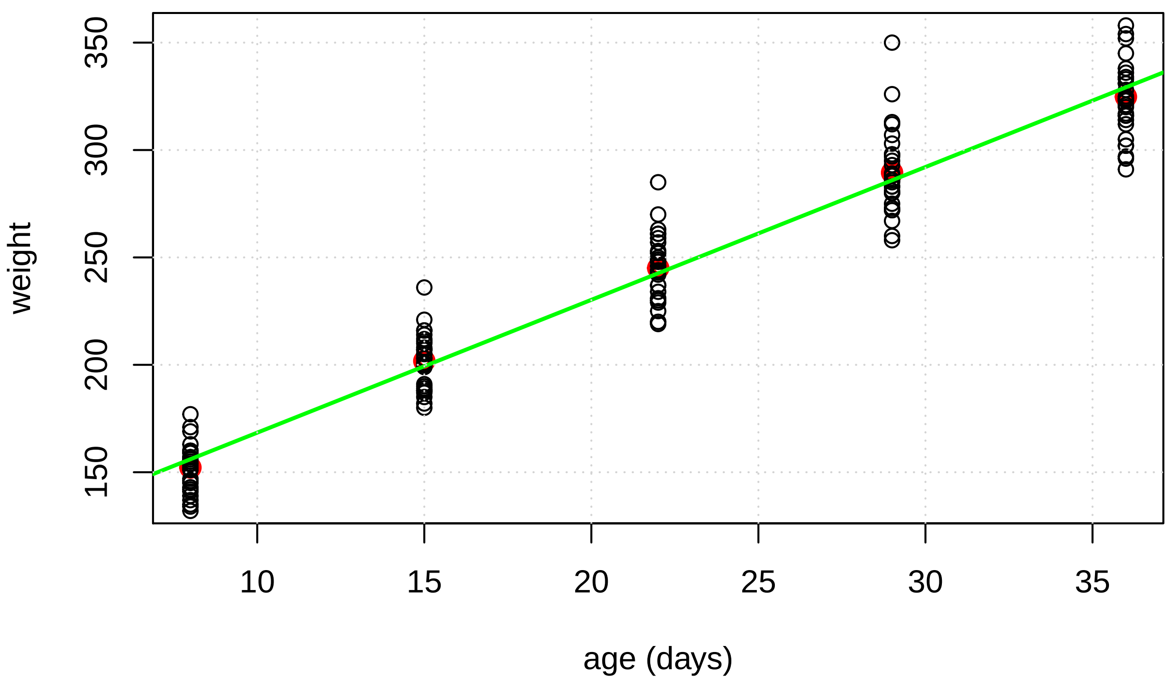
```r
# mean of the intercept
mod2$BUGSoutput$summary[,"mean"]["alpha.c"]
```

```
##  alpha.c
## 106.5845
```

```r
# mean of the slope
mod2$BUGSoutput$summary[,"mean"]["beta.c"]
```

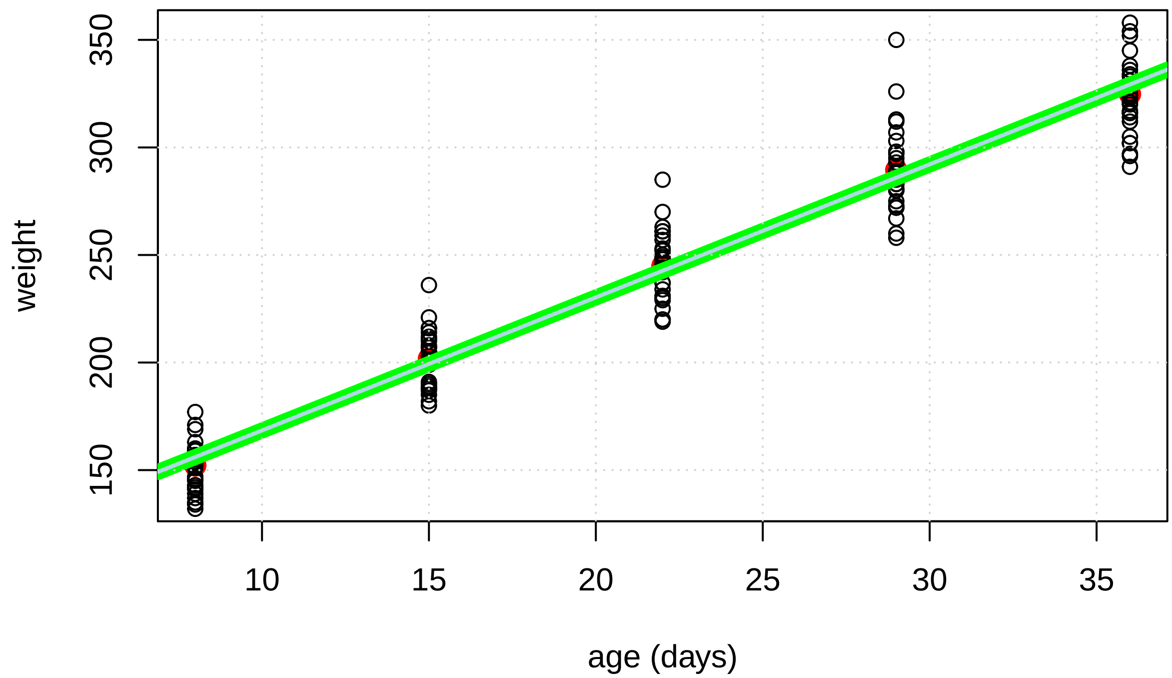
```
##  beta.c
## 6.18362
```

```r
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod2$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod2$BUGSoutput$summary[,"mean"]["beta.c"], col="green", lwd=2)
grid()
```




```r
# comparison with frequentistic approach
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod2$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod2$BUGSoutput$summary[,"mean"]["beta.c"], col="green", lwd=8)
```

```r
abline(mean_alpha, mean_beta, col="lightblue", lwd=2)
grid()
```



Third model

In this model we use the same priors of the first model, but in this case we fix the intercept to a global value. The fixed value for the intercept is 200 and we will see successively if this model is better or not respect to the first model (where the itercept is not fixed)

```
# third model: using the priors of the first model and fixing the intercept to a global value

model3 <- function()
{
  for (i in 1:N)
  {
    for (j in 1:T)
    {
      Y[i,j] ~ dnorm(mu[i,j], tau.c)
      mu[i, j] <- alpha.c + beta[i] * (x[j])
    }
    beta[i] ~ dnorm(beta.c, tau.beta)
  }
  alpha.c = 200
  beta.c ~ dnorm(0, 1.0E-6)
  tau.c ~ dgamma(1.0E-3, 1.0E-3)
  tau.beta ~ dgamma(1.0E-3, 1.0E-3)
  sigma.c <- 1.0/sqrt(tau.c)
  xbar <- mean(x[])
  alpha0 <- alpha.c - beta.c*xbar
}

## Read in the rats data for JAGS
rats.data.list  <- list("Y", "x", "T", "N")

## Name the JAGS parameters
rats.params <- c("tau.c", "alpha.c", "tau.beta", "beta.c")

## Define the starting values for JAGS
rats.inits <- function(){
  list(beta  = c(6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
                 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6),
       beta.c = 10,
       tau.c = 1, tau.beta = 1)
}

mod3 <- jags(data=rats.data.list, inits=rats.inits, rats.params, n.chains=2, n.iter=10000,
             n.burnin=1000, n.thin = 1, model.file=model3, DIC=TRUE)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 150
##    Unobserved stochastic nodes: 33
##    Total graph size: 505
##
## Initializing model
```

```r
mod3
```

```
## Inference for Bugs model at "C:/Users/valer/AppData/Local/Temp/RtmpERaEBF/model172864686974.txt", fi
##  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%
## alpha.c   200.000   0.000  200.000  200.000  200.000  200.000  200.000
## beta.c      2.649   0.136    2.382    2.557    2.649    2.741    2.912
## tau.beta  287.735 467.784    5.463   29.845  106.323  334.069 1625.040
## tau.c       0.001   0.000    0.000    0.001    0.001    0.001    0.001
## deviance 1545.450   2.556 1540.916 1543.999 1545.080 1546.605 1551.554
##           Rhat n.eff
## alpha.c  1.000     1
## beta.c   1.006   330
## tau.beta 1.006   350
## tau.c    1.001  5300
## deviance 1.007  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 = 3.3 and DIC = 1548.7
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```r
mod3$BUGSoutput$DIC
```

```
## [1] 1548.715
```

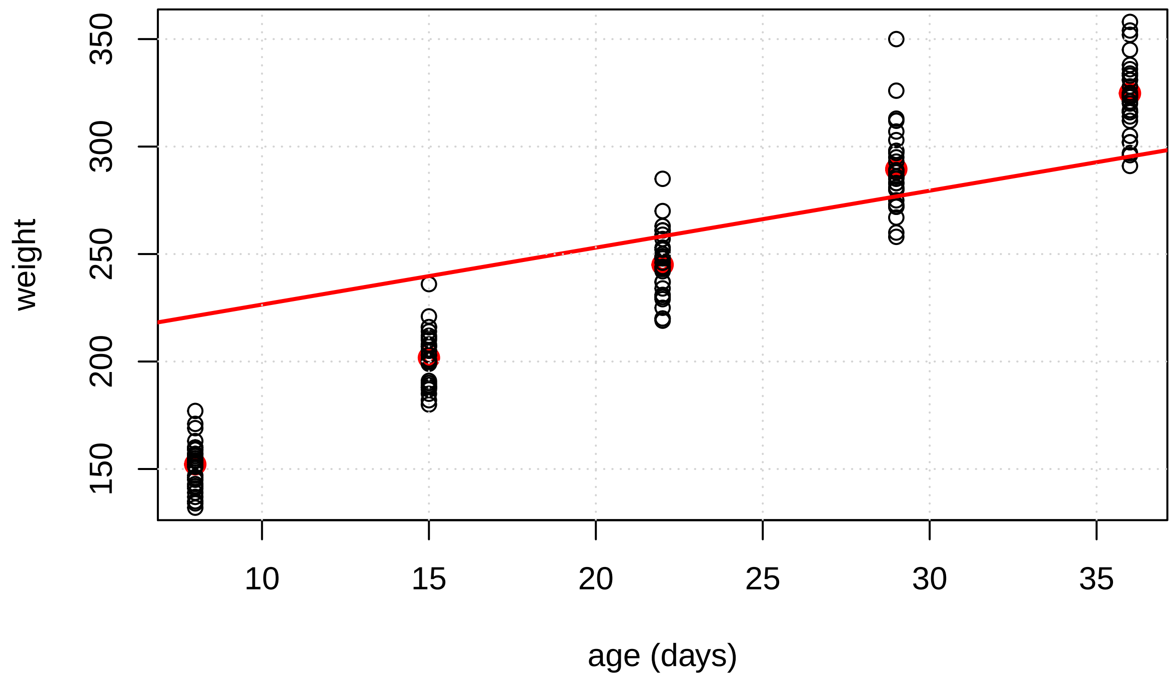
```r
# mean of the intercept
mod3$BUGSoutput$summary[,"mean"]["alpha.c"]
```

```
## alpha.c
##     200
```

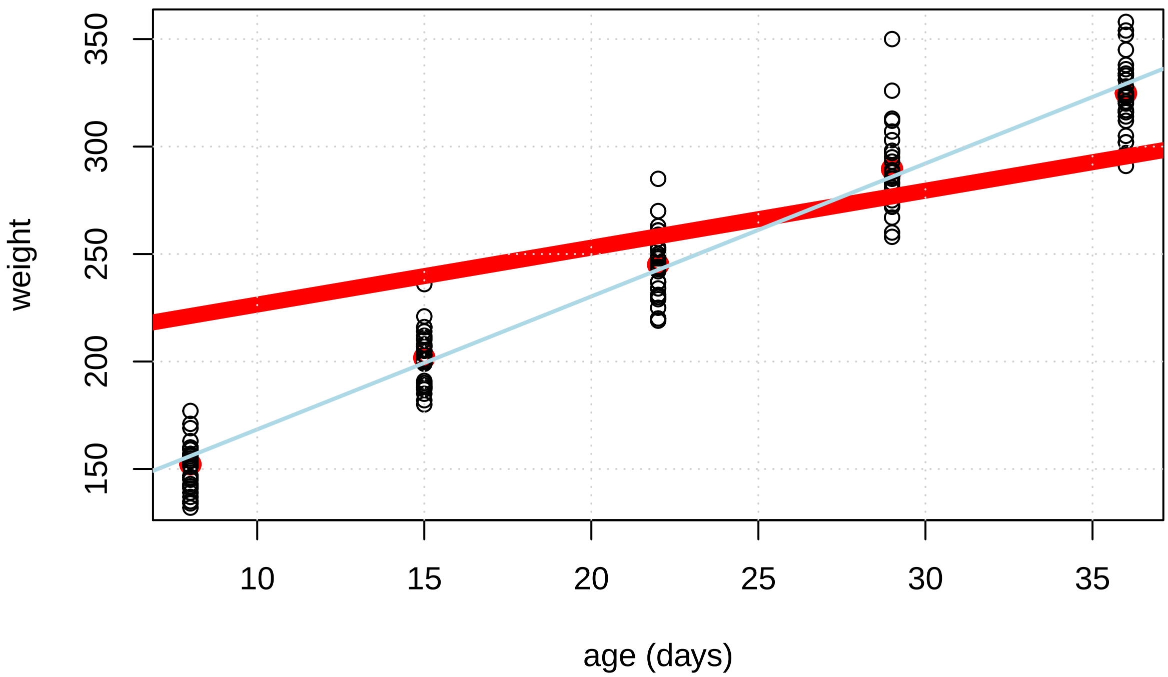
```r
# mean of the slope
mod3$BUGSoutput$summary[,"mean"]["beta.c"]
```

```
##   beta.c
## 2.648985
```

```r
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod3$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod3$BUGSoutput$summary[,"mean"]["beta.c"], col="red", lwd=2)
grid()
```




```
# comparison with frequentistic approach
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod3$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod3$BUGSoutput$summary[,"mean"]["beta.c"], col="red", lwd=8)
abline(mean_alpha, mean_beta, col="lightblue", lwd=2)
grid()
```


weight
350
300
250
200
150
10
15
20
25
30
35
age (days)

Forth model

In this model we use the same priors of the first model, but in this case we fix the slope to a global value. The fixed value for the slope is 3.5 and we will see successively if this model is better or not respect to the first model (where the slope is not fixed)

```
# forth model: using the priors of the prior model and fixing the slope to a global value

model4 <- function()
{
  for (i in 1:N)
  {
    for (j in 1:T)
    {
      Y[i,j] ~ dnorm(mu[i,j], tau.c)
      mu[i, j] <- alpha[i] + beta.c * (x[j]);
    }
    alpha[i] ~ dnorm(alpha.c, tau.alpha);
  }
  alpha.c ~ dnorm(0, 1.0E-6)
  beta.c <- 3.5
  tau.c ~ dgamma(1.0E-3, 1.0E-3)
  tau.alpha ~ dgamma(1.0E-3, 1.0E-3)
  sigma.c <- 1.0/sqrt(tau.c)
  xbar <- mean(x[])
  alpha0 <- alpha.c - beta.c*xbar
}

## Read in the rats data for JAGS
rats.data.list  <- list("Y", "x", "T", "N")

## Name the JAGS parameters
rats.params <- c("tau.c", "alpha.c", "tau.alpha","beta.c")


## Define the starting values for JAGS
rats.inits <- function(){
  list(alpha = c(250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250,
                 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250, 250),
       alpha.c = 150,
       tau.c = 1, tau.alpha = 1)
}

mod4 <- jags(data=rats.data.list, inits=rats.inits, rats.params, n.chains=2, n.iter=10000,
             n.burnin=1000, n.thin = 1, model.file=model4, DIC=TRUE)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 150
##    Unobserved stochastic nodes: 33
##    Total graph size: 359
##
## Initializing model
```

```r
mod4
```

```
## Inference for Bugs model at "C:/Users/valer/AppData/Local/Temp/RtmpERaEBF/model1728302b3f2c.txt", fi
##  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%
## alpha.c   165.952   2.644  161.397  163.853  166.005  167.827  170.615
## beta.c      3.500   0.000    3.500    3.500    3.500    3.500    3.500
## tau.alpha  99.473 307.473    0.010    0.184    4.835   52.350  897.951
## tau.c       0.001   0.000    0.001    0.001    0.001    0.001    0.001
## deviance 1457.099   3.737 1446.604 1456.395 1457.459 1459.079 1462.891
##           Rhat n.eff
## alpha.c   1.185    13
## beta.c    1.000     1
## tau.alpha 1.035    49
## tau.c     1.001 18000
## deviance  1.014 18000
##
## 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 = 7.0 and DIC = 1464.1
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```r
mod4$BUGSoutput$DIC
```

```
## [1] 1464.083
```

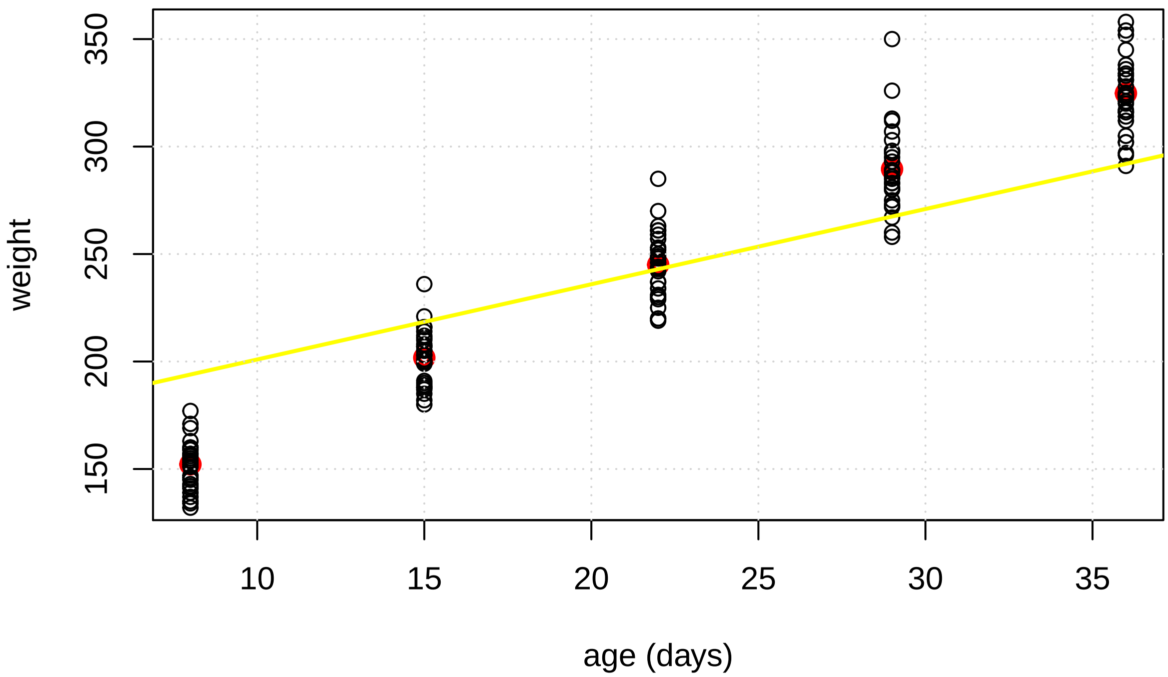
```r
# mean of the intercept
mod4$BUGSoutput$summary[,"mean"]["alpha.c"]
```

```
##  alpha.c
## 165.9519
```

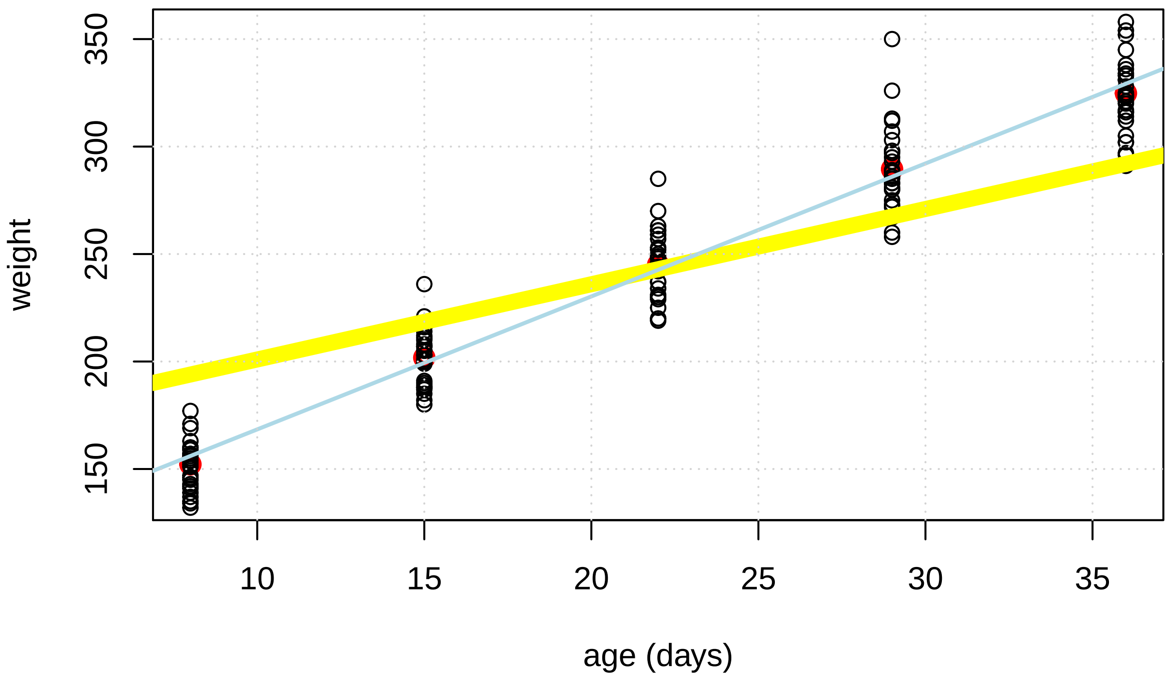
```r
# mean of the slope
mod4$BUGSoutput$summary[,"mean"]["beta.c"]
```

```
## beta.c
##    3.5
```

```r
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod4$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod4$BUGSoutput$summary[,"mean"]["beta.c"], col="yellow", lwd=2)
grid()
```




```
# comparison with frequentistic approach
plot(rats.data$x,colMeans(rats.data$Y), lwd=4, xlab = "age (days)", ylab = "weight",
     col="red", ylim=c(135,355))
points(rep(rats.data$x[1],N), rats.data$Y[,1])
points(rep(rats.data$x[2],N), rats.data$Y[,2])
points(rep(rats.data$x[3],N), rats.data$Y[,3])
points(rep(rats.data$x[4],N), rats.data$Y[,4])
points(rep(rats.data$x[5],N), rats.data$Y[,5])
abline(mod4$BUGSoutput$summary[,"mean"]["alpha.c"],
       mod4$BUGSoutput$summary[,"mean"]["beta.c"], col="yellow", lwd=8)
abline(mean_alpha, mean_beta, col="lightblue", lwd=2)
grid()
```

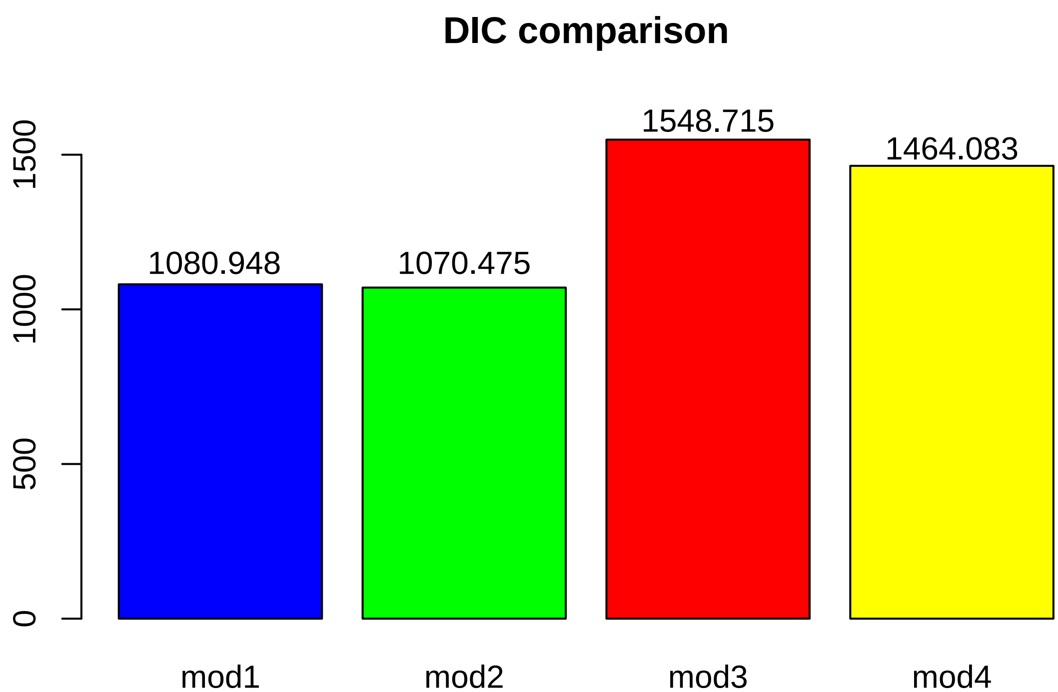

weight
150
200
250
300
350
10
15
20
25
30
35
age (days)

Deviance Information Criterion, DIC

```
DIC_array <- cbind(mod1.DIC=mod1$BUGSoutput$DIC,
                   mod2.DIC=mod2$BUGSoutput$DIC,
                   mod3.DIC=mod3$BUGSoutput$DIC,
                   mod4.DIC=mod4$BUGSoutput$DIC
)
DIC_array
```

```
##      mod1.DIC mod2.DIC mod3.DIC mod4.DIC
## [1,] 1080.948 1070.475 1548.715 1464.083
```

```
# comparison of DIC
DIC = c(mod1$BUGSoutput$DIC, mod2$BUGSoutput$DIC, mod3$BUGSoutput$DIC, mod4$BUGSoutput$DIC)
barplot(DIC, col=c("blue", "green", "red", "yellow"), main="DIC comparison",
        names.arg = c("mod1", "mod2", "mod3", "mod4"), ylim=c(0,1650))
text(0.67, 1150,round(mod1$BUGSoutput$DIC,3))
text(1.9, 1150, round(mod2$BUGSoutput$DIC,3))
text(3.1, 1610, round(mod3$BUGSoutput$DIC,3))
text(4.3, 1520, round(mod4$BUGSoutput$DIC,3))
```

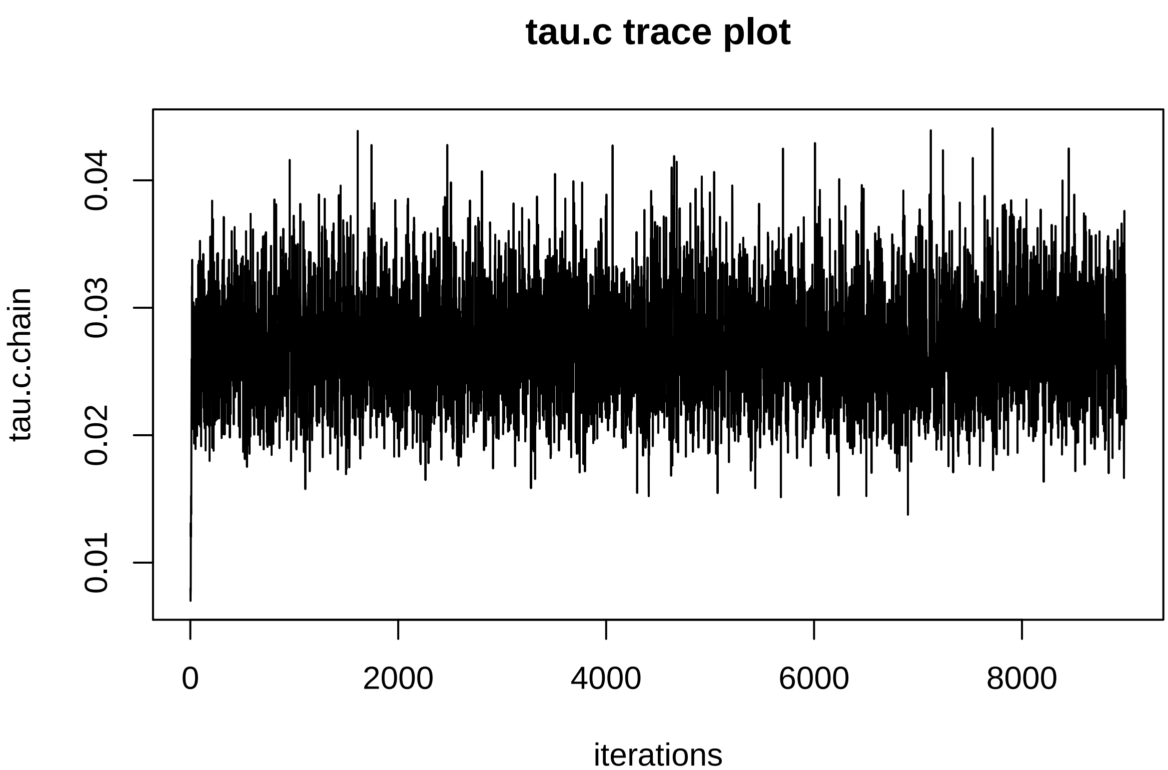


The first and the second model present similar DIC, even if the second model has the smallest value. The third and forth model instead present values of DIC higher respect to the previous models.

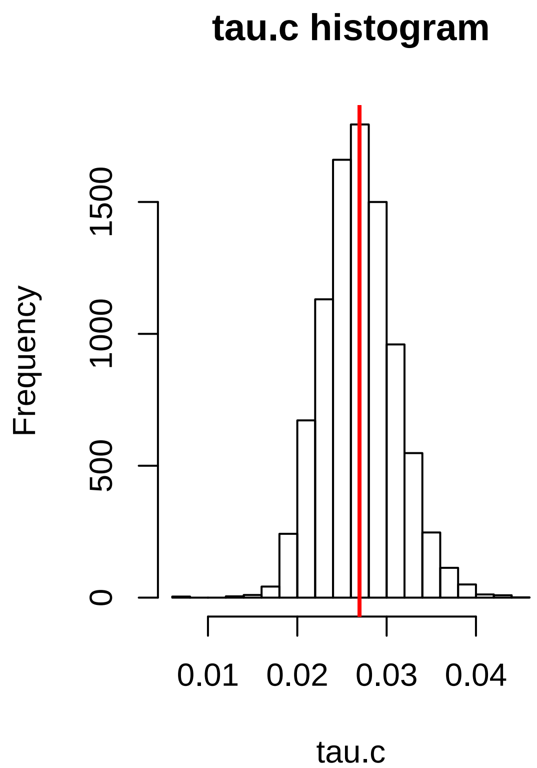
Analysis based on the first model

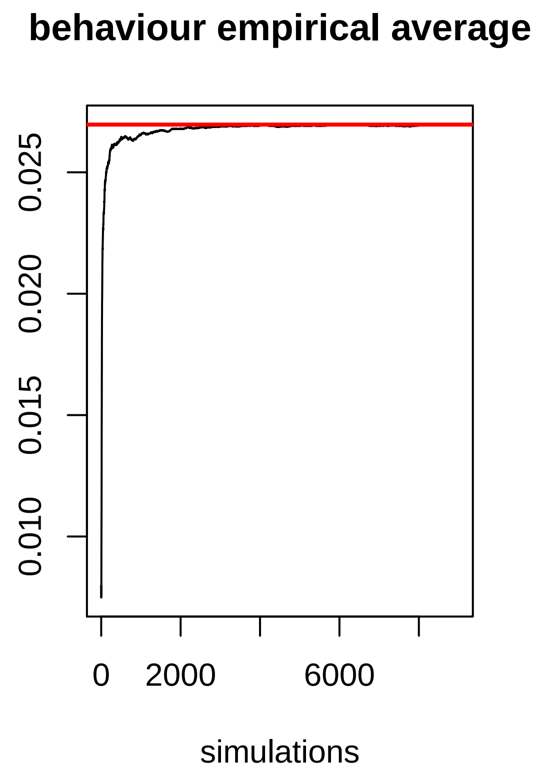
From the theory of Markov chains, we expect our chains to eventually converge to the stationary distribution, which is also our target distribution. However, there is no guarantee that our chain has converged after a given number of iterations. We will check the convergence through some tools

```
# let's see every parameter individually through traceplot, histogram, behaviour of
# the empirical mean and approximation error

# tau.c
tau.c.chain <- mod1$BUGSoutput$sims.array[,1,"tau.c"]
plot(tau.c.chain, xlab = "iterations", main="tau.c trace plot",type="l")
```

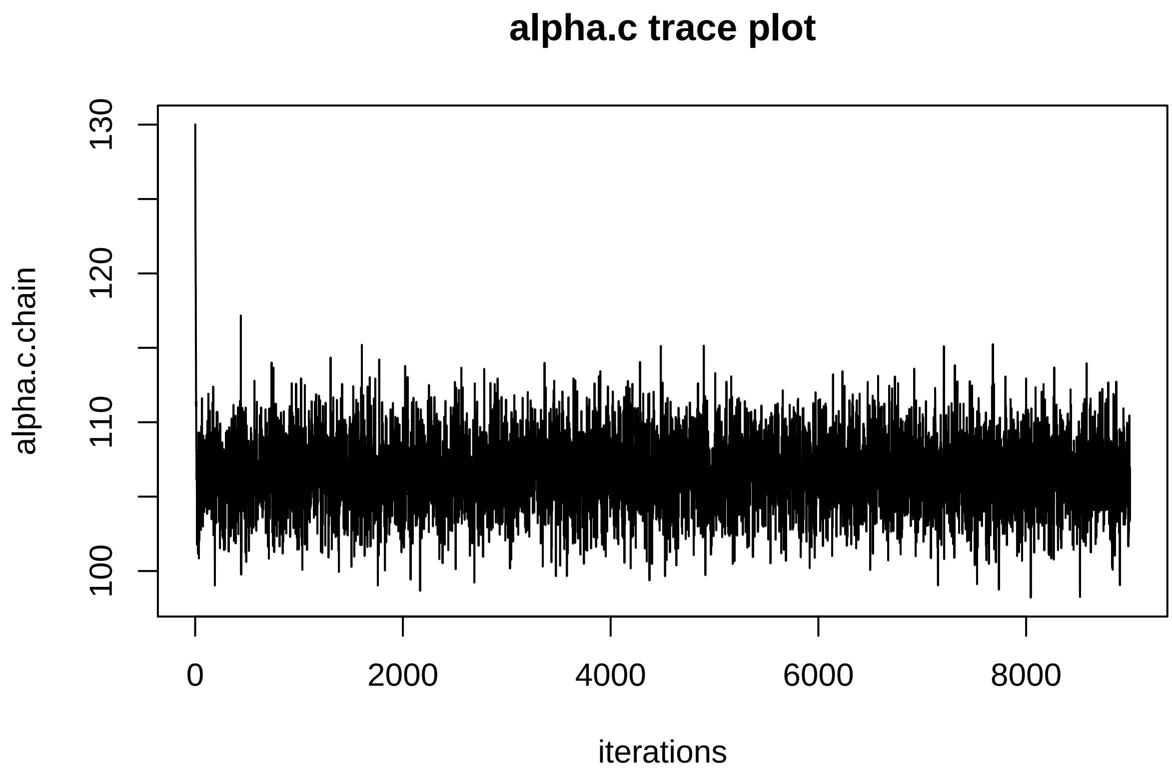



```
par(mfrow=c(1,2))
hist(tau.c.chain, main= "tau.c histogram", xlab = "tau.c")
abline(v=mean(tau.c.chain), col="red", lwd=2)
plot(cumsum(tau.c.chain)/(1:length(tau.c.chain)), type="l", ylab="",
     main="behaviour empirical average", xlab="simulations")
abline(h=mean(tau.c.chain), col="red", lwd=2)
```

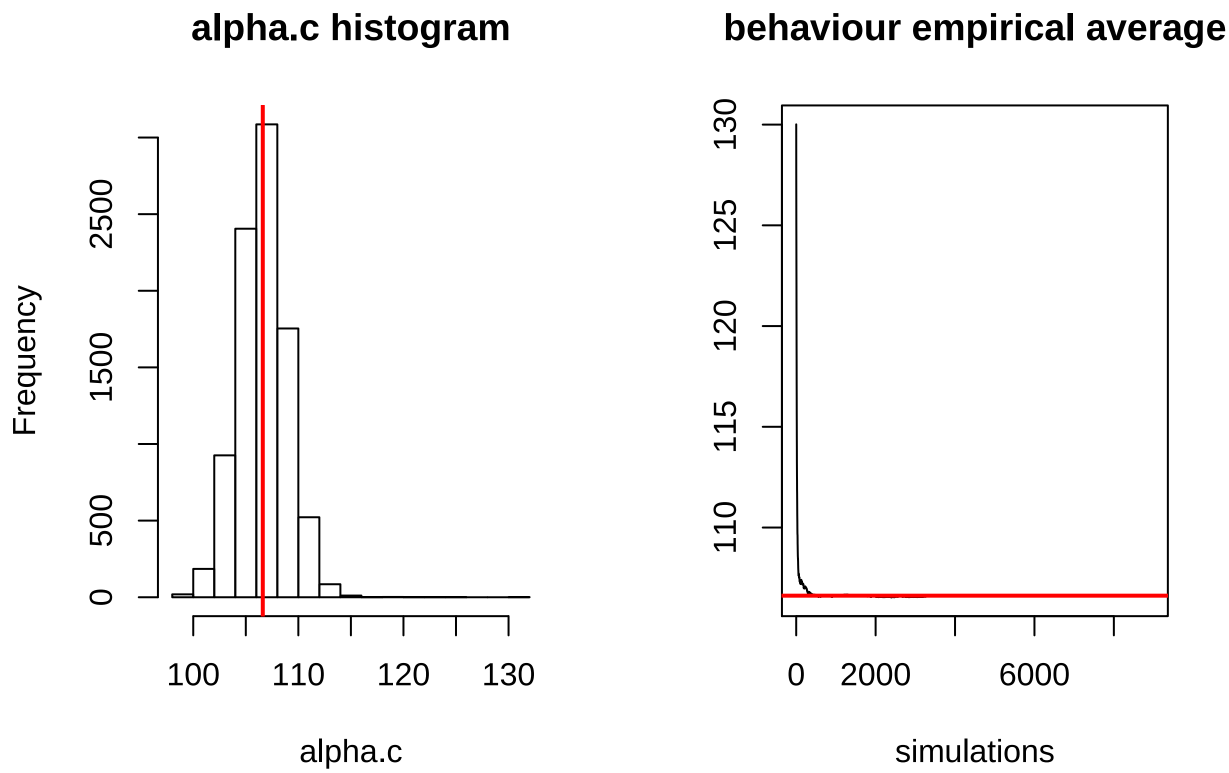




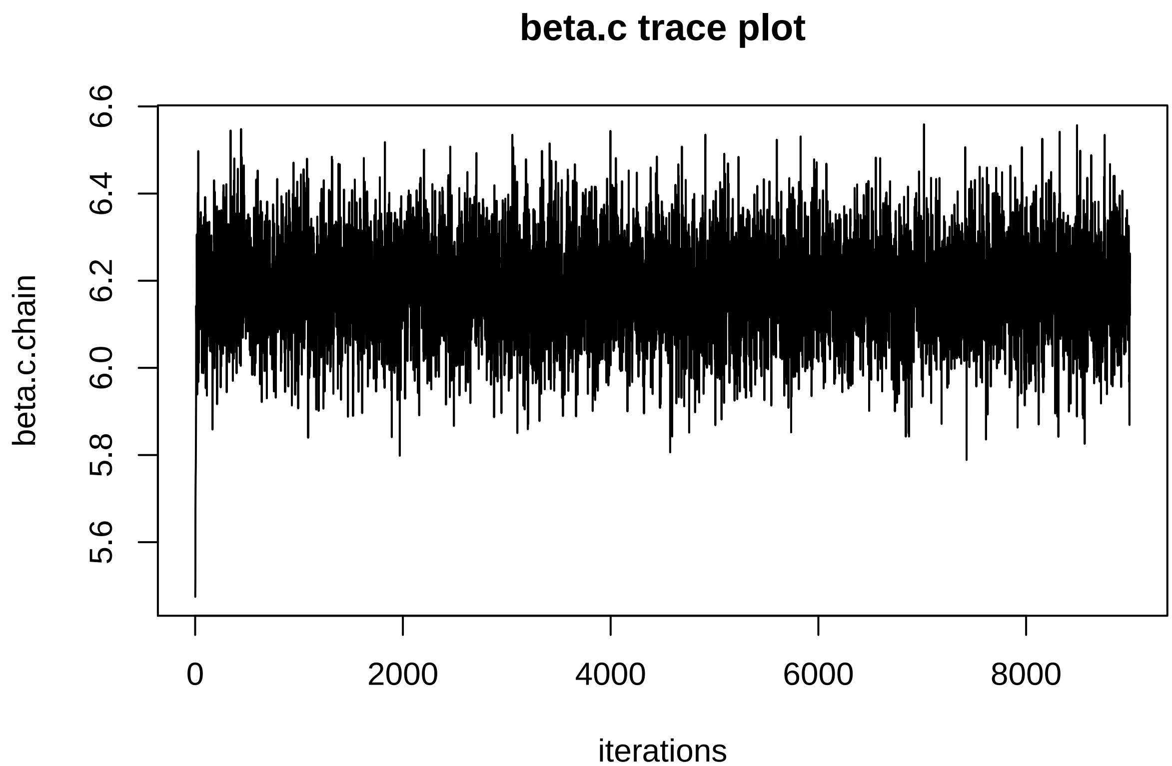

```
par(mfrow=c(1,1))

# alpha.c
alpha.c.chain <- mod1$BUGSoutput$sims.array[,1,"alpha.c"]
plot(alpha.c.chain, xlab = "iterations", main="alpha.c trace plot",type="l")
```



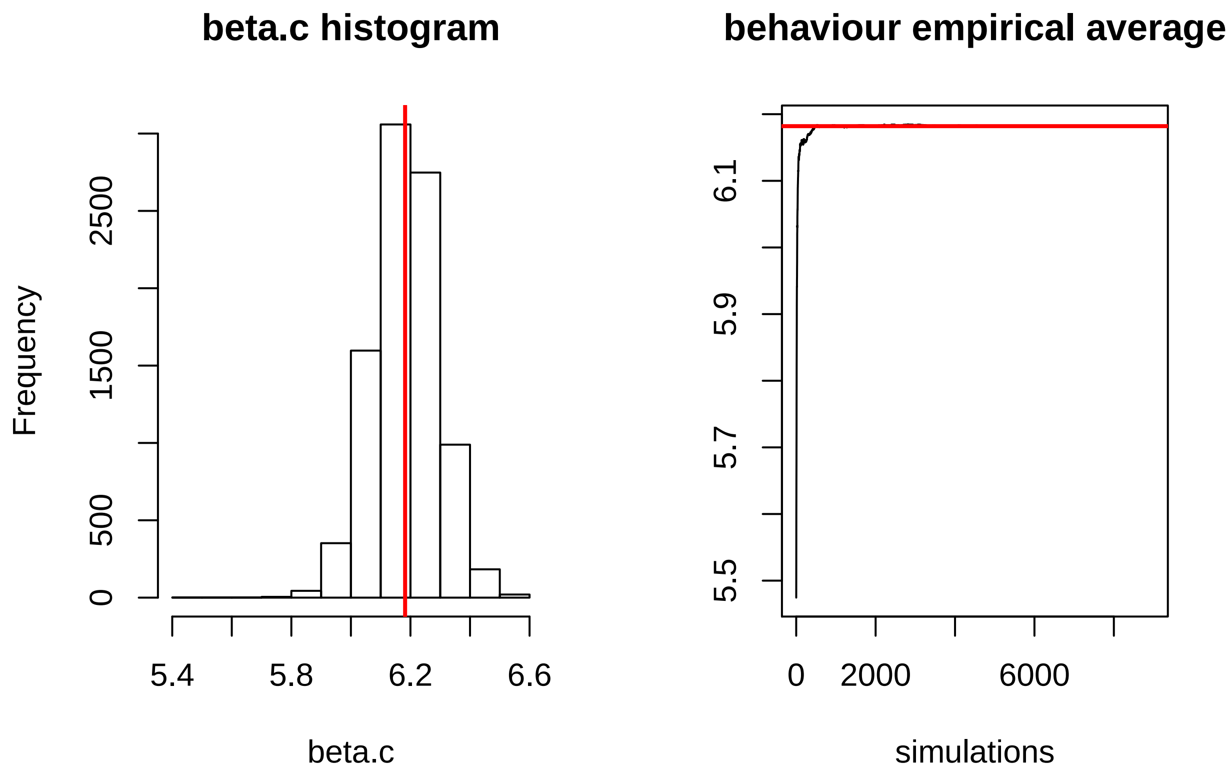

```
par(mfrow=c(1,2))
hist(alpha.c.chain, main= "alpha.c histogram", xlab = "alpha.c")
abline(v=mean(alpha.c.chain), col="red", lwd=2)
plot(cumsum(alpha.c.chain)/(1:length(alpha.c.chain)), type="l", ylab="",
     main="behaviour empirical average", xlab="simulations")
abline(h=mean(alpha.c.chain), col="red", lwd=2)
```



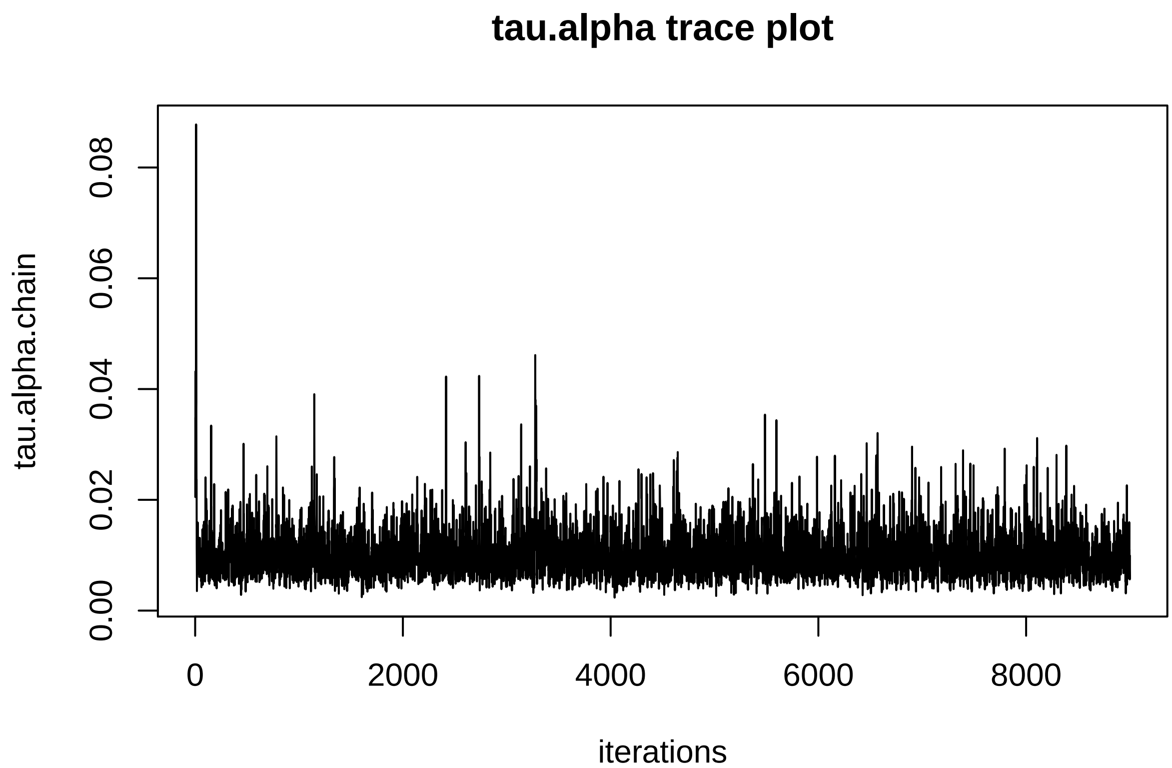

```
par(mfrow=c(1,1))

# beta.c
beta.c.chain <- mod1$BUGSoutput$sims.array[,1,"beta.c"]
plot(beta.c.chain, xlab = "iterations", main="beta.c trace plot",type="l")
```



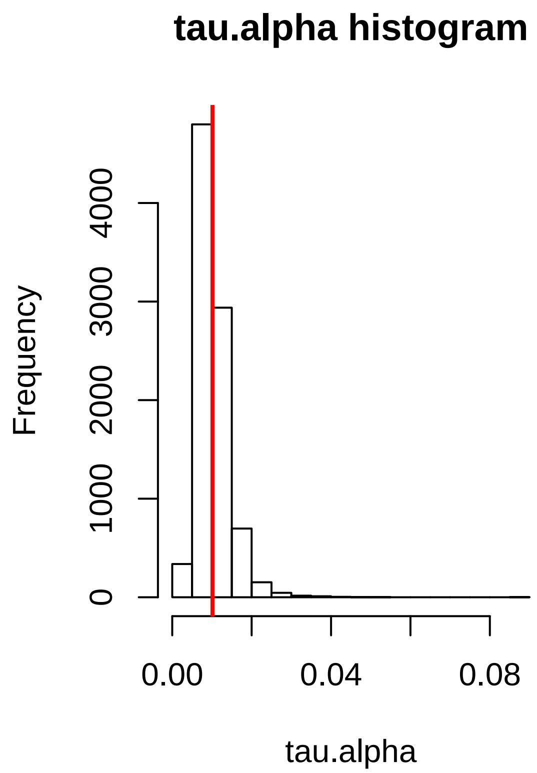

```r
par(mfrow=c(1,2))
hist(beta.c.chain, main= "beta.c histogram", xlab = "beta.c")
abline(v=mean(beta.c.chain), col="red", lwd=2)
plot(cumsum(beta.c.chain)/(1:length(beta.c.chain)), type="l", ylab="",
     main="behaviour empirical average", xlab="simulations")
abline(h=mean(beta.c.chain), col="red", lwd=2)
```

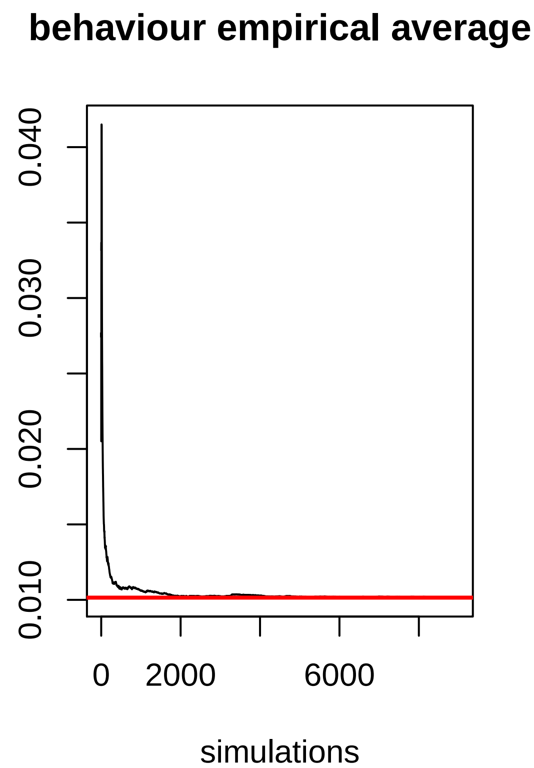



```r
par(mfrow=c(1,1))

# tau.alpha
tau.alpha.chain <- mod1$BUGSoutput$sims.array[,1,"tau.alpha"]
plot(tau.alpha.chain, xlab = "iterations", main="tau.alpha trace plot",type="l")
```

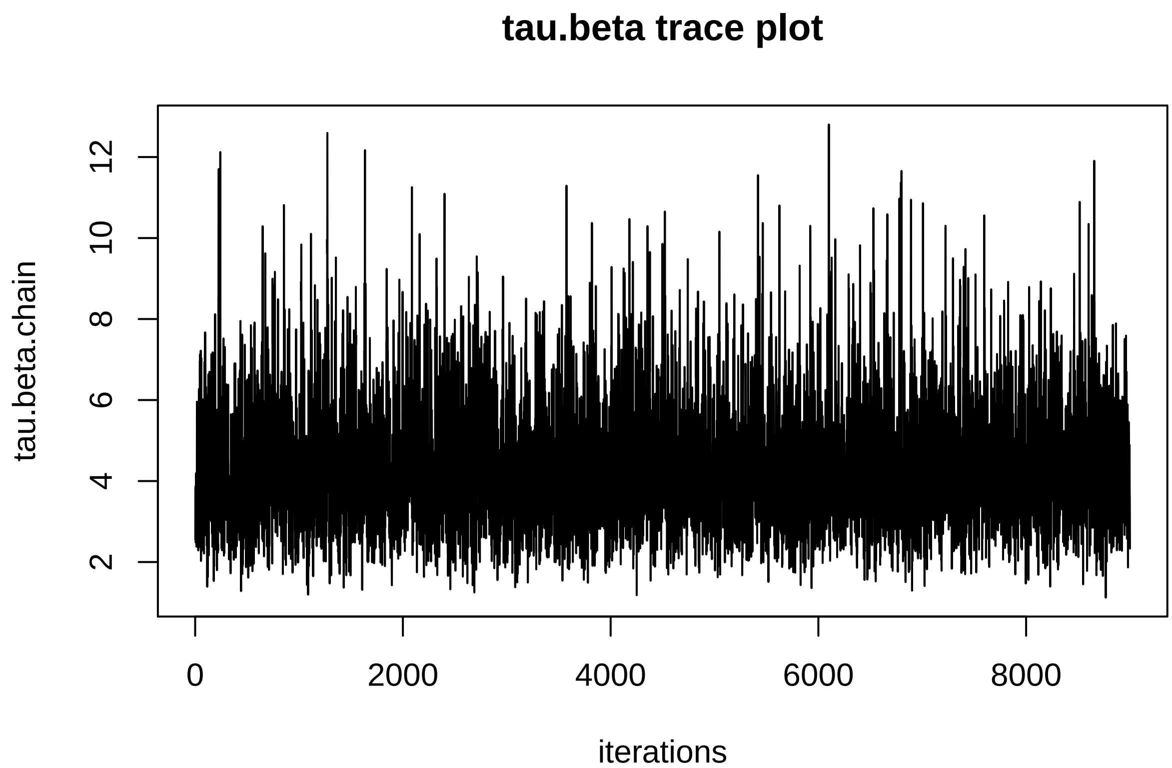



```r
par(mfrow=c(1,2))
hist(tau.alpha.chain, main= "tau.alpha histogram", xlab = "tau.alpha")
abline(v=mean(tau.alpha.chain), col="red", lwd=2)
plot(cumsum(tau.alpha.chain)/(1:length(tau.alpha.chain)), type="l", ylab="",
     main="behaviour empirical average", xlab="simulations")
abline(h=mean(tau.alpha.chain), col="red", lwd=2)
```

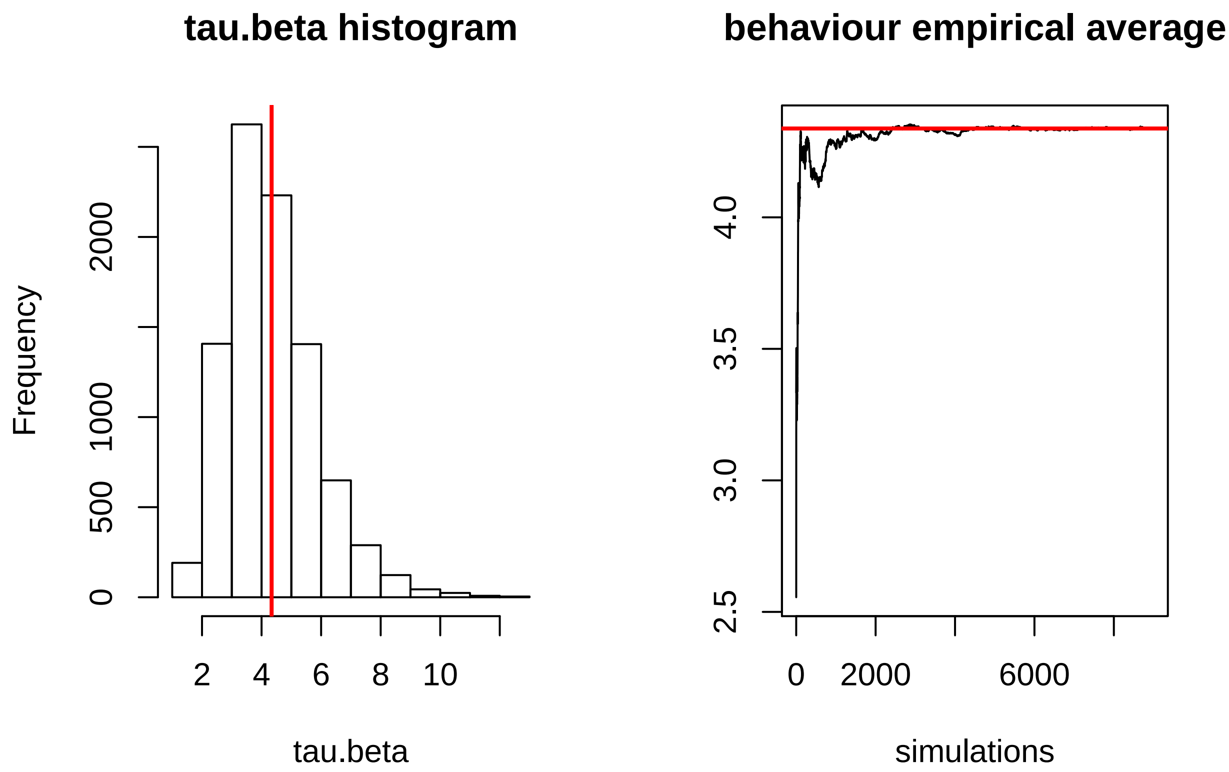





```
par(mfrow=c(1,1))

# tau.beta
tau.beta.chain <- mod1$BUGSoutput$sims.array[,1,"tau.beta"]
plot(tau.beta.chain, xlab = "iterations", main="tau.beta trace plot",type="l")
```

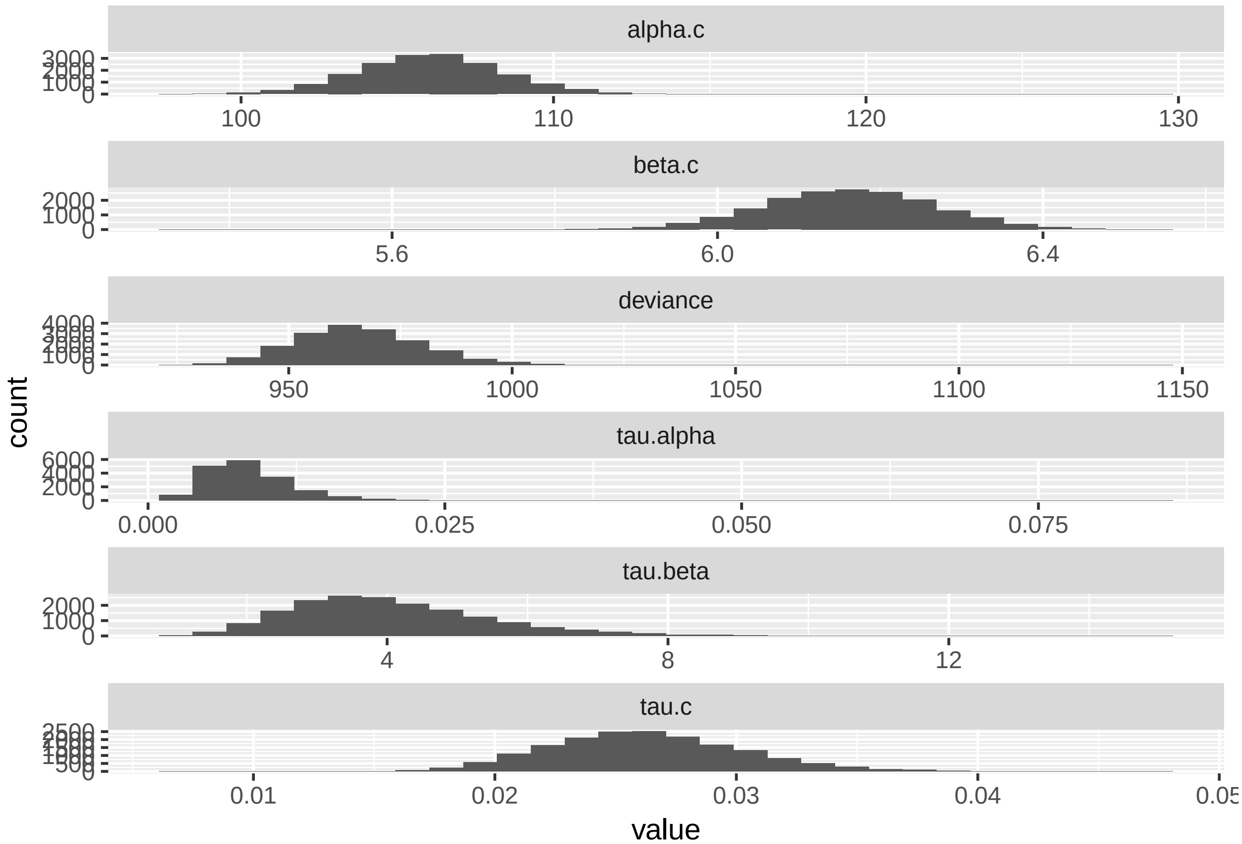



```r
par(mfrow=c(1,2))
hist(tau.beta.chain, main= "tau.beta histogram", xlab = "tau.beta")
abline(v=mean(tau.beta.chain), col="red", lwd=2)
plot(cumsum(tau.beta.chain)/(1:length(tau.beta.chain)), type="l", ylab="",
     main="behaviour empirical average", xlab="simulations")
abline(h=mean(tau.beta.chain), col="red", lwd=2)
```

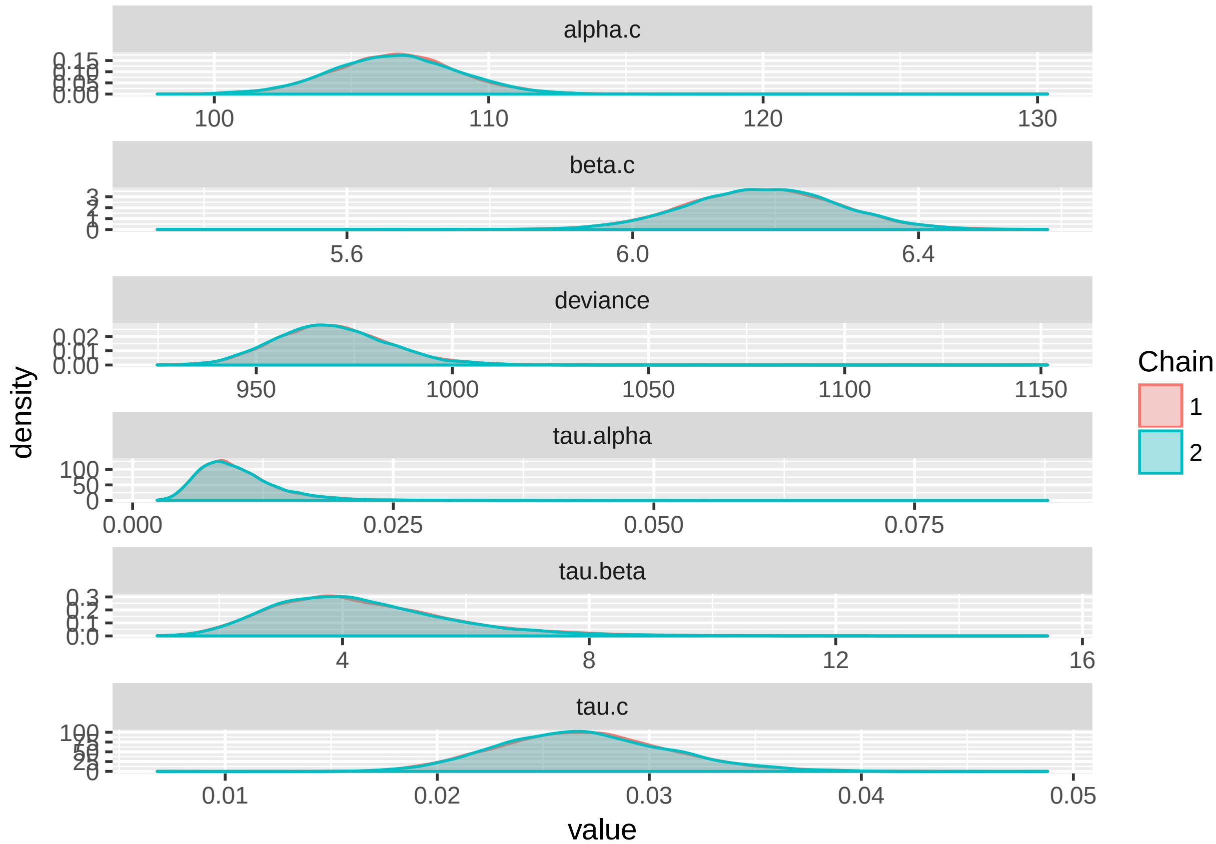



```r
par(mfrow=c(1,1))
```

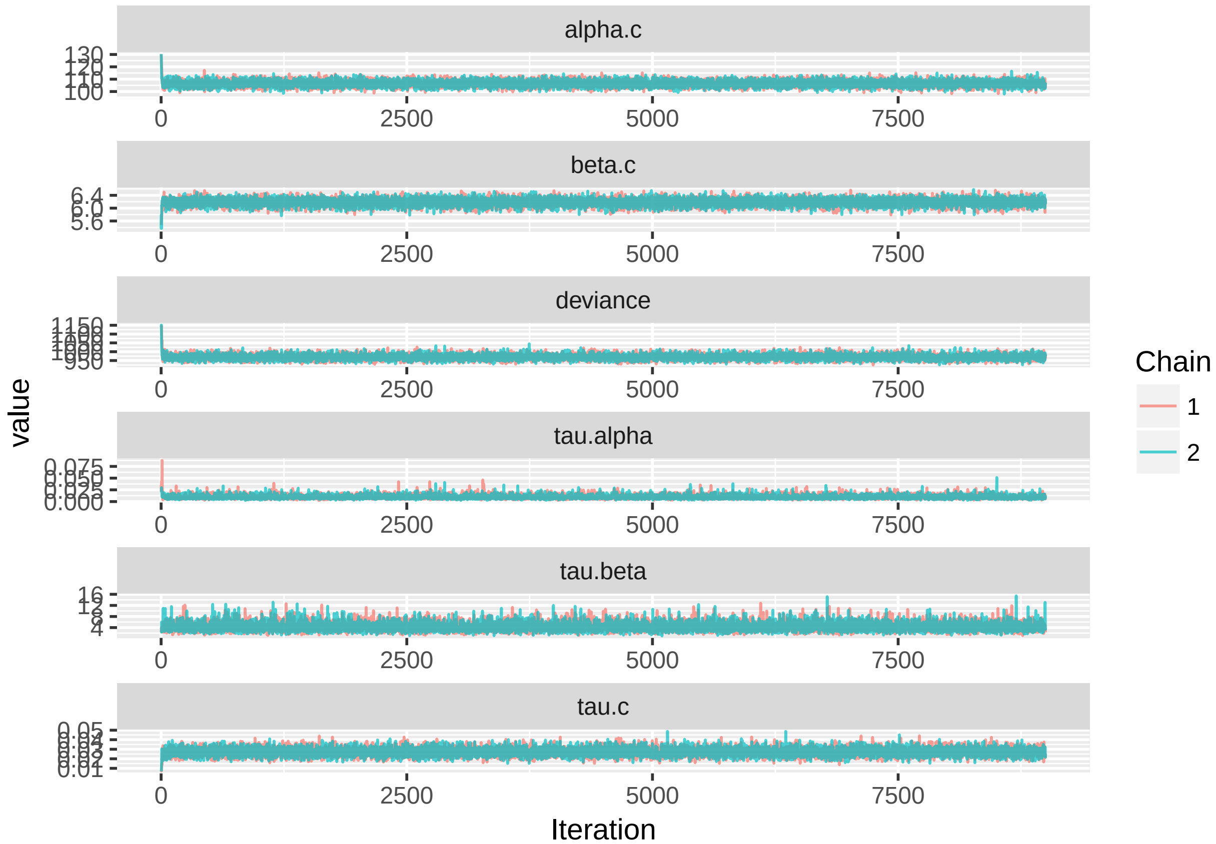
```r
# Instead of seeing every parameter individually, we can analyze
# all the parameters in a single plot

# use of the library ggmcmc that allows us to work with ggs object
mod1.fit.gg <- ggs(as.mcmc(mod1))

# histograms of the parameters
ggs_histogram(mod1.fit.gg)
```




```r
ggs_density(mod1.fit.gg)
```

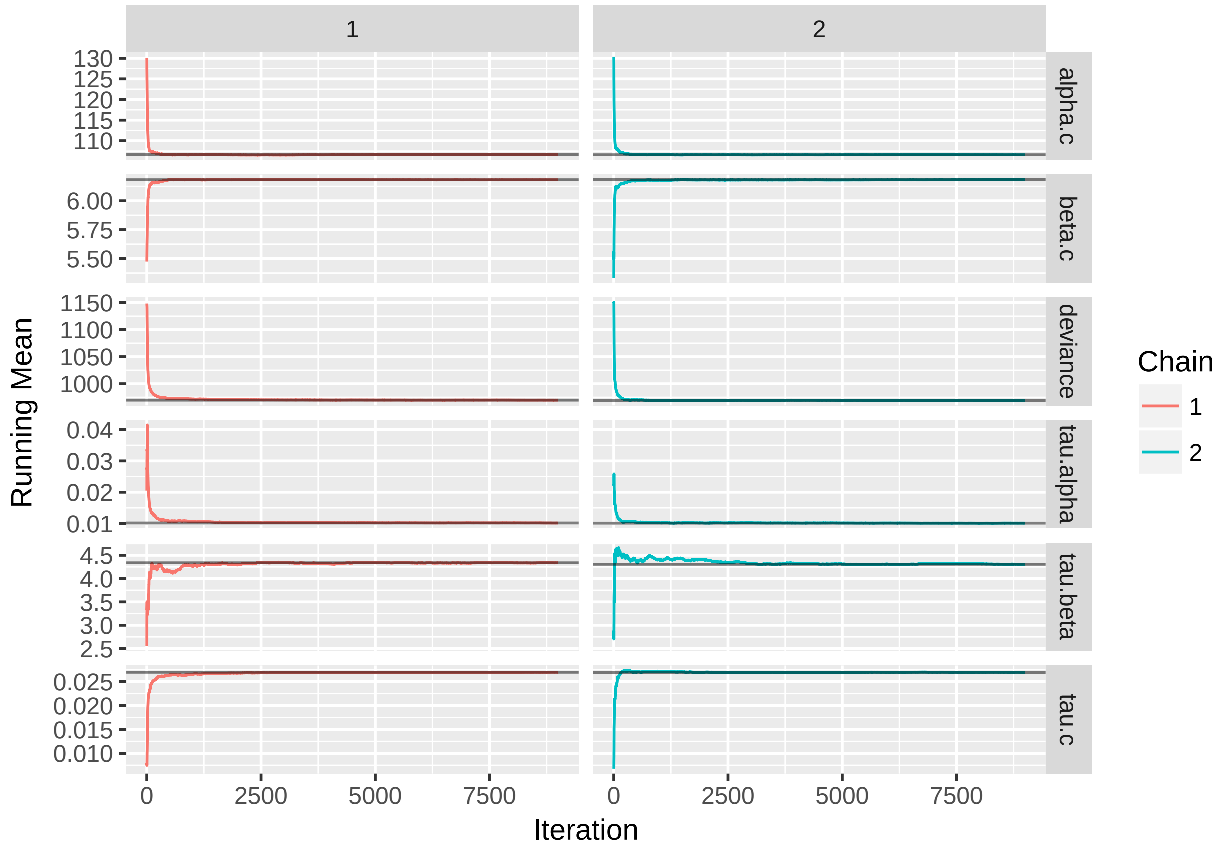



```
ggs_traceplot(mod1.fit.gg)
```




```r
ggs_running(mod1.fit.gg)
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```



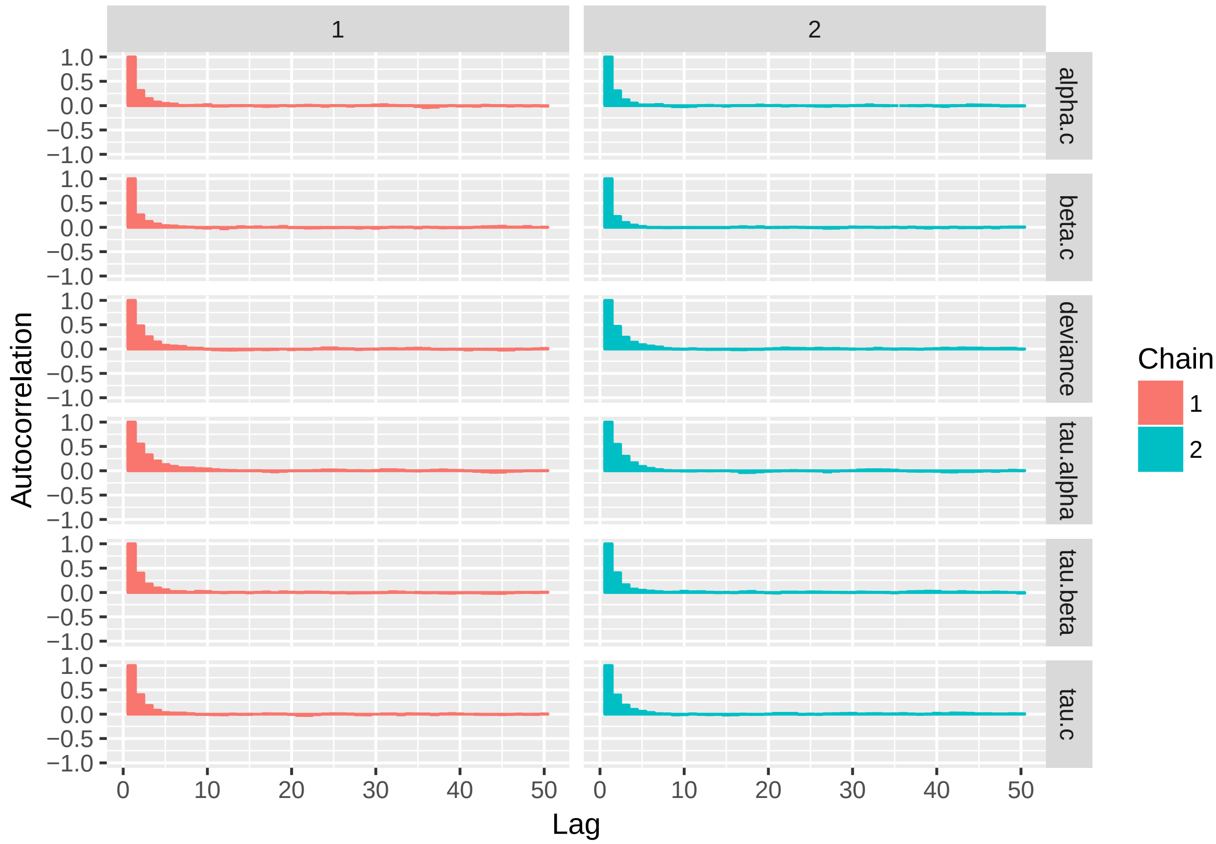
Monitoring autocorrelations is also very useful since low or high values indicate fast or slow convergence, respectively.

The lag k autocorrelation ρ_k is the correlation between every draw and its kth lag:

$$\rho_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

We would expect the kth lag autocorrelation to be smaller as k increases

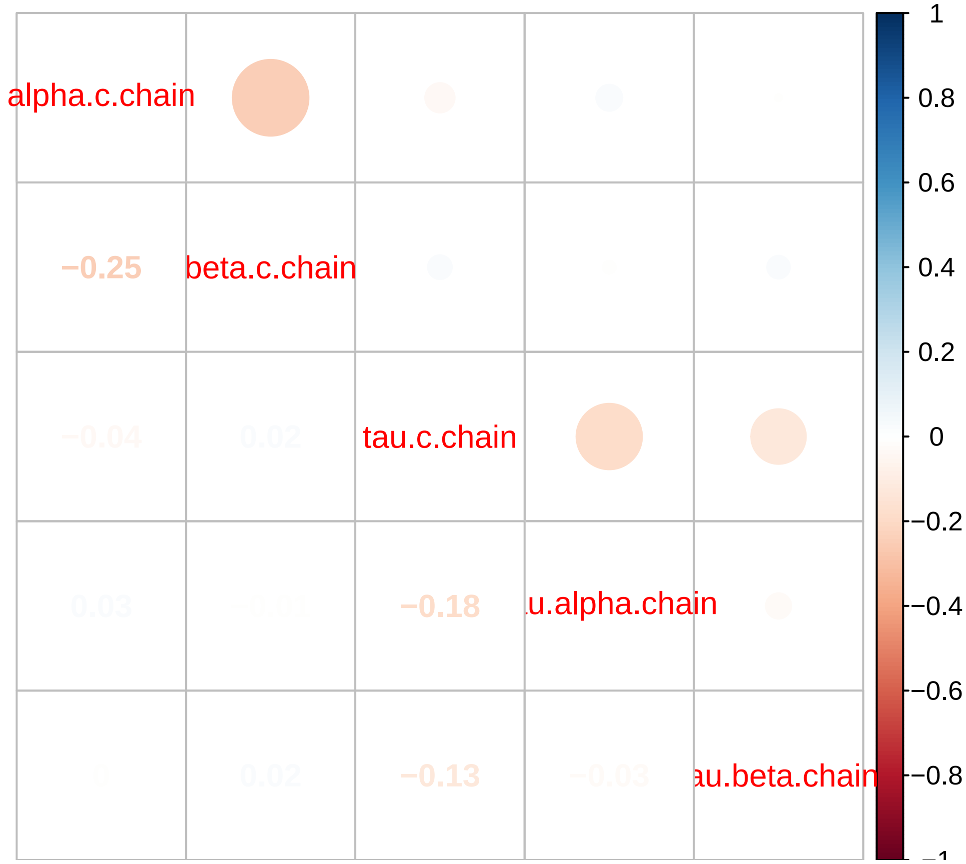
```
ggs_autocorrelation(mod1.fit.gg)
```




```r
# correlation
mod1.parameters = cbind(alpha.c.chain, beta.c.chain,
                        tau.c.chain, tau.alpha.chain, tau.beta.chain)
correlation=cor(mod1.parameters)
correlation
```

```
##                 alpha.c.chain beta.c.chain tau.c.chain tau.alpha.chain
## alpha.c.chain      1.00000000 -0.246546298 -0.03690857     0.028938519
## beta.c.chain      -0.24654630  1.000000000  0.02422755    -0.007050389
## tau.c.chain       -0.03690857  0.024227548  1.00000000    -0.183734230
## tau.alpha.chain    0.02893852 -0.007050389 -0.18373423     1.000000000
## tau.beta.chain    -0.00210440  0.022060545 -0.12794196    -0.027422050
##                 tau.beta.chain
## alpha.c.chain      -0.00210440
## beta.c.chain        0.02206054
## tau.c.chain        -0.12794196
## tau.alpha.chain    -0.02742205
## tau.beta.chain      1.00000000
```

```r
corrplot.mixed(correlation)
```


alpha.c.chain
beta.c.chain
tau.c.chain
u.alpha.chain
au.beta.chain
−0.25
−0.18
−0.13
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8

Prediction

```
prediction <- function(x){
  alpha <- rep(NA, 9000)
  beta <- rep(NA, 9000)
  y.pred <- rep(NA, 9000)
  for(i in 1:9000){
    alpha[i] = rnorm(1,alpha.c.chain[i], tau.alpha.chain[i])
    beta[i] = rnorm(1,beta.c.chain[i], tau.beta.chain[i])
    y.pred[i] = alpha[i]+beta[i]*x
  }
  return(y.pred)
}


# mean with age egual to 8
mean(prediction(8))
```

```
## [1] 156.6524
```

```
# mean with age egual to 15
mean(prediction(15))
```

```
## [1] 199.9859
```

```
# mean with age egual to 22
mean(prediction(22))
```

```
## [1] 242.4791
```

```
# mean with age egual to 29
mean(prediction(29))
```

```
## [1] 284.7272
```

```
# mean with age egual to 36
mean(prediction(36))
```

```
## [1] 329.9798
```