WinBUGS : part 2

Bruno Boulanger
Jonathan Jaeger
Astrid Jullion
Philippe Lambert

Agenda

- Hierarchical model: linear regression example
- R2WinBUGS

Linear Regression

- Bayesian linear regression model :

  - Likelihood

$$y_i = \alpha + \beta x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

$$\Updownarrow$$

$$y_i \sim N(\alpha + \beta x_i, \sigma^2) \qquad for \quad i = 1, \ldots, n$$

  - Prior (non-informative):

$\alpha \sim N(0 , 10^4)$

$\beta \sim N(0 , 10^4)$

$\tau \sim Gamma(0.0001,0.0001)$ with $\tau = 1/\sigma^2$

Linear Regression

- In WinBUGS :

```
model {
   for (i in 1:n){
         y[i]~dnorm(mu[i],tau)
          mu[i] <- alpha + beta * x[i]
   }
```

- **Prior :**

```
    alpha~dnorm(0, 0.0001)
    beta ~ dnorm (0, 0.0001)
    tau ~ dgamma (0.0001, 0.0001) }
```

Hierarchical model

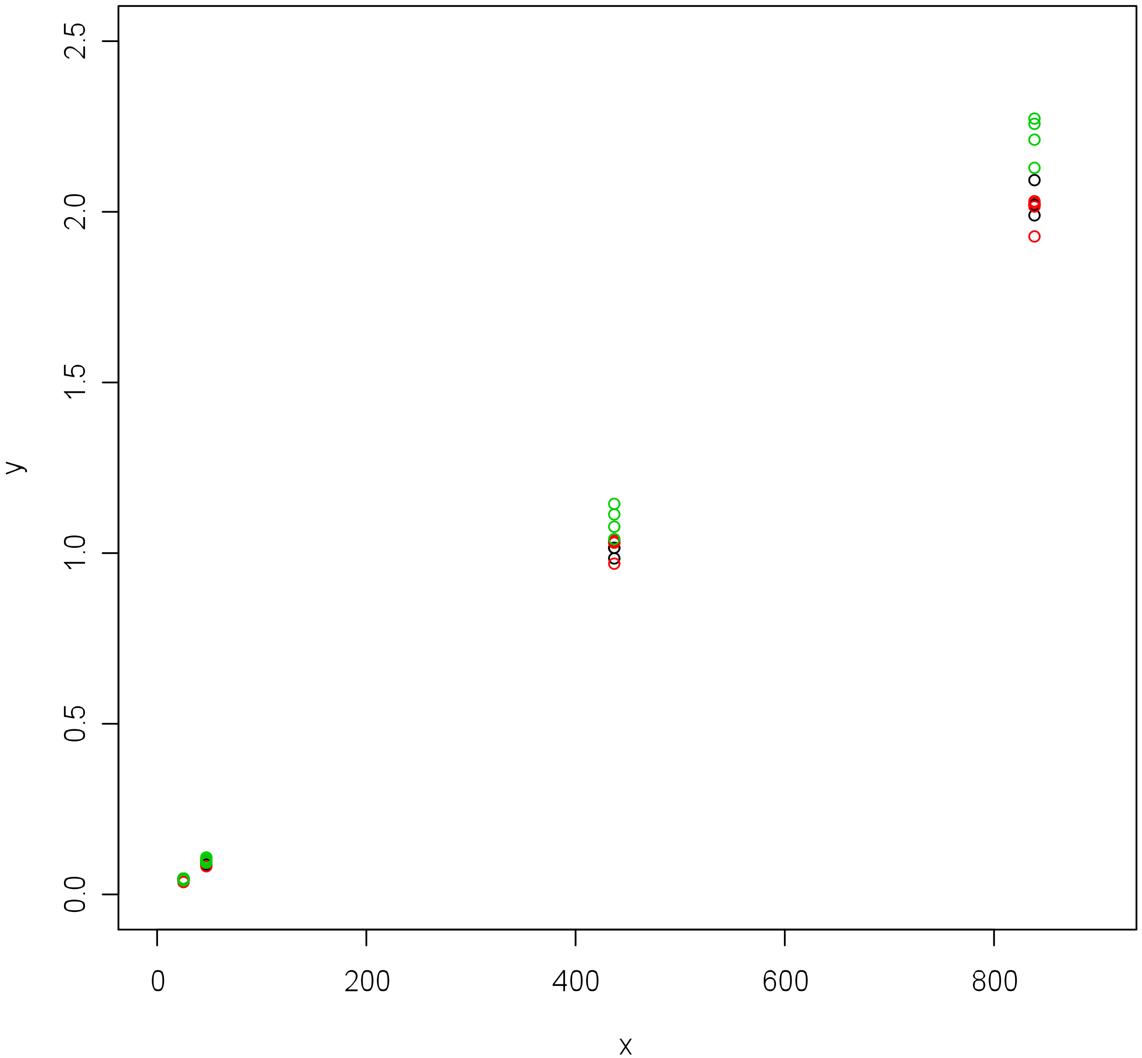
- Calibration experiment:

- A new calibration curve is established every day
  - 3 DAYS of calibration
    - 4 levels of concentrations
    - 4 repetitions by level

- Linear calibration curve

- Calibration curve (intercept and slope) will slightly vary from day to day.

Hierarchical model



3 days for calibration

Hierarchical model

- Data :Practical exercices\Hierarchical reg\data_orig.xls

Hierarchical model

- Bayesian hierarchical linear regression model :

$$y_{ij} = \alpha_j + \beta_j x_i + \varepsilon_{ij}$$

$$\alpha_j \sim N(\alpha_{mean}, \sigma_\alpha^2)$$

$$\beta_j \sim N(\beta_{mean}, \sigma_\beta^2)$$

$$\varepsilon_{ij} \sim N(0, \sigma^2)$$

for i=1,…,n and j=1,…,m

n observations per day

m days

Hierarchical model

- Bayesian hierarchical linear regression model :

  - Prior (non-informative):

$$\alpha_{mean} \sim N(0\ ,\ 10000)$$

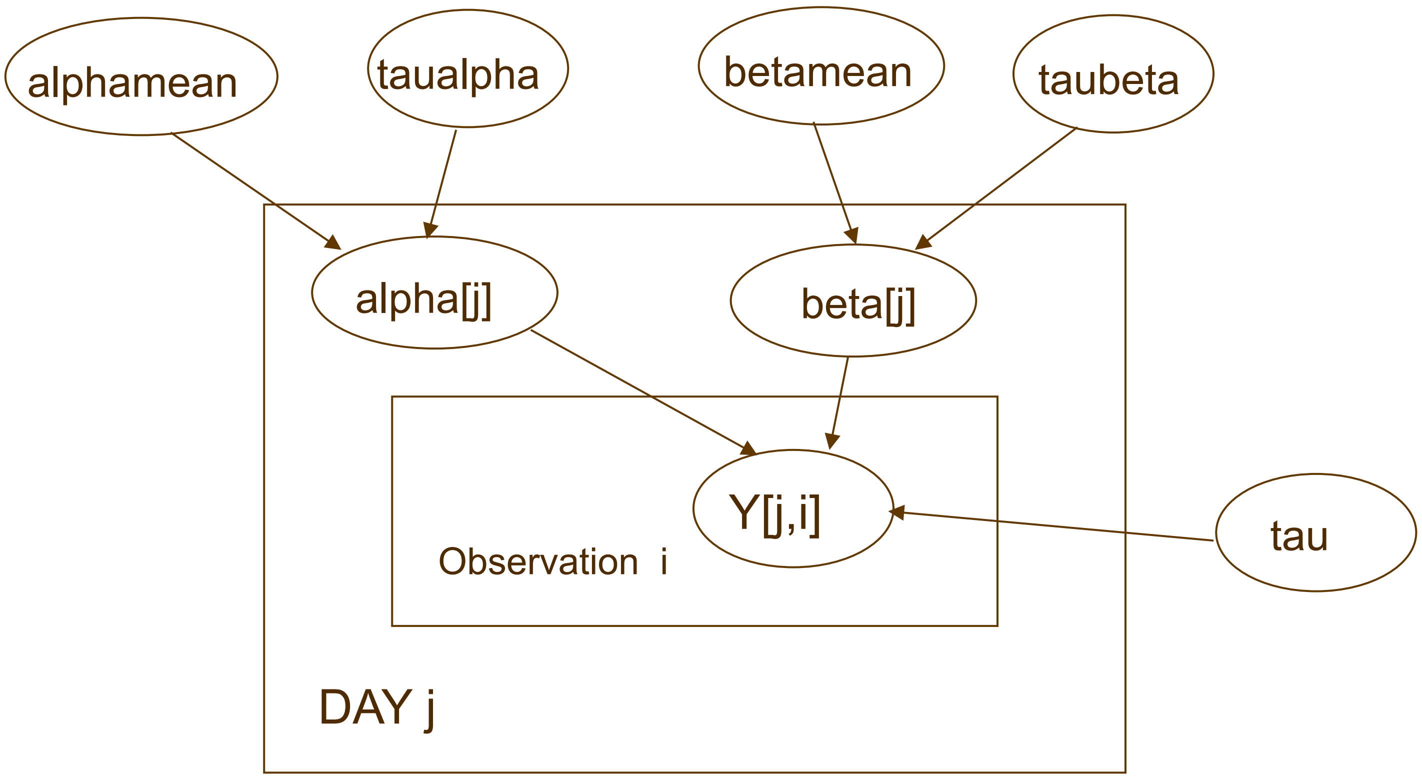
$$\beta_{mean} \sim N(0\ ,\ 10000\ )$$

$$\tau_{\alpha} \sim \mathrm{Gamma}(1{,}0.001) \quad \text{with } \tau_{\alpha} = 1/\sigma^2_{\alpha}$$

$$\tau_{\beta} \sim \mathrm{Gamma}(1{,}0.001) \quad \text{with } \tau_{\beta} = 1/\sigma^2_{\beta}$$

$$\tau \sim \mathrm{Gamma}(1{,}0.001) \quad \text{with } \tau = 1/\sigma^2$$

Graphical illustration



In WinBUGS

WinBUGS model

```
model{
for(i in 1:n){
        y[i]~dnorm(mu[i], tau)
        mu[i]<-alpha[serie[i]] + (beta[serie[i]]*(x[i]-mean(x[])))}

    for(j in 1:J){
        alpha[j]~dnorm(alphamean, taualpha)
        beta[j]~dnorm(betamean, taubeta) }

alphamean~dnorm(0, 0.0001)
 betamean~dnorm(0, 0.0001)

 tau~dgamma(1,0.001)

  taualpha~dgamma(1,0.001)
  taubeta~dgamma(1,0.001)

  sigma<-1/sqrt(tau)
  sigmaalpha<-1/sqrt(taualpha)
  sigmabeta<-1/sqrt(taubeta)}
```

Loop on the observations

Residual variance

Individual parameters normally distributed around the population mean with precision taualpha/ taubeta

Model

Prior for the mean parameters

Prior for the residual variability

Prior for the "inter-day variability"

Prior

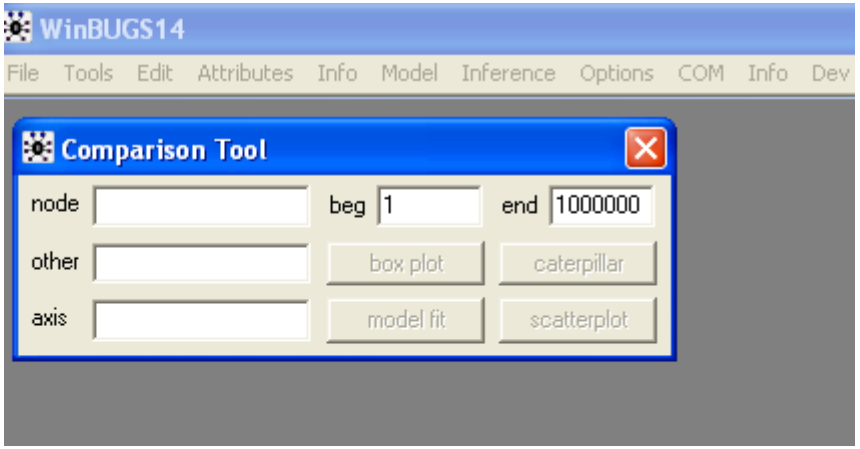
Derive parameters

Exercice

- Open data.txt, inits1.txt, inits2.txt (and model.txt)
- Run the model in WinBUGS with 1000 iterations for burnin, 5000 iterations for inference
- Monitor the parameters:
  - alpha, beta,
  - alphamean,betamean,
  - taualpha,taubeta,
  - mu,
  - tau

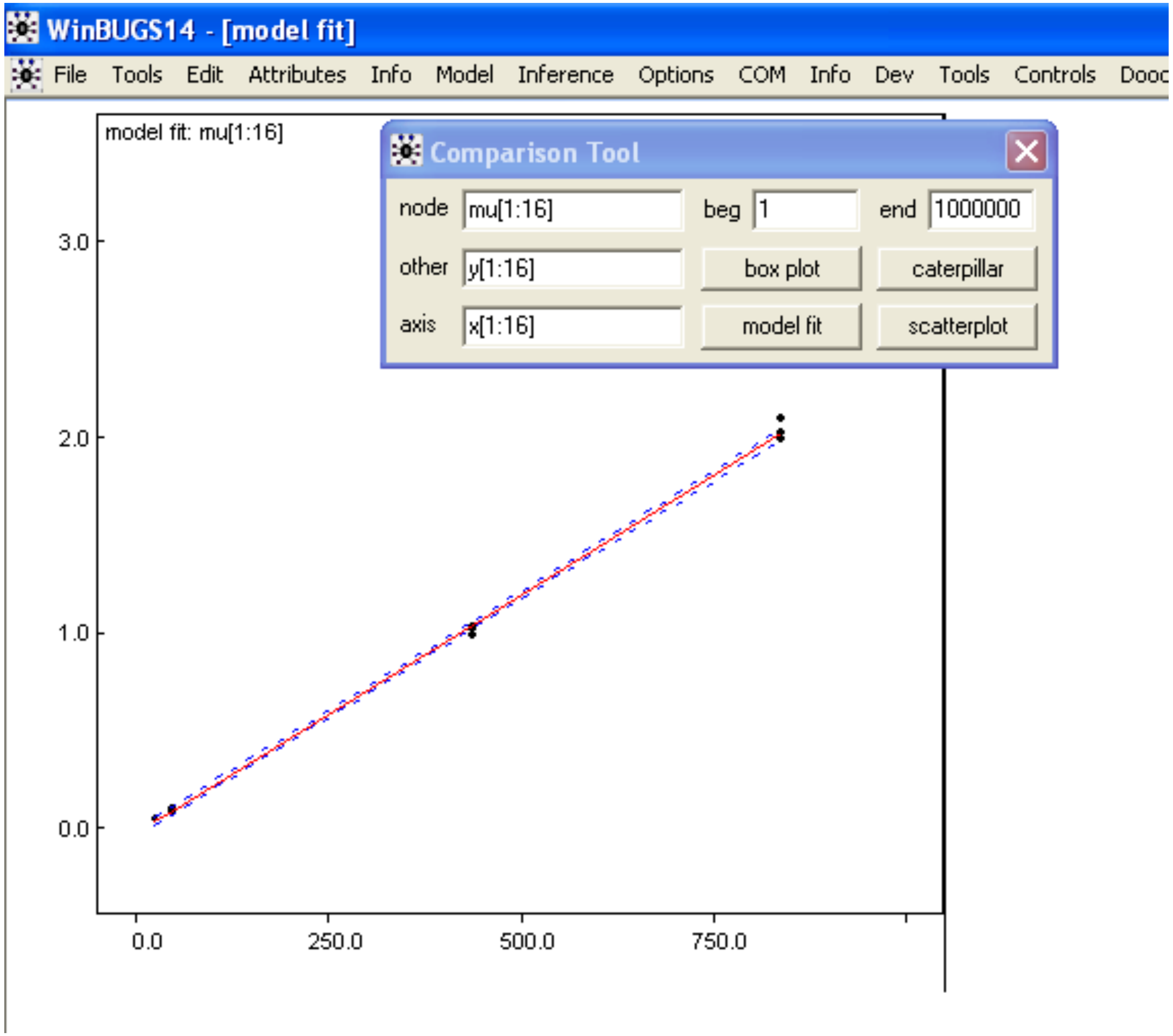
Exercice

- To check the fit for serie 1:
  - Go into: *Inference -> compare*



  - *In node: mu[1:16]*
  - *In other: y[1:16]*
  - *In axis: x[1:16]*
  - *Click on "model fit"*

Exercice


WinBUGS14 - [model fit]
File Tools Edit Attributes Info Model Inference Options COM Info Dev Tools Controls Dooc
model fit: mu[1:16]
Comparison Tool
node mu[1:16]
beg 1
end 1000000
other y[1:16]
box plot
caterpillar
axis x[1:16]
model fit
scatterplot
3.0
2.0
1.0
0.0
0.0
250.0
500.0
750.0

R2WinBUGS

R2WinBUGS: presentation

- Drawbacks with WinBUGS:
  - You have to write the data and initial values.
  - You have to specify the parameters to be monitored in each run.
  - The outputs are standards.
- Interesting to save the output and read it into R for further analyses :
  - R2WinBUGS allows WinBUGS to be run from R
  - Possibility to have the results of the MCMC and work from them (plot, convergence diagnostics…)

R2WinBUGS: steps

1- Create a .txt file

- The model is saved in a .txt file :

```
model{
   for(i in 1:n){
            y[i]~dnorm(mu[i], tau)
            mu[i]<-alpha[serie[i]] + (beta[serie[i]]*(x[i]-mean(x[])))}

         for(j in 1:J){
            alpha[j]~dnorm(alphamean, taualpha)
            beta[j]~dnorm(betamean, taubeta) }

 alphamean~dnorm(0, 0.0001)
 betamean~dnorm(0, 0.0001)

 tau~dgamma(1,0.001)
 taualpha~dgamma(1,0.001)
 taubeta~dgamma(1,0.001)

 sigma<-1/sqrt(tau)
 sigmaalpha<-1/sqrt(taualpha)
 sigmabeta<-1/sqrt(taubeta)}
```

R2WinBUGS: steps

- 2- In **R**, the working directory is the one where the model is saved :

setwd("C:\\BAYES2010\\Exercices\\WinBUGS_Part2\\Exercice”)

R2WinBUGS: steps

- 3- Load `the R2WinBUGS` packages :

```
library(R2WinBUGS)
```

- 4- Create the data for WinBUGS :

```
donnee=read.table("data_orig.txt",header=TRUE)
x=donnee$concentration
y=donnee$resp
serie=donnee$serie
n=length(y)

data <- list(n=n,J=3, x=x, y=y,serie=serie)
```

R2WinBUGS: steps

- 5- Load the initials values in a list :
  - One list for each set of initial values
  - “One list of the lists”

```
inits1=list(alphamean=0,betamean=1,tau=1,alpha=rep(0,3),beta=
  rep(1,3),taualpha=1,taubeta=1)
inits2=list(alphamean=0,betamean=1,tau=0.1,alpha=rep
  (0,3),beta=rep(0.5,3),taualpha=10,taubeta=10)
inits=list(inits1,inits2)
```

- 6- Specify the parameters to monitor in a list.

```
parameters=list("alpha","beta","tau","alphamean","betamean",
  "taualpha","taubeta")
```

R2WinBUGS: steps

- 7- Create a bugs object called "sims1"

```
sims1<-bugs(data=data,inits=inits,parameters=parameters,
   model.file="model.txt",n.chains=2,n.iter=5000,n.burnin=
   1000)
```

Outputs

Gabriele,
living with rheumatoid arthritis

R2WinBUGS: outputs

- names(sims1):

- "n.chains" "n.iter" "n.burnin”
- "sims.array" "sims.list" "sims.matrix" : all the iterations
- "summary" "mean" "sd" "median"
- "pD" "DIC"

R2WinBUGS: get the chains

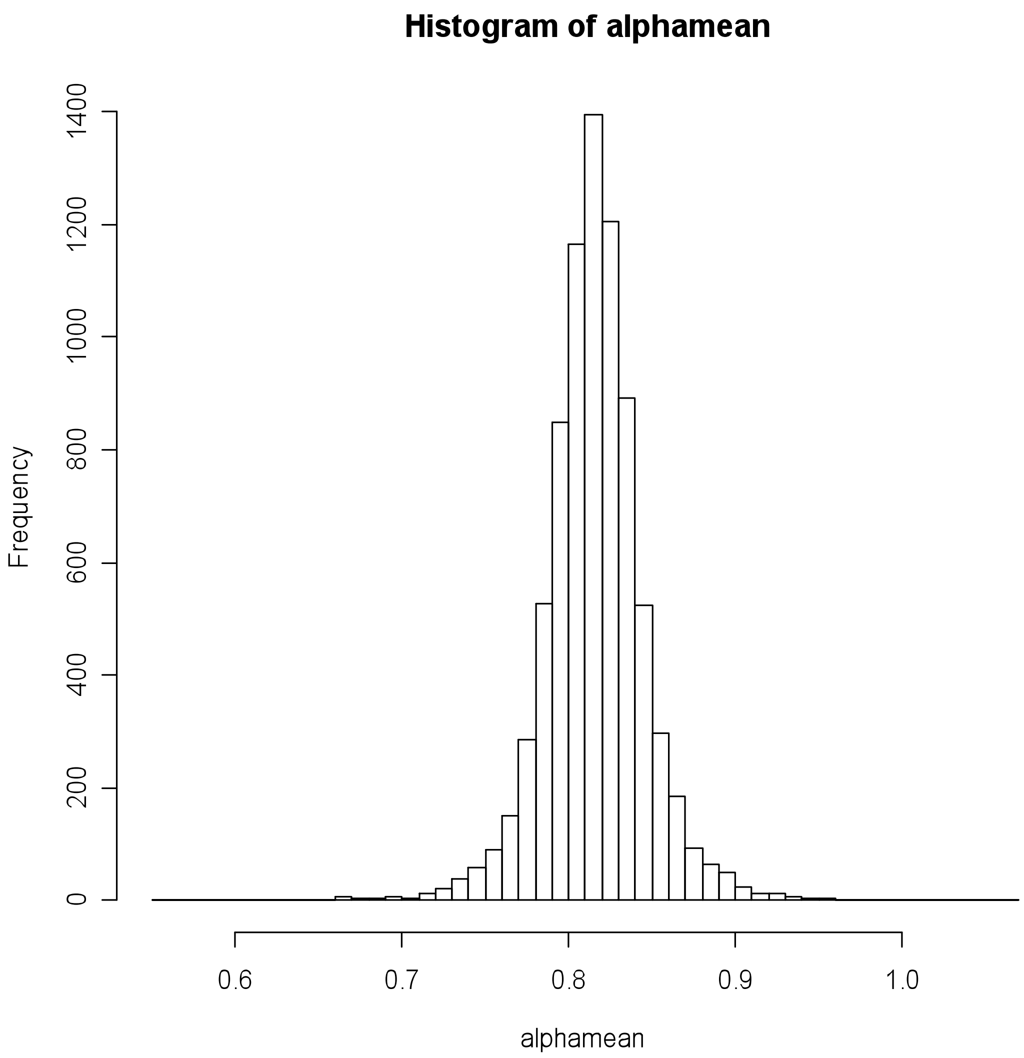
- 3 ways to get the chains :

```
dim(sims1$sims.array)
[1] 4000    2    12
dim(sims1$sims.matrix)
[1] 8000    12
 names(sims1$sims.list)
 "alpha"      "beta"       "tau"        "alphamean" "betamean"
   "taualpha"  "taubeta"   "deviance"
```

R2WinBUGS: Get the chains of the different parameters


```
alphamean<-sims1$sims.list$alphamean

betamean<-sims1$sims.list$betamean

alpha<-sims1$sims.list$alpha

beta<-sims1$sims.list$beta

tau<-sims1$sims.list$tau
```

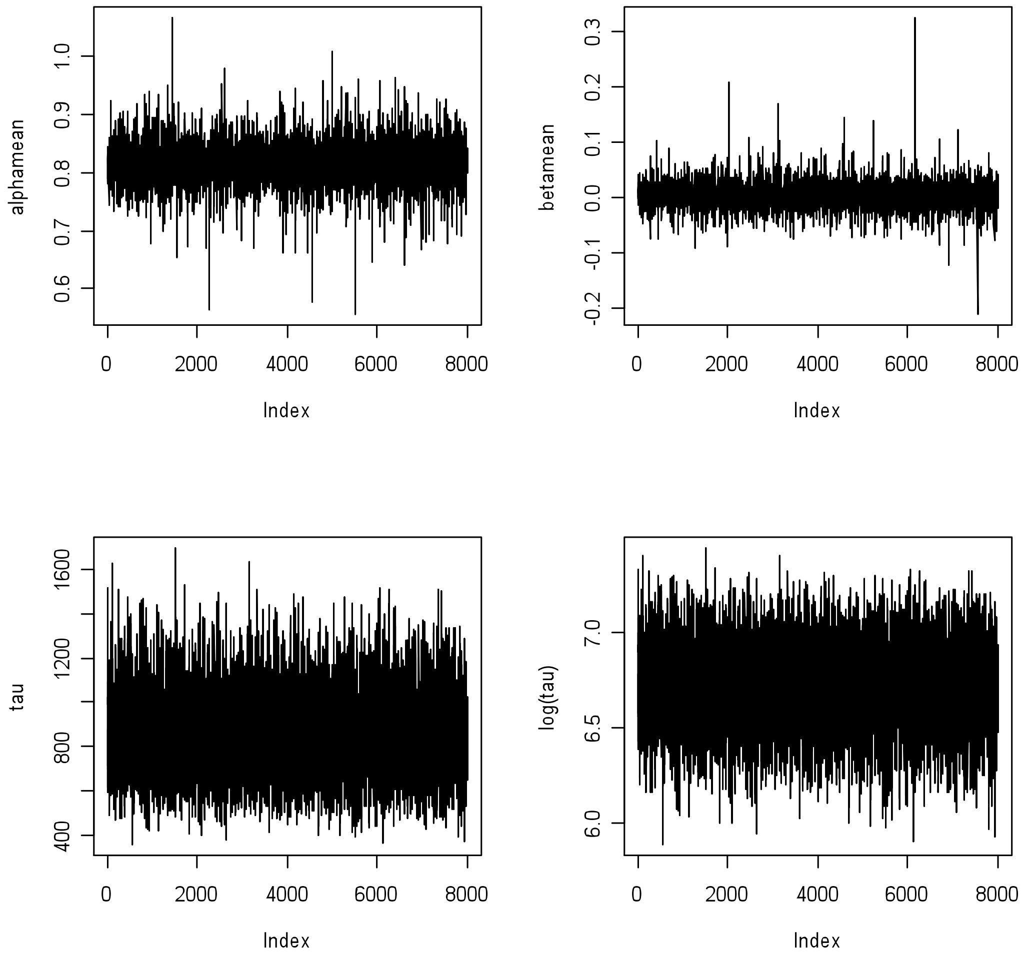
hist(alphamean)



R2WinBUGS: Draw some traces


```
par(mfrow=c(2,2))
plot(alphamean,type='l')
plot(betamean,type='l')
plot(tau,type='l')
plot(log(tau),type='l')
```

R2WinBUGS: Draw the traces



R2WinBUGS: fitted curves


```
#Compute the median a posteriori
alphaest<-apply(alpha,2,function(x){quantile(x,0.5)})
betaest<-apply(beta,2,function(x){quantile(x,0.5)})
alphameanest<-quantile(alphamean,0.5)
betameanest<-quantile(betamean,0.5)

#Graphical illustration
plot(data$x[serie==1],data$y[serie==1],type="n",xlab="x",ylab="y",xlim=c
    (0,900),ylim=c(0,2.5))
for (j in 1:3)
{
lines(data$x[serie==j],alphaest[j]+betaest[j]*(data$x[serie==j]),col=j)
points(data$x[serie==j],data$y[serie==j],col=j)
}

}
```

R2WinBUGS: fitted curves