An introduction to Bayesian modelling with JAGS and R

Francisco Rodriguez-Sanchez (@frod_san)

April 2015

This is a practical workshop

BUT do read the literature e.g.

- Data analysis using regression and multilevel/hierarchical models
- Bayesian data analysis
- Bayesian methods for ecology
- The BUGS book
- Introduction to WinBUGS for ecologists
- Models for ecological data
- and many more

Bayesian modelling software

- WinBUGS/OpenBUGS
- JAGS
- STAN
- Filzbach
- Nimble
- Many R packages: MCMCpack, MCMCglmm, LaplacesDemon, r-inla, etc (see Bayesian task view)

Why JAGS?

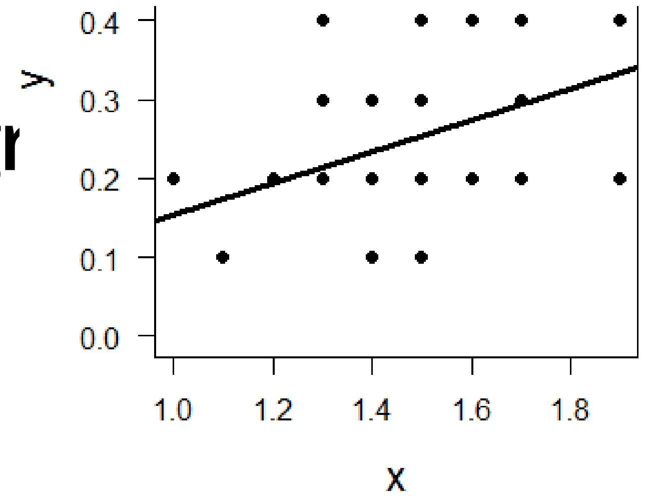
- Very similar to BUGS, both very popular
- Gate to other software e.g. STAN, Filzbach, etc
- Easy to start, can deal with complex models too (open-ended modelling)
- But look for specific implementations of your analysis (e.g. hSDM)

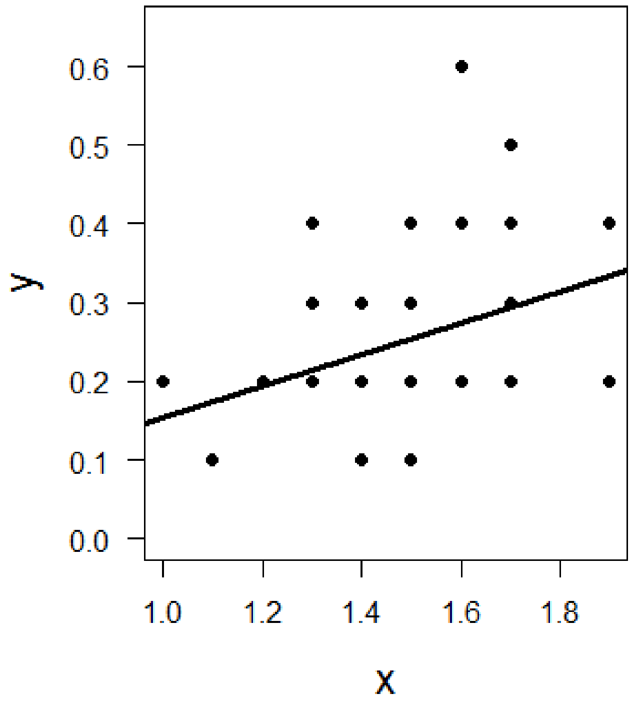
Why R?

Just kidding

:)

The very basics: linear regr

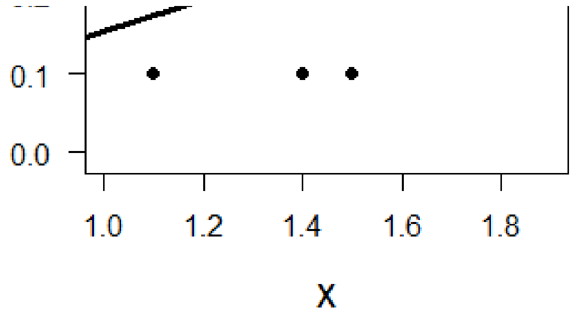




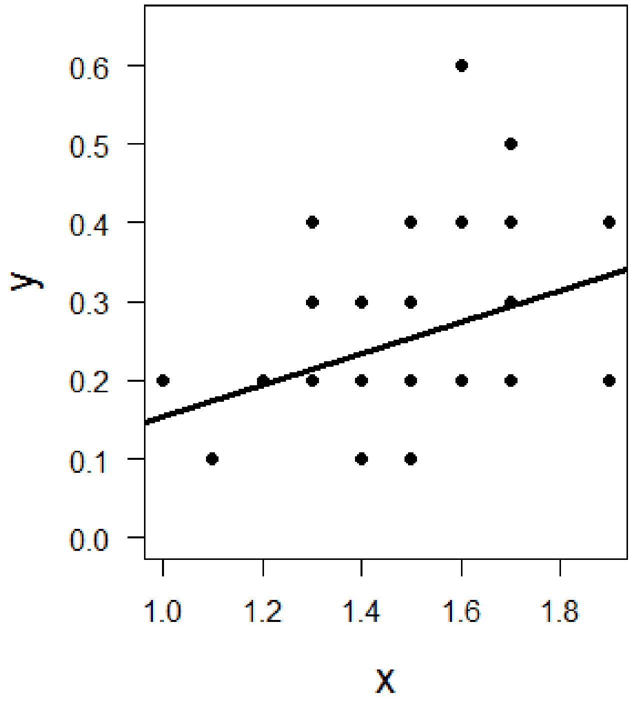
$$y_i = a + bx_i + \epsilon_i$$

How many parameters?

6/73



The very basics: linear regr



$$y_i = a + bx_i + \epsilon_i$$

$$\epsilon \sim N\left(0, \sigma^2\right)$$

Or also

$$y_i \sim N\left(\mu_i, \sigma^2\right)$$

$$\mu_i = a + bx_i$$

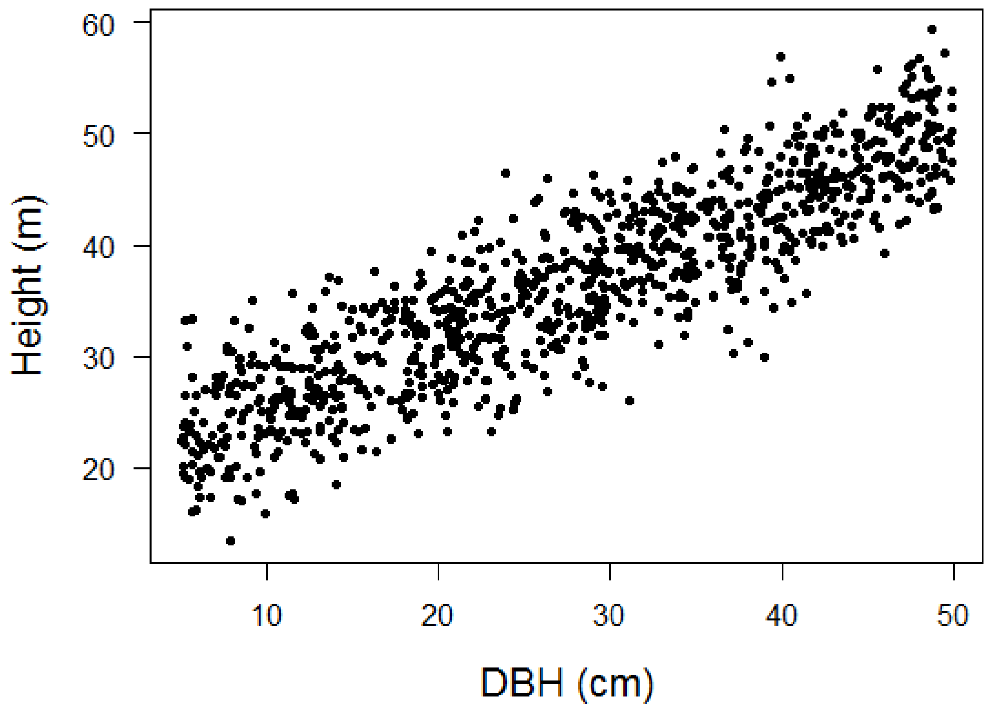
Our dataset: tree heights and DBH

- One species
- 10 plots
- 1000 trees
- Number of trees per plot ranging from 4 to 392

```
trees <- read.csv("trees.csv")
summary(trees[,1:3])
```

```
      plot           dbh              height     
 Min.   : 1.0   Min.   : 5.06   Min.   :13.40  
 1st Qu.: 1.0   1st Qu.:17.69   1st Qu.:29.68  
 Median : 2.0   Median :28.62   Median :36.55  
 Mean   : 2.7   Mean   :27.88   Mean   :36.51  
 3rd Qu.: 4.0   3rd Qu.:38.97   3rd Qu.:43.33  
 Max.   :10.0   Max.   :49.92   Max.   :59.30  
```

What's the relationship between DBH and height?



First step: linear regression (lm)

```
simple.lm <- lm(height ~ dbh, data=trees)
arm::display(simple.lm)
```

```
lm(formula = height ~ dbh, data = trees)
            coef.est coef.se
(Intercept) 19.34     0.31
dbh          0.62     0.01
---
n = 1000, k = 2
residual sd = 4.09, R-Squared = 0.79
```

Interpretation?

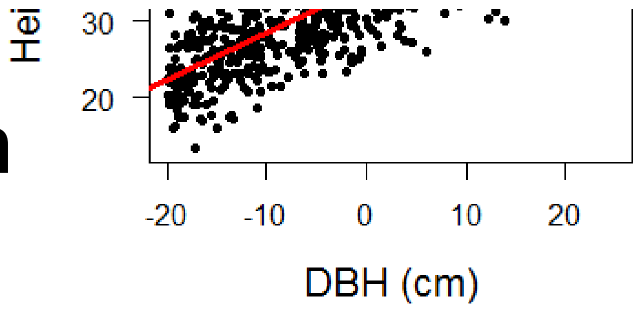
Always centre continuous variables

```
summary(trees$dbh)
```

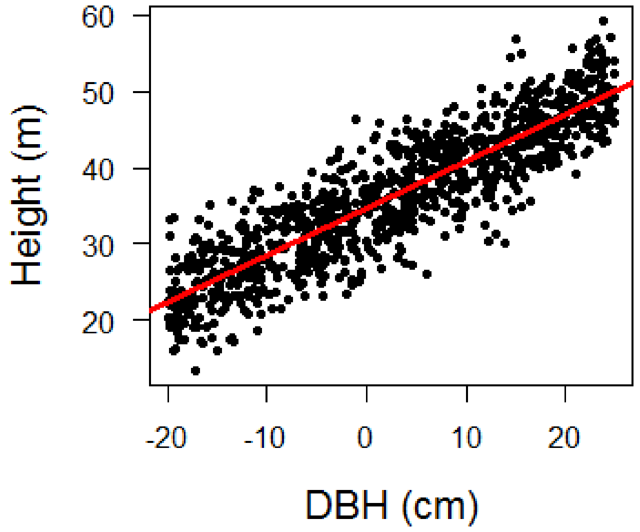
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.06   17.69   28.62   27.88   38.96   49.92
```

```
trees$dbh.c <- trees$dbh - 25
```

So, all parameters will be referred to a 25 cm DBH tree.



Linear regression with cen



```
lm(formula = height ~ dbh.c, data = trees)
            coef.est coef.se
(Intercept) 34.73    0.13
dbh.c        0.62    0.01
---
n = 1000, k = 2
residual sd = 4.09, R-Squared = 0.79
```

Let's make it Bayesian

Things we'll need

- Data
- A function describing the model (including **priors**)
- Decide number of MCMC chains
- Define initial values
- Decide number of iterations (and burnin)
- Choose parameters to save

Specify the model as an R function

```
model1 <- function(){

  # LIKELIHOOD
  for (i in 1:length(height)){
    height[i] ~ dnorm(mu[i], tau)    # tau = precision (inverse of variance)
    mu[i] <- alfa + beta*dbhc[i]     # centred diameter
  }

  # PRIORS (vague or weakly informative)
  alfa ~ dunif(1, 100)      # prior for average height of a 25-cm-DBH tree
  beta ~ dunif(0, 10)       # how much do we expect height to scale with DBH?
  tau <- pow(sigma, -2)       # tau = 1/sigma^2
  sigma ~ dunif(0, 50)      # residual standard deviation
}
```

A note on priors

Avoid 'non-informative' priors (see this and this)

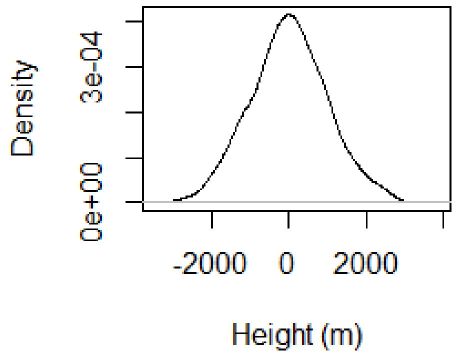
Use *weakly informative* (e.g. bounded Uniform, Normal with reasonable parameters, Cauchy…)

or *strongly informative* priors based on previous knowledge and common sense.

Example: estimating people height across countries

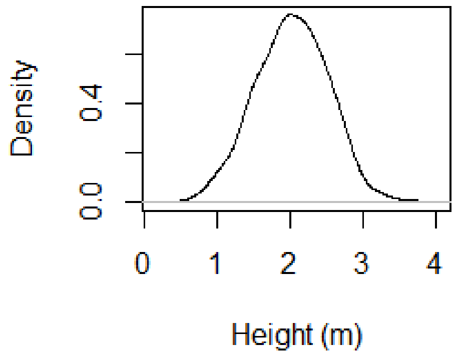
Unreasonable prior

```
plot(density(rnorm(1000, 0, 1000)),
     main="", xlab="Height (m)")
```



Reasonable prior

```
plot(density(rnorm(1000, 2, 0.5)),
     main="", xlab="Height (m)")
```



(*from STAN manual*)

Next step: create list with data

```
data <- list(height = trees$height,
             dbhc = trees$dbh.c)
```

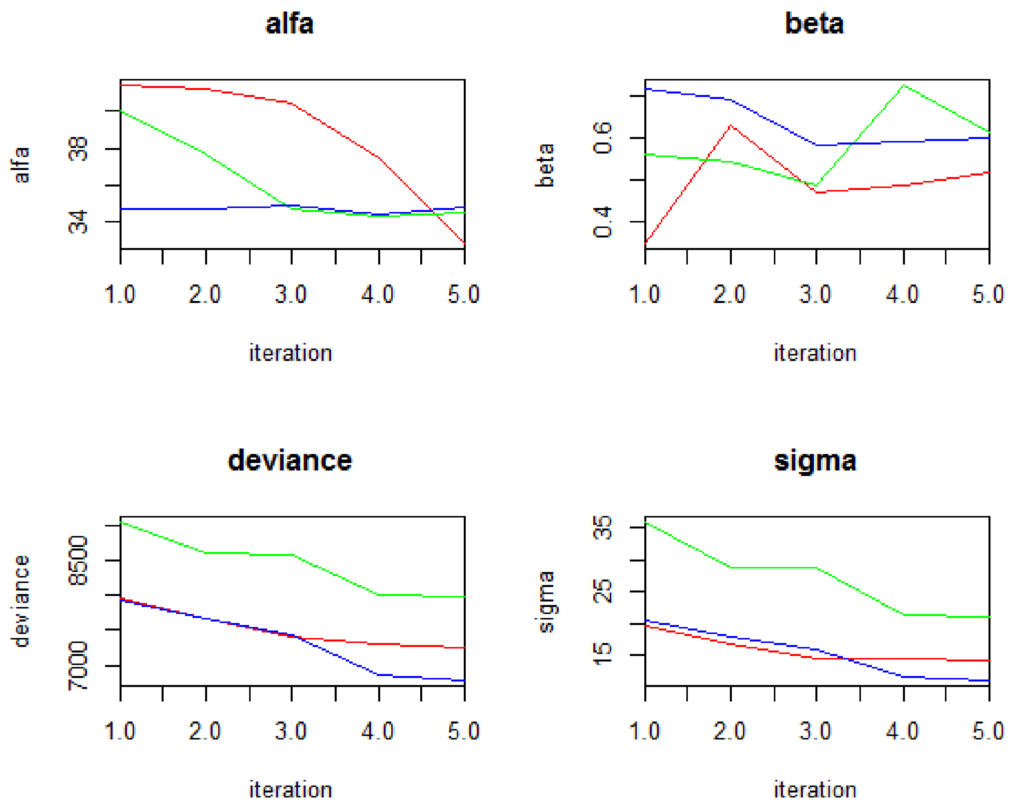
Now call JAGS to run the model

```
m1 <- jags(data,
           model.file=model1,
           parameters.to.save = c("alfa", "beta", "sigma"),
           n.chains=3,
           inits=NULL,
           n.iter=10,
           n.burnin=5)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 3787

Initializing model
```

Viewing MCMC in action

```
traceplot(m1, ask=FALSE, mfrow=c(2,2))
```



Obviously we haven't achieved convergence yet...

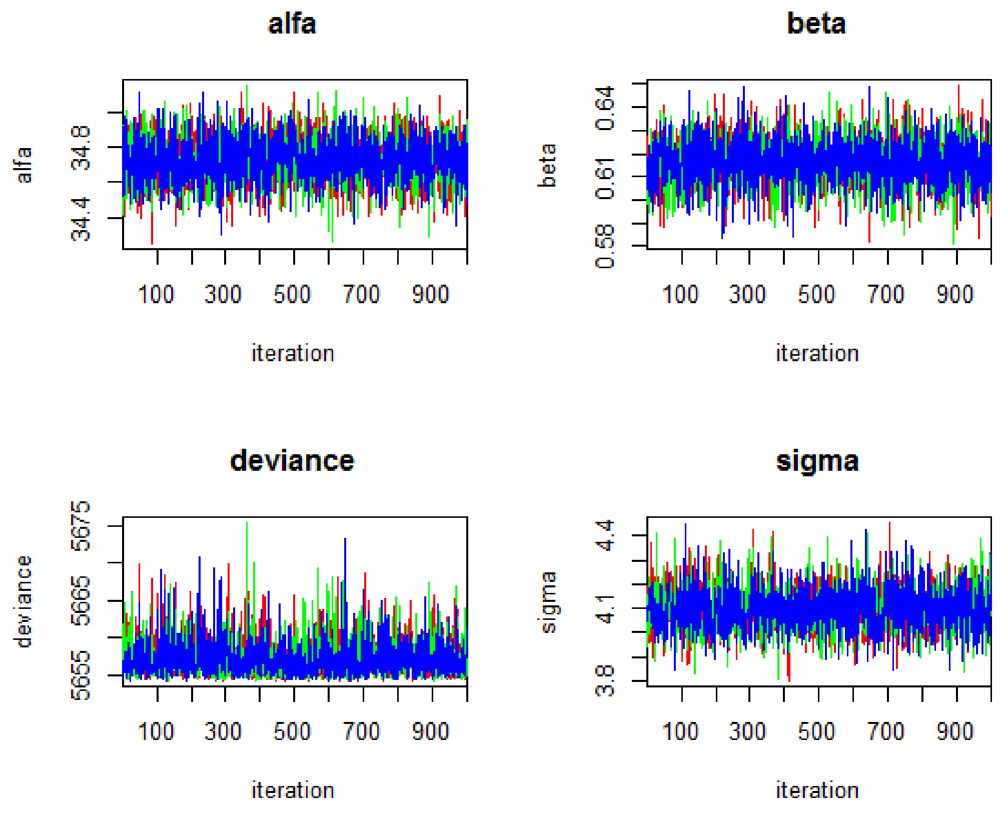
Let's run JAGS for longer

```
m1 <- jags(data,
           model.file=model1,
           parameters.to.save = c("alfa", "beta", "sigma"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 3787

Initializing model
```

Traceplots

```
traceplot(m1, ask=FALSE, mfrow=c(2,2))
```



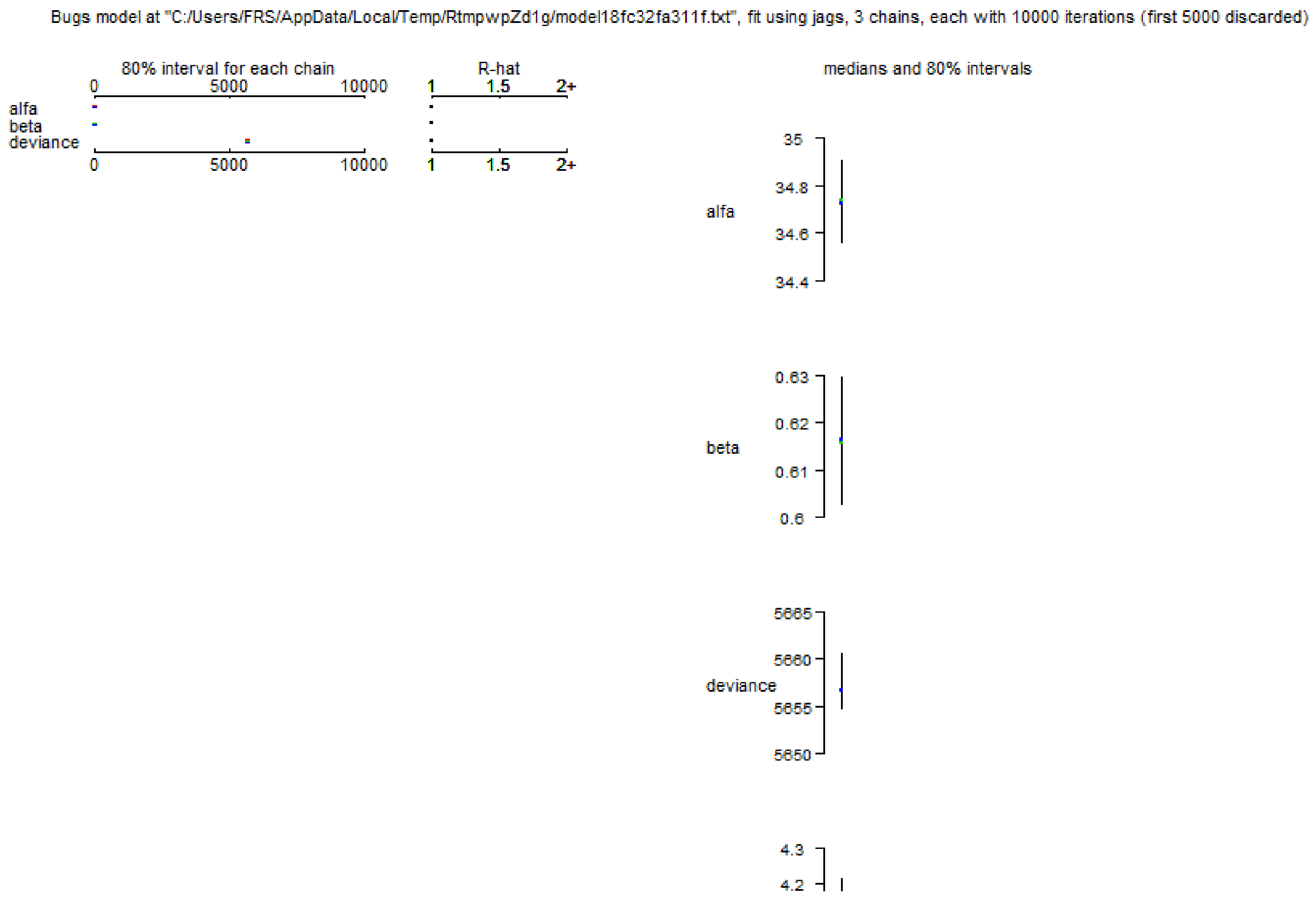
Results

```
Inference for Bugs model at "C:/Users/FRS/AppData/Local/Temp/RtmpwpZd1g/model18fc32fa311f.txt", f
 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
 n.sims = 3000 iterations saved
         mu.vect sd.vect     2.5%    97.5%  Rhat n.eff
alfa      34.729   0.134   34.464   34.980 1.001  3000
beta       0.616   0.010    0.596    0.636 1.002  1700
sigma      4.099   0.092    3.926    4.280 1.001  3000
deviance 5657.303   2.529 5654.453 5663.886 1.001  2300

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 = 3.2 and DIC = 5660.5
DIC is an estimate of expected predictive error (lower deviance is better).
```

Results pretty similar to simple.lm (because of vague priors)

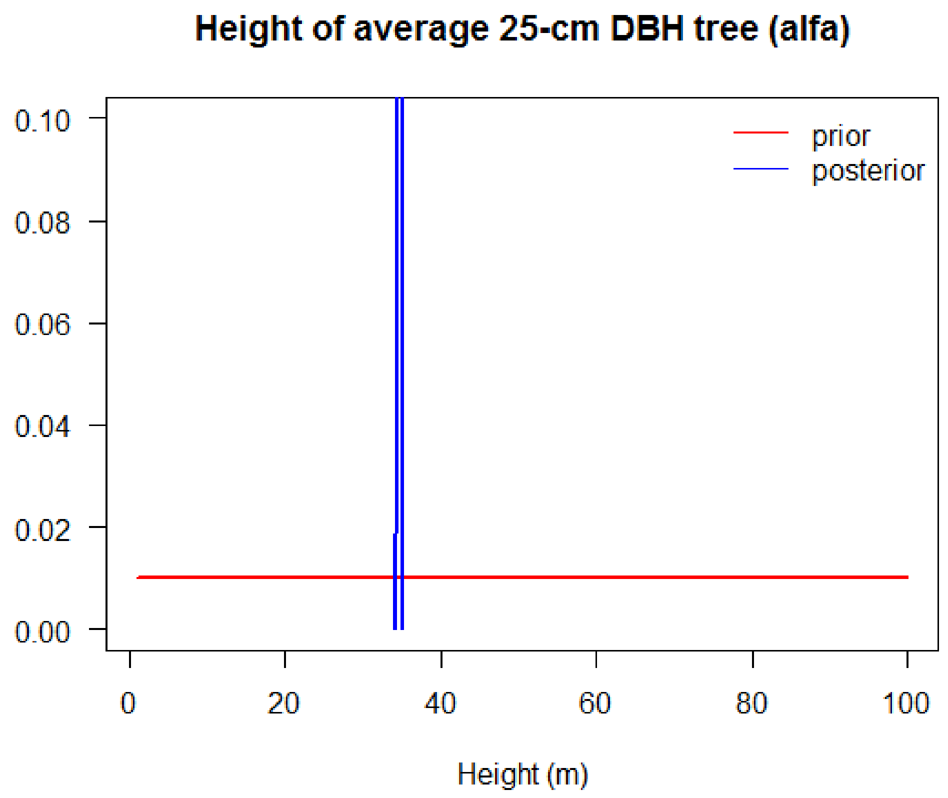
A plot of the whole model

```
plot(m1)
```

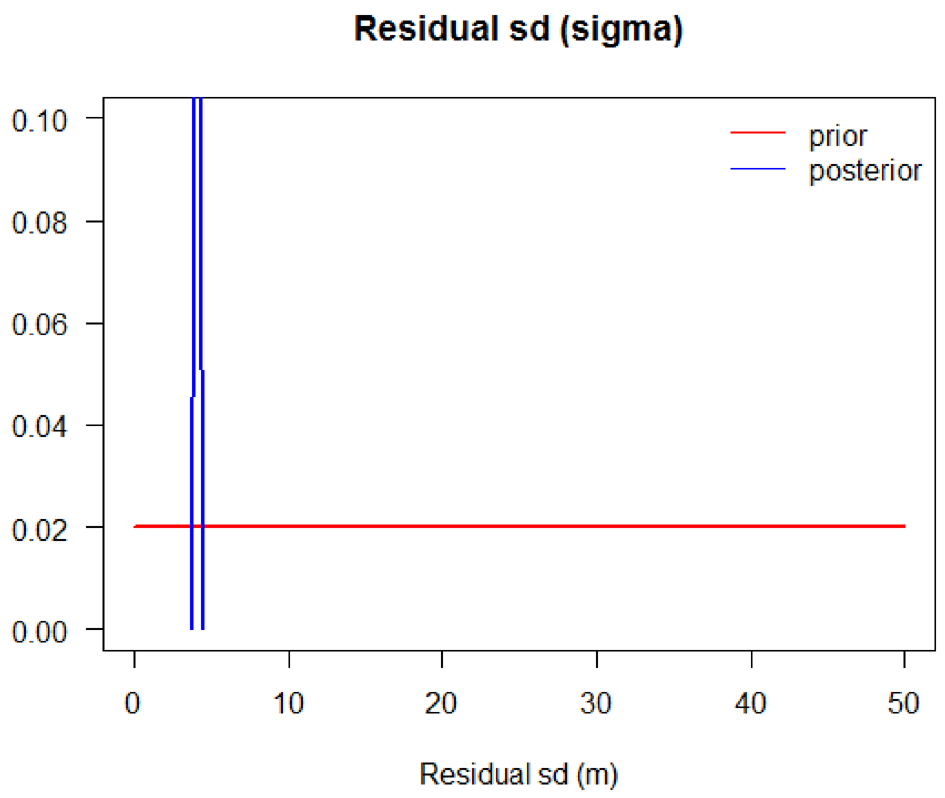


23/73

Comparing prior and posterior densities



Comparing prior and posterior densities



Now using Normal vague priors

Model with Normal priors

```
model1b <- function(){

  # LIKELIHOOD
  for (i in 1:length(height)){
    height[i] ~ dnorm(mu[i], tau)    # tau = precision (inverse of variance)
    mu[i] <- alfa + beta*dbhc[i]     # centred diameter
  }

  # PRIORS
  alfa ~ dnorm(0, 0.001)      # prior for intercept
  beta ~ dnorm(0, 0.001)       # prior for beta (slope)
  tau <- pow(sigma, -2)       # tau = 1/sigma^2
  sigma ~ dunif(0, 50)      # residual standard deviation
}
```

Calling JAGS

```
m1 <- jags(data,
           model.file=model1b,
           parameters.to.save = c("alfa", "beta", "sigma"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 3785

Initializing model
```

Results

```
Inference for Bugs model at "C:/Users/FRS/AppData/Local/Temp/RtmpwpZd1g/model18fcecc61de.txt", fi
 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
 n.sims = 3000 iterations saved
         mu.vect sd.vect     2.5%    97.5%  Rhat n.eff
alfa      34.732   0.131   34.470   34.980 1.001  3000
beta       0.615   0.010    0.595    0.636 1.003   950
sigma      4.098   0.095    3.918    4.286 1.001  3000
deviance 5657.292   2.526 5654.481 5664.219 1.004   600

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 = 3.2 and DIC = 5660.5
DIC is an estimate of expected predictive error (lower deviance is better).
```

Very similar

29/73

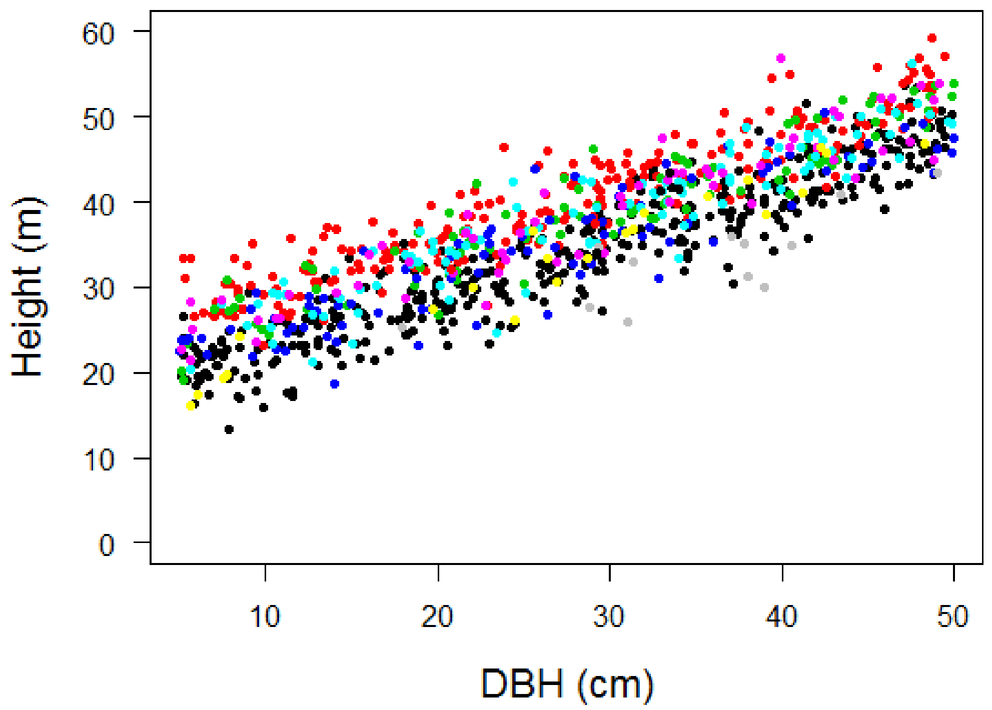
Bayesian inference

$$p(\theta \mid x) = \frac{p(x \mid \theta) p(\theta)}{p(x)}$$

$$\text{Posterior distribution} \propto \text{Likelihood} \times \text{Prior distribution}$$

Varying-intercept models

Accounting for plot effects



Do it yourself using lm

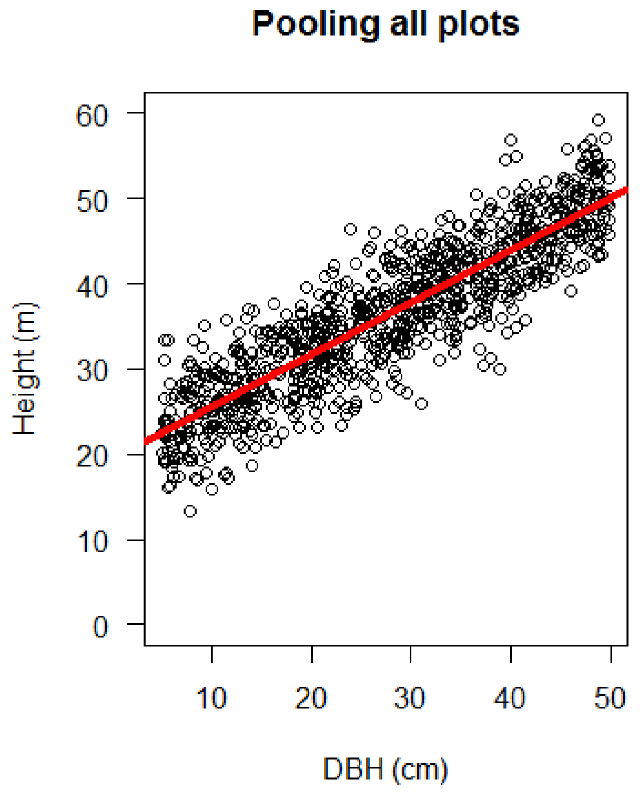
lm results

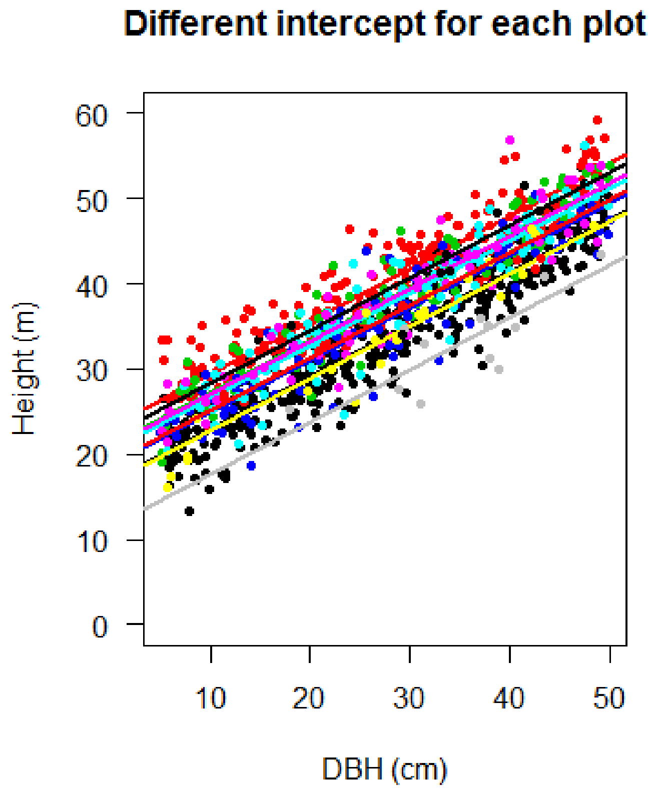
```
lm.plot <- lm(height ~ factor(plot) + dbh.c, data=trees)
```

```
lm(formula = height ~ factor(plot) + dbh.c, data = trees)
               coef.est coef.se
(Intercept)    32.13     0.16  
factor(plot)2   6.50     0.26  
factor(plot)3   4.36     0.35  
factor(plot)4   1.93     0.36  
factor(plot)5   3.64     0.34  
factor(plot)6   4.20     0.42  
factor(plot)7  -0.18     0.67  
factor(plot)8  -5.31     0.89  
factor(plot)9   5.44     1.09  
factor(plot)10  2.26     1.37  
dbh.c           0.62     0.01  
---
n = 1000, k = 11
residual sd = 3.04, R-Squared = 0.88
```

Interpretation?

Single vs varying intercept





Let's make it Bayesian

Things we'll need

- Data
- A function describing the model (including **priors**)
- number of MCMC chains
- initial values
- number of iterations (and burnin)
- parameters to save

Bayesian varying-intercept model with no pooling

```
model2 <- function(){
  # LIKELIHOOD
  for (i in 1:length(height)){
    height[i] ~ dnorm(mu[i], tau)    # tau = precision (inverse of variance)
    mu[i] <- alfa[plot[i]] + beta*dbhc[i]     # centred diameter
  }
  # PRIORS
  #alfa ~ dnorm(0, .001)
  for (j in 1:10){
    alfa[j] ~ dnorm(0, .001)  # Plot effects drawn from Normal distribution
                              # with large **fixed** variance
  }
  beta ~ dnorm(0, .001)
  tau <- pow(sigma, -2)       # tau = 1/sigma^2
  sigma ~ dunif(0, 50)
}
```

This fits same model as `lm.plot`

Call JAGS

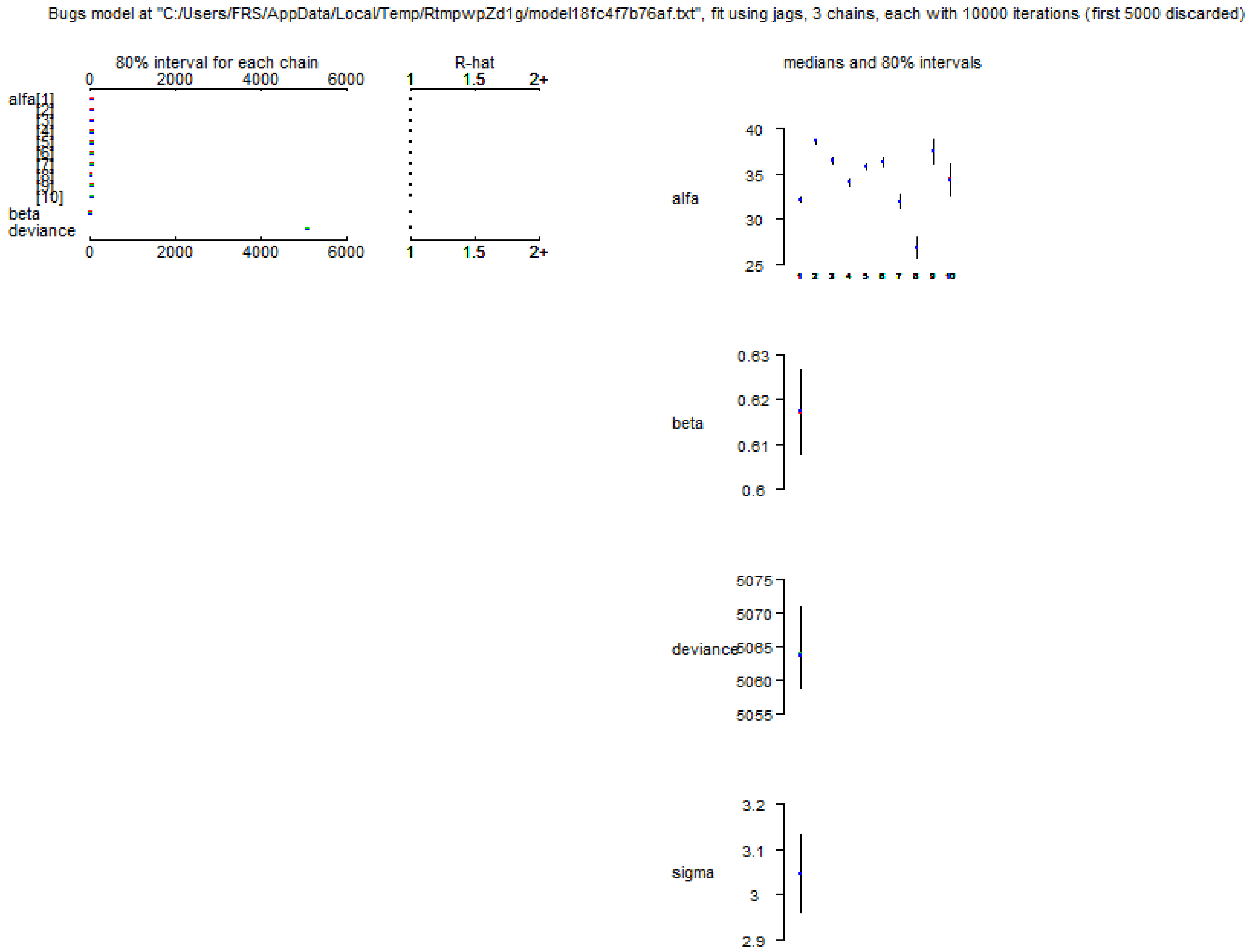
```
data <- list(height=trees$height,
             dbhc=trees$dbh.c,
             plot=trees$plot)
m2 <- jags(data,
           model.file=model2,
           parameters.to.save = c("alfa", "beta", "sigma"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 4880

Initializing model
```

Results

```
Inference for Bugs model at "C:/Users/FRS/AppData/Local/Temp/RtmpwpZd1g/model18fc4f7b76af.txt", f
 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
 n.sims = 3000 iterations saved
          mu.vect sd.vect     2.5%    97.5%  Rhat n.eff
alfa[1]    32.126   0.152   31.828   32.420 1.001  3000
alfa[2]    38.628   0.208   38.219   39.037 1.003   750
alfa[3]    36.472   0.315   35.871   37.085 1.001  2700
alfa[4]    34.058   0.319   33.425   34.683 1.002  1000
alfa[5]    35.760   0.310   35.137   36.372 1.002  1500
alfa[6]    36.314   0.399   35.571   37.082 1.001  3000
alfa[7]    31.949   0.640   30.680   33.210 1.002  1800
alfa[8]    26.790   0.870   25.086   28.452 1.001  3000
alfa[9]    37.498   1.057   35.490   39.596 1.001  3000
alfa[10]   34.322   1.384   31.648   37.016 1.002  1900
beta        0.617   0.007    0.602    0.632 1.001  3000
sigma       3.047   0.068    2.915    3.188 1.001  3000
deviance 5064.492   4.914 5056.952 5075.810 1.001  3000

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)
```

Plot whole model

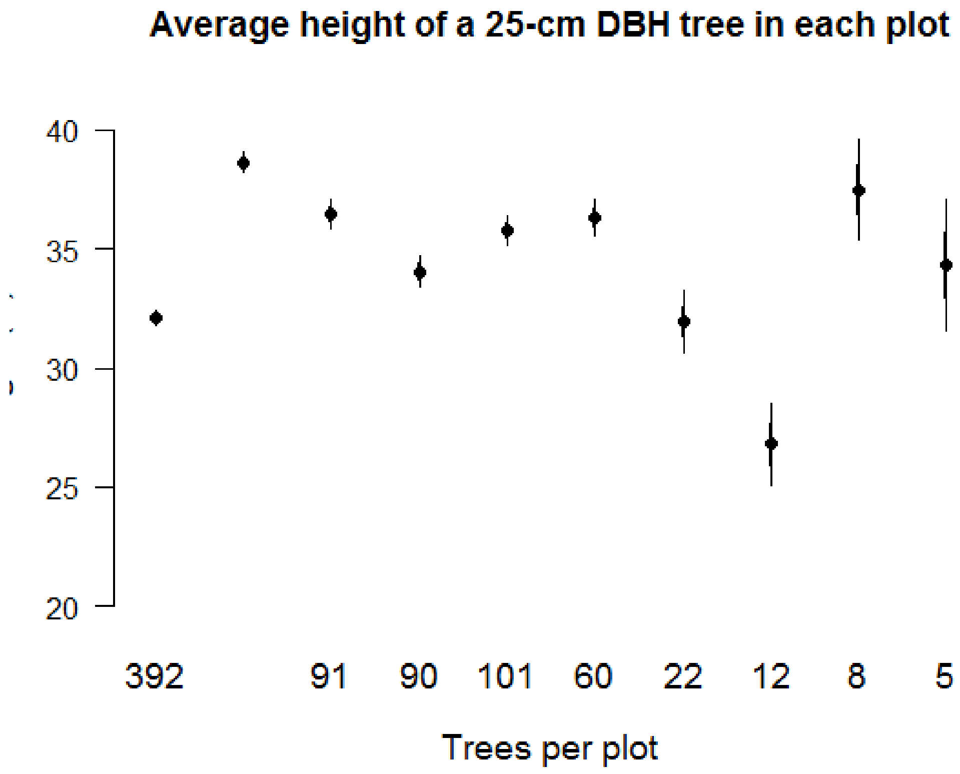


The varying-intercept model is much better

DIC(m1) = 5660

DIC(m2) = 5077

Estimation of plot effects improves with sample size



Varying-intercepts with pooling

(mixed/multilevel/hierarchical model)

Multilevel model with varying intercepts

$$\begin{aligned} y_i &= a_j + bx_i + \varepsilon_i \\ a_j &\sim N\left(0, \tau^2\right) \\ \varepsilon_i &\sim N\left(0, \sigma^2\right) \end{aligned}$$

In our example:

$$\begin{aligned} Height_i &= plot_j + bDBH_i + \varepsilon_i \\ plot_j &\sim N\left(0, \tau^2\right) \\ \varepsilon_i &\sim N\left(0, \sigma^2\right) \end{aligned}$$

Fitting mixed models with lmer

```
mixed <- lmer(height ~ dbh.c + (1|plot), data = trees)
```

```
lmer(formula = height ~ dbh.c + (1 | plot), data = trees)
            coef.est coef.se
(Intercept) 34.43    1.08
dbh.c        0.62    0.01

Error terms:
 Groups   Name        Std.Dev.
 plot     (Intercept) 3.35
 Residual             3.04
---
number of obs: 1000, groups: plot, 10
AIC = 5116.3, DIC = 5096.3
deviance = 5102.3
```

lmer coefficients

```
coef(mixed)
```

```
$plot
   (Intercept)     dbh.c
1     32.13118 0.6169271
2     38.61479 0.6169271
3     36.46547 0.6169271
4     34.06404 0.6169271
5     35.75313 0.6169271
6     36.30517 0.6169271
7     32.04003 0.6169271
8     27.30620 0.6169271
9     37.27097 0.6169271
10    34.39546 0.6169271

attr(,"class")
[1] "coef.mer"
```

Bayesian varying-intercept model with pooling across plots

```
model3 <- function(){
    # LIKELIHOOD
  for (i in 1:length(height)){
    height[i] ~ dnorm(mu[i], tau)    # tau = precision (inverse of variance)
    mu[i] <- alfa[plot[i]] + beta*dbhc[i]     # centred diameter
  }
    # PRIORS
  for (j in 1:10){
    alfa[j] ~ dnorm(grandmu, tauplot)   # Now we are estimating the plot variance!
  }
  grandmu ~ dnorm(0, .001)     # Overall mean height across all plots
  tauplot <- pow(sigmaplot, -2)
  sigmaplot ~ dunif(0, 20)    # between-plot variance
  beta ~ dnorm(0, .001)
  tau <- pow(sigma, -2)
  sigma ~ dunif(0, 50)     # residual variance
}
```

Call JAGS

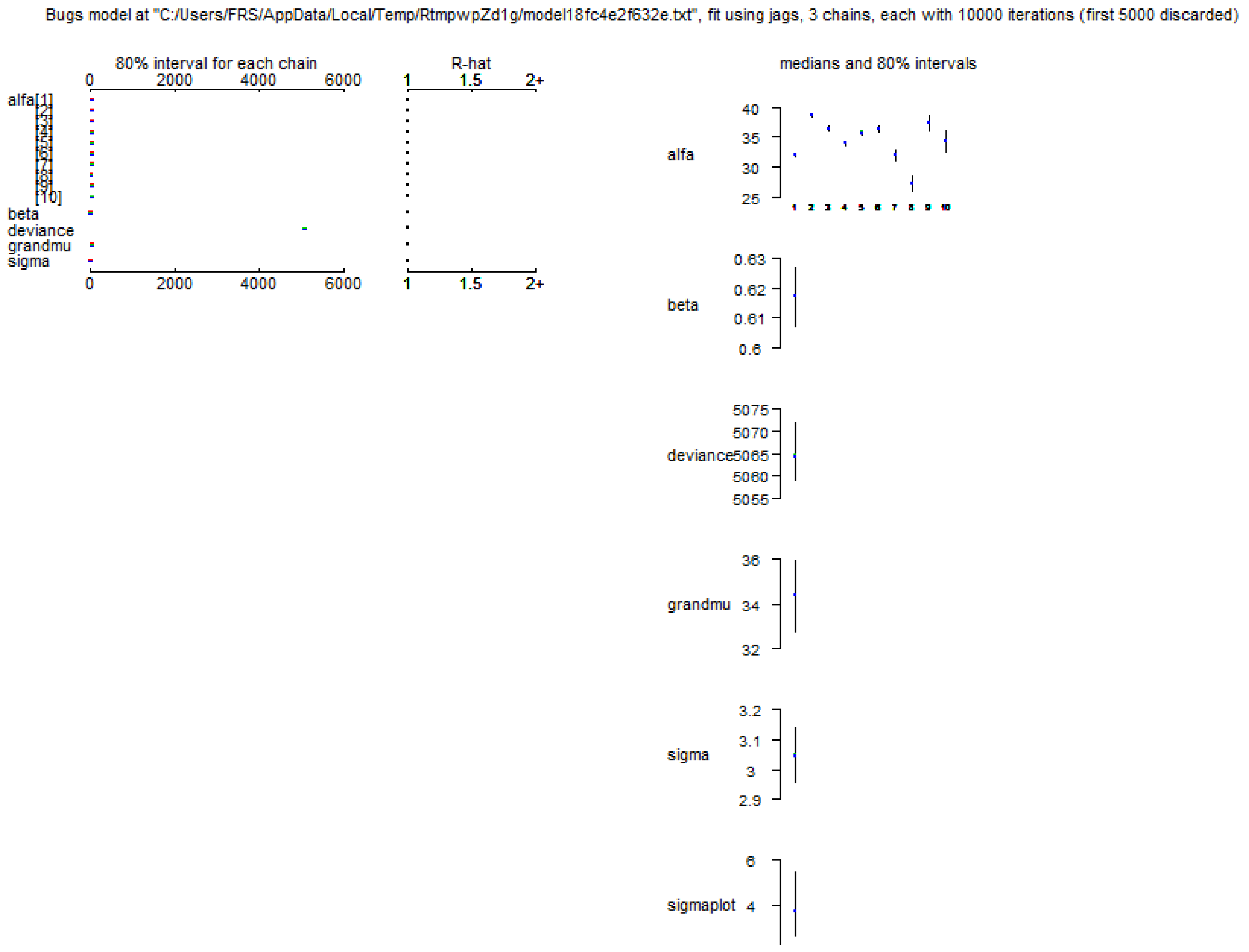
```
data <- list(height=trees$height,
             dbhc=trees$dbh.c,
             plot=trees$plot)
m3 <- jags(data,
           model.file=model3,
           parameters.to.save = c("alfa", "beta", "sigma", "grandmu", "sigmaplot"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 4884

Initializing model
```

Results

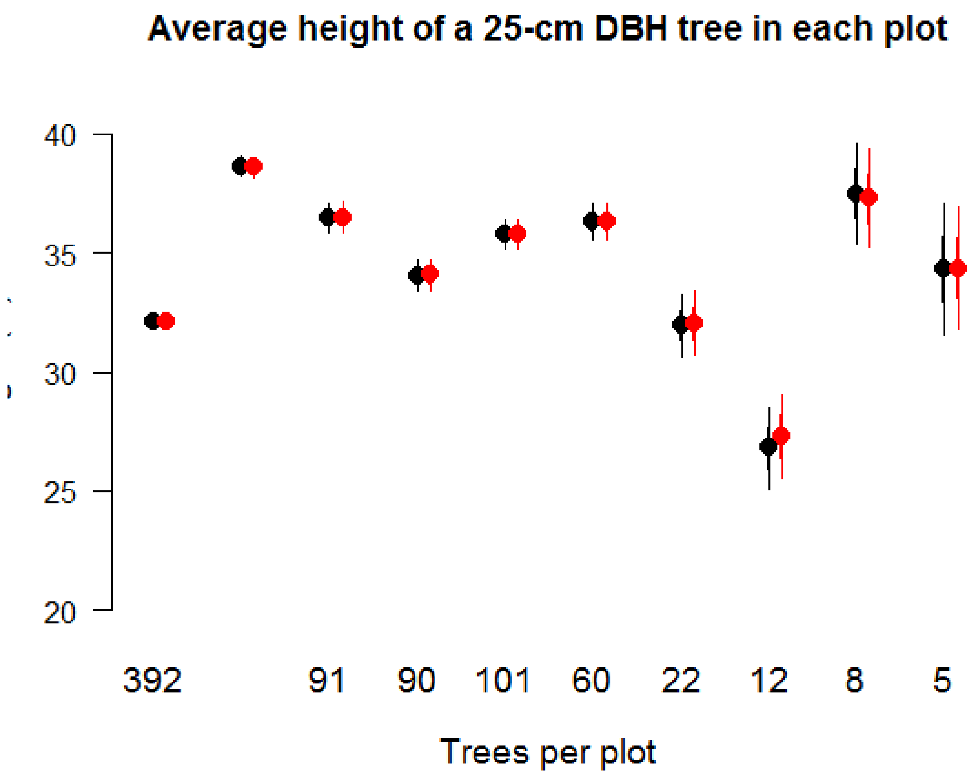
```
Inference for Bugs model at "C:/Users/FRS/AppData/Local/Temp/RtmpwpZd1g/model18fc4e2f632e.txt", f
 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
 n.sims = 3000 iterations saved
           mu.vect sd.vect     2.5%    97.5%  Rhat n.eff
alfa[1]     32.131   0.156   31.822   32.421 1.001  3000
alfa[2]     38.611   0.211   38.206   39.029 1.002  1400
alfa[3]     36.474   0.325   35.835   37.105 1.001  2100
alfa[4]     34.068   0.318   33.448   34.694 1.001  3000
alfa[5]     35.746   0.305   35.158   36.352 1.002  1500
alfa[6]     36.305   0.392   35.499   37.068 1.001  3000
alfa[7]     32.044   0.667   30.731   33.345 1.001  3000
alfa[8]     27.288   0.896   25.541   29.064 1.002  1400
alfa[9]     37.294   1.037   35.254   39.425 1.001  3000
alfa[10]    34.363   1.281   31.897   36.861 1.001  3000
beta         0.617   0.008    0.602    0.632 1.001  2700
grandmu     34.336   1.304   31.599   36.871 1.001  3000
sigma        3.048   0.071    2.916    3.186 1.001  3000
sigmaplot    3.942   1.237    2.304    6.982 1.001  3000
deviance  5064.988   5.105 5057.256 5076.492 1.002  1500

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).
```

A plot of the whole model



49/73

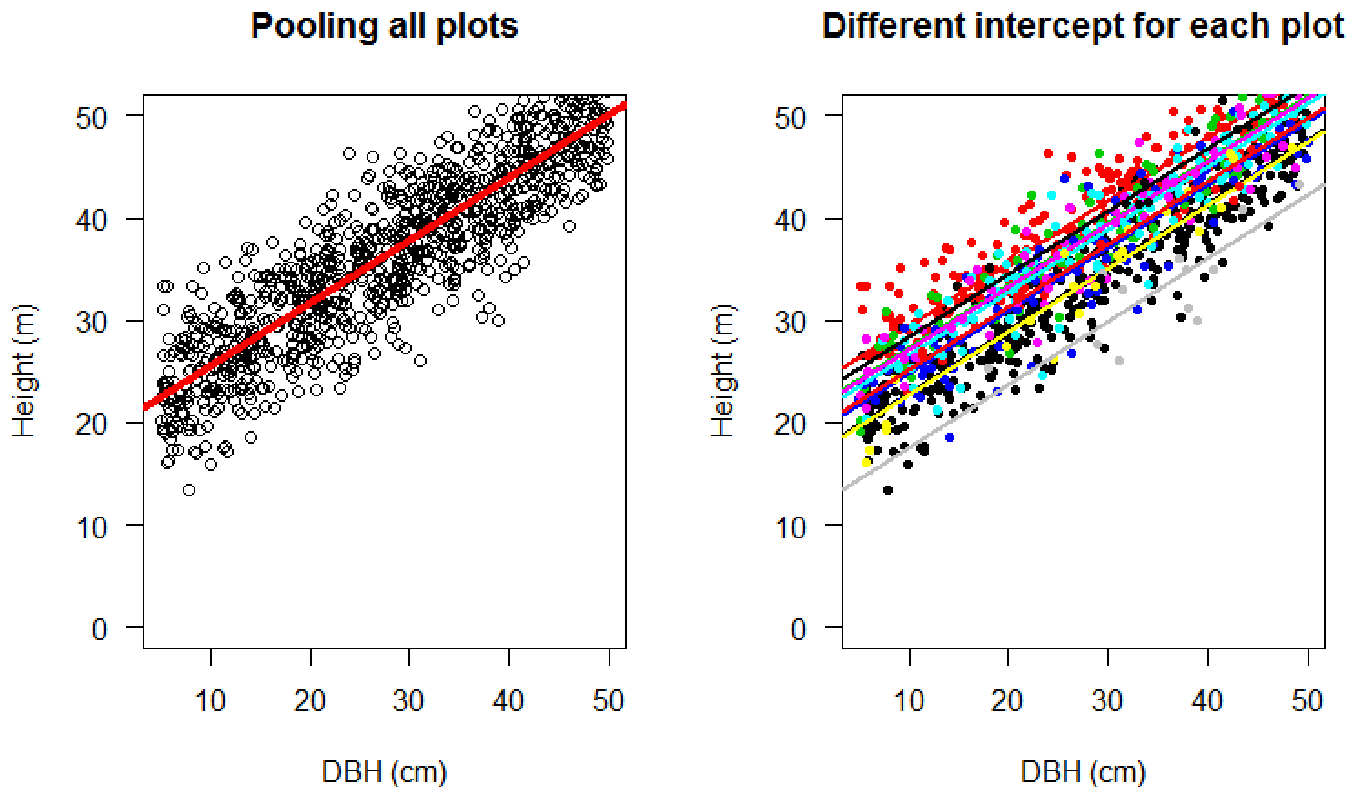
Comparing plot coefficients



50/73

A gradient from complete to no pooling

The multilevel model (with pooling) is somewhere between the complete-pooling (single intercept) and the no-pooling (one intercept for each plot, without shrinkage) models.



51/73

Growing the hierarchy: adding plot-level predictors

Model with group-level predictors

We had:

$$\begin{aligned} y_i &= a_j + bx_i + \varepsilon_i \\ a_j &\sim N\left(0, \tau^2\right) \\ \varepsilon_i &\sim N\left(0, \sigma^2\right) \end{aligned}$$

Now

$$\begin{aligned} y_i &= a_j + bx_i + \varepsilon_i \\ a_j &\sim N\left(\mu_j, \tau^2\right) \\ \mu_j &= \gamma + \delta \cdot predictor_j \\ \varepsilon_i &\sim N\left(0, \sigma^2\right) \end{aligned}$$

Reading plot data

```
plotdata <- read.csv("plotdata.csv")
temp.c <- plotdata$temp - 15
```

Model with group-level predictors

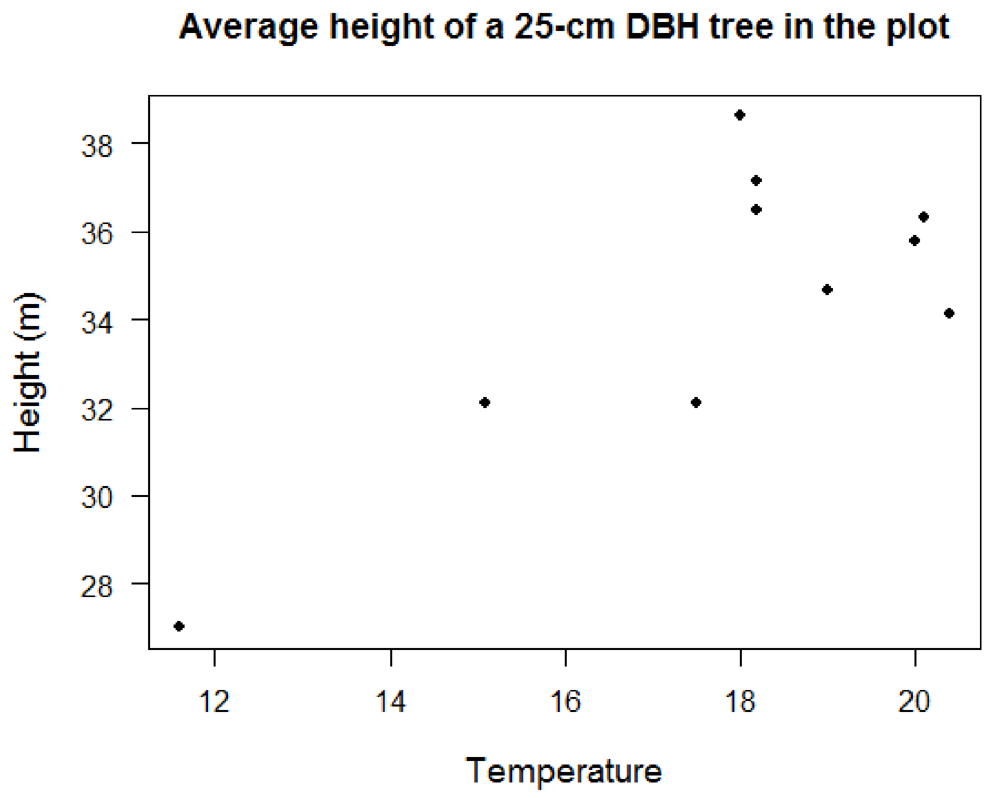
```
model4 <- function(){
  # LIKELIHOOD
  for (i in 1:length(height)){
    height[i] ~ dnorm(mu[i], tau)
    mu[i] <- alfa[plot[i]] + beta*dbhc[i]
  }
  # PRIORS
  for (j in 1:10){
    alfa[j] ~ dnorm(grandmu + beta.temp*tempc[j], tauplot)
  }
  beta.temp ~ dnorm(0, .001)   # slope for temperature effects
  grandmu ~ dnorm(0, .001)
  tauplot <- pow(sigmaplot, -2)
  sigmaplot ~ dunif(0, 20)
  beta ~ dnorm(0, .001)
  tau <- pow(sigma, -2)
  sigma ~ dunif(0, 50)
}
```

running JAGS…

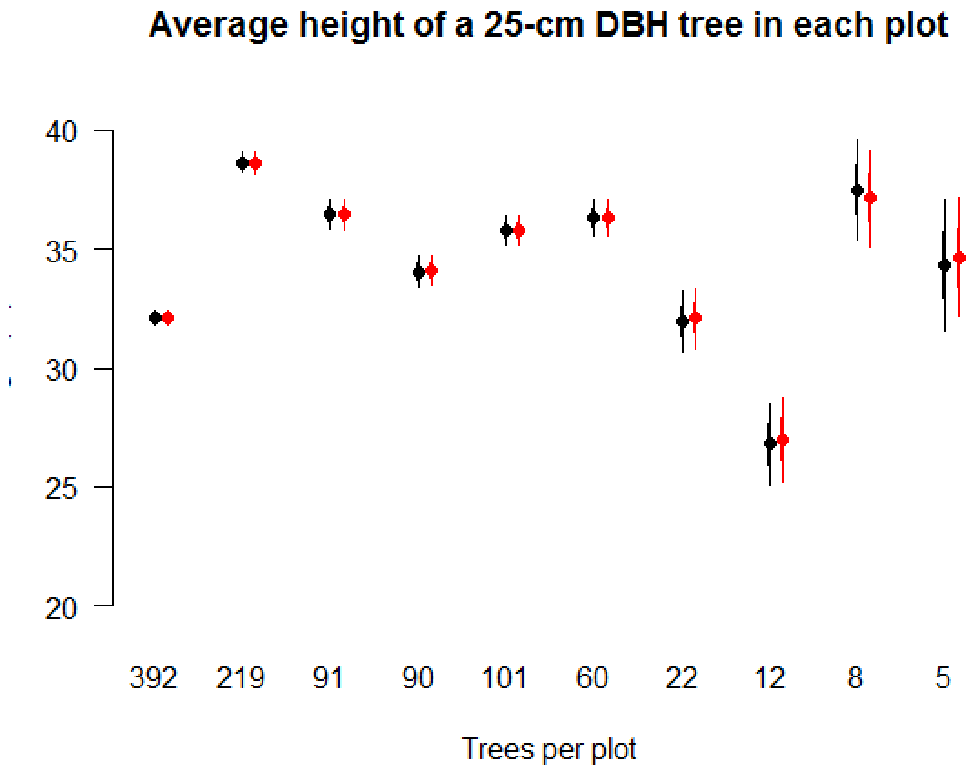
```
data <- list(height=trees$height,
             dbhc=trees$dbh.c,
             plot=trees$plot,
             tempc=temp.c)
m4 <- jags(data,
           model.file=model4,
           parameters.to.save = c("alfa", "beta", "sigma", "grandmu", "sigmaplot", "beta.temp"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 4913

Initializing model
```

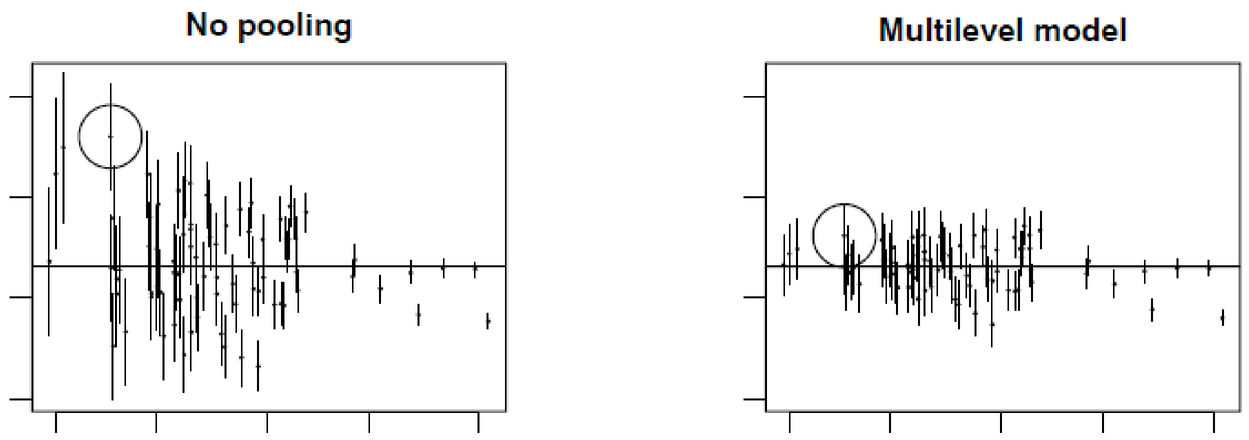
Average heights among plots related to temperature



Adding plot-level predictors (pooling) may improve parameter estimation



Adding plot-level predictors (pooling) may improve parameter estimation



From Gelman & Hill p. 253

Slopes can also vary…

- and coefficients be estimated with pooling
- but the correlation between slopes and intercepts must be modelled explicitly
- see e.g. Gelman & Hill 2007, ch. 13.

So what's a multilevel/hierarchical model?

Parameters/coefficients are given a probability model (with their own hyperparameters estimated from data).

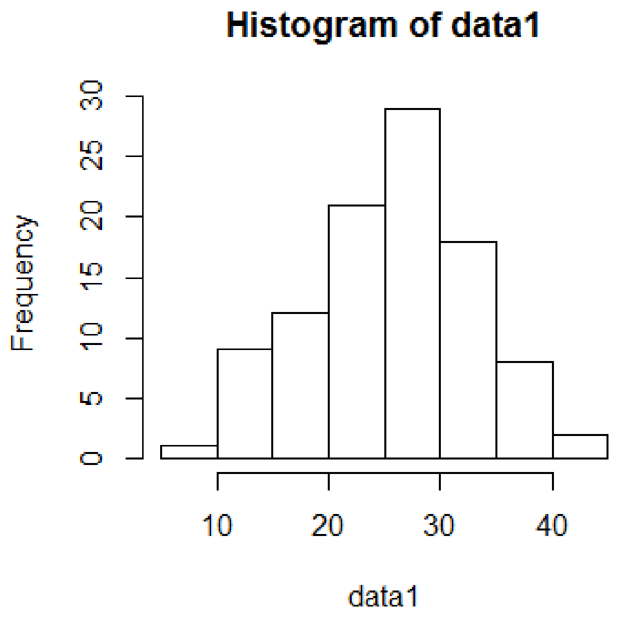
Intercepts and/or slopes may vary, and can be modelled (sometimes including their own predictors).

Advantages of hierarchical Bayes

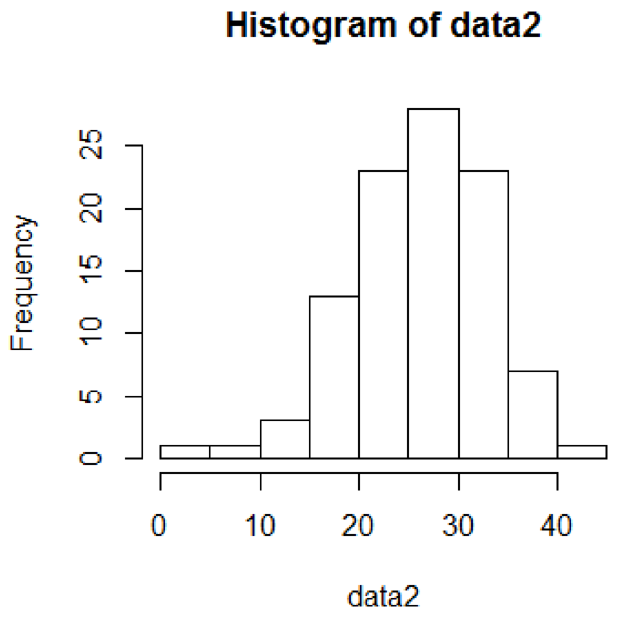
- Perfect for structured data (space-time)
- Predictors enter at the appropriate level
- Accommodate variation in treatment effects
- More efficient inference of regression parameters
- Using all the data to perform inferences for groups with small sample size
- Predictions fully accounting for uncertainty and variability
- Prior information

Datasets are stochastic realisations of a process

```
data1=rnorm(100, 2 + 1.6*x, 5)
```

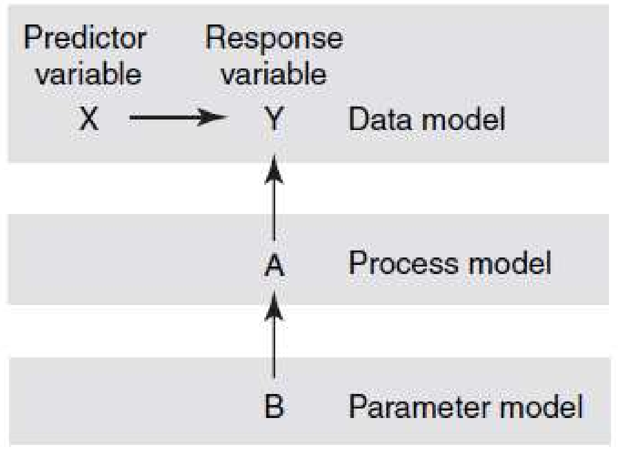


```
data2=rnorm(100, 2 + 1.6*x, 5)
```



These two datasets are different, even though they arise from same process

Hierarchical Bayes: data, process, parameters



$$\begin{aligned} &f(\text{data,process,parameters}) \\ &\quad \propto f(\text{data}|\text{process, parameters}) \\ &\qquad \times f(\text{process}|\text{parameters}) \\ &\qquad \times f(\text{parameters}). \end{aligned}$$

Clark et al. 2006, Clark 2007

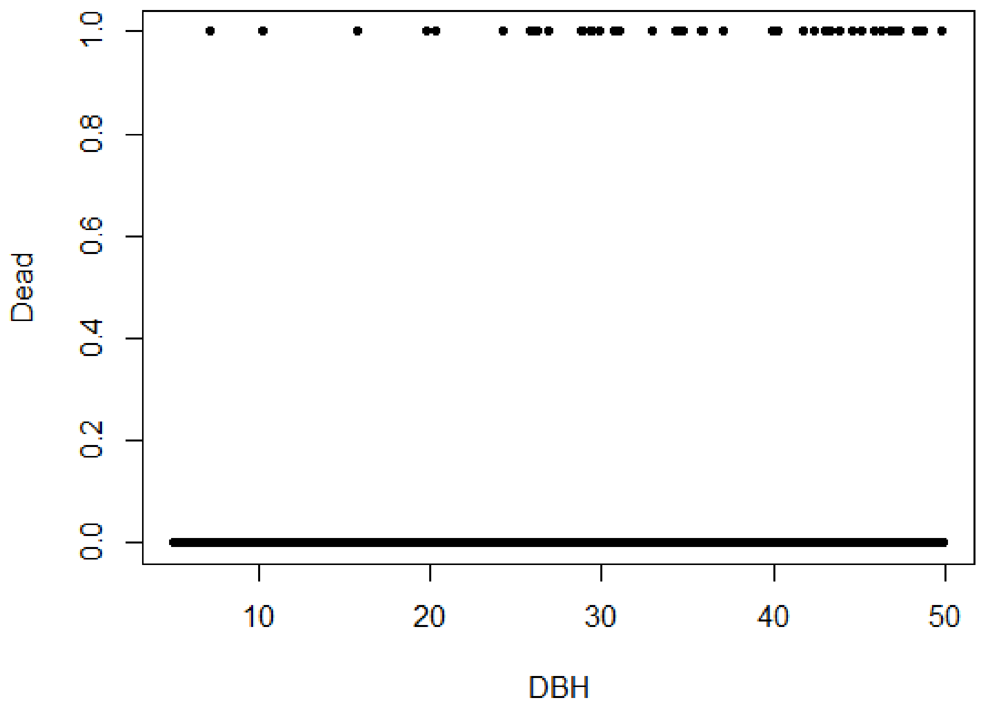
Exercise

Does sex influence height?

Do it yourself

Bayesian logistic regression

Relationship between tree size and mortality



Logistic regression model

```
model5 <- function(){

  # LIKELIHOOD
  for (i in 1:length(dead)){
    dead[i] ~ dbern(pdeath[i])
    logit(pdeath[i]) <- mu + beta*dbhc[i]
  }

  # PRIORS
  mu ~ dnorm(0, .001)
  beta ~ dnorm(0, .001)
}
```

Calling JAGS

```
data <- list(dead=trees$dead,
             dbhc=trees$dbh.c)
m5 <- jags(data,
           model.file=model5,
           parameters.to.save = c("mu", "beta"),
           n.chains=3,
           inits=NULL,
           n.iter=10000,
           n.burnin=5000)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 4668

Initializing model
```

Results

```
Inference for Bugs model at "C:/Users/FRS/AppData/Local/Temp/RtmpwpZd1g/model18fc588f6cea.txt", f
 3 chains, each with 10000 iterations (first 5000 discarded), n.thin = 5
 n.sims = 3000 iterations saved
         mu.vect sd.vect    2.5%   97.5%  Rhat n.eff
beta       0.055   0.014   0.029   0.084 1.002  1600
mu        -3.461   0.222  -3.916  -3.076 1.002  1300
deviance 345.865   5.831 343.745 351.241 1.001  3000

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 = 17.0 and DIC = 362.9
DIC is an estimate of expected predictive error (lower deviance is better).
```

Compare with glm

```
logreg <- glm(dead ~ dbh.c, data = trees, family = binomial)
display(logreg)
```

```
glm(formula = dead ~ dbh.c, family = binomial, data = trees)
            coef.est coef.se
(Intercept) -3.44     0.21
dbh.c        0.05     0.01
---
  n = 1000, k = 2
  residual deviance = 343.7, null deviance = 360.9 (difference = 17.2)
```

END