Bayesian Methods for Estimating Concentrations in Serial Dilution Assays

Alessandra della Fazia

1/7/2021

Summary

Serial dilution assay is a technique for estimating concentration of compounds, based on combining measurements of several different dilutions of a unknown sample. The relation between concetrations and measurements is non linear and therefore it is not suitable to weight the measurements equally. We use a Bayesian approach for estimating jontly the calibration curve and the unknown concentrations. In particular, we set up two models: the first model assumes that measurements are *homoscedastic* with respect to concentration. We will see how this model is not able to estimate too small concentrations. The second model drops this assumption and, compared to the first, estimates have much lower standard errors and give results for any concentration. We test the ability of our models to recover models parameters simulating data from the models it self. Finally, we compare the Bayesian approach with the frequentist one, concluding that Bayesian estimates are much more accurate than those obtained by the frequentist method.

Introduction

A common approach for estimating concentrations of compounds in biological samples is the *serial dilution assay*, in which a small amount of sample or solute are serially diluted, mixing a amount of sample with a measured amount of a inert liquid. Then for each diluted samples we take a measurement. The reason for the serial dilutions of each sample is that the concentration in a assay is quantified by an automated optical reading of a color change, and there is only a certain range of concentrations for which the color change is informative: at low values of concetration the color change is imperceptible, at high values the color saturates. Hence, different dilutions give several opportunities for an accurate measurement. Or better, different dilutions give measurements with different accuracy.

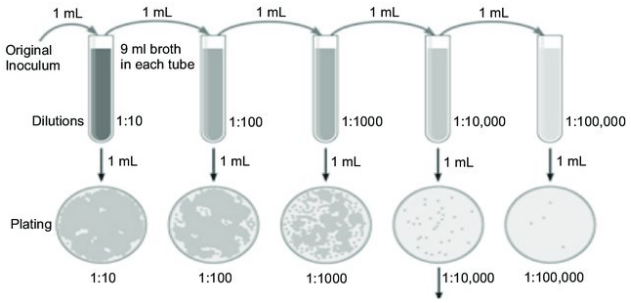


Figure 1: Example of serial dilution

Data

An assay comes on plate with a number of wells each containing a sample at a specific dilution. Dilution levels in a plate are show in Table 1, where each cell represent a well. The plate contains two type of data:

Table 1: Setup of a plate with 96 wells. The first two columns are dilutions of "Standards" (sample known concentrations), and the other column are 10 different "Unknowns". The goal of the assay is to estimate the concentrations of the"Unknowns", using the "Standards" as calibration.

std	std	unk	unk	unk	unk	unk	unk	unk	unk	unk	unk
1	1	1	1	1	1	1	1	1	1	1	1
1/2	1/2	1	1	1	1	1	1	1	1	1	1
1/4	1/4	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
1/8	1/8	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
1/16	1/16	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
1/32	1/32	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
1/64	1/64	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27
0	0	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27	1/27

the *Unknowns*, which are the samples for which we want estimates the concentration, and the *Standards*, which are compound for which the concentration is known and used to calibrate the measurements.

The data used in the experiment comes from a single plate with 96 wells in a study of cockroach allergens in the homes of a asthma sufferers. In Table 2 shows the data for the *Standards* (having know concentration of 0.64), within its measurements at several dilutions, and data for two *Unknowns*. Dilutions for *Standards* range from 1 to 0 (the numbers in Table 1). Dilution equal to 1 means no dilution (sample at full strenght), dilution equal to 0 means solution is made by 100% of inert liquid and 0% of sample.

The goal of the project is to estimates the concentration for the *Unknowns* samples, using *Standards* as calibration.

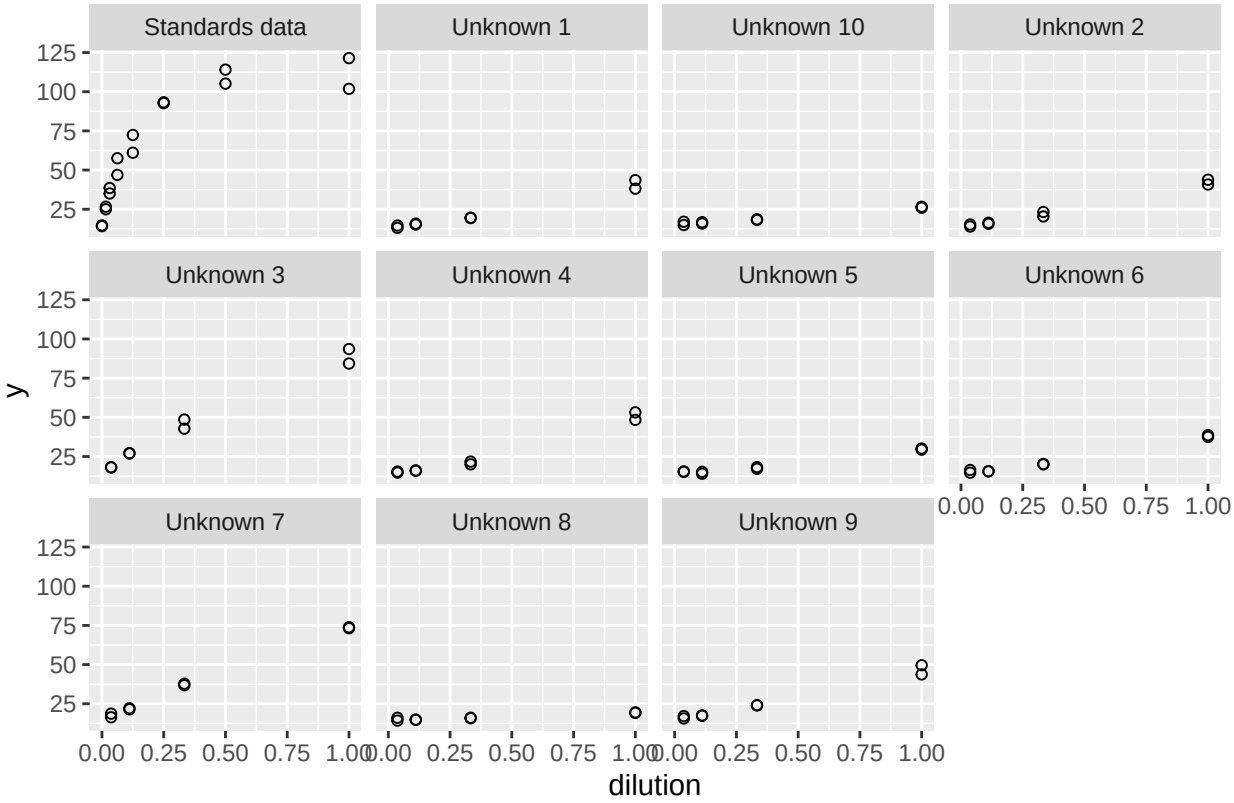


Figure 1: Data of measurements versus dilutions from a single plate.

Table 2: Example of some measurements y from a plate. On the left measurements y for "Standards" (sample with known concetration of 0.64). On the righ measurements y for two "Unknowns" ("Unknown 8" and "Unknown 9") for which we want estimate the concetration.

Standards data			Some of Unknowns data		
Dilution	Diluted.Conc.	y	Sample	Dilution	y
1	0.64	101.835	Unknown 8	1	19.164
1	0.64	121.417		1	19.522
1/2	0.32	105.167		1/3	16.090
1/2	0.32	114.084		1/3	15.789
1/4	0.16	92.698		1/9	14.860
1/4	0.16	93.287		1/9	14.755
1/8	0.08	72.406		1/27	14.274
1/8	0.08	61.068		1/27	16.063
1/16	0.04	57.546	Unknown 9	1	49.591
1/16	0.04	46.923		1	43.750
1/32	0.02	38.544		1/3	23.970
1/32	0.02	35.050		1/3	24.087
1/64	0.01	26.608		1/9	17.314
1/64	0.01	24.974		1/9	17.573
0	0.00	14.700		1/27	15.609
0	0.00	14.200		1/27	17.122

Model Structure

We use the following notation:

- y_i for observed measurement (color intesities)
- θ_j for the concetration of the samples (for the standard sample $j = 0$ and $\theta_0 = 0.64$)
- d_i level of dilution (the column *Dilution* in Table 2)
- x_i for the diluted concentration, that is $x_i = d_i \cdot \theta_j$

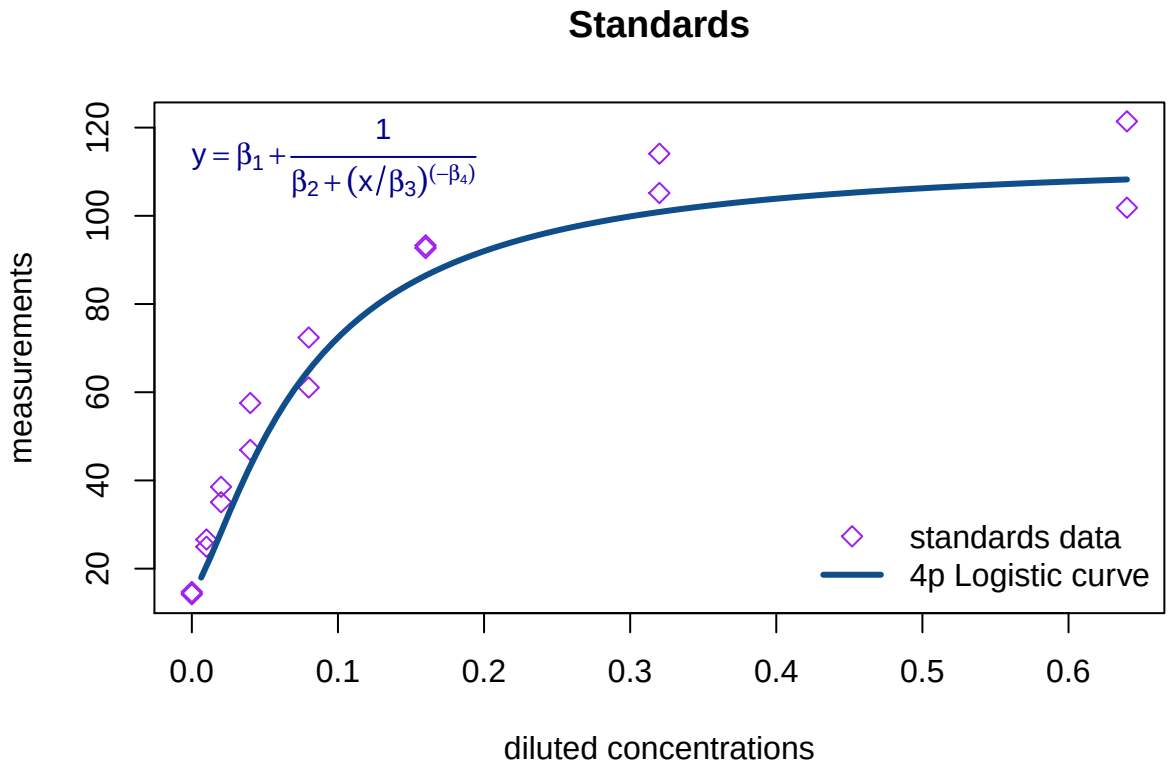
Thus, the goal is to estimates the concentration θ_j for the *Unknowns*.

The *expected value* of the measurement y is a function of the (diluted) concentration x and typically a four-parameter logistic curve is used:

$$E(y|x, \beta) = g(x, \beta) = \beta_1 + \frac{\beta_2}{1 + (x/\beta_3)^{-b4}}$$

where β_1 is the color intensity y at zero concentration, β_2 is the increase to saturation (y max is $\beta_1 + \beta_2$), β_3 is the concentration at which the curve turns, and β_4 is the rate at which saturation occurs.

All the four parameters β_k, as well as concentrations θ are positive, so we model them on a logarithmic scale.



(The curve shown are estimated from the model described later.)

Model 1: equal variance

We model the measurement as normally distributed with equal variances:

$$y_i \sim N(g(x_i, \beta), \sigma_y^2)$$

$$x_i = d_i \cdot \theta_j$$

Prior distribution We assign non informative prior distributions:

- for the parameters of the curve: $log(\beta_k) \sim Unif(-\infty, +\infty)$ for k = 1, 2, 3, 4.
- InvGamma prior distribution for σ_y^2, that is a Gamma distribution for the precision $\sigma_y^2 = \frac{1}{\tau}, \tau \sim Gamma(0.001, 0.001)$.
- for the unknown concentrations: $log(\theta_j) \sim Unif(-\infty, +\infty)$.

Inference We fitted the model in RJAGS. The definition of the model is stored on the `simple_model.txt` file but it is reported here as well.

We actually work we parameters $log(\beta)$, σ_y^2, and $log(\gamma)$ where $log(\gamma_i) = log(\theta_j/\beta_3)$. The use of γ_i in place of θ_j gets around the problem of the strong correlation between the unknown concentrations and the parameter β_3.

```
# equal variance model

model{
```

```
  for (i in 1:(zero_conc_from - 1) ) {

    y[i] ~ dnorm(expect[i], prec[i])

    expect[i] <- exp(logb1) + ( (exp(logb2)) /  (1 + pow(d[i]*exp(loggamma[index[i]]) , -exp(logb4)) ) )

    prec[i] <- precision_y
  }

  # if the dilution coeff. d_i is 0 then E[y|x] = beta_1

  for (i in zero_conc_from : zero_conc_to) {

    y[i] ~ dnorm(expect[i], prec[i])

    expect[i] <- exp(logb1)

    prec[i] <- precision_y
  }

  for (j in 1:11) {
    loggamma[j] ~ dunif(-1000,1000)
  }

  # priors
  logb1 ~ dunif(-1000,1000)
  logb2 ~  dunif(-1000,1000)
  logb4~ dunif(-1000,1000)

  precision_y ~ dgamma(0.001, 0.001)

  # transformation to orginal scale
  b1 <- exp(logb1)
  b2 <- exp(logb2)
  b4 <- exp(logb4)

  b3 <- 0.64/exp(loggamma[11])
  sd_y <- 1 / sqrt(precision_y)

  for (j in 1:11) {
    theta[j] <- exp(loggamma[j]) * b3
  }
}
```

As starting point for JAGS we used the Non Linear Lest Square (NLS) estimates of the parameters computed in Section "Frequentist".

We obtained convergence (the potential reduction factors R were below 1.1) after 50 000 interations of two parallel chains. To save memory we save every 20 th iteration of each chain, to have in the end 1250 samples per chain.

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
```

```
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 96
##    Unobserved stochastic nodes: 15
##    Total graph size: 573
##
## Initializing model
```

```
## Inference for Bugs model at "simple_model_2.txt", fit using jags,
##  2 chains, each with 50000 iterations (first 25000 discarded), n.thin = 20
##  n.sims = 2500 iterations saved
##           mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## b1         14.989   0.559  13.853  14.637  15.000  15.360  16.058 1.001  2500
## b2        103.703   2.944  98.286 101.726 103.595 105.608 109.886 1.001  2500
## b3          0.068   0.005   0.059   0.064   0.067   0.070   0.078 1.001  2500
## b4          1.325   0.069   1.196   1.278   1.321   1.371   1.463 1.001  2500
## sd_y        3.244   0.250   2.814   3.061   3.220   3.406   3.764 1.004   510
## theta[1]    0.028   0.003   0.023   0.026   0.028   0.030   0.034 1.001  2500
## theta[2]    0.030   0.003   0.025   0.028   0.030   0.032   0.037 1.001  2500
## theta[3]    0.120   0.008   0.105   0.114   0.119   0.125   0.136 1.001  2500
## theta[4]    0.039   0.003   0.032   0.036   0.039   0.041   0.045 1.001  2500
## theta[5]    0.017   0.003   0.012   0.015   0.016   0.018   0.022 1.001  2500
## theta[6]    0.025   0.003   0.020   0.024   0.025   0.027   0.031 1.001  2500
## theta[7]    0.080   0.006   0.070   0.076   0.080   0.084   0.092 1.001  2500
## theta[8]    0.000   0.001   0.000   0.000   0.000   0.000   0.000 1.010  2500
## theta[9]    0.036   0.003   0.030   0.034   0.036   0.038   0.042 1.001  2500
## theta[10]   0.014   0.003   0.008   0.012   0.013   0.015   0.019 1.001  2500
## deviance  498.203   5.959 488.933 493.922 497.464 501.642 511.609 1.003   670
##
## 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 = 17.7 and DIC = 515.9
## DIC is an estimate of expected predictive error (lower deviance is better).
```

We note how the estimate for "theta 8" (concentration of "Unknown 8") is zero. A casual glance at the data, (see Unknown 8 in Figure 2) might suggest that these data are all noise (the curve that interpolates the data is quite flat), and we therefore we could conclude that concentration is zero. However, a carefull look at the numbers reveals that the measurements decline consistently from concentration of 1 to 1/3 to 1/9 (see Table 2), with only the final dilutions lost in noise (the measurements at 1/27 are indeed no lower that at 1/9).

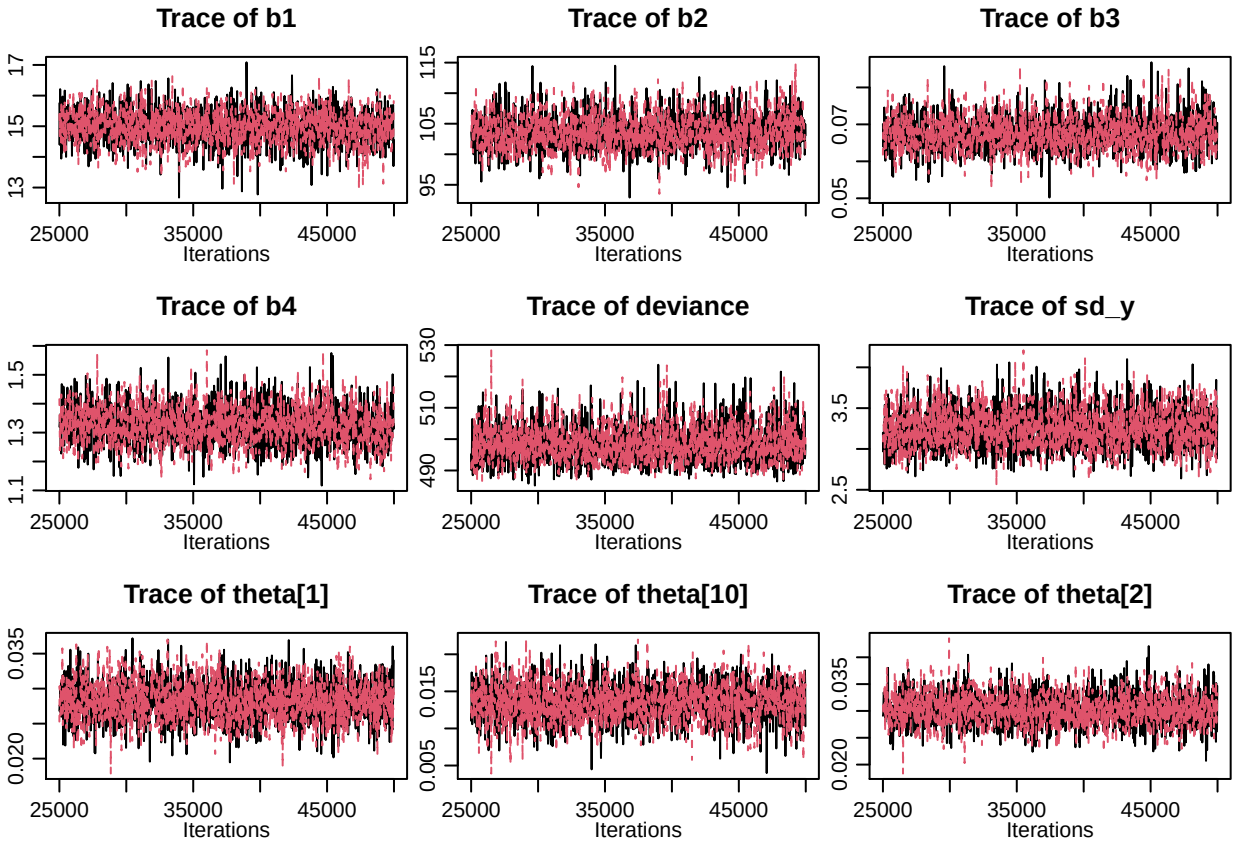
In measurements of allergens, even low concentration can be important for asthma sufferers, and we need to be able to distinguish between zero concentrations and values that are merely low.

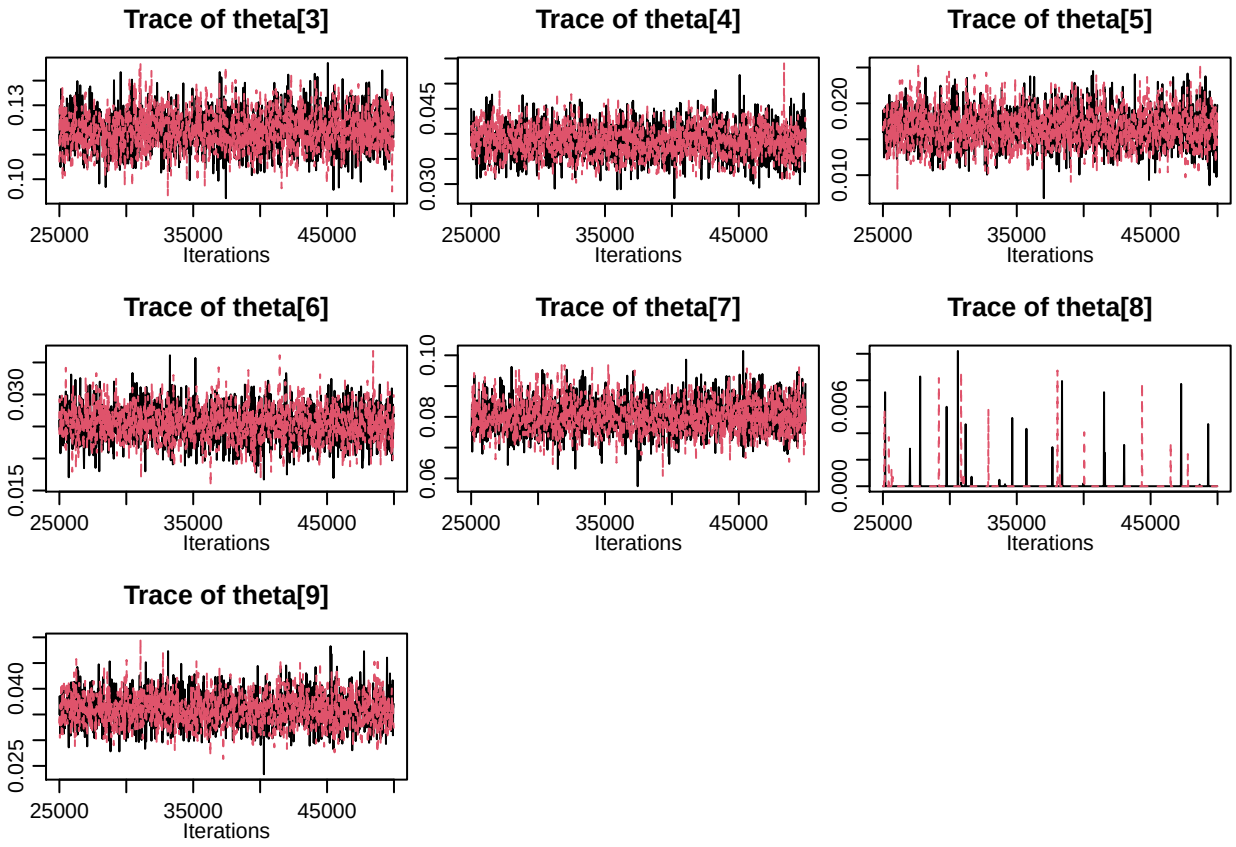
Converge Diagnostic

Once the samples from the *posterior distributions* are obtained, **the convergence check is needed**. If the Markov Chians converges, it means it has reached its equilibrium (target) distribution and the generated samples comes from the target distribution. Hence, monitoring the convergence of the algorithm is essential for producing results from the posterior distribution.

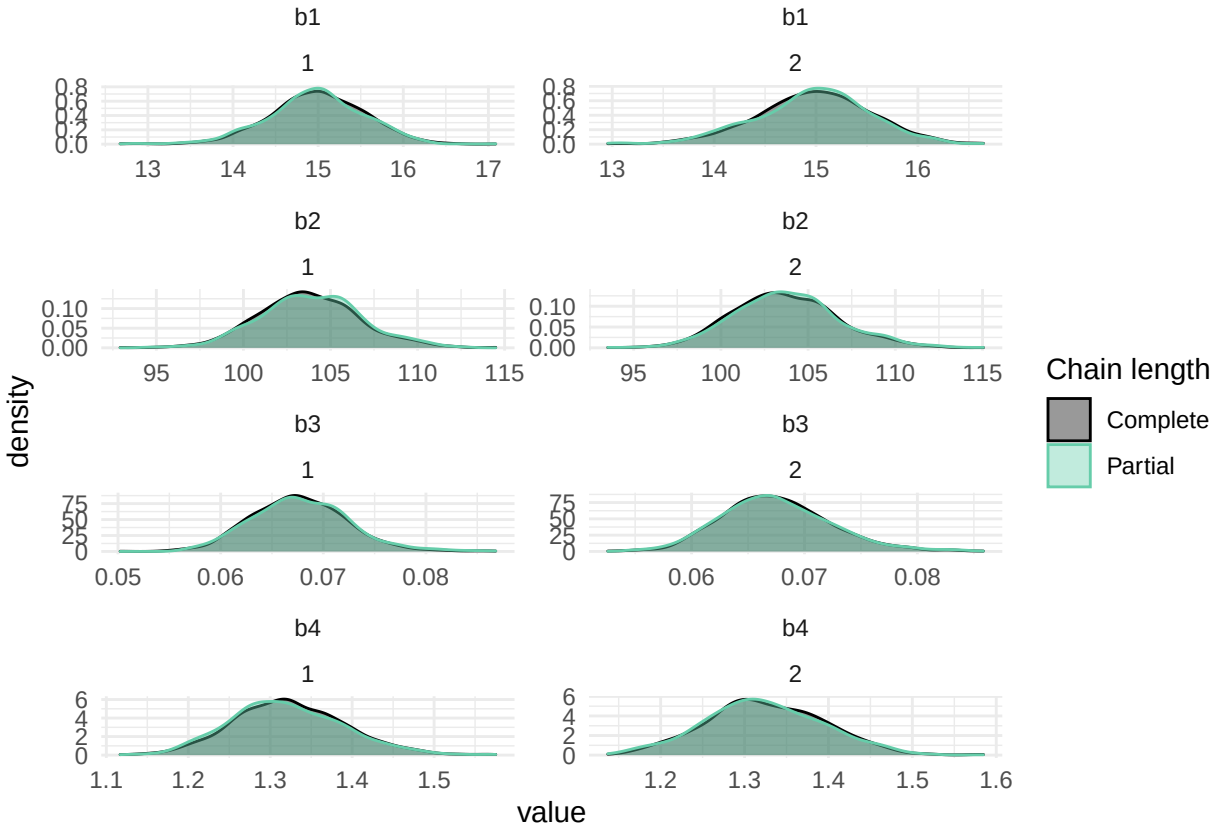
Trace plots Trace plots show the values the parameter took during the runtime of che chain. If the samples comes from the same distribution, the trace plots of different chains (that are show with different colour) mix with each other. It seems our case. Moreover, the range of values is not too wide.

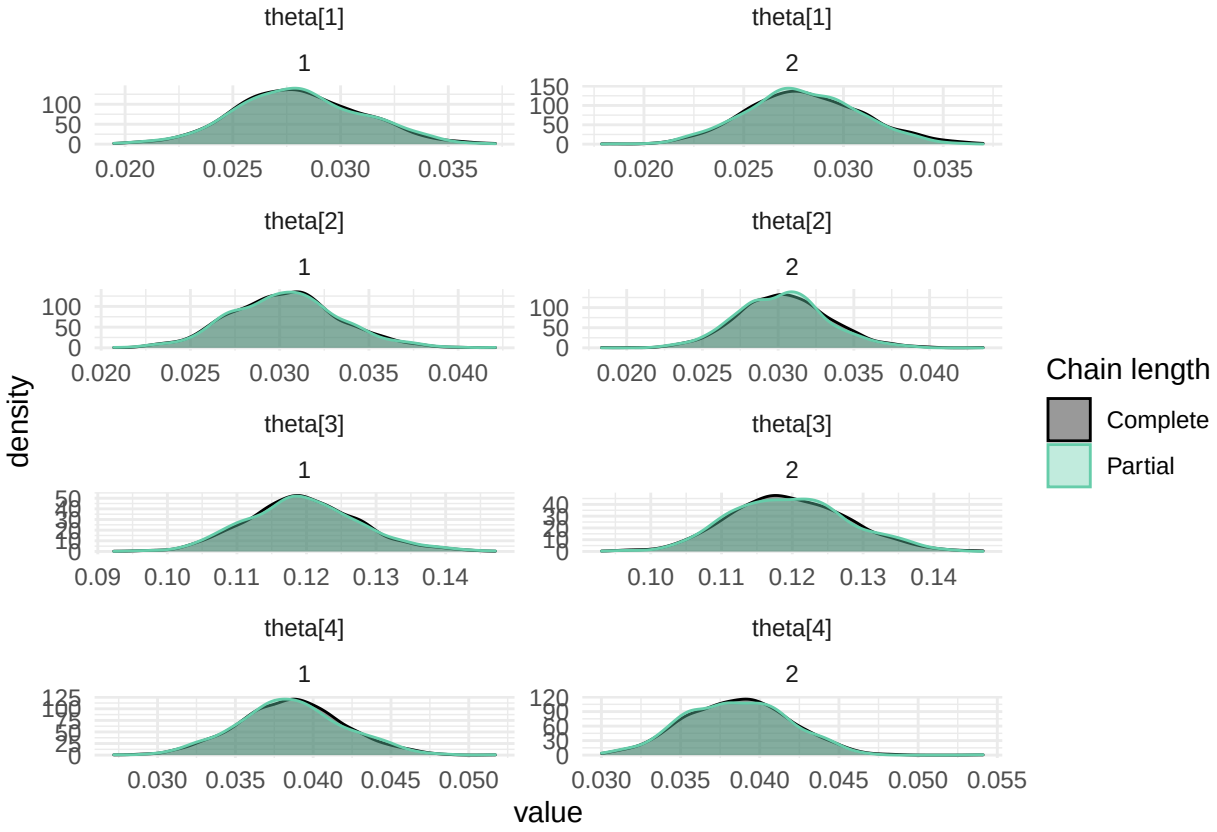
Again, we see how the plot seems suitable for all variables, except concentration of "Unknown 8."

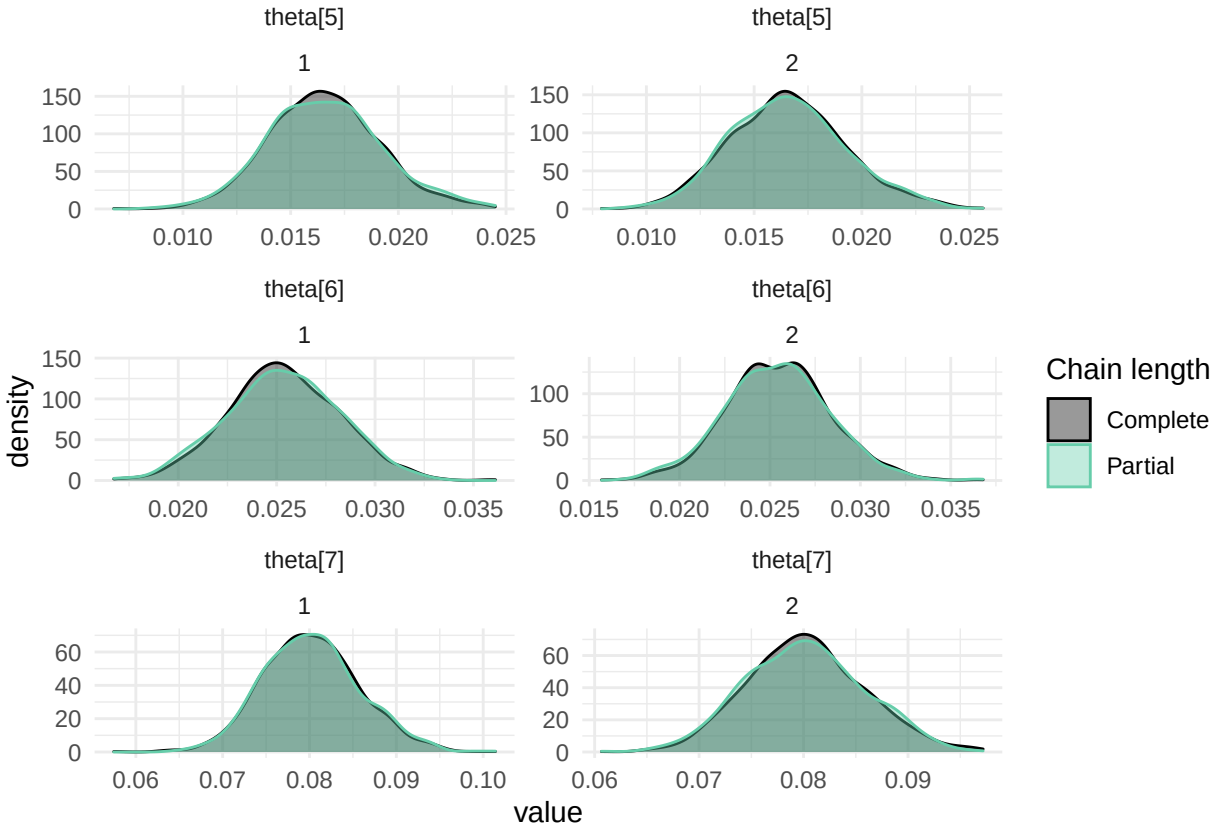


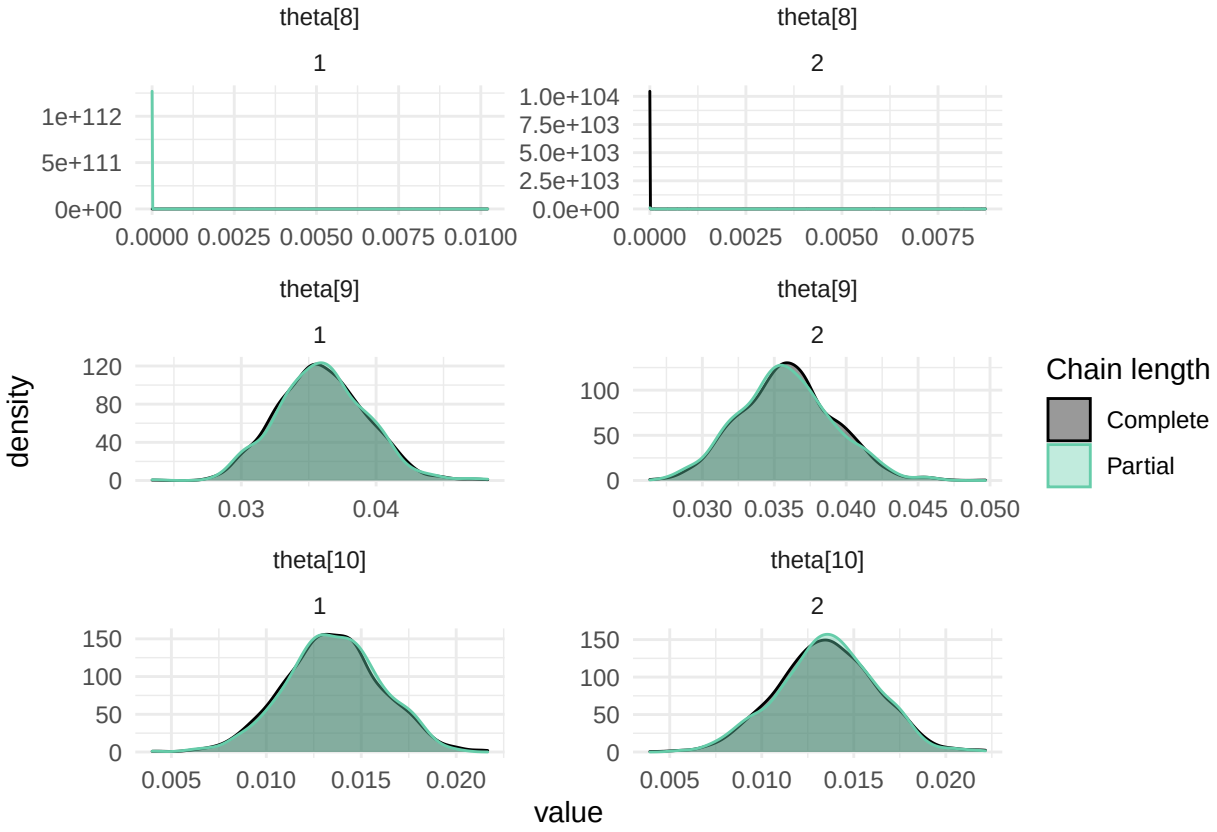


Density Plot In the following graphs, we compare the density coming from last part of the chain (the last 50 percent of the values, in green) with the whole chain (black), where the two columns refers to the two chains. Ideally, the initial and final parts of the chain have to be sampling in the same target distribution, so the overlapped densities should be similar. As before, we note that the graphs are reassuring, except for Unknown 8.


b1
1
b1
2
b2
1
b2
2
b3
1
b3
2
b4
1
b4
2
density
value
Chain length
Complete
Partial


theta[1]
1
theta[1]
2
theta[2]
1
theta[2]
2
theta[3]
1
theta[3]
2
theta[4]
1
theta[4]
2
density
value
Chain length
Complete
Partial


theta[5]
1
2
theta[6]
theta[7]
density
value
Chain length
Complete
Partial



Prediction

The model used is not able to make inferences on Unknown 8, even if informations seems to be present in the data which decline consistently with dilution. In order to find the weak point of the model, we check its prediction ability, comparing predictions of the model on y with the measurements.

Recall that for each level of dilution two measurements are provided (see Table 2). In Figure 2 we try to predict the values of y accroding to our model, and we compare them with the data showed before in Figure 1. Looking at the plot in Figure 1, we note that, as the dilution level increases, there is more dispersion in the measurements than in our prediction, (the plot for measurements has the shape of a funnel). This is a clear signal that the variability of the random disturbance is different across measurements and we should taking in to account. Therefore we drop our assumption that the modelling errors all have the same variance.

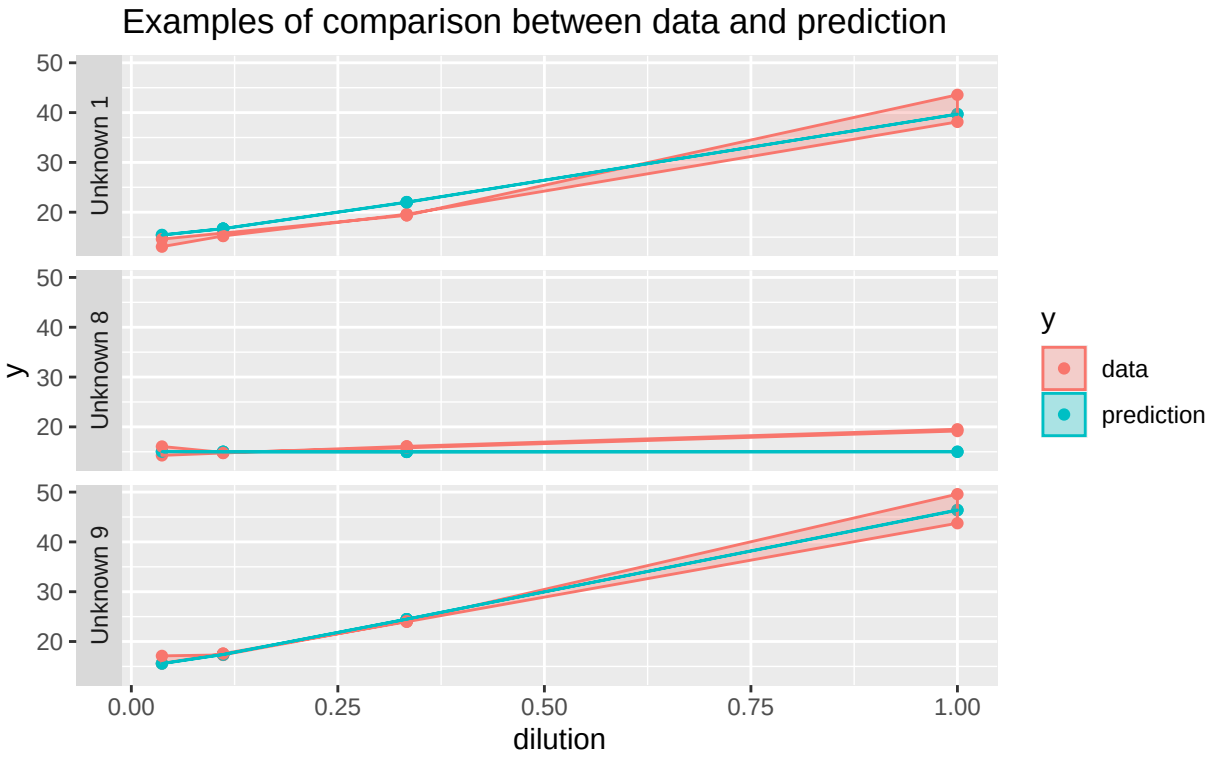


Figure 2. For each level of dilution we have two measures. However, data (pink) appear very spaced, while predictions (lightblue) are so close that they quite coincide.

Model 2: unequal variance

In this section, we follow *Higgins et al. (1998)* model the measurement error as normally distributed with unequal variances:

$$y_i \sim N\left(g(x_i,\beta), \left(\frac{g(x_i,\beta)}{A}\right)^{2\alpha}\sigma_y^2\right), \qquad x_i = d_i \cdot \theta_j$$

where:

- the parameter $\alpha > 0$ models the pattern that variances are higher for larger measurements.
- the constant A is arbitrary and is set to some value in the middle of the range of the data (we set to the geometric mean). It is included so that the parameter σ_y has a more direct interpretation as the standard deviation of a "typical" measurement.

Prior For the parameters β, σ_y^2 and for the unknown concentrations θ we use a non informative prior distributions, as did before. For the parameter α we assign a uniform prior distribution in the range $[0, 2]$. To summarize:

- $log(\beta) \sim Unif(-\infty, +\infty)$ for k = 1, 2, 3, 4.
- $\sigma_y^2 = \frac{1}{\tau}, \tau \sim Gamma(0.001, 0.001)$.
- $log(\theta_j) \sim Unif(-\infty, +\infty)$.
- $\alpha \sim Unif(0, 2)$.

Inference We fit the following model in JAGS. We obtain converges ($R < 1.1$) for all the paramenters after 50 000 iterations of two parallel chains.

```
model{

  for (i in 1: (zero_conc_from - 1) ) {

    y[i] ~ dnorm(expect[i], prec[i])

    expect[i] <- exp(logb1) + ( (exp(logb2)) /  (1 + pow(d[i]*exp(loggamma[index[i]]) , -exp(logb4)) ) )

    prec[i] <- ((A/expect[i])^(2*alpha)) * tau_y
  }

  for (i in zero_conc_from : zero_conc_to) {

    y[i] ~ dnorm(expect[i], prec[i])

    expect[i] <- exp(logb1)

    prec[i] <- ((A/expect[i])^(2*alpha)) * tau_y

  }

  for (j in 1:11) {
    loggamma[j] ~ dunif(-1000,1000)
  }

  # priors
  logb1 ~ dunif(-1000,1000)
  logb2 ~ dunif(-1000,1000)
  logb4 ~ dunif(-1000,1000)

  tau_y ~ dgamma(0.001, 0.001)

  alpha ~ dunif(0,1)

  # transformation to orginal scale
  b1 <- exp(logb1)
  b2 <- exp(logb2)
  b4 <- exp(logb4)

  b3 <- 0.64/exp(loggamma[11])

  sd_y <- 1/sqrt(tau_y)


  for (j in 1:10) {
    theta[j] <- exp(loggamma[j]) * b3
  }
}
```

```
A <- round(gm_mean(y))

dat.jags <- list("y", "d", "index","A", "zero_conc_from", "zero_conc_to")
```

```r
mod.params <- c("b1", "b2","b3","b4", "theta", "alpha","sd_y")

# Starting values
mod.inits <- function(){
  list("logb1" = log(15),
       "logb2"= log(108),
       "logb4"=log(1.12),
       "tau_y"=0.03,
       "alpha" = 0.5,
       "loggamma" = rep(1, 11)
  )
}

# Run JAGS

mod.fit.complex <- jags(data = dat.jags,                                          # DATA
                        model.file = "complex_model_2.txt", inits = mod.inits,                # MODEL
                        parameters.to.save = mod.params,
                        n.chains = 2, n.iter = 50000, n.thin=20)# MCMC
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 96
##    Unobserved stochastic nodes: 16
##    Total graph size: 722
##
## Initializing model
```

```r
mod.fit.complex
```

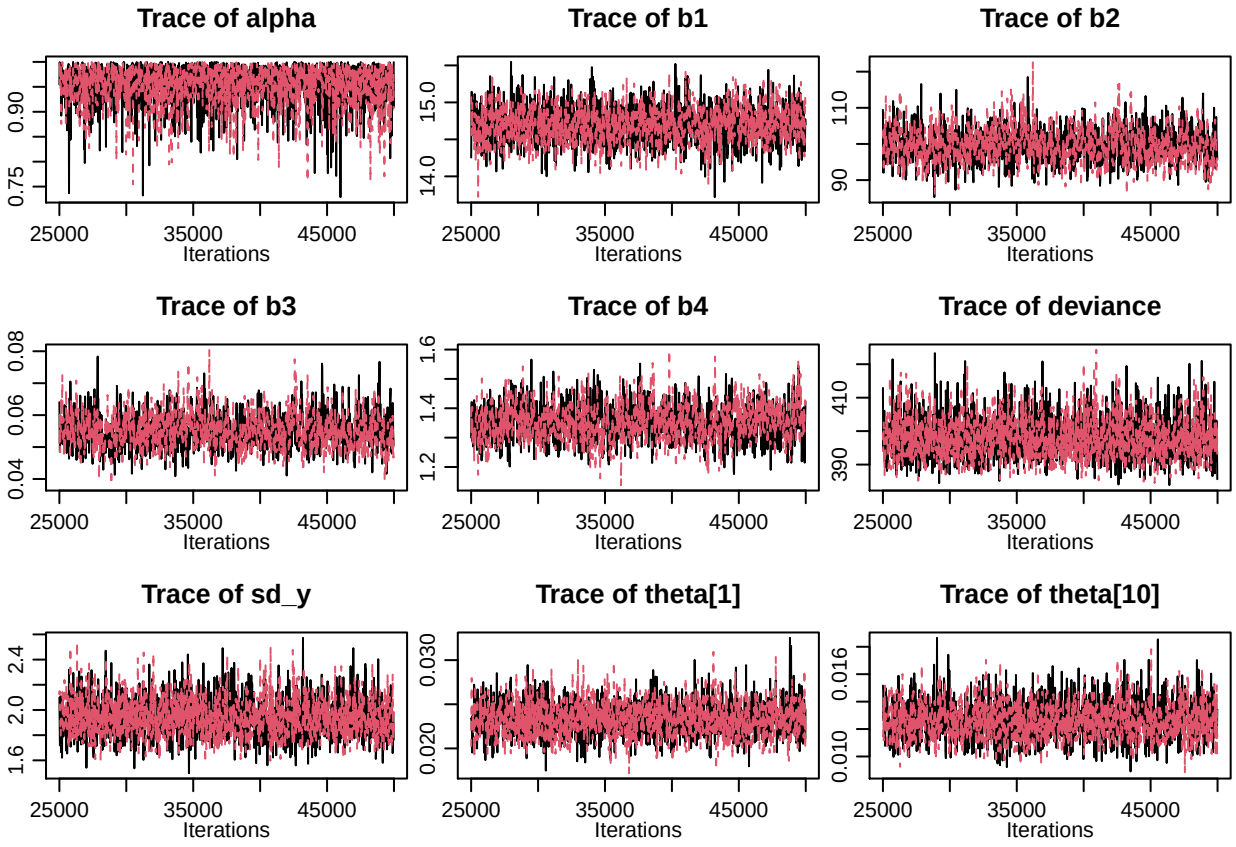
```
## Inference for Bugs model at "complex_model_2.txt", fit using jags,
##  2 chains, each with 50000 iterations (first 25000 discarded), n.thin = 20
##  n.sims = 2500 iterations saved
##           mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## alpha       0.955   0.041   0.844   0.935   0.966   0.986   0.999 1.001  2500
## b1         14.705   0.252  14.206  14.534  14.708  14.876  15.189 1.001  2500
## b2         99.809   4.846  91.039  96.417  99.678 102.892 109.874 1.002  2400
## b3          0.055   0.006   0.045   0.051   0.055   0.059   0.067 1.002  1300
## b4          1.351   0.063   1.233   1.307   1.350   1.394   1.483 1.001  2500
## sd_y        1.939   0.158   1.665   1.827   1.925   2.041   2.269 1.001  2300
## theta[1]    0.023   0.002   0.020   0.022   0.023   0.025   0.027 1.002  1200
## theta[2]    0.026   0.002   0.022   0.025   0.026   0.027   0.030 1.001  2500
## theta[3]    0.105   0.008   0.090   0.100   0.105   0.111   0.122 1.001  2500
## theta[4]    0.031   0.002   0.027   0.029   0.031   0.033   0.036 1.002   820
## theta[5]    0.015   0.001   0.012   0.014   0.015   0.016   0.018 1.001  2500
## theta[6]    0.022   0.002   0.018   0.021   0.022   0.023   0.026 1.001  2500
## theta[7]    0.070   0.006   0.060   0.066   0.070   0.074   0.082 1.001  2500
## theta[8]    0.006   0.001   0.004   0.005   0.006   0.006   0.008 1.001  2500
## theta[9]    0.031   0.003   0.027   0.030   0.031   0.033   0.037 1.001  2500
## theta[10]   0.013   0.001   0.010   0.012   0.013   0.014   0.016 1.004   530
## deviance  398.025   6.338 387.635 393.385 397.173 401.987 412.404 1.002  1600
##
```

```
## 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 = 20.1 and DIC = 418.1
## DIC is an estimate of expected predictive error (lower deviance is better).
```

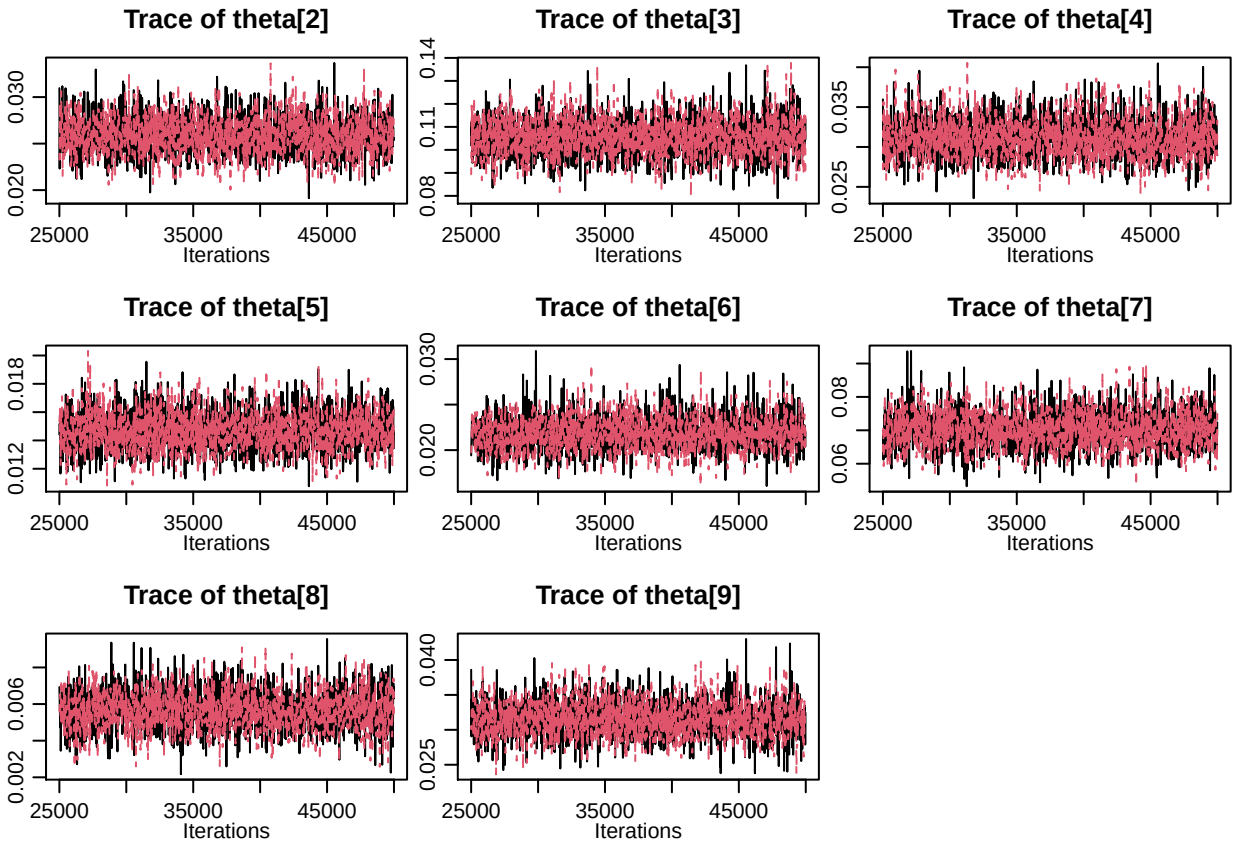
Convenge Diagnostic

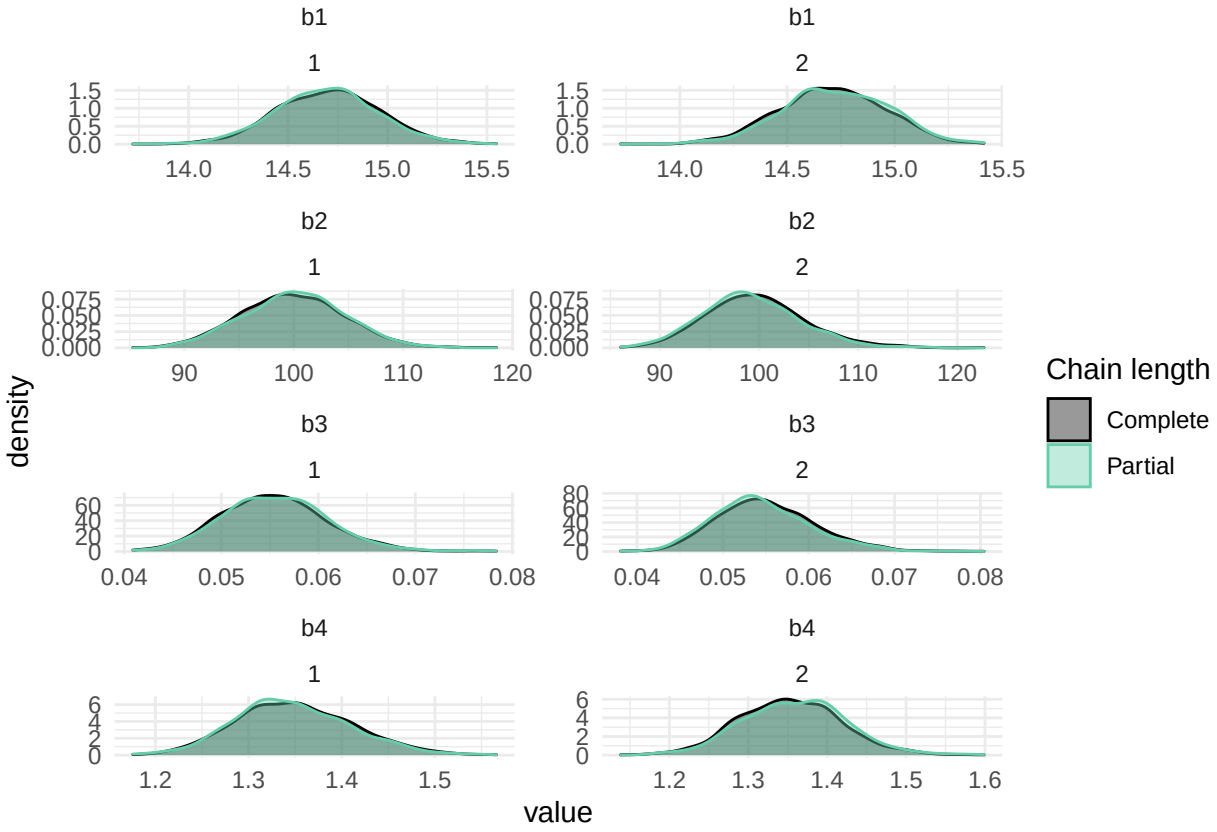
Trace plot and posterior density Looking at the trace plots we notice how now the convergence seem achieved for all parameters (Unknown 8 inclusive). The posterior distributions obtained with last half of the chain and with the complete chain is compared.

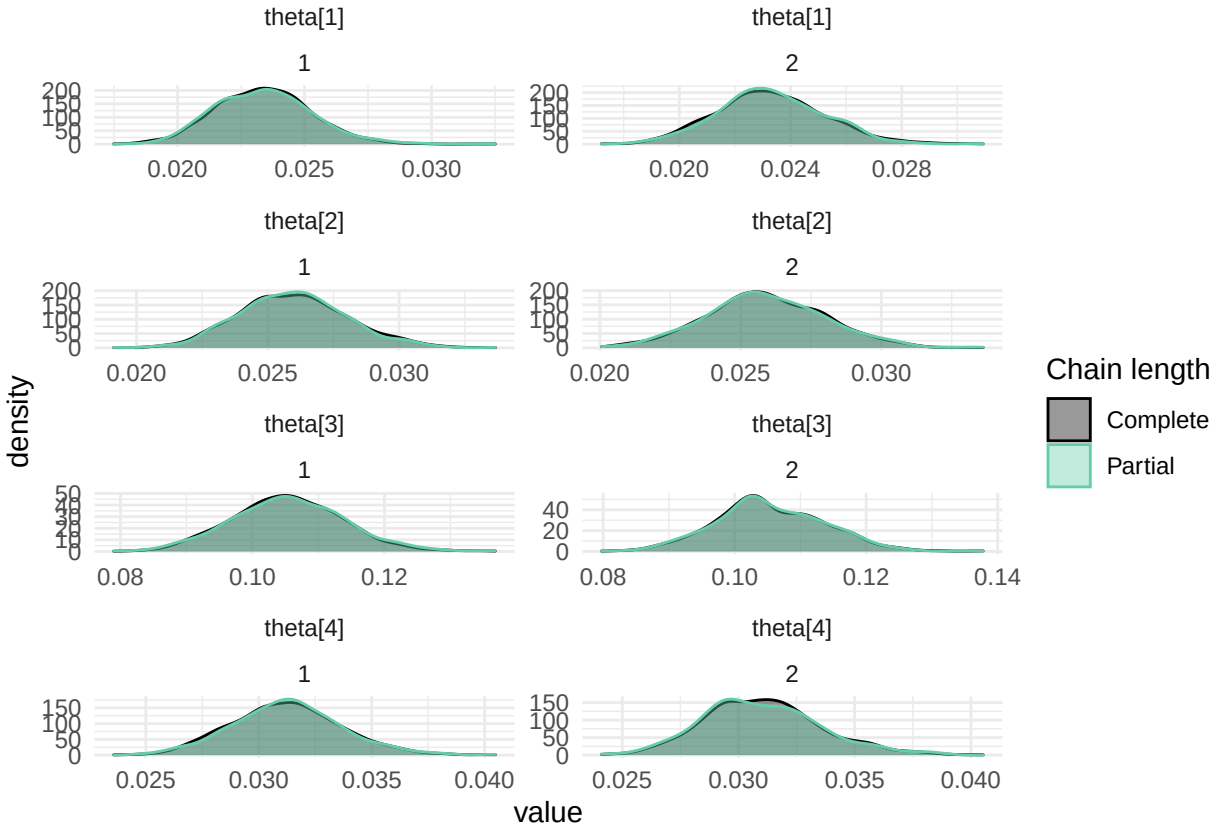
```
mod.fit.mcmc <- as.mcmc(mod.fit.complex)
chain_array <- mod.fit.complex$BUGSoutput$sims.array
chain_array_df <- ggs(mod.fit.mcmc)

par(mfrow= c(3,3), mgp=c(1.5,.7,0), oma = c(0,0.5,0,0), mar = c(2.5,1,3.1,1))
coda::traceplot(mod.fit.mcmc[, 1:9])
```

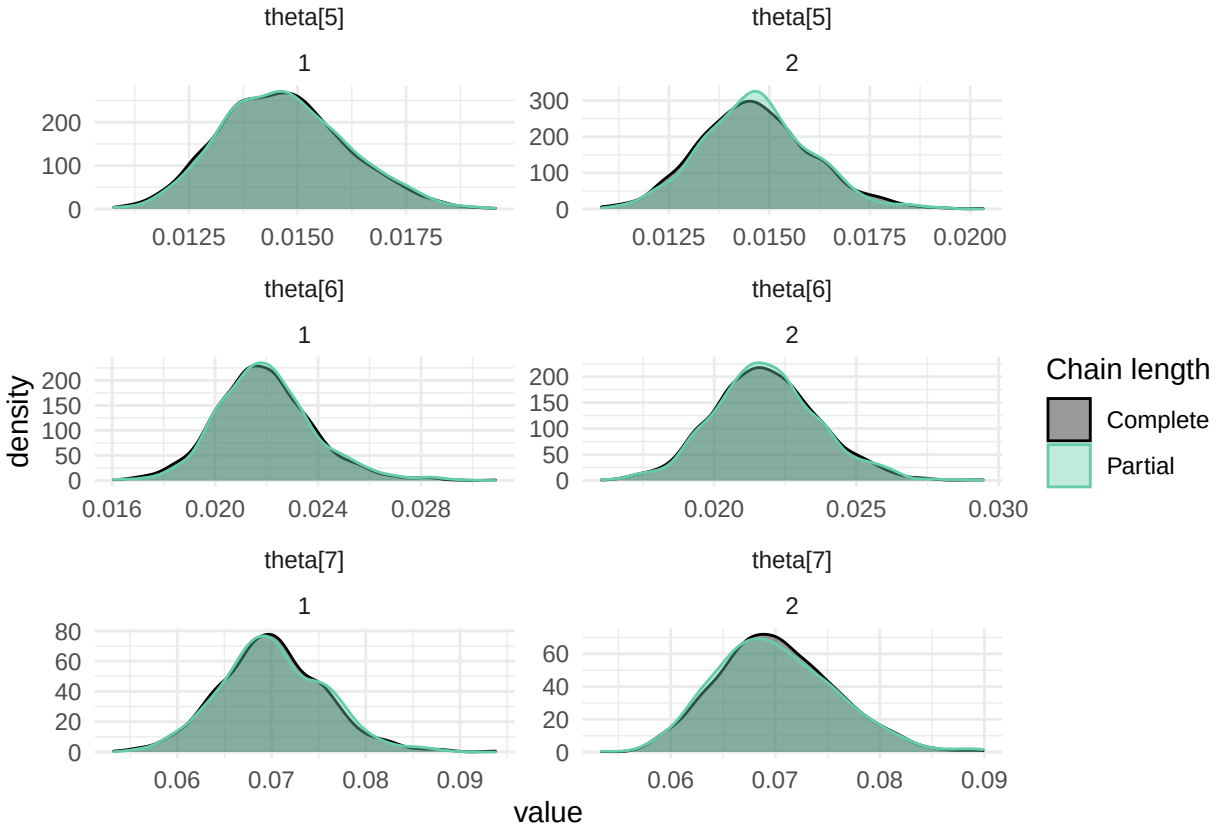


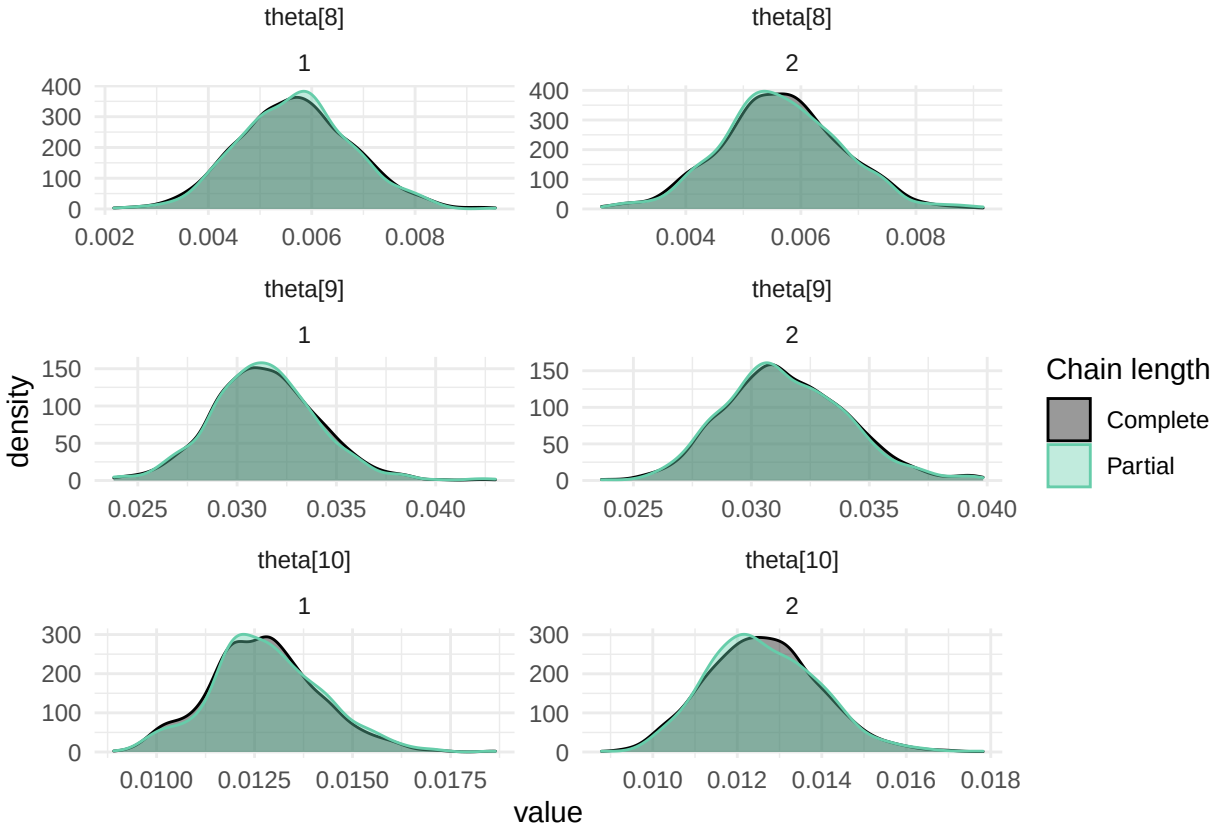

```
coda::traceplot(mod.fit.mcmc[, 10:17])
```


Trace of theta[2]
0.020
0.030
25000
35000
45000
Iterations
Trace of theta[3]
0.08
0.11
0.14
25000
35000
45000
Iterations
Trace of theta[4]
0.025
0.035
25000
35000
45000
Iterations
Trace of theta[5]
0.012
0.018
25000
35000
45000
Iterations
Trace of theta[6]
0.020
0.030
25000
35000
45000
Iterations
Trace of theta[7]
0.06
0.08
25000
35000
45000
Iterations
Trace of theta[8]
0.002
0.006
25000
35000
45000
Iterations
Trace of theta[9]
0.025
0.040
25000
35000
45000
Iterations


b1
1
2
b2
b3
b4
density
value
Chain length
Complete
Partial


theta[1]
1
theta[1]
2
theta[2]
1
theta[2]
2
theta[3]
1
theta[3]
2
theta[4]
1
theta[4]
2
density
value
Chain length
Complete
Partial


theta[5]
1
2
theta[6]
theta[7]
density
value
Chain length
Complete
Partial



We test if the chains have reached convergence comparing densities produced by the first half of the chain with complete chain.

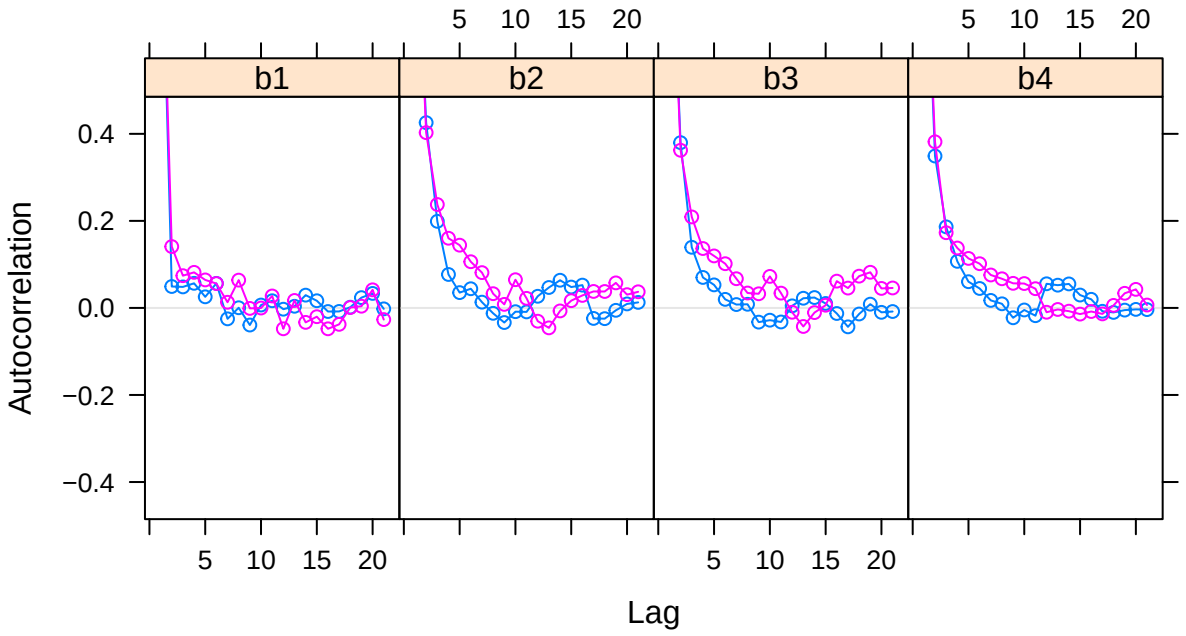
With this model all the parameters pass the verification, Unknown 8 included.

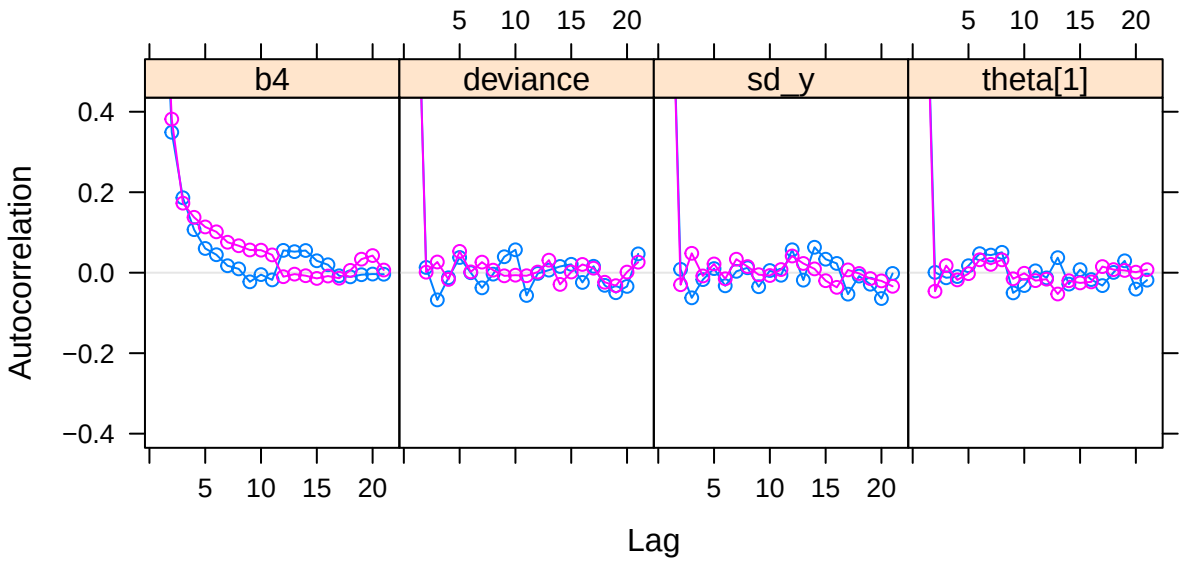
Autocorrelation In order to get good approximation, from the generated sequences of posterior samples, is necessary that the *autocorrelation* (correlation between consecutive values of the chain) is low. A high degree of autocorrellation obstructs the parameter from jumping between different regions of the parameter space in one step (the chains get "stuck" is some region) and therefore the convergences to the target distribution is slower.

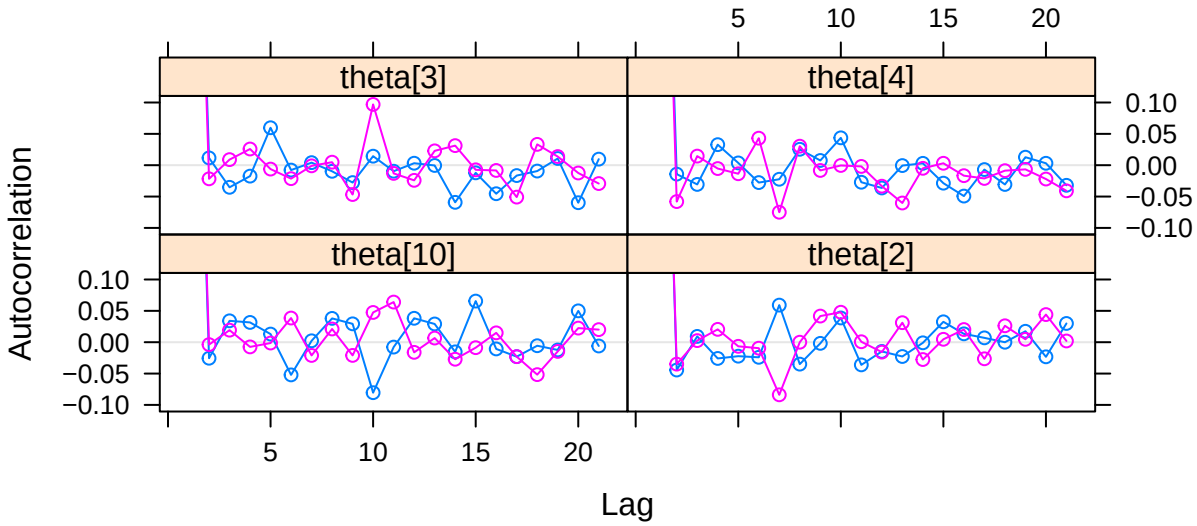
For a generic sequence of S numbers $\theta_1, \theta_2, ..., \theta_S$ the lag-t *sample autocorrelation* estimates the correlation between elements of the sequence that are t steps apart:

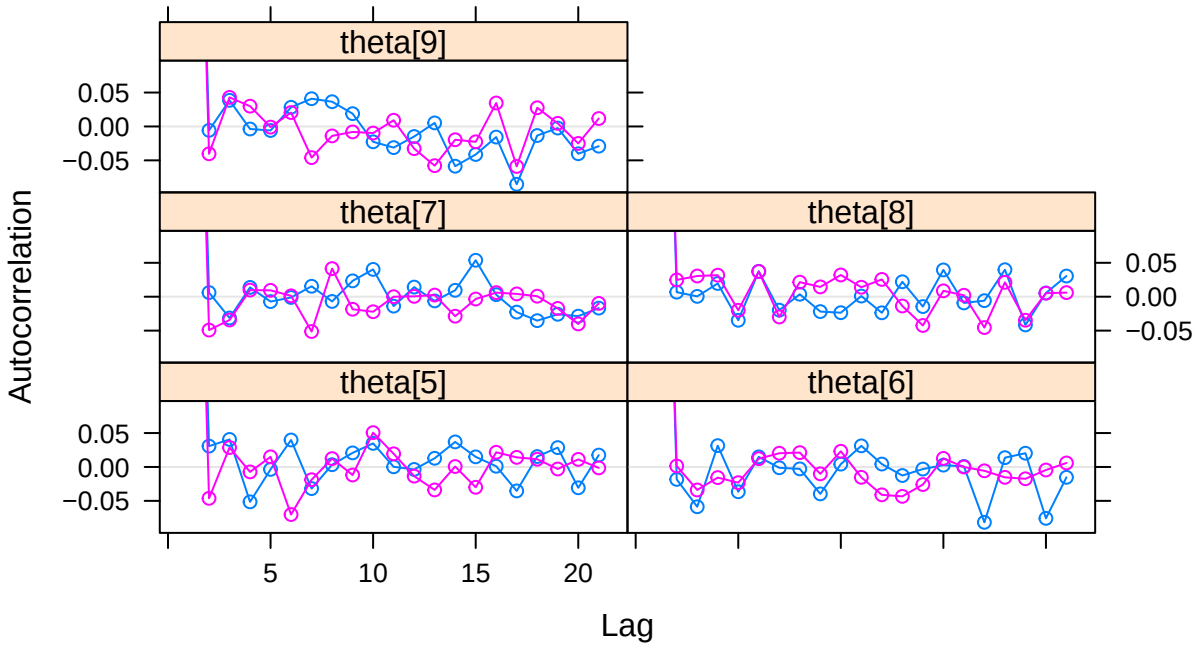
$$acf_t(\theta) = \frac{\frac{1}{S-t}\sum_{s=1}^{S-t}(\theta_s - \bar{\theta})(\theta_{s+t} - \bar{\theta})}{\frac{1}{S}\sum_{s=1}^{S}(\theta_s - \bar{\theta})^2}$$

The autocorrelation plots allows us to detect the presence of some misbehaviour of chains and whetever chains needs more time to converge.




b4
deviance
sd_y
theta[1]
Autocorrelation
0.4
0.2
0.0
−0.2
−0.4
5
10
15
20
Lag


theta[3]
theta[4]
theta[10]
theta[2]
Autocorrelation
Lag
5
10
15
20
0.10
0.05
0.00
−0.05
−0.10



Gelman–Rubin diagnostic (Potential Scale Reduction Factor) The Gelman–Rubin diagnostic assess convergence of a Monte Carlo Markov Chain comparing the estimated between-chains and within-chain variances of several independent simulated Markov chains, for each model parameter. If chain have converged, the ratio of *inter-chain* to *intra-chain variances* for all the parameters sampled is close to 1.

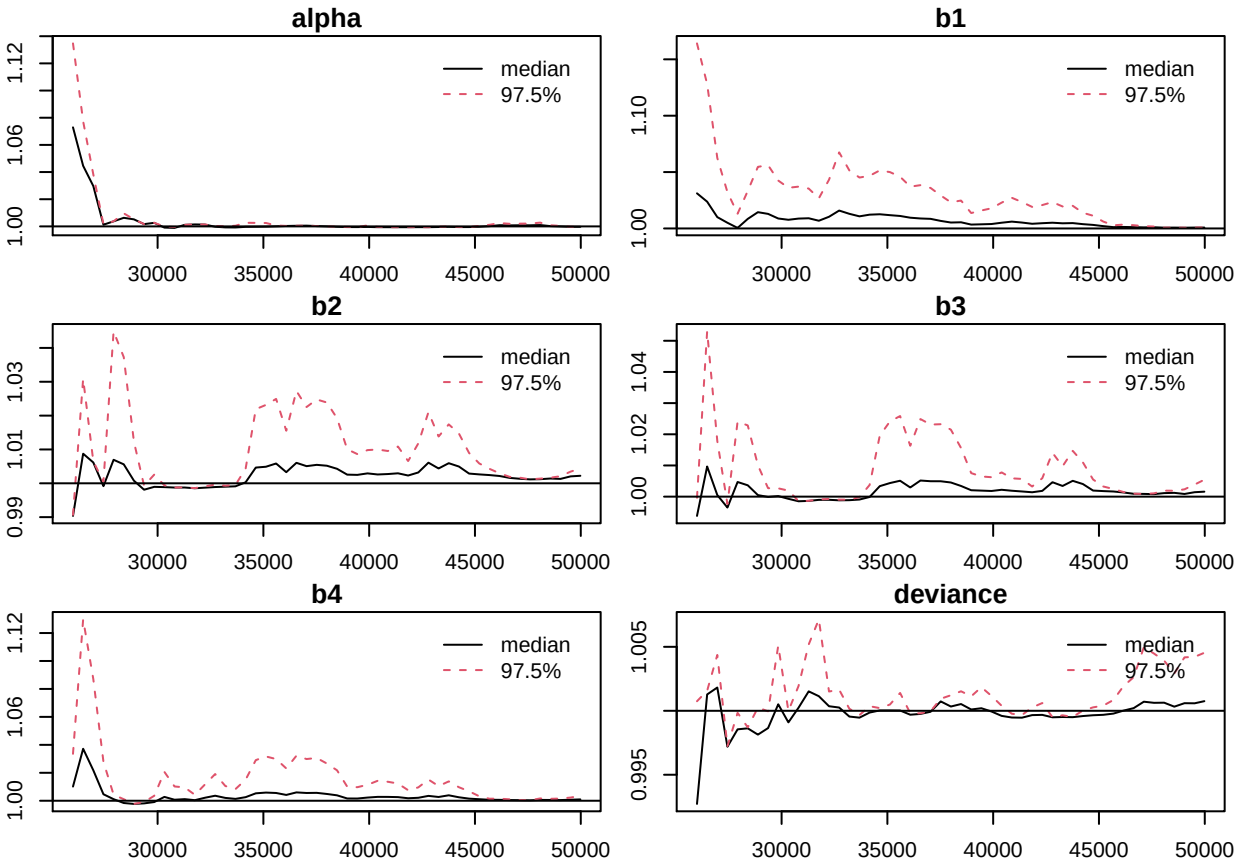
The *Potential Scale Reduction Factor (PSRF)* for a model parameter θ is defined to be the ratio of the *estimates of the posterior variance* of θ and the *within-chain variances*.

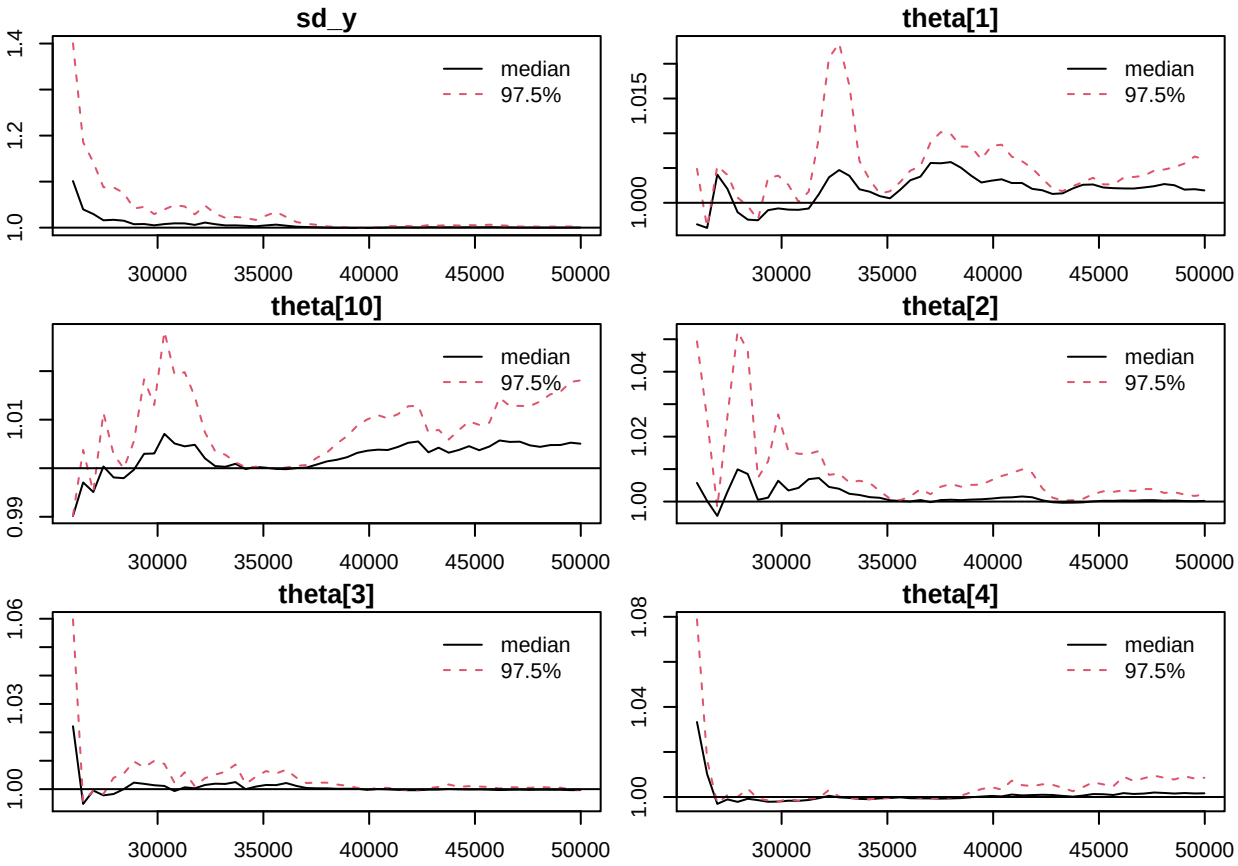
The PSRF estimates the potential decrease in the between-chains variability with respect to the within-chain variability.

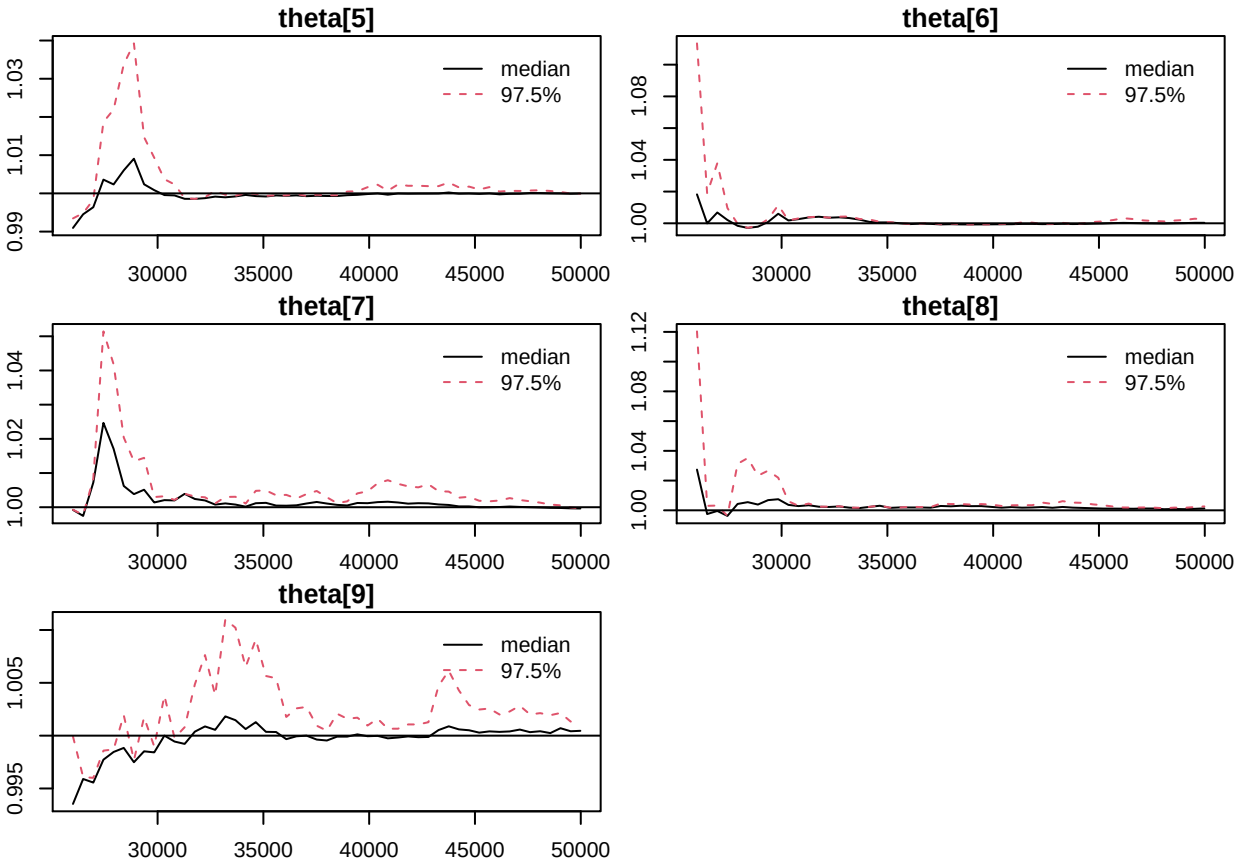
The following graphs show the PSRF median (black), within the upper 95% confidence limits (red), as the number of iterations increases.

After 30 000 iterations all R values are below the threshold of 1.1 and they do not exceed their corresponding upper confidence limits at the 95% confidence level.

```
par(mfrow= c(3,2), mgp=c(3,1,0), oma = c(0,0,0,0), mar = c(2,2,1.5,1))
coda::gelman.plot(mod.fit.mcmc, auto.layout = F)
```


alpha
median
97.5%
1.00
1.06
1.12
30000
35000
40000
45000
50000
b1
median
97.5%
1.00
1.10
30000
35000
40000
45000
50000
b2
median
97.5%
0.99
1.01
1.03
30000
35000
40000
45000
50000
b3
median
97.5%
1.00
1.02
1.04
30000
35000
40000
45000
50000
b4
median
97.5%
1.00
1.06
1.12
30000
35000
40000
45000
50000
deviance
median
97.5%
0.995
1.005
30000
35000
40000
45000
50000


sd_y
median
97.5%
1.0
1.2
1.4
30000
35000
40000
45000
50000
theta[1]
median
97.5%
1.000
1.015
30000
35000
40000
45000
50000
theta[10]
median
97.5%
0.99
1.01
30000
35000
40000
45000
50000
theta[2]
median
97.5%
1.00
1.02
1.04
30000
35000
40000
45000
50000
theta[3]
median
97.5%
1.00
1.03
1.06
30000
35000
40000
45000
50000
theta[4]
median
97.5%
1.00
1.04
1.08
30000
35000
40000
45000
50000


theta[5]
median
97.5%
0.99
1.01
1.03
30000
35000
40000
45000
50000
theta[6]
median
97.5%
1.00
1.04
1.08
30000
35000
40000
45000
50000
theta[7]
median
97.5%
1.00
1.02
1.04
30000
35000
40000
45000
50000
theta[8]
median
97.5%
1.00
1.04
1.08
1.12
30000
35000
40000
45000
50000
theta[9]
median
97.5%
0.995
1.005
30000
35000
40000
45000
50000

Correlation

Looking a the column "b3" of the correlation matrix plot, we can see that the *Unknown concentration* θ have a strong posterior correlation with the parameter β_3. We recall that β_3 is the inflection point of the logistic curve, the concentration at which the curvature line turns (see *Model Structure* Section). This fact leads us to work with the parameters γ when running the Gibbs algorithm, instead of β, where $\gamma = \frac{\theta}{\beta_3}$.

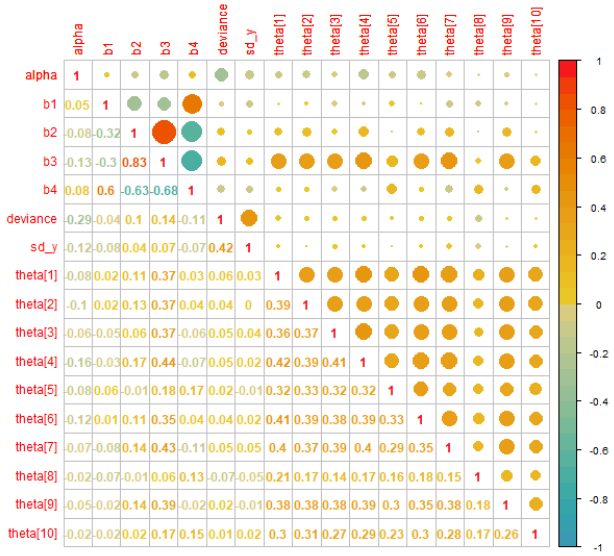


Figure 2: half-size correlation matrix

Table 3: Point estimates comparison for the two model.

	Equal Variance Model						Unequal Variance Model					
				95 % HPD						95 % HPD		
	mean	sd	MCSE	low	up	width	mean	sd	MCSE	low	up	width
b1	14.9892	0.5589	0.0111	13.9481	16.1213	2.1732	14.7051	0.2524	0.0049	14.2639	15.2338	0.9699
b2	103.7033	2.9445	0.0579	97.9983	109.5275	11.5291	99.8087	4.8459	0.1078	90.9714	109.6938	18.7224
b3	0.0675	0.0047	0.0001	0.0590	0.0772	0.0182	0.0553	0.0056	0.0001	0.0450	0.0664	0.0214
b4	1.3248	0.0690	0.0014	1.1943	1.4609	0.2667	1.3514	0.0633	0.0013	1.2259	1.4736	0.2477
sd_y	3.2436	0.2503	0.0050	2.7978	3.7399	0.9421	1.9393	0.1582	0.0033	1.6519	2.2507	0.5989
theta[1]	0.0281	0.0029	0.0001	0.0226	0.0339	0.0113	0.0233	0.0019	0.0000	0.0195	0.0270	0.0075
theta[2]	0.0304	0.0030	0.0001	0.0248	0.0367	0.0119	0.0259	0.0021	0.0000	0.0221	0.0303	0.0081
theta[3]	0.1196	0.0081	0.0002	0.1042	0.1356	0.0314	0.1054	0.0084	0.0002	0.0889	0.1212	0.0322
theta[4]	0.0386	0.0033	0.0001	0.0322	0.0451	0.0128	0.0311	0.0025	0.0000	0.0267	0.0362	0.0095
theta[5]	0.0166	0.0026	0.0001	0.0115	0.0218	0.0104	0.0146	0.0014	0.0000	0.0121	0.0176	0.0055
theta[6]	0.0254	0.0028	0.0001	0.0202	0.0315	0.0113	0.0219	0.0019	0.0000	0.0183	0.0257	0.0074
theta[7]	0.0802	0.0056	0.0001	0.0697	0.0913	0.0216	0.0701	0.0056	0.0001	0.0590	0.0806	0.0216
theta[8]	0.0001	0.0006	0.0000	0.0000	0.0000	0.0000	0.0057	0.0011	0.0000	0.0036	0.0078	0.0042
theta[9]	0.0359	0.0033	0.0001	0.0297	0.0421	0.0124	0.0315	0.0026	0.0001	0.0266	0.0367	0.0102
theta[10]	0.0135	0.0026	0.0001	0.0084	0.0185	0.0101	0.0127	0.0014	0.0000	0.0099	0.0152	0.0053

Table 4: Point estimates comparison for the two model.

	DIC
Equal Variance Model	515.9398
Unequal Variance Model	418.1082

Comparison of the models

We illustrate the inference for the data given by the two models.

Since convergence were assested, the parallel chains MCMC are merged.

Table 3 reports: the *posterior mean estimates* of the parameters of the calibration curve $\hat{\beta}$, estimates for the unknown concentrations $\hat{\theta}_j$, the posterior standard deviation, the Monte Carlo Standard Error and *95% HPD confidence interval*.

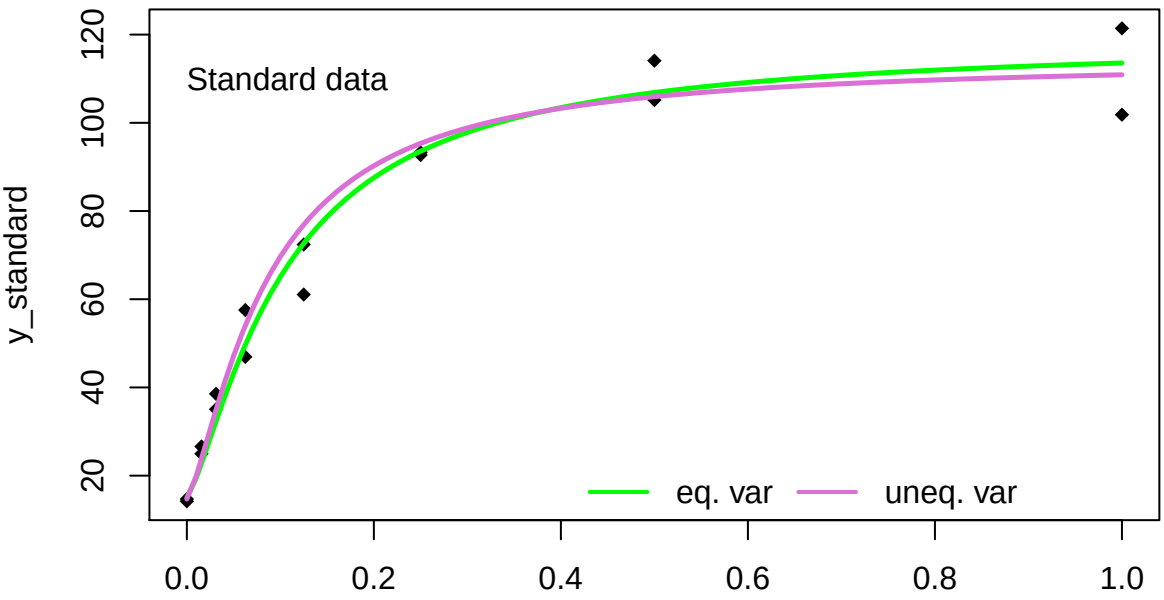
The sd column reports the marginal posterior standard deviation.

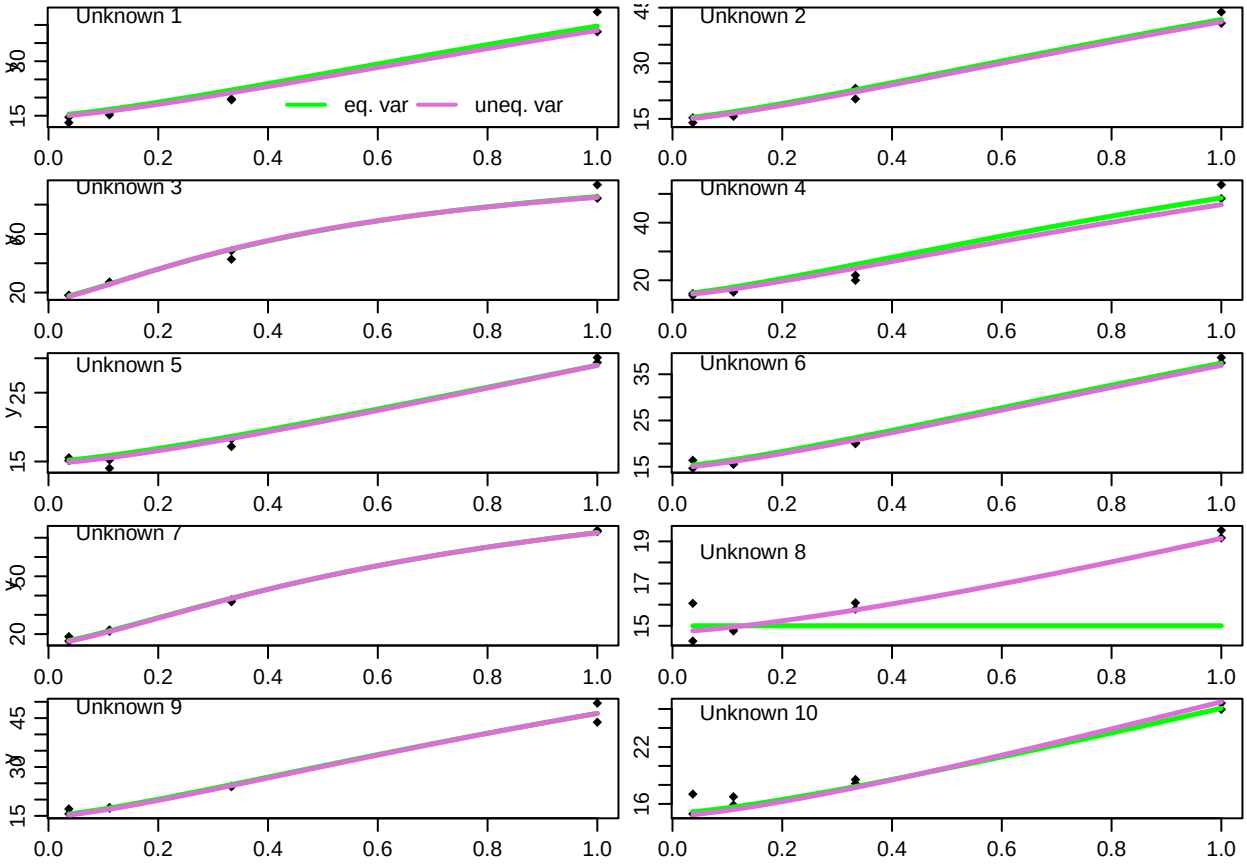
The **MCSE** column reports the **Monte Carlo Standard Error**, that estimates of inaccuracy of Monte Carlo samples regarding the marginal posterior mean. The MCSE take into account non-independence in the generated sequence. Non-independence is estimated with the ESS function for Effective Sample Size. The MCSE can be then estimated modifing the formula for standard error $se = \frac{\sigma}{\sqrt{n}}$ in $MCSE = \frac{\sigma}{\sqrt{ESS}}$. However, we use the `MCSE` function in the *LaplacesDemon* package that estimates the MCSE using the Geyer's Initial Monotone Positive Sequence (IMPS) estimator *(Geyer, 1992)*.

We notice that the second model (unequal variance) is generally characterized by a smaller MCSE and a narrower HPD confidence interval.

The estimates define a curve $g(x, \hat{\beta})$,used to draw curves for each unknowns displayed below, that allows us to make comparisons.

As expected, the curve goes throught the data (black points in the plot). The estimated curve are similar, except that there is no estimate for Unknown 8 for the simpler model (equal variance).


Standard data
y_standard
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
1.0
eq. var
uneq. var

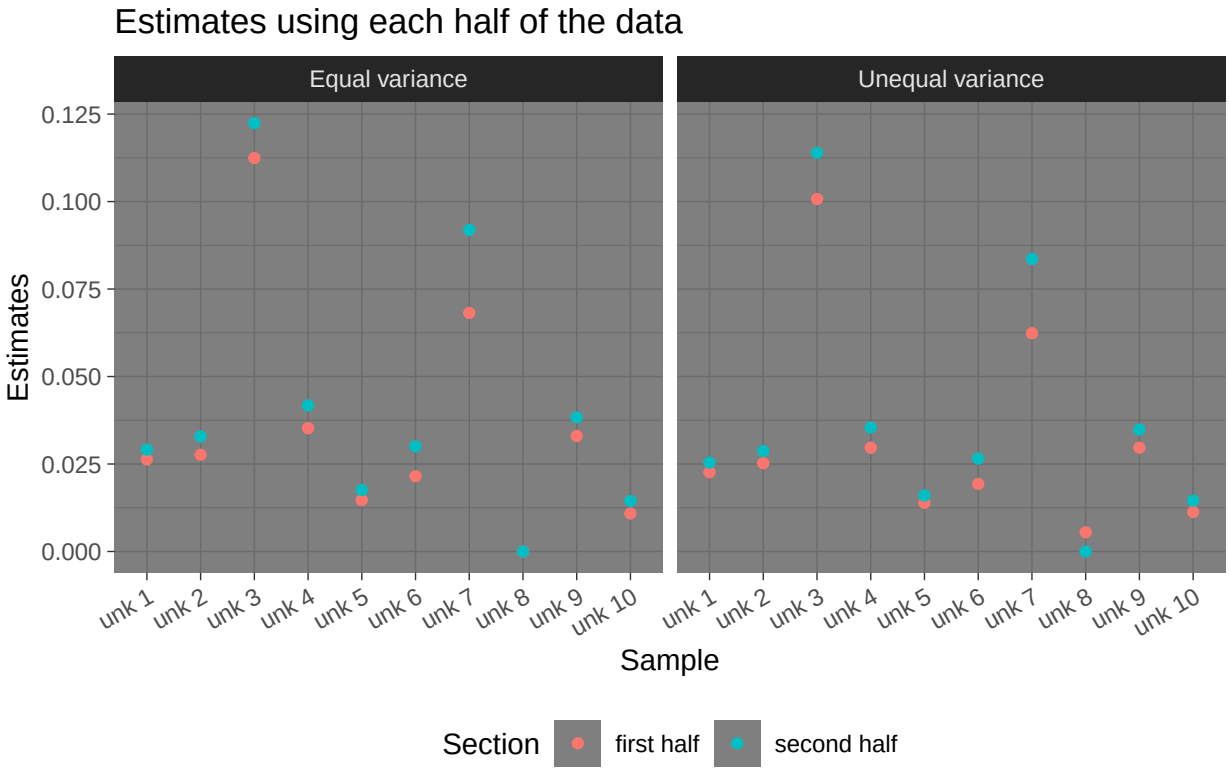


Comparison of the models: spitting the dataset In this section we compare the inference made by the two models in different ways.

Remember that in our dataset for each dilution level two measurements are provided.

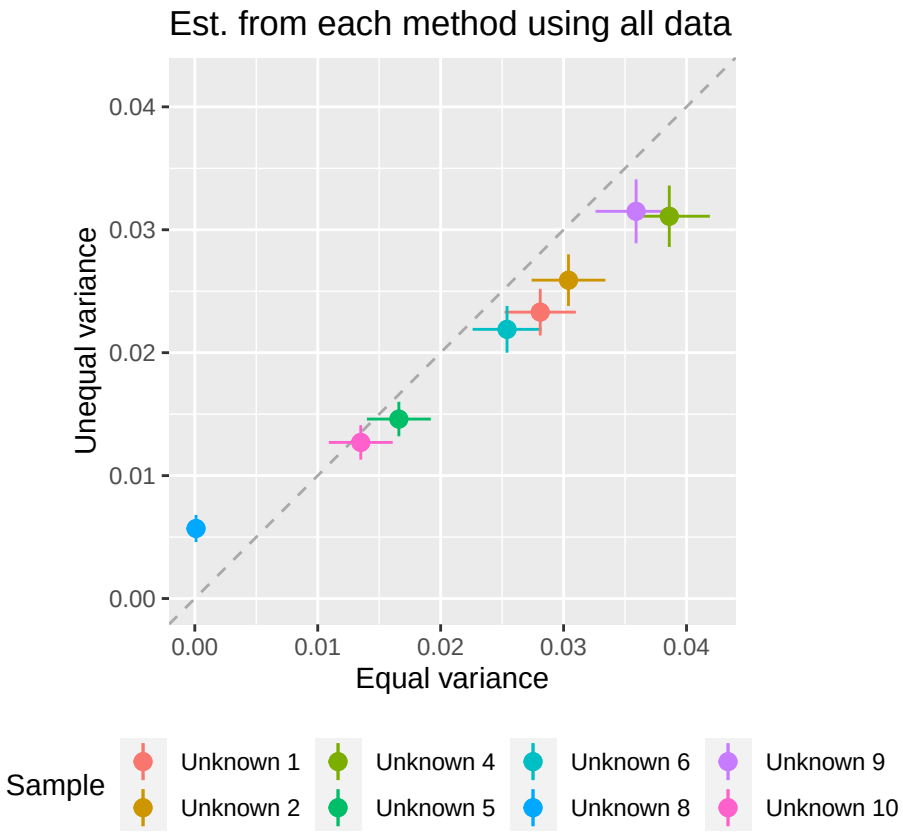
In order to compare the estimates produced by the two models:

- We divide the dataset into two halves (the first 4 rows and the last 4 rows in table 1).
- With each half we fit the first model (equal variance) and the most complex model (unequal variance).
- Finally we compare the estimates given for each of the two halves, in order to see which of the two models is more consistent.

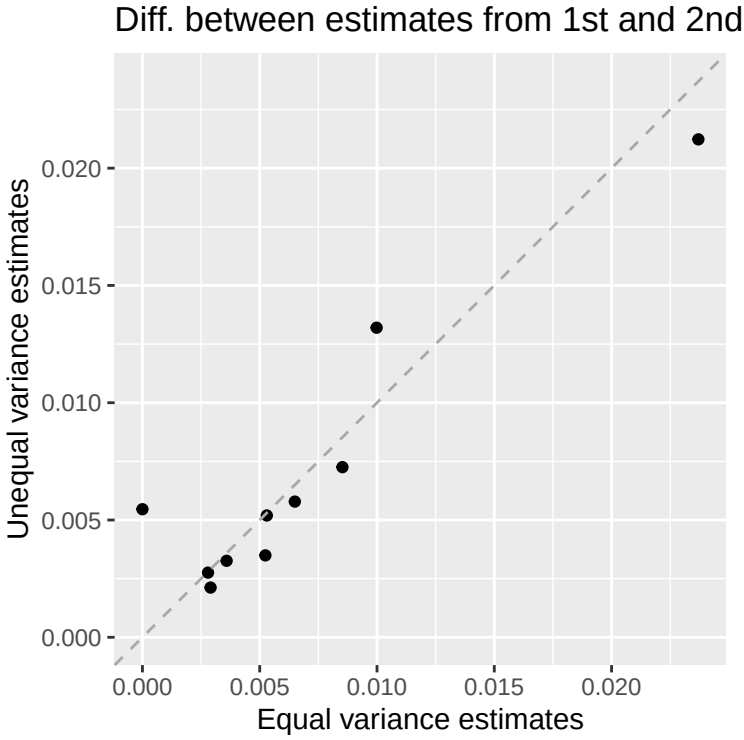


The two estimates, using just the first or second half of the data,are closer under the 'unequal varaince model' than the 'equal variance' model.

The plot above shows the estimates from each of the two halves of the data. For each model, the two estimates are similar, but the consistency is much stronger for the second model (unequal variance).



The above graphs, instead, compare the two estimates directly, reporting the estimates of the second model (on the y axis) versus the estimates of the first model (x axis). The plot shows that the two models give similar estimates (the points are placed along the bisector), however the second model has a lower standard errors (vertical bars are shorter than horizontal bars).



	diff_eq_var	diff_uneq_var
1	0.0028	0.00276
2	0.00524	0.00349
3	0.00998	0.0132
4	0.0065	0.00579
5	0.00291	0.00212
6	0.00853	0.00725
7	0.0237	0.02123
8	0	0.00546
9	0.0053	0.00519
10	0.00359	0.00326

The above plot displays the asbolute difference between the first half and second half estimates under each model. We note how most of the time, the second model is more reliable.

Simulation

In order to assest whetever the Bayesian model defined is able to recover model parameters, we simulated from the second model.

We select ten values from the "true" concentrations in the range (0.004, 0.22). The range is chosen to be wider and includes even much smaller values than the previously used data.

We recall here the definition of the model, with the "true" parameters choosed:

$$E(y|x,\beta) = g(x,\beta) = \beta_1 + \frac{\beta_2}{1 + (x/\beta_3)^{-b4}}$$

$$y_i \sim N(g(x_i,\beta), \sigma_y^2), \quad x_i = d_i * \theta_j$$

As calibration data we use the *Standard data* provided previously.

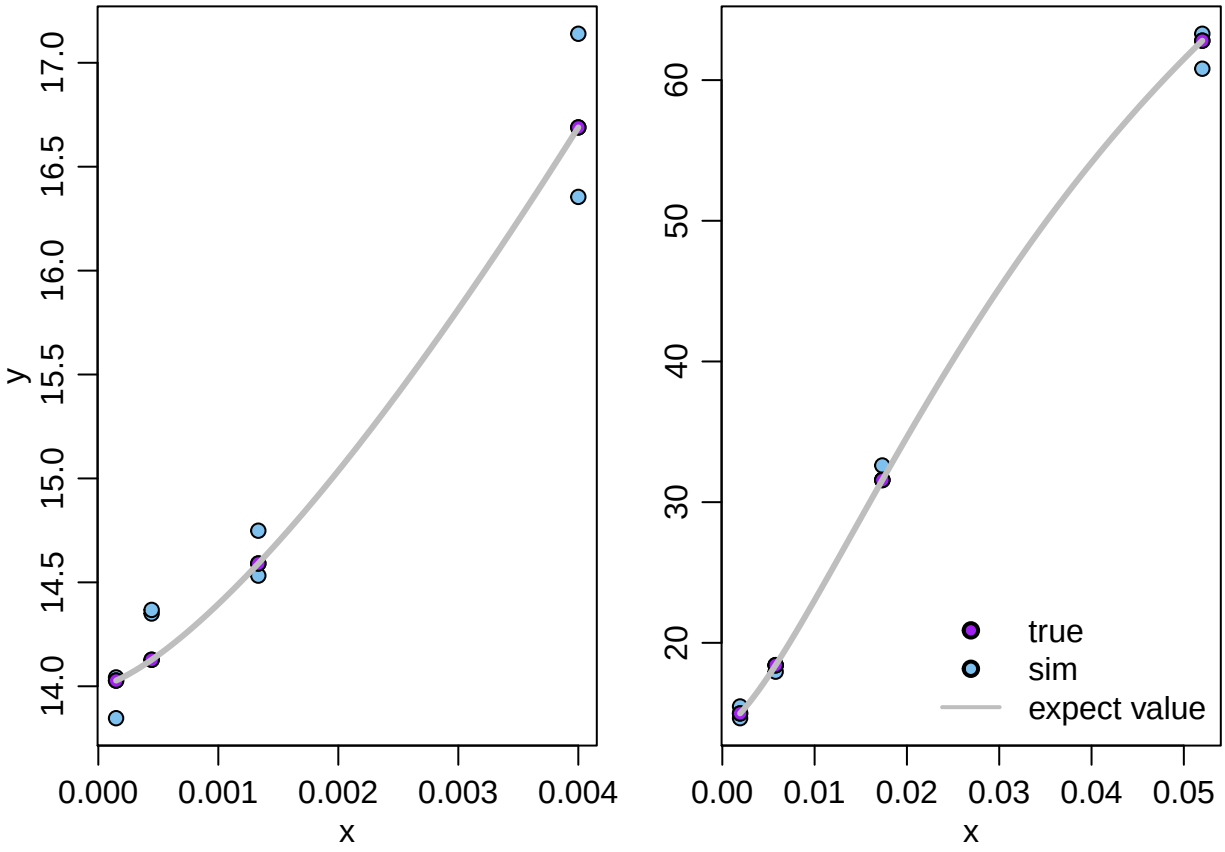
In order to simulated the "unknown" data:

For each "true" concentration θ:

1. we multiple θ by the each dilution coefficents (1, 1/2, 1/4 ...) d_i showed in Table 1 (under columns "Unknown"). In this way we obtain x_i.
2. we compute the expected value as $g(x,\beta) = \beta_1 + \frac{\beta_2}{1+(x/\beta_3)^{-b4}}$

3. we compute the variance of the measurements, that models the *measurement error*, as $\sigma^2 = \left(\frac{g(x_i,\beta)}{A}\right)^{2\alpha} \sigma_y^2$
4. we simulated the measurement as $y \sim dnorm(\mu = g(x,\beta), \sigma^2 = \left(\frac{g(x_i,\beta)}{A}\right)^{2\alpha} \sigma_y^2)$

Example of simulated data, within the expected value curve, is showed:




```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 96
##    Unobserved stochastic nodes: 16
##    Total graph size: 722
##
## Initializing model
```

```
## Inference for Bugs model at "complex_model_2.txt", fit using jags,
##  2 chains, each with 50000 iterations (first 25000 discarded), n.thin = 20
##  n.sims = 2500 iterations saved
##           mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat n.eff
## alpha       0.970   0.029   0.892   0.959   0.979   0.991   0.999 1.001  2500
## b1         13.862   0.200  13.469  13.730  13.859  13.992  14.255 1.005   330
## b2        100.532   2.429  96.110  98.833 100.442 102.124 105.488 1.009   180
## b3          0.053   0.003   0.048   0.051   0.053   0.055   0.059 1.012   150
## b4          1.327   0.032   1.265   1.305   1.328   1.349   1.392 1.011   150
## sd_y        1.839   0.148   1.584   1.734   1.832   1.933   2.148 1.001  2500
```

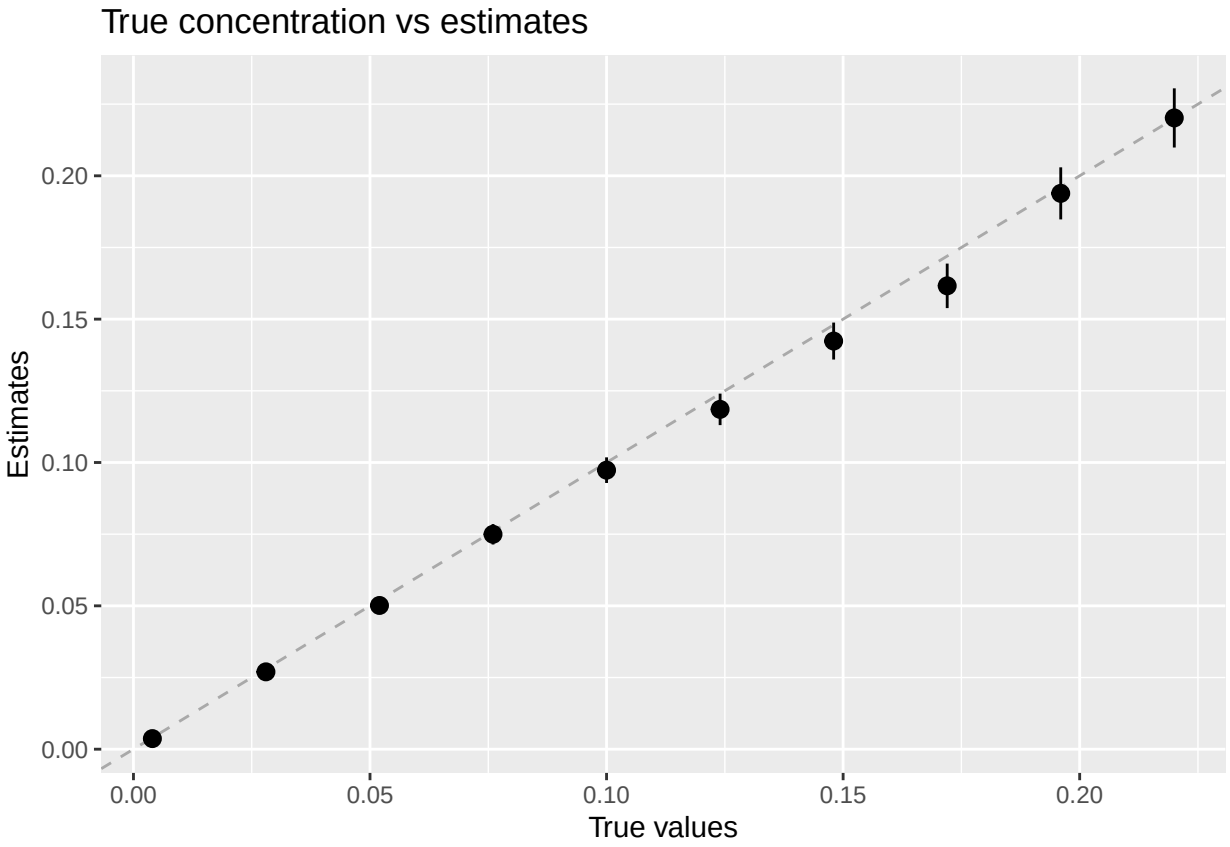
```
## theta[1]    0.004   0.001   0.003   0.003   0.004   0.004   0.005 1.002  1400
## theta[2]    0.027   0.001   0.024   0.026   0.027   0.028   0.030 1.001  1900
## theta[3]    0.050   0.002   0.046   0.049   0.050   0.052   0.055 1.001  2500
## theta[4]    0.075   0.004   0.068   0.072   0.075   0.077   0.082 1.005   770
## theta[5]    0.097   0.004   0.089   0.094   0.097   0.100   0.106 1.008   210
## theta[6]    0.119   0.005   0.108   0.115   0.118   0.122   0.130 1.003   640
## theta[7]    0.142   0.006   0.130   0.138   0.142   0.146   0.155 1.001  2500
## theta[8]    0.162   0.008   0.147   0.156   0.161   0.167   0.178 1.001  2500
## theta[9]    0.194   0.009   0.177   0.188   0.194   0.200   0.213 1.001  2300
## theta[10]   0.220   0.010   0.201   0.213   0.220   0.227   0.241 1.002  1100
## deviance  369.391   6.413 359.259 364.768 368.677 373.134 383.972 1.001  2500
##
## 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 = 20.6 and DIC = 390.0
## DIC is an estimate of expected predictive error (lower deviance is better).
```

After running JAGS we compare the "true" values with the estimates:

Table 5: Point estimates simulated data.

	true_value	mean	sd	MCSE
b1	14.000	13.8620	0.1995	0.0040
b2	95.000	100.5317	2.4290	0.0471
b3	0.050	0.0533	0.0029	0.0001
b4	1.400	1.3273	0.0325	0.0007
theta[1]	0.004	0.0037	0.0006	0.0000
theta[2]	0.028	0.0270	0.0013	0.0000
theta[3]	0.052	0.0501	0.0024	0.0001
theta[4]	0.076	0.0750	0.0036	0.0001
theta[5]	0.100	0.0973	0.0045	0.0001
theta[6]	0.124	0.1185	0.0055	0.0001
theta[7]	0.148	0.1424	0.0064	0.0001
theta[8]	0.172	0.1616	0.0078	0.0002
theta[9]	0.196	0.1939	0.0091	0.0002
theta[10]	0.220	0.2202	0.0103	0.0002

We note how the estimates are pretty close to the true values. Plotting the true values versus the estimates ones, we see how they quite line up along the bisector line.


True concentration vs estimates
Estimates
0.00
0.05
0.10
0.15
0.20
True values
0.00
0.05
0.10
0.15
0.20

Frequentist Approach

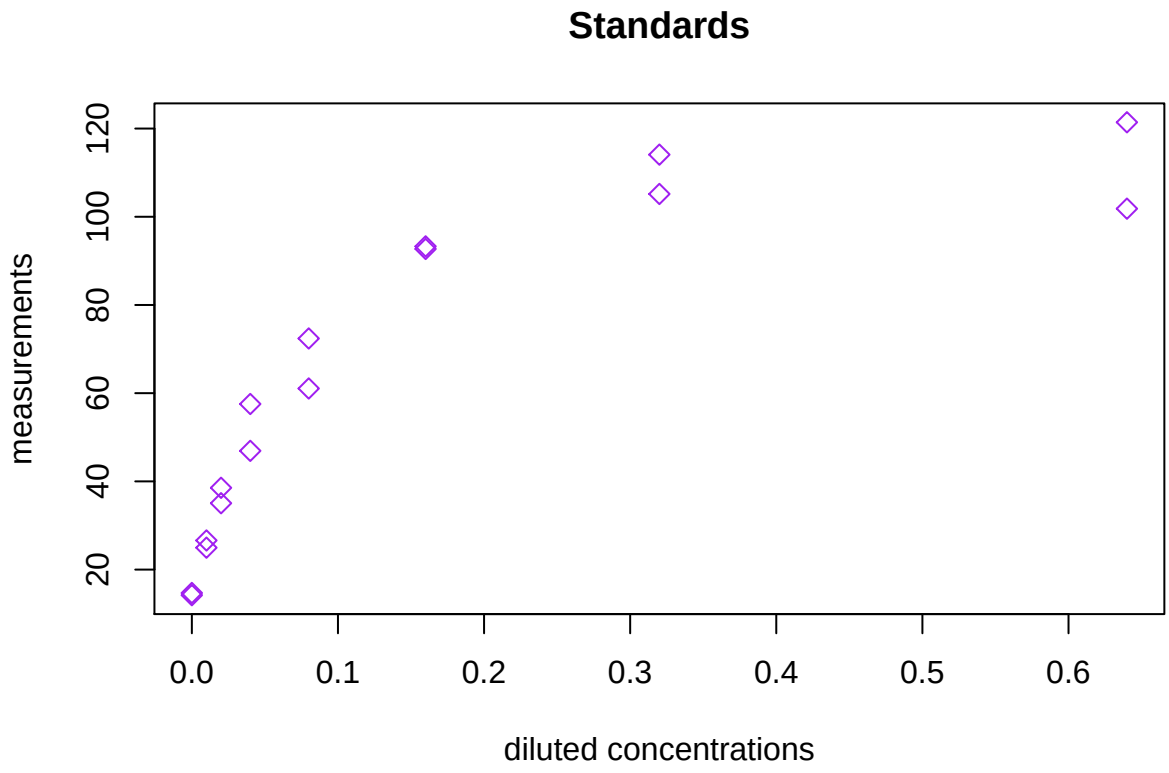
A usual frequentist approach to analysis of dilution assays follow two step.

First standards data are used to estimate the curve relating concentrations to measurements (the four-parameter logistic function), using least squares, $y = \beta_1 + \frac{\beta_2}{1+(x/\beta_3)^{-b4}}$, where y are measurements and x diluted concentration.

Second, this fitted curve is used to read off the (diluted) concetration that corrisponds to each of the measurements of the unknowns. Estimates of diluted concentration are scaled back to the original scale dividing by the dilution factors, and these are averaged to obtain an estimated concentration for each unknown.

For instances, looking at Table 7, the estimate concentration from *Unknown 8* is: $\frac{1}{4}(0.0040 + 0.0043 + 3 \times 0.0009 + 3 \times 0.0006) = 0.0032$. Data from dilutions 1/9 and 1/27 are not used in the estimate since those mesaurements are not available.

The second step (estimating the unknown concentrations) present serious problems. In reading concentrations directly off a curve, the method ignores measurement error, which is particularly serious for every high level of dilution, where the curve is flat.




```
nlsfit <- nls(y_standard ~ (b1 + b2/(1 + (x_standard/b3)^(-b4))),data =df, start = b_start)
summary(nlsfit)
```

```
##
## Formula: y_standard ~ (b1 + b2/(1 + (x_standard/b3)^(-b4)))
##
## Parameters:
##     Estimate Std. Error t value Pr(>|t|)
```

Table 6: Estimates Frequentist Approach

	estimates	estimates_Std_Error	lower_ci	upper_ci	width
b1	15.3018	4.1513	7.9031	22.7006	14.7975
b2	108.1761	10.5750	89.3285	127.0238	37.6953
b3	0.0748	0.0149	0.0482	0.1014	0.0532
b4	1.1233	0.2023	0.7628	1.4839	0.7211

Table 7: Examples of inferences

d_unk	y_unk	index	estimate_dil_unk
1	19.164	8	0.0040
1	19.522	8	0.0043
1/3	16.090	8	0.0009
1/3	15.789	8	0.0006
1/9	14.860	8	NaN
1/9	14.755	8	NaN
1/27	14.274	8	NaN
1/27	16.063	8	0.0009
1	49.591	9	0.0378
1	43.750	9	0.0299
1/3	23.970	9	0.0085
1/3	24.087	9	0.0086
1/9	17.314	9	0.0022
1/9	17.573	9	0.0024
1/27	15.609	9	0.0004
1/27	17.122	9	0.0020

```
## b1  15.30184    4.15125   3.686 0.003114 **
## b2 108.17613   10.57499  10.229  2.8e-07 ***
## b3   0.07480    0.01492   5.012 0.000303 ***
## b4   1.12334    0.20229   5.553 0.000125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.229 on 12 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 2.347e-06
```

Looking at some examples of inferences for two unknown, we see how the current model is unable to estimate a value for sample "Unknown 8" when it is very diluted. The model is not precise enough to estimate smaller concentration.

Once beta parameters have been estimated, given the measurements y we obtain the diluted concentration x with the inverse formula:

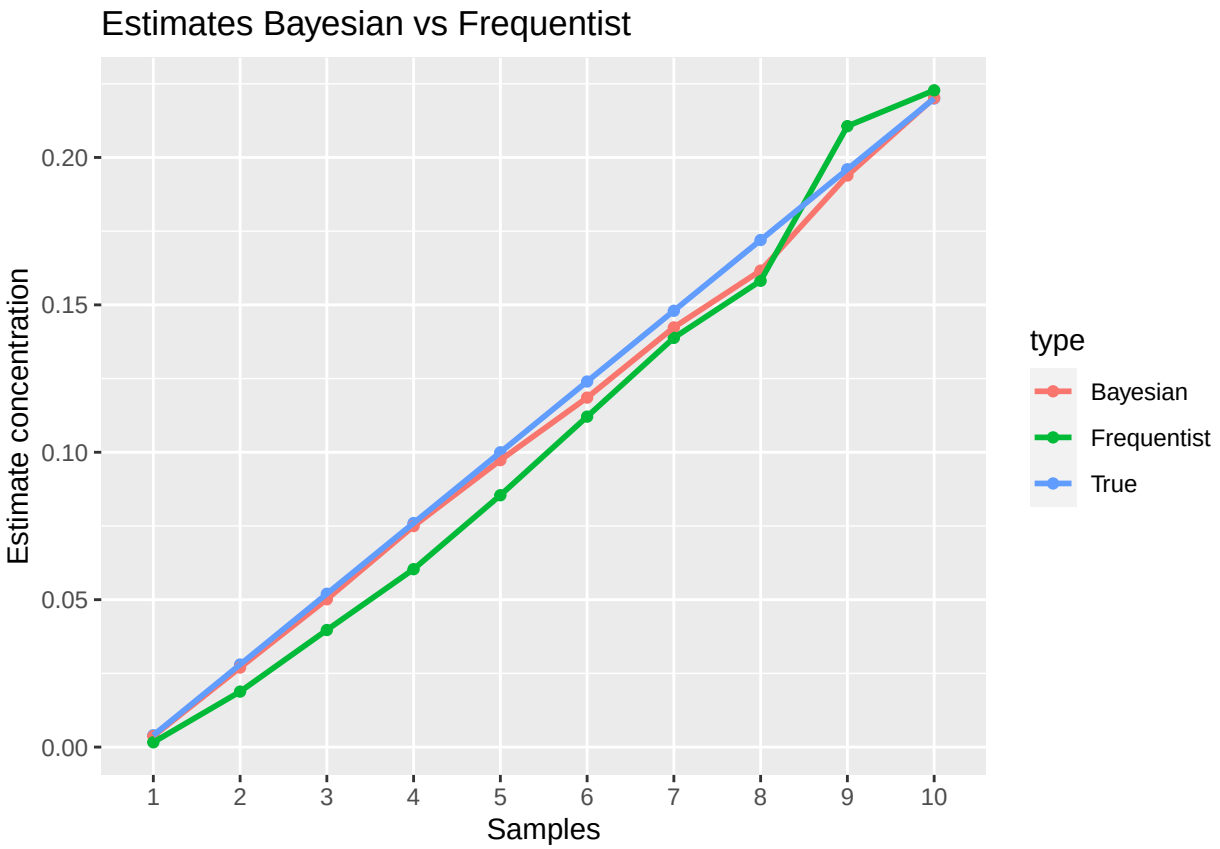
$$x = \frac{1}{\beta_3}\left(\frac{\beta_2}{y-\beta_1} - 1\right)^{-\beta_4}$$

Simulation for frequnetist approach In order to test the frequentist model and compare it with the Bayesian method, we estimate the parameters of interest with simulated data (of which we know the true value).

In the plot below we compare the ten simulated concentrations ("True" values in blue), with the estimates for those given by the frequentist approach (green), with the Bayesian estimates (red). We see clearly the Bayesian model gives estimates closer to true values.

Table 8: Bayes vs Frequentist: point estimates simulated data.

	true_value	bayes_estimates	freq_estimates
b1	14.000	13.862	15.302
b2	95.000	100.532	108.176
b3	0.050	0.053	0.075
b4	1.400	1.327	1.123
theta[1]	0.004	0.004	0.002
theta[2]	0.028	0.027	0.019
theta[3]	0.052	0.050	0.040
theta[4]	0.076	0.075	0.060
theta[5]	0.100	0.097	0.085
theta[6]	0.124	0.119	0.112
theta[7]	0.148	0.142	0.139
theta[8]	0.172	0.162	0.158
theta[9]	0.196	0.194	0.211
theta[10]	0.220	0.220	0.223



Discussion and further work

We have seen that, for the purpose of estimating unknown concentrations with serial dilution, a Bayesian formulation of the standard four-parameter logic model is able to make inference with reasonable accuracy. The model can be improved so that it will be more accurate at the extremes of very low measurements. We have see how inferences for those measurements are sensitive to the assumed variance.

References

- Gelman, A., Ginger L. G., Shnaidman M. (2004), "Bayesian Analysis of Serial Dilution Assays".
- Gelman, A., Carlin, J. B., Stern, H. S.„ Rubin, D. B. (2004). "Bayesian Data Analysis". Chapman and Hall/CRC.