

## NIMBLE AND MY MCMC

I have used Nimble with the code in `MCMC_Nimble()` function because I was interested in seeing if my implementation would be trustable (Just for information: the problem arose since in the implementation of the BLRM of Neuenschwander et al.

(<https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.3230>) the posterior probability of the probability of DLT at each dose is analysed for some intervals and the EWOC approach is applied. This approach means that if the probability of a certain interval is over a certain threshold then we can not use the specific dose. In the paper a threshold of 0.25 is used but this results are too strict for the probabilities that I obtain. Given so I was looking for possible problems and justifications at each level of the implementation. Spoiler: even using the Nimble code, the EWOC is too strict so I have to ask to Alexandre for this).

**Problem:** If the Nimble code is correct then I obtain some different results with respect to what I have in my implementation and I am not sure on how to interpret the differences and where the problem might be.

### Context:

Adaptive MCMC done with global adaptive scaling. With references:

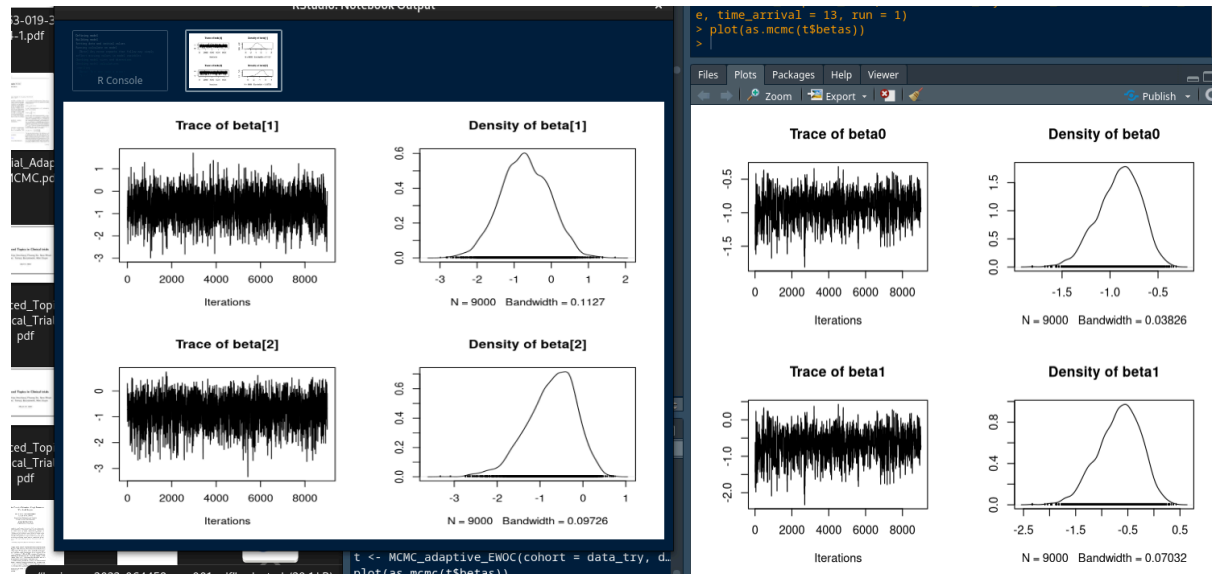
1. [https://biodatascience.github.io/statcomp/advmcmc/advmcmc.html#24\\_Adaptive\\_Metropolis\\_Algorithm](https://biodatascience.github.io/statcomp/advmcmc/advmcmc.html#24_Adaptive_Metropolis_Algorithm),
2. <https://journals.sagepub.com/doi/pdf/10.1177/1536867X1401400309#cite.AT08@-11>
3. <https://link.springer.com/article/10.1007/s11222-008-9110-y>

Set the delta parameter to 0.01 so that my acceptance rate is at 0.24.

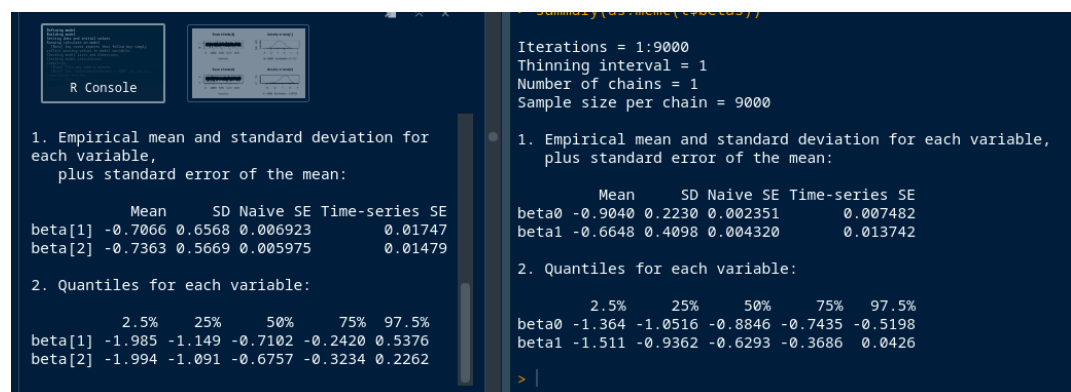
The lambda proposal is set at the beginning to  $\log((2.38^2)/2)$  as seen in the references.

### Examples

## NIMBLE + ADAPTIVE MCMC (see MCMC\_adaptive\_EWOC())



With summary stats

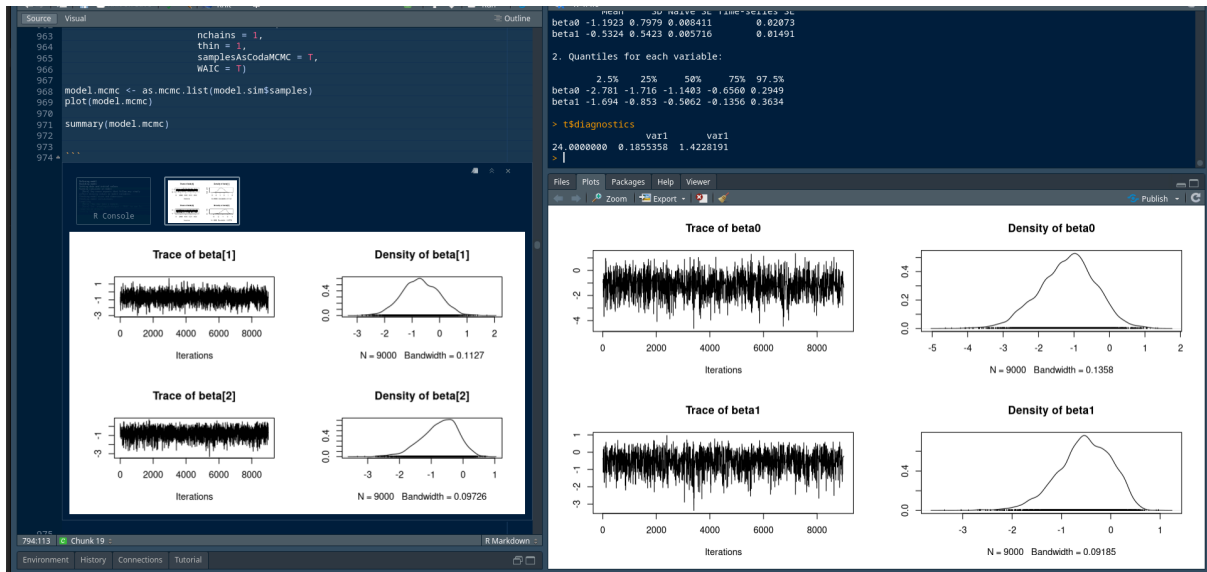


## Questions and considerations:

- Beta1 (beta[2]) shows quite an agreement: is this enough?
- Beta0 (beta[1]) does not have such an agreement: why? Why do I have these differences in general? Due to Adaptive MCMC?
- Very different SD: due to adaptive? Gain of precision: is this okay?
- My betas distributions are similar between them while the Nimble ones are quite different

## Changing delta parameter

If I change the delta to 0.5 (instead that 0.01) I obtain

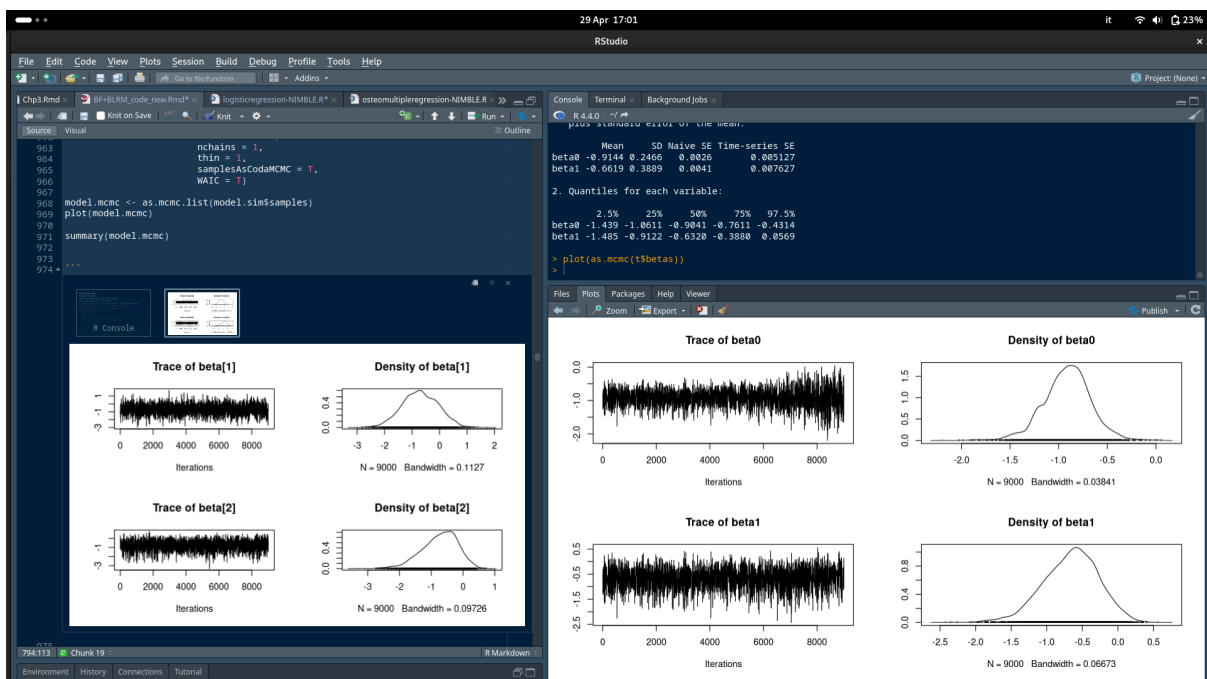


The summary stats for Nimble are as before as nothing changed there.

### Questions and considerations:

- Reduced agreement between the parameters in both cases: do I have to change the delta parameter? But this is just an example of an MCMC so how can I tune delta based on this? It seems not the way but then: the tuning of Delta should be based on the acceptance rate only? (Note: with these values I have ac of 0.24 and good geweke diagnostics)
- Increased SE

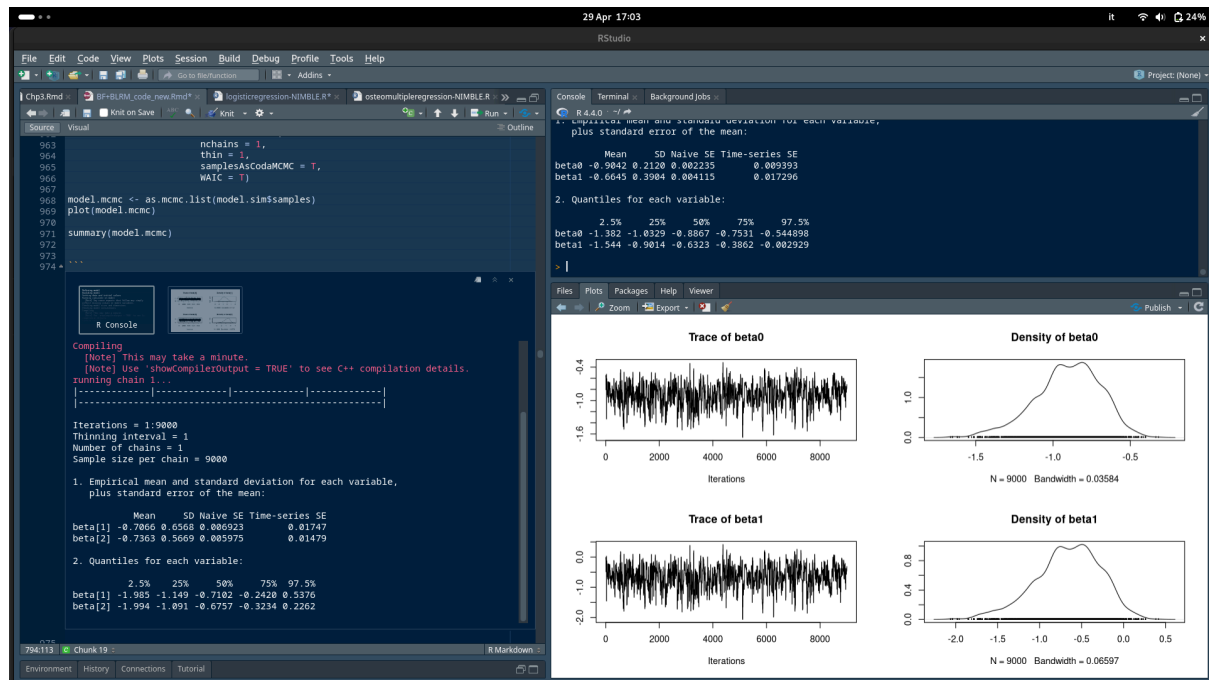
If I change delta = 1



## Questions and considerations:

- Increased agreement between the parameters and reduced SE but more variable ac (not perfectly 0.24)

If I change delta = 0.001

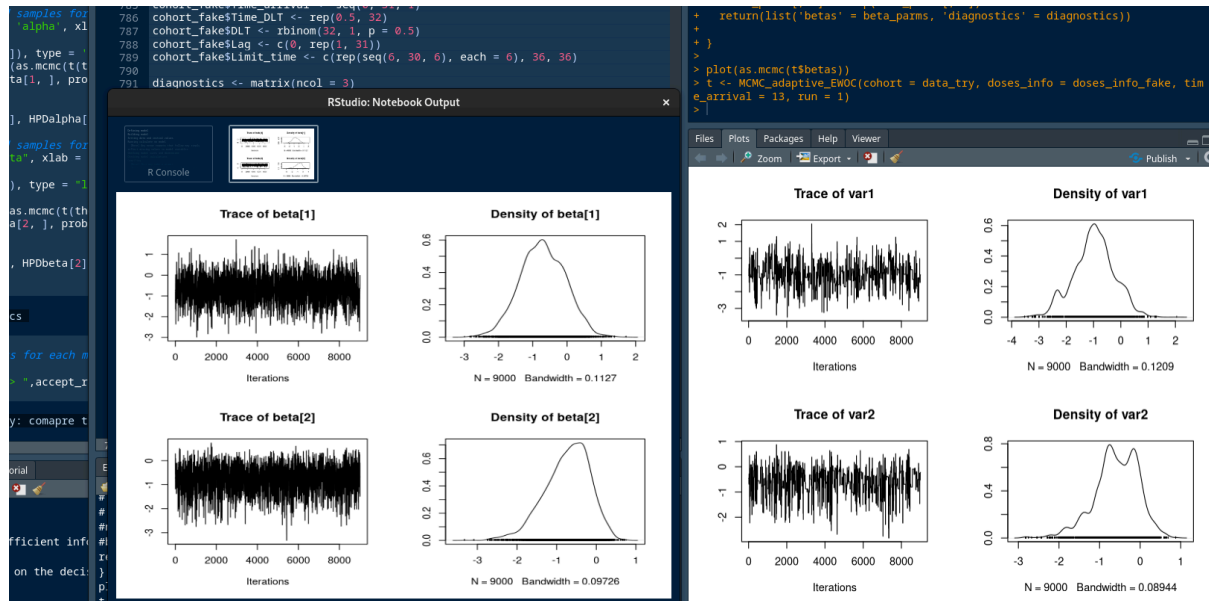


## Questions and considerations:

- Quite an agreement but not in the tails: I should consider the entire distribution and not just the mean as I use the entire distribution to derive the posterior distribution of the probabilities at each dose level.

What if the distributions are different because of the adaptive MCMC? Try with the not-adaptive MCMC())

## NIMBLE + MCMC (not adaptive. See function MCMC())



## Questions and considerations:

- The two distributions are somewhat quite similar so possibly what I saw before was due to the adaptive mechanism(?)
- Why does Nimble not have the same problem I have for the acceptance rate? The trace-plots are much better in the Nimble code: what am I missing?

## Question regarding the adaptive (MCMC\_adaptive\_EWOC())

- I have 'tuned' the delta parameter as to have ac of 24: now for every MCM I run I have 24 as ac, is this normal? I do not have a variability there: is this possible? Since I want to have that specific ac and I am looking for a specific value to obtain it, then it might sound reasonable, but I am not sure about the absence of the variability.
- For the delta parameter: should I just focus on the good ac and good geweke diagnostics to tune it? Given the results with Nimble I am not sure, but I think that I might be over-focusing on a single example.
- In one of the implementations that I have tried (# Scenario 5: `true_pDLT <- c(0.02, 0.05, 0.08, 0.11, 0.25, 0.39)`, `true_presp <- c(0.05, 0.1, 0.15, 0.2, 0.3, 0.4)`, `response_10 <- BFBLRM(doses, true_pDLT, true_presp, cohortsize, n_max, n_cap, n_stop, DLT_time, lambda, i_simulation = 4)`) I had a problem with the adaptation of the VarCov matrix that became 0 at the first cycle. So I have added a small quantity (0.001, see `MCMC_adaptive_EWOC()`) as I have seen from a reference that can be an approach (to avoid being stuck <https://keichenseer.com/post/a-metropolis-algorithm-in-r-part-2-adaptive-proposals/>, see image below). Is this a valid approach? (Honestly, I could not find any reference at end of the mentioned link but the idea sounded quite reasonable)

```

if(i>=3) {
  weights = allWeights[(nAdapt-i+2):nAdapt-1] # select weights for current iteration
  sum_weights = sum(weights)
  weighted_var_coeff <- apply(coefficients[2:i,], 2, # calculate weighted variance
    function(f) weighted_var(
      f, weights = weights, sum_weights = sum_weights))

  for(v in 1:4) {if(weighted_var_coeff[v]==0) { # if variance is zero, reduce sd
    proposal_sd[i+1,v] <- sqrt(proposal_sd[i,v]^2/5)
  } else proposal_sd[i+1,v] <- 2.4/sqrt(4) * sqrt(weighted_var_coeff[v]) + 0.0001
  # note that a small value is added to prevent that the adaptation gets stuck.
}
}

```