

# Simulation Results

See [https://github.com/stefanvermeent/ddm\\_simulations](https://github.com/stefanvermeent/ddm_simulations) for code.

## General approach

I take the following six steps for the simulation analyses:

1. Based on observed mean RTs, accuracy rates, and correlations between task conditions, simulate 10,000 trials per participant following a Gamma distribution (to get the signature right-skewed RT distributions).
2. Estimate DDM parameters based on simulated trials (which will be highly reliable because of the large number of trials).
3. For each DDM parameter, calculate the mean and SD, and use these to simulate *ground-truth* DDM parameters for individual participants
4. Simulate RTs/accuracies based on *ground-truth* DDM parameters, with the same number of trials as the original study
5. Apply Hierarchical Bayesian DDM to the simulated RTs/accuracies
6. Compare estimated DDM parameters with simulated *ground-truth* DDM parameters, under different conditions (e.g., different model specifications)

For each task, I simulated two scenarios using the number of trials in the mixed block and using the same number of participants as in the real dataset. In simulation 1, a single boundary separation (response caution) is estimated across conditions (i.e., assuming children have the same response caution on congruent and incongruent trials). It would be theoretically reasonable to constrain boundary separation this way, because the different trial types are intermixed randomly, with no advance cue of whether the current trial is going to be congruent or incongruent. Therefore, children have little time to adjust their speed-accuracy trade-off accordingly. A practical advantage is that it reduces the number of parameters to be estimated, which generally improve model fit. Nevertheless, I also looked into model performance when all three parameters are freely estimated in each condition (see simulation 2). Drift rate and non-decision time were always estimated separately for congruent and incongruent condition.

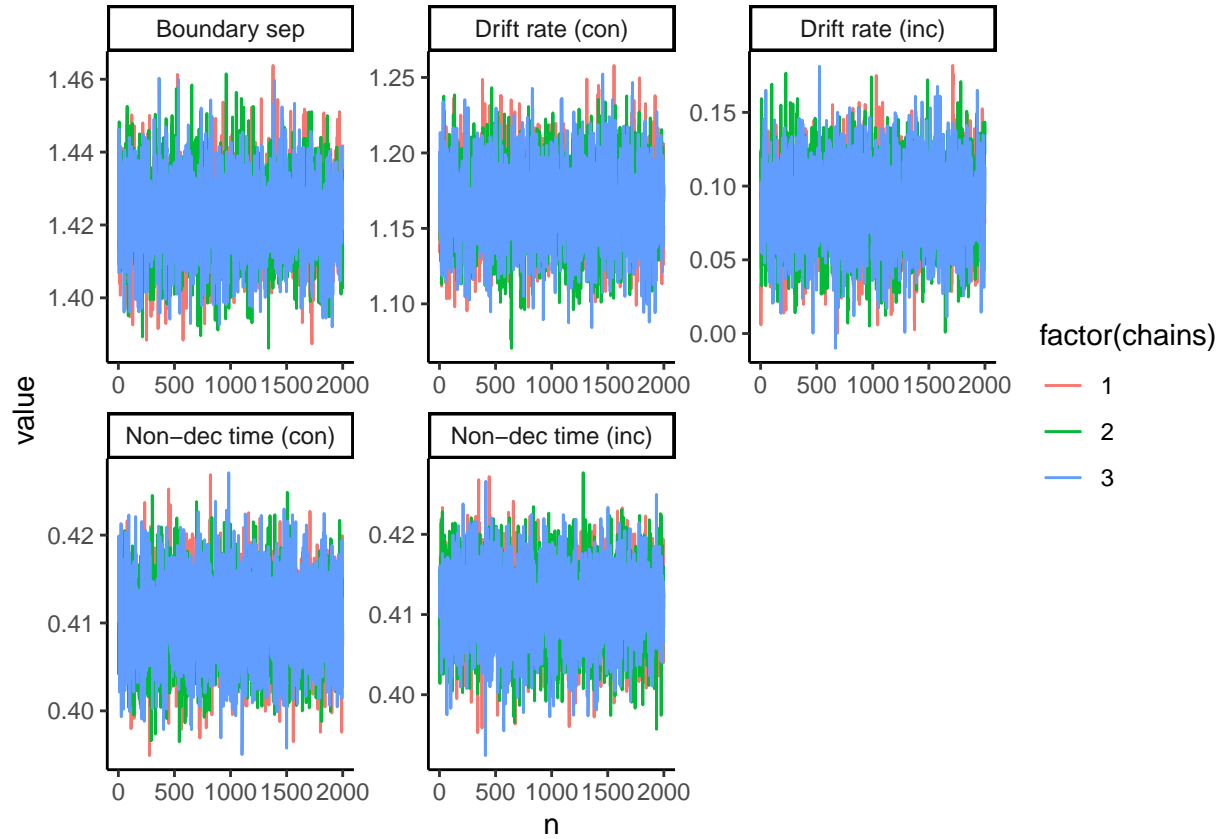
For each simulation, I present two Figures. The first figure shows model convergence (i.e., whether all Markov Chain Monte Carlo (MCMC) chains were able to converge on a stable solution). In the case of good model convergence, the plots should resemble a “thick, hairy caterpillar”. This is the case for all the simulations presented here. The second figure shows parameter recovery (i.e., how closely the estimated DDM parameters resemble the “true”, underlying DDM parameters). Because the data are simulated, we actually know the groundtruth DDM parameters. Higher correlations indicate better parameter recovery.

## Flanker

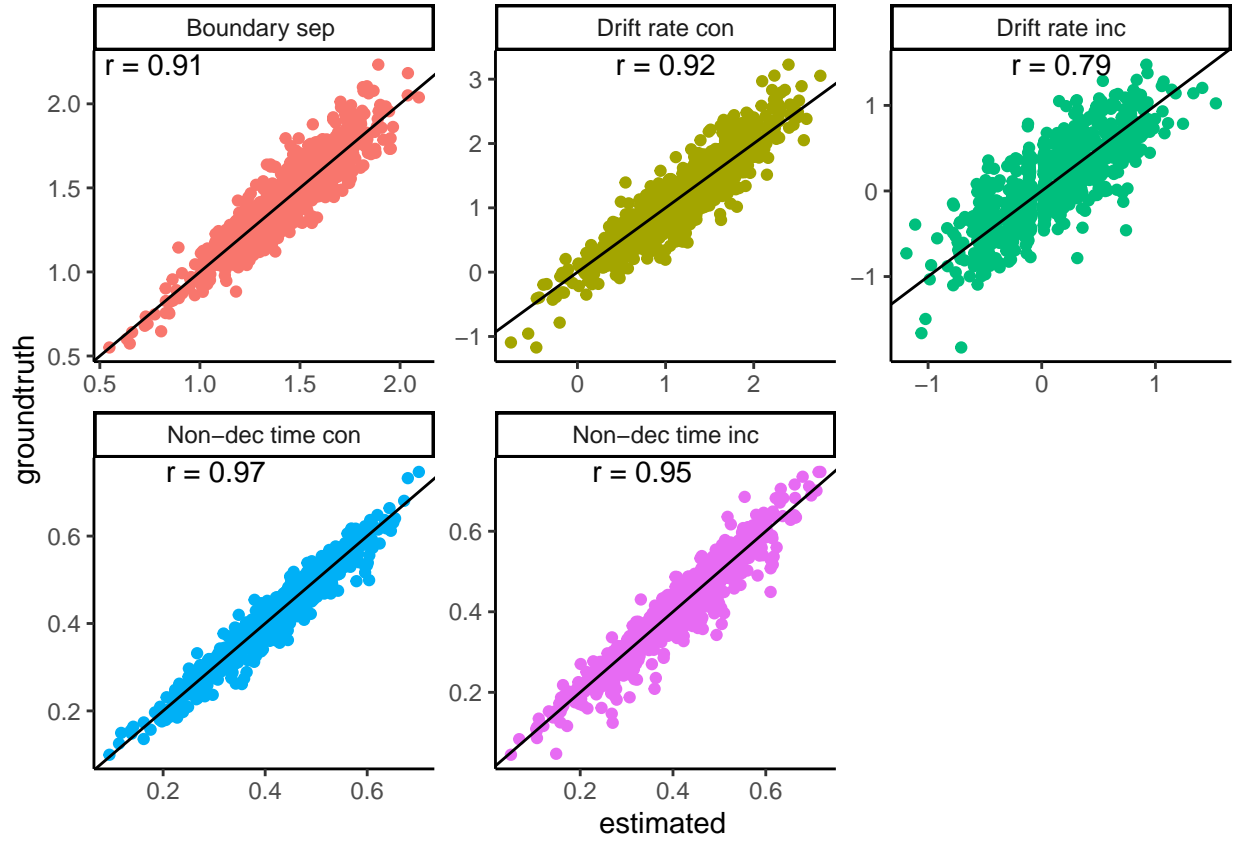
### Simulation 1: Constrain boundary separation across conditions

For simulation 1, I simulated 29 congruent and 16 incongruent trials. This corresponds with the number of trials in mixed blocks, averaged across blue and pink trials. Drift rate and non-decision time were estimated separately for congruent and incongruent condition, but boundary separation was fit across both conditions.

The HDDM model converged normally (plots should resemble a “thick, hairy caterpillar”):



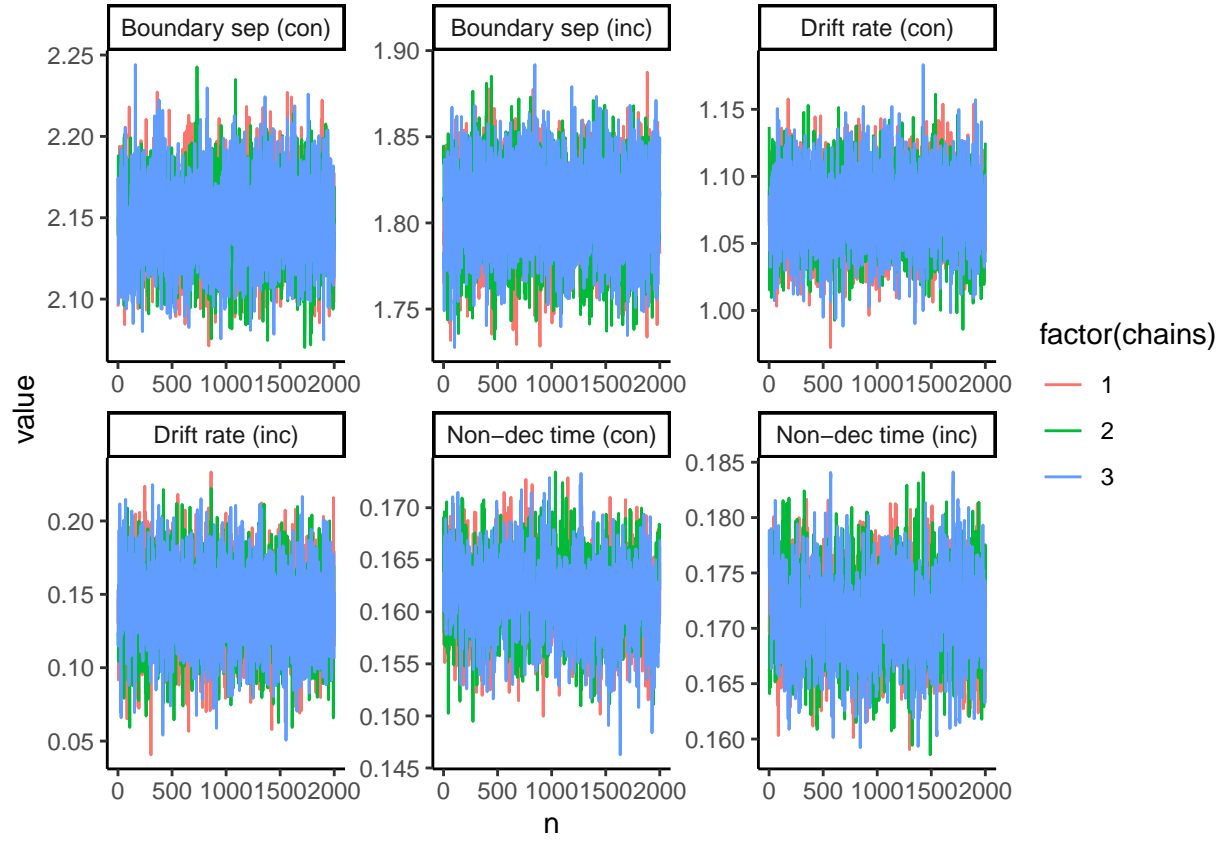
Parameter recovery was good for all parameters, showing that the DDM model is able to accurately recover the “true” simulated parameter values.



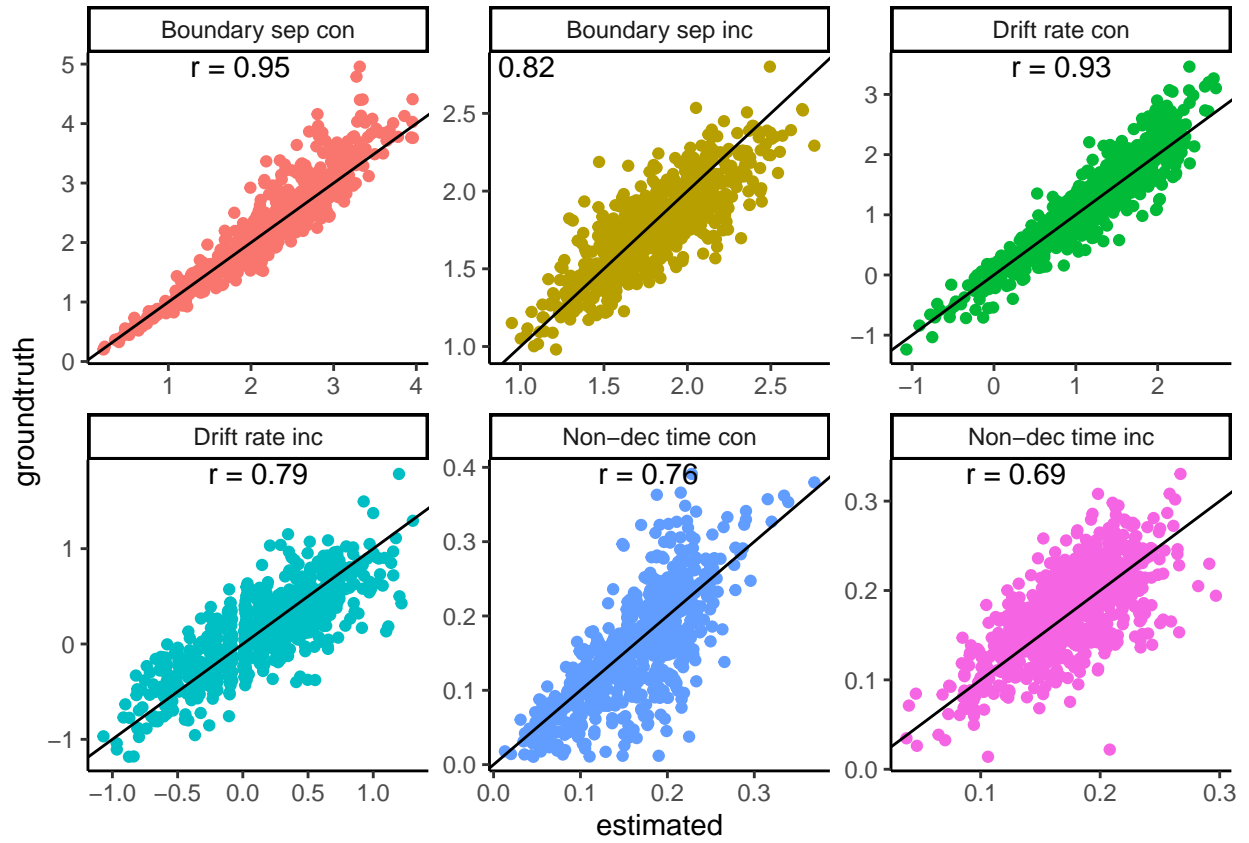
## Simulation 2: Boundary separation estimated freely for each condition

Like simulation 1, I simulated 29 congruent and 16 incongruent trials.

The HDDM model converged normally:

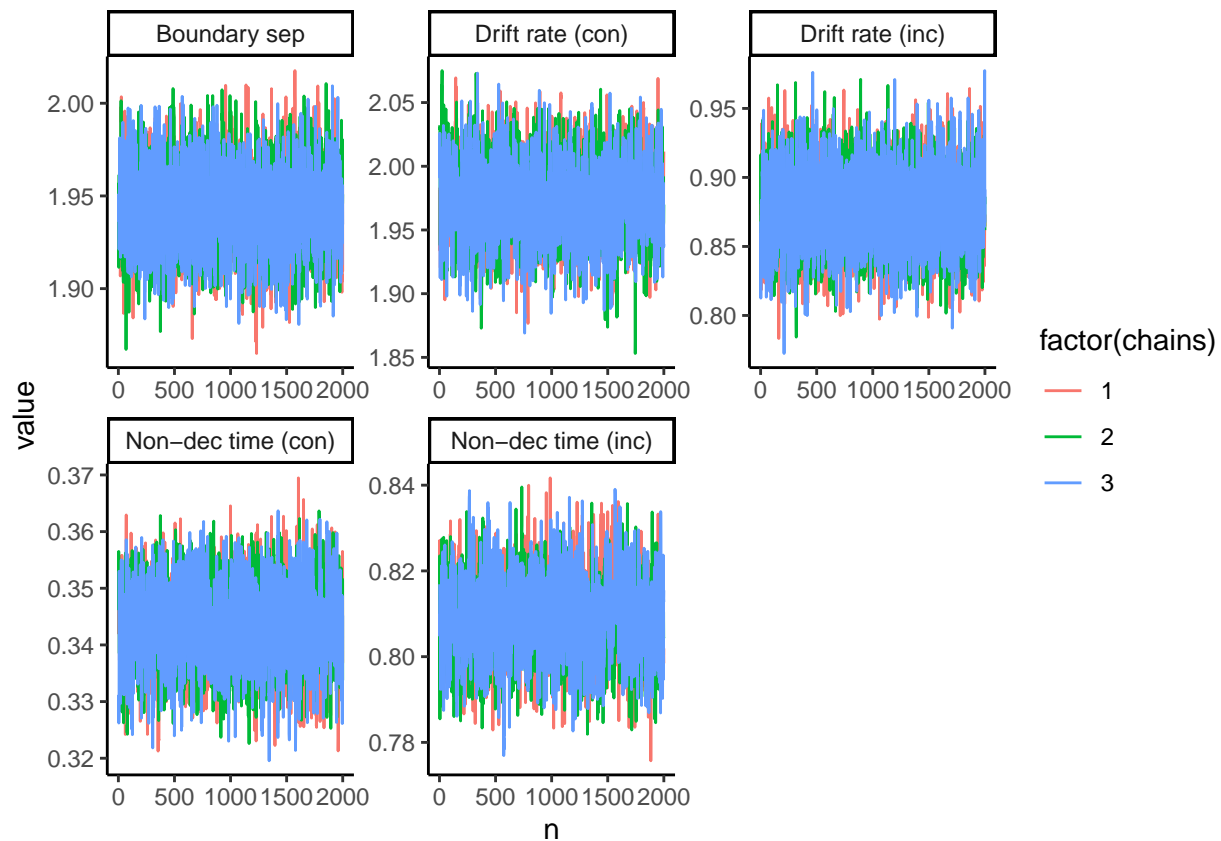


Drift rates and boundary separations were still recovered with high accuracy. Recovery of non-decision time was comparatively lower, but still reasonable.

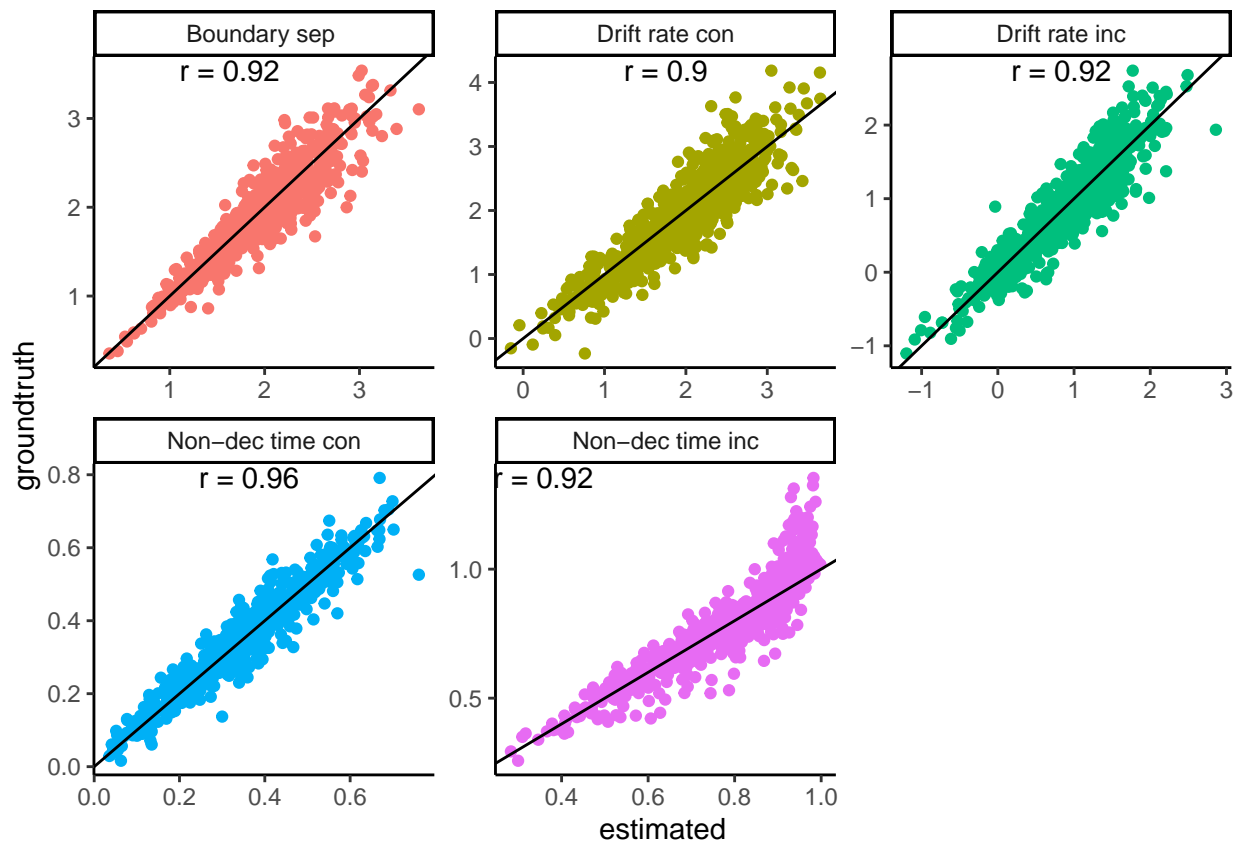


## MSIT

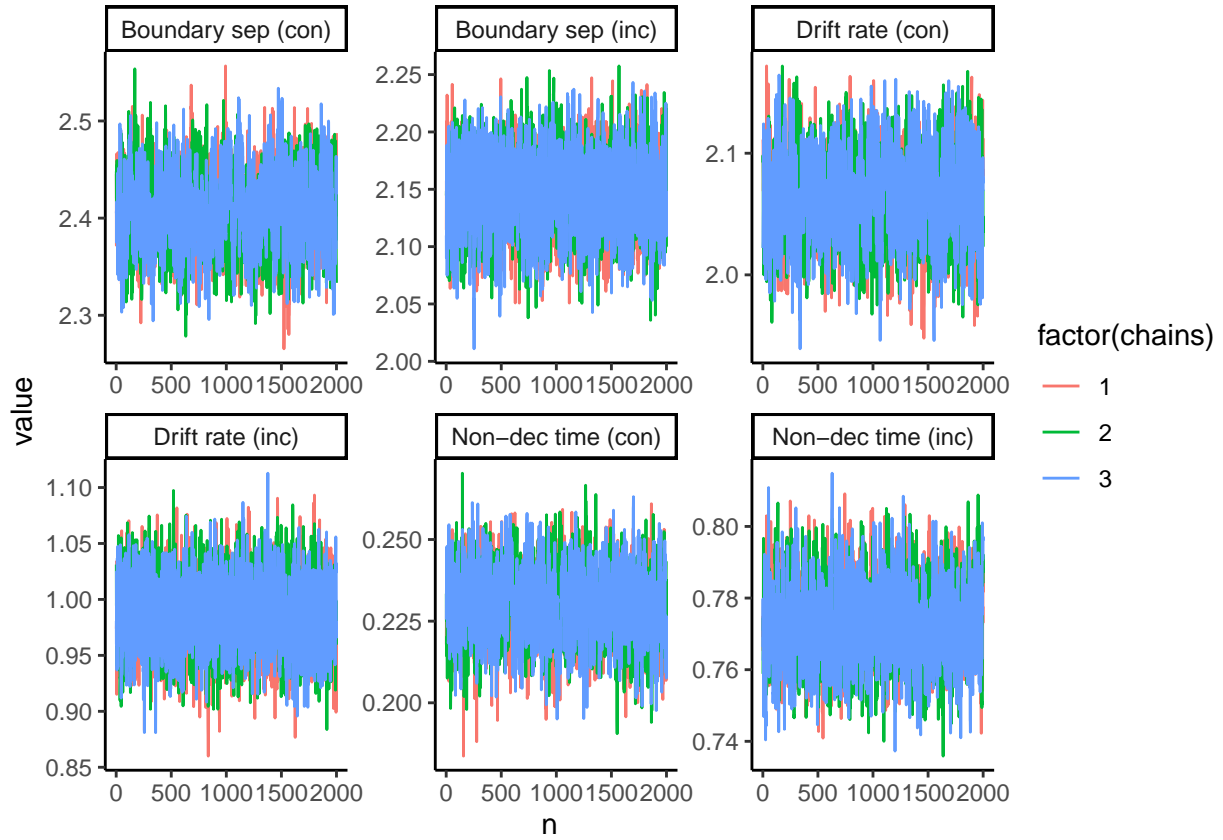
For the MSIT, I simulated 24 congruent and 24 incongruent trials. The model converged normally.



Just like with the Flanker, recovery was high for all parameters.



## Simulation 2



Recovery of all DDM parameters remained high when boundary separation was estimated separately for each condition.



