

# A Bayesian Hierarchical Diffusion Model Decomposition of Performance in Approach-Avoidance Tasks: Supplementary Material

Angelos-Miltiadis Kryptos<sup>1,2</sup>, Tom Beckers<sup>1,2,3</sup>, Merel Kindt<sup>1,2</sup>  
& Eric-Jan Wagenmakers<sup>1,2</sup>

<sup>1</sup>University of Amsterdam, <sup>2</sup>Amsterdam Brain and Cognition, <sup>3</sup> KU Leuven

## Abstract

In this document we provide (1) the R code for simulating Approach–Avoidance Reaction Time (AAT) data and for computing AAT indices according to the different strategies described in the main paper, (2) an example R code for conducting Bayesian repeated measures Analyses of Variance (ANOVA), (3) an example Python code for fitting a Bayesian hierarchical drift diffusion model (BHDDM) to AAT data, (4) details on how we fitted the BHDDM that we describe in the main paper to the first and the second data set, (5) the posterior kernel density estimates of the posterior distributions for every group parameter, (6) the Markov chains for every group parameter as well as convergence assessment statistics, and (7) the quality of fit of each model to the data.

### **R code used for simulating AAT data and computing AAT indices according to the different strategies**

Below is the R (R Core Team, 2013) code for generating data from an ex-Gaussian distribution and for computing the AAT index according to the different strategies described in the main paper. R can be freely obtained online (visit <http://cran.r-project.org/>).

```
# Load gamlss.dist package
library(gamlss.dist)

# Generate data
set.seed(1000)
apP <- rexGAUS(1000, mu=550, nu=200, sigma=35)
avP <- rexGAUS(1000, mu=580, nu=500, sigma=35)
apN <- rexGAUS(1000, mu=580, nu=200, sigma=35)
avN <- rexGAUS(1000, mu=550, nu=200, sigma=35)

# Strategy 1
con <- mean(c(apP, avN))
inc <- mean(c(apN, avP))
index1 <- con - inc

# Strategy 2 & 3
pos <- mean(avP) - mean(apP)
neg <- mean(avN) - mean(apN)
index2 <- pos - neg
```

### **Bayesian repeated measures ANOVA**

For the Bayesian repeated measures ANOVA, we used the BayesFactor package (Morey & Rouder, 2013; Rouder, Morey, Speckman, & Province, 2012; see also <http://bayesfactorpcl.r-forge.r-project.org>) available for the R programming language (R Core Team, 2013). Below we provide an example R script for running a Bayesian repeated measures ANOVA with AAT data.

```
# Load packages.
library(BayesFactor)
library(foreign)
library(reshape2)

# Read data
d <- read.spss("AART1final.sav", to.data.frame = T)

# Demonstration of the first 5 rows of the data.
# ID AppCSMin AvoidCSMin AppCSPlus AvoidCSPlus
# 1 598.20 811.85 651.35 668.90
# 2 843.60 1408.40 869.60 821.95
# 3 534.45 521.05 523.80 467.45
# 4 386.65 351.80 304.80 393.85
# 5 519.85 800.25 836.40 580.60
# 6 598.40 713.70 655.25 647.95

# Transform data to long format.
ID <- factor(rep(rownames(d), 4))
Stim <- rep(c("CSM", "CSP"), each = nrow(d)*2)
Resp <- rep(c("App", "Avoi", "App", "Avoi"), each = nrow(d))
d <- cbind(ID, melt(d), Stim, Resp)

# Perform the Bayesian repeated measures ANOVA.
bf1 <- anovaBF(value ~ Stim*Resp + ID, data = d, whichRandom="ID")

# Compute the main effects and the interaction term.
interactionTerm <- bf1[4]/bf1[3]
StimEff <- bf1[3]/bf1[2]
RespEff <- bf1[3]/bf1[1]
```

### Example Python code for fitting the BHDDM to the data

To fit all the described models to the data we used the hddm package (Wiecki, Sofer, & Frank, 2012, 2013; see also [http://ski.clps.brown.edu/hddm\\_docs](http://ski.clps.brown.edu/hddm_docs)) for the Python programming language (Lutz & Ascher, 2003). The hddm package allows users to estimate parameters of the diffusion model, in a hierarchical Bayesian manner, with minimal use of computer code. The hddm estimation routine is based on MCMC sampling, meaning that the posterior distributions are approximated by repeatedly drawing very many samples from those distributions.

Below is an example Python script for running a single MCMC of the Bayesian hierarchical diffusion model in which drift rate differs per condition. In order to run the model one needs to have Python as well as the hddm module installed.

```
# Import module
import hddm

# Load Data
data = hddm.load_csv('data.csv')

# Define model
m = hddm.HDDMRegressor(data, "v~C(coninc, Treatment('avoiCSM'))",
group_only_regressors=False)

# Find starting values
m.find_starting_values()

# Run model
m.sample(120000, burn = 20000, thin = 10, tune_throughout=False)

# Inspect results
m.print_stats()
```

### **BHDDM: Drift rate model for data set 1 and 2**

In the following section we describe how we fitted a BHDDM in which drift rate differed per condition to data set 1 and 2. This is the main model described in our paper.

#### **Model specification – Data set 1**

All observations for every participant were entered in our analysis. As also described in the main paper, for the present model we assumed that differences across the four conditions stemmed from differences in the drift rate parameter ( $v$ ). Regarding the remaining parameters, we assumed a symmetric starting point,  $z = a/2$ , and we did not allow the parameters  $a$  and  $Ter$  to vary across conditions. As described in the main paper, for the specific model we chose an arbitrary drift rate – here the one corresponding to the “Avoid CS<sup>-</sup>” condition – with the other three drift rates estimated as differences from the baseline. We point that any other condition could have served as a valid “baseline” condition, with results being essentially the same.

#### **Convergence assessment – Data set 1**

When using the MCMC method, it is important to ensure that the sampled values have converged from the random starting value to the posterior distribution. To assess convergence we ran three chains. Each single chain (i.e., a sequence of samples) consisted of 120,000 samples, and we discarded the first 20,000 samples as burn in. We then used a thinning factor of 10, so that the posterior distributions are based on a total of  $(120,000 - 20,000)/10 = 10,000$  samples. The markov chains were initialized using the default “find\_starting\_values()” function in the hddm package.

Convergence was assessed by computing the R-hat Gelman-Rubin statistic,

with values below 1.1 indicating successful converge, and by visually inspecting the chains in order to see whether they resembled “fat hairy caterpillars” (see Figure 1).

All R-hat values for the present model were below 1.1 indicating successful convergence for all chains. Furthermore, we visually inspected the chains and confirmed that they resembled “fat hairy caterpillars”. All chains for each group parameter can be found below (see Figure 2). See the y-axis of each plot to see the parameter that each plot refers to.

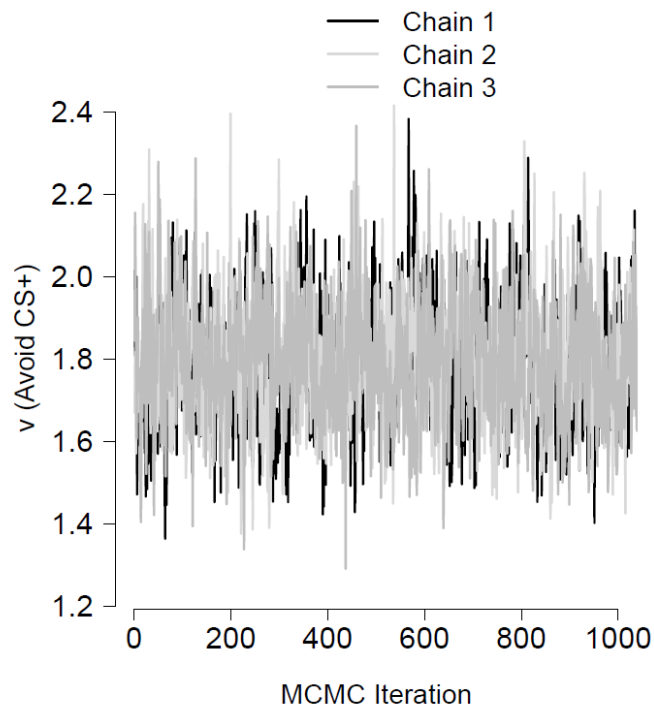
### **Kernel density estimates of the posterior distributions – Data set**

**1.** The kernel density estimate of the posterior distribution of each parameter can be found below (see Figure 3). Each density refers to a group parameter. See the y-axis of each plot for the parameter that each plot refers to.

**Quality of model fit– Data set 1** We assessed the quality of the model fit by plotting the real data against simulated data for the .1, .3, .5, and .9 RT quantiles for each stimulus by response condition.

For the simulated data, we simulated error-free model predictions by generating 8,000,000 data points for each condition for each individual, based on the posterior values (from the 1 to the 9001 point, with steps of 1000, of the first Markov samples) for each one of the individual parameters. We then computed the .1, .3, .5, and .9 RT quantiles for each individual and averaged those quantiles across the whole sample. Finally we compared the .1, .3, .5, and .9 RT quantiles generated by the model against those from the observed data. Error bars on the observed data were computed by a double bootstrap technique (Efron & Tibshirani, 1993); on each of 100 bootstrap runs, we randomly selected, with replacement, 32 individuals. We then sampled, again with replacement, the observed data for each participant in each of the four conditions. Then, for each bootstrap data set, we computed the RT quantiles; the standard deviation across 100 bootstrap RT quantiles was then used as an estimate for the standard error. As can be seen in Figure 4, the model fits the

observed data well.



*Figure 1.* Three Markov chains for the group parameter  $v(\text{Avoid CS}^+)$  for data set 1. The three chains resemble a “fat, hairy caterpillar” suggesting successful convergence.

#### **AAT index– Data set 1**

Please see the main text for the corresponding AAT index.

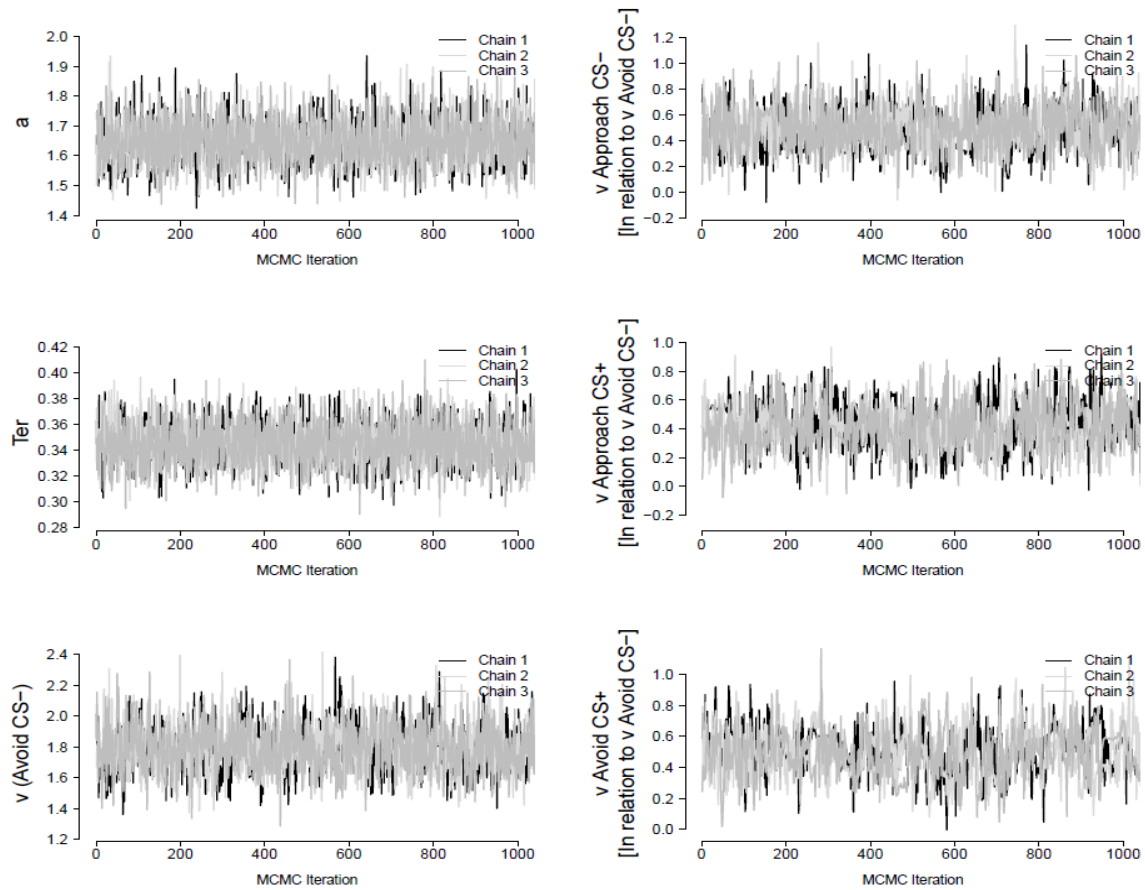
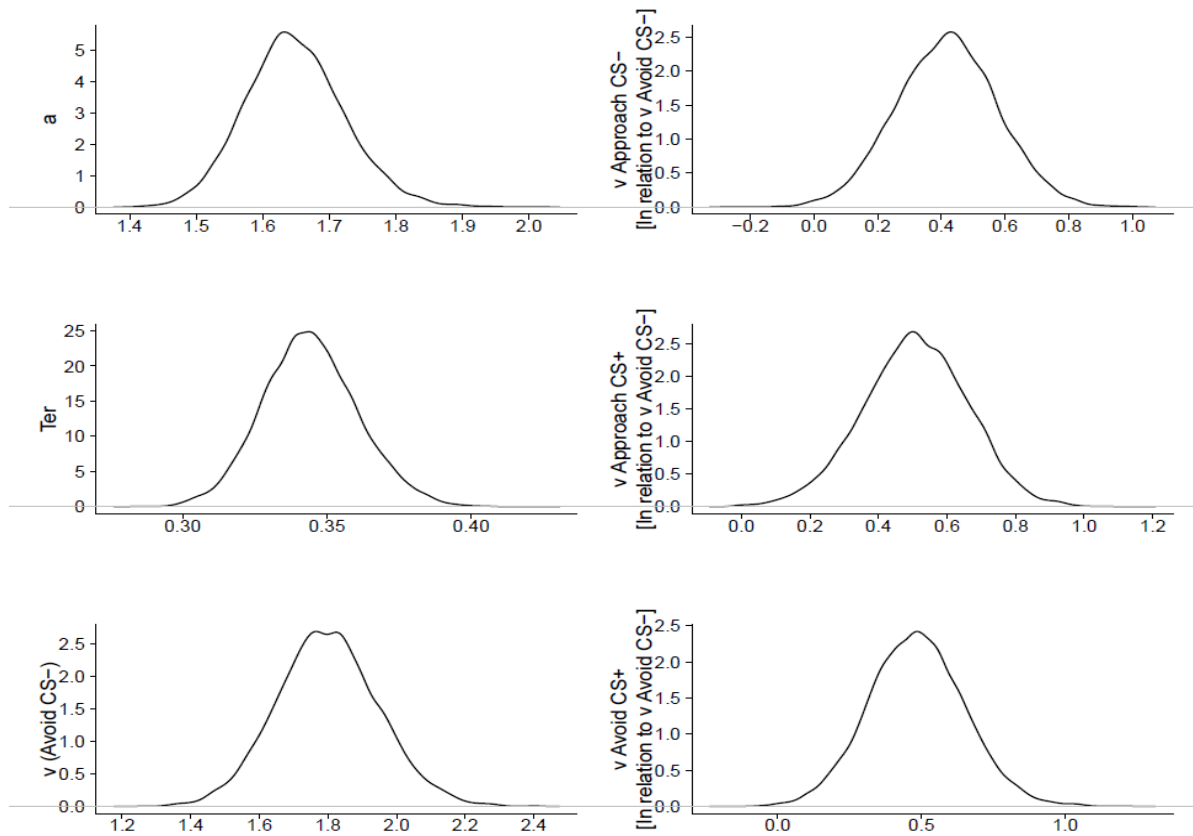
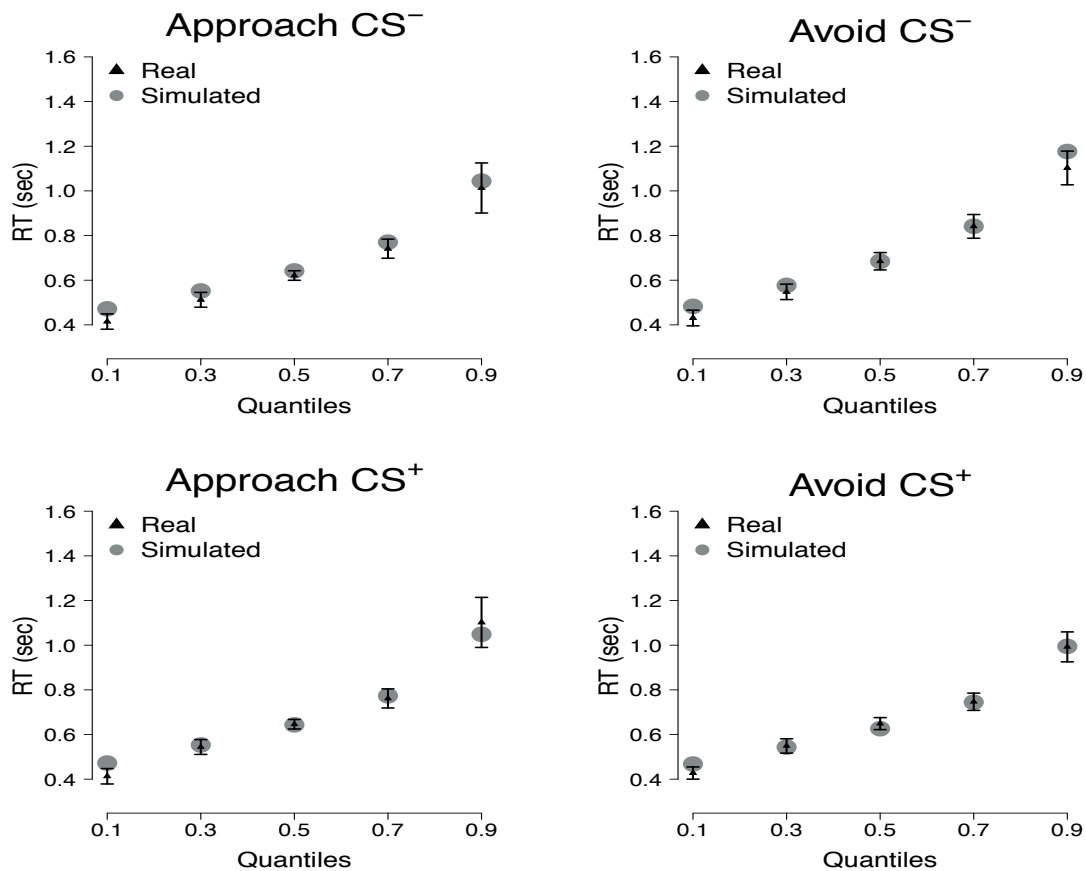


Figure 2. The Markov chains for each group parameter of the “drift rate model” for the first data set. See the y-axis of each plot to see the parameter that each plot refers to.





*Figure 3.* Kernel density estimates of the posterior distributions for every parameter of the “drift rate model” for the first data set. See the y-axis of each plot to see the parameter that each plot refers to.



*Figure 4.* Comparison of the simulated data generated by the “drift rate model” parameters and the observed data for each stimulus (CS<sup>+</sup> vs. CS<sup>-</sup>) by response (Approach vs. Avoidance) combination for the .1, .3, .5, and .9 RT quantiles. The model seems to capture the data quite well. Error bars represent bootstrapped standard error of means for each quantile.

**Model specification – Data set 2**

Model specifications were the same as for the first data set with the exception that the drift rate corresponding to the “Approach CS<sup>+</sup>” condition served as baseline. Any other condition could have served as a baseline, with results being essentially the same. Note that we fitted the model to the data of each group (i.e., AAA and ABA) separately.

**Convergence assessment – Data set 2**

MCMC convergence was assessed similarly to the “drift rate model” of data set 1.

All chains for every parameter had an R-hat statistic below 1.1 suggesting successful convergence. All chains were also visually inspected and they also resembled “fat hairy caterpillars”. The Markov chains for each group parameter and for each group can be found below (see Figure 5 for the AAA group and Figure 6 for the ABA group). See the y-axis of each plot for the parameter that the plot refers to.

**Kernel density estimates of the posterior distributions – Data set 2**

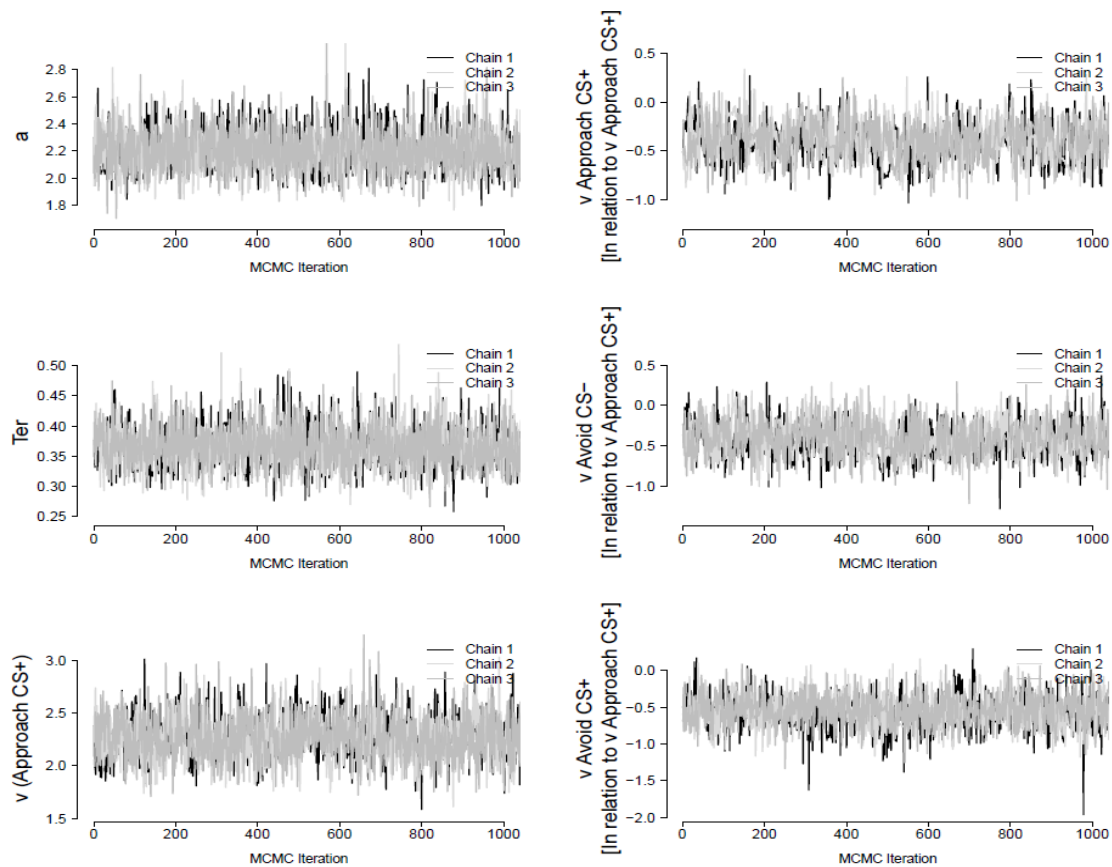
The kernel density estimates of the posterior distributions of each group parameter, separately for each group, can be found below (see Figure 7 for the AAA group and Figure 8 for the ABA group). See y-axis of each plot for the parameter that each plot refers to.

**Quality of model fit – Data set 2**

The quality of model fit was assessed by plotting the RT quantiles of the real against the simulated data, for each group separately, in a manner similar to the “drift rate model” of the first data set. Error bars were also computed as before. Figure 9 for group AAA and Figure 10 for the ABA group show that model predictions fit the data quite well.

**AAT index– Data set 2**

Please see the main text for the corresponding AAT index for each group of the second data set.



*Figure 5.* The Markov chains for every group parameter of the "drift rate model" for the AAA group of the second data set. See the y-axis of each plot for the parameter that each plot refers to.

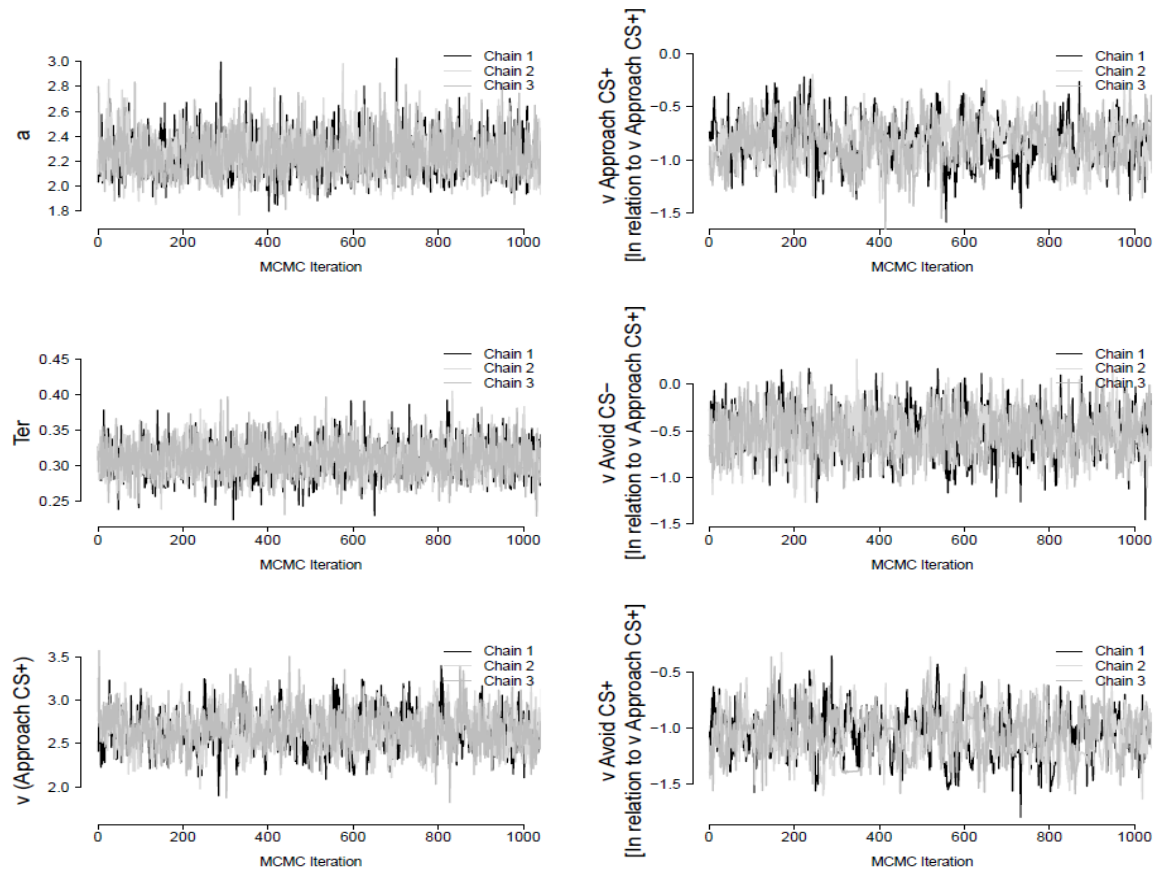
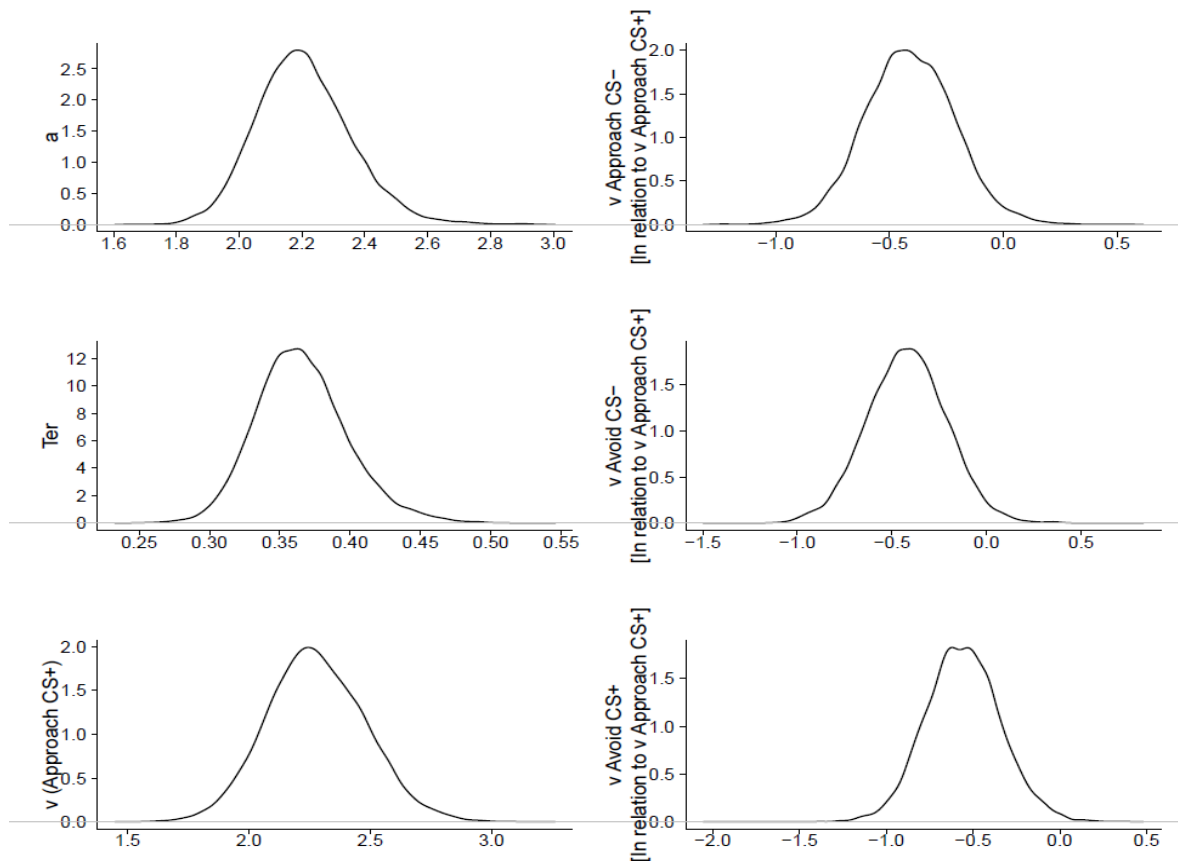
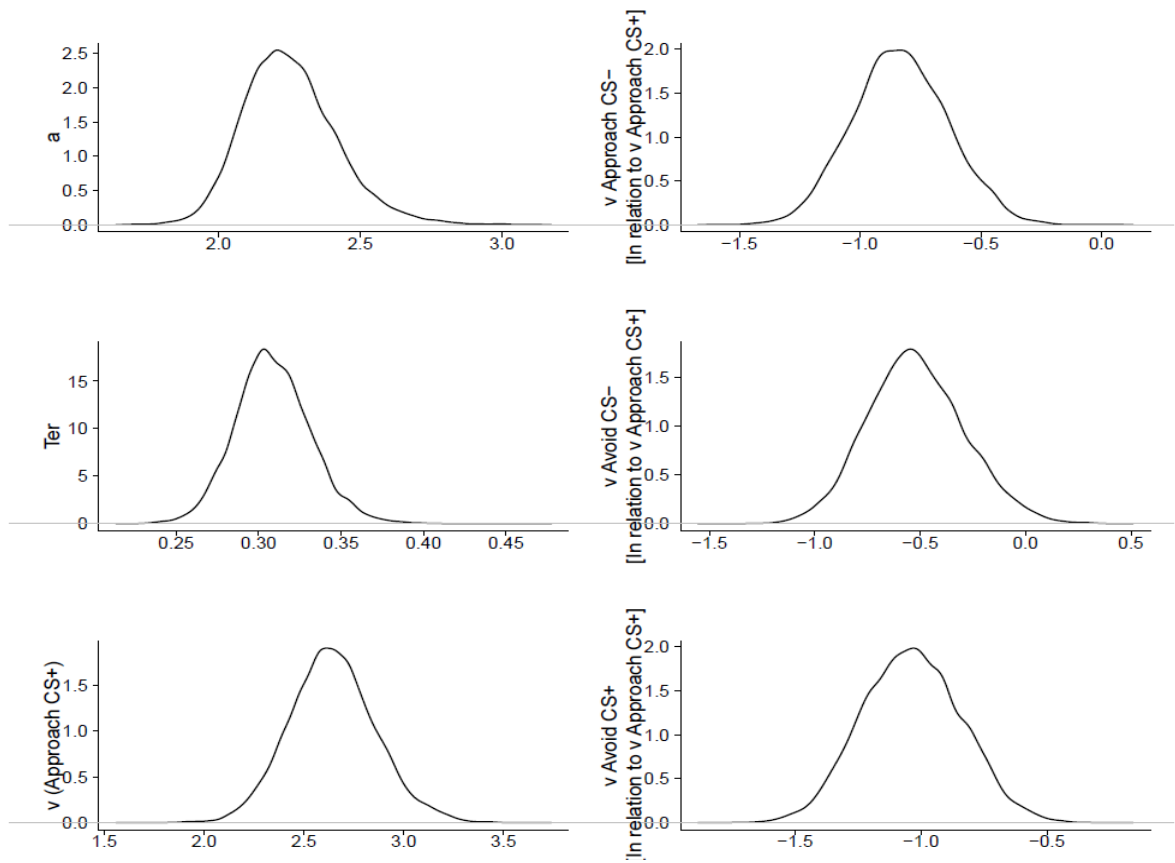


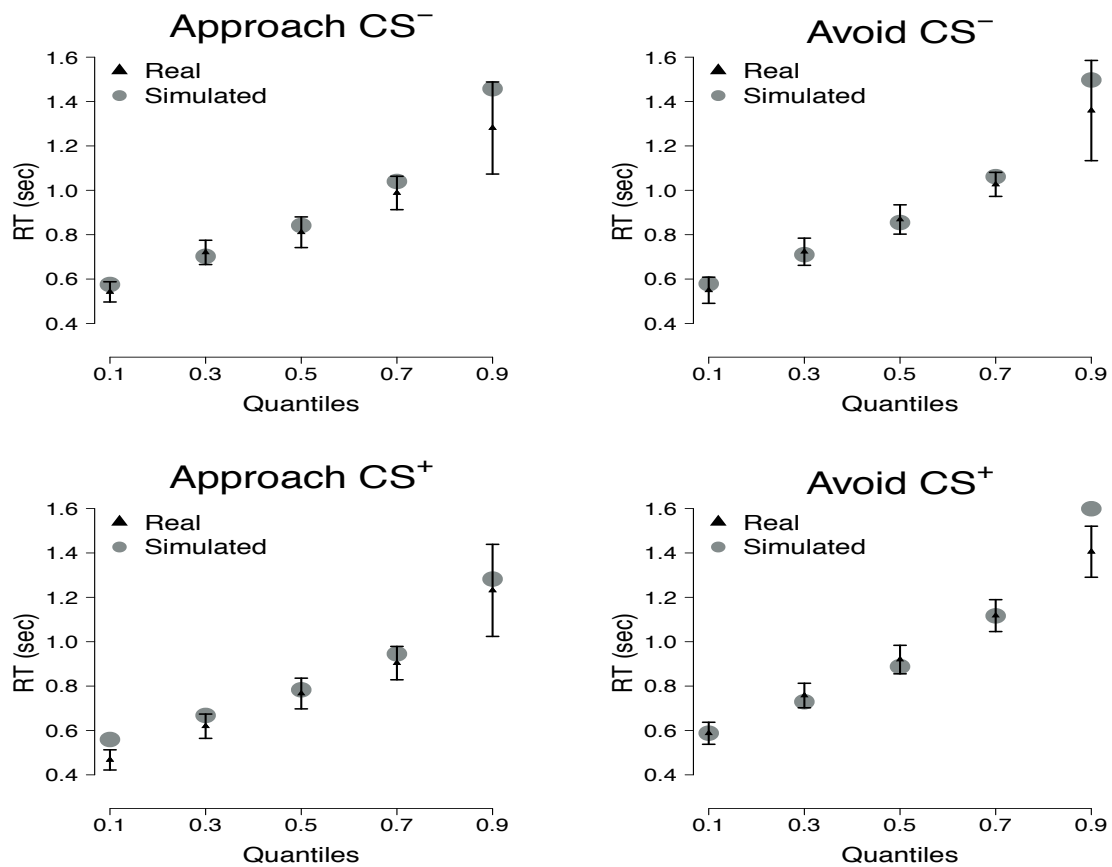
Figure 6. The Markov chains for every group parameter of the “drift rate model” for the ABA group of the second data set. See the y-axis of each plot for the parameter that each plot refers to.



*Figure 7.* Kernel density estimates of the posterior distributions for every group parameter of the “drift rate model” for the AAA group of the second data set. See the y-axis of each plot for the parameter that each plot refers to.



*Figure 8.* Kernel density estimates of the posterior distributions for every group parameter of the “drift rate model” for the ABA group of the second data set. See the y-axis of each plot for the parameter that each plot refers to.



*Figure 9.* Comparison of the simulated data generated by the “drift rate model” parameters and the observed data, group AAA of the second data set, for each stimulus (CS<sup>+</sup> vs. CS<sup>-</sup>) by response (Approach vs. Avoid) combination for the .1, .3, .5, and .9 RT quantiles. The model seems to capture the data quite well. Error bars represent bootstrapped standard error of means for each quantile.



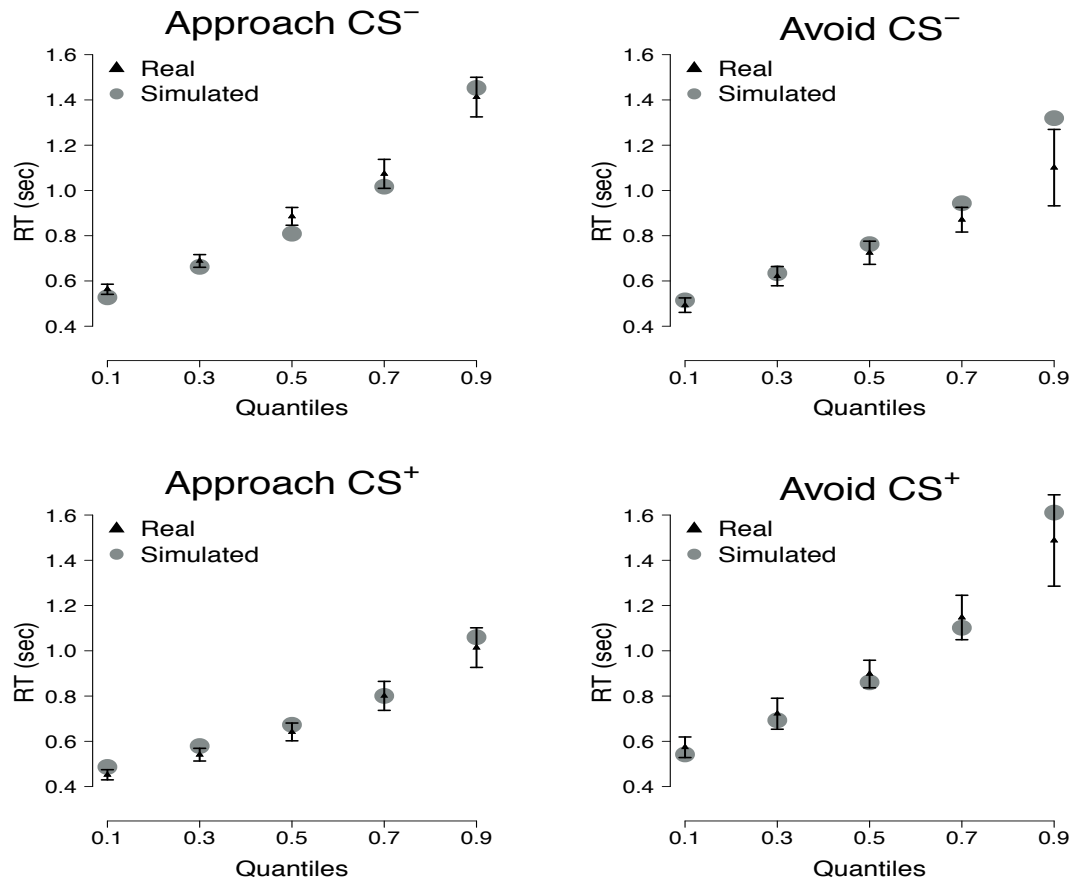


Figure 10. Comparison of the simulated data generated by the “drift rate model” parameters and the observed data, group ABA of the second data set, for each stimulus (CS<sup>+</sup> vs. CS<sup>-</sup>) by response (Approach vs. Avoid) combination for the .1, .3, .5, and .9 RT quantiles. The model seems to capture the data quite well. Error bars represent bootstrapped standard error of means for each quantile.

## References

- Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. New York: Chapman & Hall.
- Lutz, M., & Ascher, D. (2003). *Learning Python*. O'Reilly Media, Incorporated.
- Morey, R. D., & Rouder, J. N. (2013). Bayesfactor: Computation of bayes factors for common designs [Computer software manual]. Available from <http://CRAN.R-project.org/package=BayesFactor> (R package version 0.9.4).
- R Core Team. (2013). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Available from <http://www.R-project.org/>
- Rouder, J. N., Morey, R. D., Speckman, P. L., & Province, J. M. (2012). Default bayes factors for anova designs. *Journal of Mathematical Psychology*, 56, 356–374.
- Wiecki, T., Sofer, I., & Frank, M. J. (2012). Hierarchical Bayesian estimation of the drift diffusion model: Quantitative comparison with maximum likelihood. In *Program no. 494.13/ccc30. 2012 Neuroscience Meeting Planner. New Orleans, LA: Society for Neuroscience*.
- Wiecki, T., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the drift–diffusion model in Python. *Frontiers in Neuroinformatics*, 7:14.