BDA - Final Project

Zachary Himmelberger

Contents

Contact Information

Hello and thank you for reviewing this project! I know I've appreciated feedback throughout the course. I am hoping to publish these results, so I would be especially interested in your thoughts. I am especially unsure if I specified the model correctly.

If you want to contact me, my info is below. Cheers to a great semester!

Zach Himmelberger zach.himmelberger@maryvillecollege.edu

Load data and packages

The data has been pre-processed in another script. I will load in the data. Note that the script uses the **tidyverse** package, so I will not re-load it.

```
# set seed for the example to be reproducible
set.seed(210521)

# load data from file on my local machine
source("/Users/zach.himmelberger/OneDrive - Maryville College/Research/Maddie/organizing data.R")

# selecting a subset of the dataframe
project.data <- route.df[c("participant", "trial", "choicePoint", "saliency", "correct")]
```

We are also using the following packages.

```
library(ggplot2) # plotting
library(loo) # PSIS-LOO implementation
library(rstan) # for model diagnostics
library(brms) # to run the model
```

Problem Statement

People tend to use landmarks when learning to navigate in a previously unseen (i.e., novel) environment. Certain characteristics of landmarks may promote faster learning. In particular, landmarks that are more salient (e.g., perceptually contrasted from the background or conceptually distinct). We were interested in how the salience of landmarks affects how people learn to navigate in novel environments.

Participants ($n = 42$) were tasked with navigating a path through a virtual maze on a computer. The maze consisted of 15 decision points (i.e., junctions where the participant must choose a direction), which are called choice points in the. A landmark was located at ten of the decision points (thus, five had no landmark). Half of the landmarks were salient and half were non-salient. If a participant made an incorrect turn at a landmark, they would reach a dead end and need to continue down a different path. Participants were stopped from returning to an earlier part of the maze and only one error could be recorded at each decision point.

The experiment started with a learning trial, where arrows indicated the correct path through the maze. Participants then completed the task ten additional times without the arrows to aid them. Thus, we have 150 data points for each participant (15 decision points per trial X 10 trials).

It is helpful to visualize the dataset. Note that decision points are called **choicePoint** in the dataset. The names are interchangeable in the literature. Also note that the first trial is coded 0. **correct** refers to whether the participant went the correct way at that decision point.

```
str(project.data)
```

```
## 'data.frame':    6450 obs. of  5 variables:
##  $ participant: Factor w/ 42 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ trial      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ choicePoint: Factor w/ 15 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ saliency   : Factor w/ 3 levels "none","non-salient",..: 3 2 1 3 2 1 3 1 2 3 ...
##  $ correct    : num  1 1 1 1 1 0 1 1 1 1 ...
```

```
head(project.data, n = 20)
```

```
##    participant trial choicePoint    saliency correct
## 1            1     0           1     salient       1
## 2            1     0           2 non-salient       1
## 3            1     0           3        none       1
## 4            1     0           4     salient       1
## 5            1     0           5 non-salient       1
## 6            1     0           6        none       0
## 7            1     0           7     salient       1
## 8            1     0           8        none       1
## 9            1     0           9 non-salient       1
## 10           1     0          10     salient       1
## 11           1     0          11 non-salient       1
## 12           1     0          12        none       1
## 13           1     0          13     salient       1
## 14           1     0          14 non-salient       0
## 15           1     0          15        none       0
```

```
## 16          1     1          1     salient       1
## 17          1     1          2 non-salient       1
## 18          1     1          3        none       1
## 19          1     1          4     salient       1
## 20          1     1          5 non-salient       0
```

General Analytic Strategy

I will build four models to analyze the data. The first two are unconditional in that they do not include landmark salience as a predictor. The second two are conditional. All models are hierarchical. Models 1, 2, and 3 are nested within model 4.

In the unconditional means model (Model 1), I will ignore the effect of trial. This will help understand how much variation there is between individual participants and between each decision point. In the unconditional linear growth model (Model 2), I will investigate the linear growth that occurs across trials. In the conditional growth model (Model 3), I will examine the effect of landmark salience on learning across trials. Finally, in the interaction growth model (Model 4), I will allow landmark salience and trial to interact.

The models are given by the following mathematical notation. Note that the priors are not included in this notation.

Model 1: Unconditional Means

$$
\begin{aligned}
\text{correct}_i &\sim \text{Bernoulli}(p_i) \\
\text{logit}(p_i) &= \alpha_{j[i],k[i]} \\
\alpha_j &\sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{ for participant j} = 1, \ldots, \text{J} \\
\alpha_k &\sim N\left(\mu_{\alpha_k}, \sigma^2_{\alpha_k}\right), \text{ for choicePoint k} = 1, \ldots, \text{K}
\end{aligned}
$$

Model 2: Unconditional Linear Growth

$$
\begin{aligned}
\text{correct}_i &\sim \text{Bernoulli}(p_i) \\
\text{logit}(p_i) &= \alpha_{j[i],k[i]} + \beta_{1j[i],k[i]}(\text{trial}) \\
\begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} &\sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}} \end{pmatrix}\right), \text{ for participant j} = 1, \ldots, \text{J} \\
\begin{pmatrix} \alpha_k \\ \beta_{1k} \end{pmatrix} &\sim N\left(\begin{pmatrix} \mu_{\alpha_k} \\ \mu_{\beta_{1k}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_{1k}} \\ \rho_{\beta_{1k}\alpha_k} & \sigma^2_{\beta_{1k}} \end{pmatrix}\right), \text{ for choicePoint k} = 1, \ldots, \text{K}
\end{aligned}
$$

Model 3: Conditional Linear Growth

$$
\begin{aligned}
\text{correct}_i &\sim \text{Bernoulli}(p_i) \\
\text{logit}(p_i) &= \alpha_{j[i],k[i]} + \beta_{1j[i],k[i]}(\text{trial}) \\
\begin{pmatrix} \alpha_j \\ \beta_{1j} \\ \gamma_{1j} \\ \gamma_{2j} \end{pmatrix} &\sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \\ \mu_{\gamma_{1j}} \\ \mu_{\gamma_{2j}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} & \rho_{\alpha_j\gamma_{1j}} & \rho_{\alpha_j\gamma_{2j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}} & \rho_{\beta_{1j}\gamma_{1j}} & \rho_{\beta_{1j}\gamma_{2j}} \\ \rho_{\gamma_{1j}\alpha_j} & \rho_{\gamma_{1j}\beta_{1j}} & \sigma^2_{\gamma_{1j}} & \rho_{\gamma_{1j}\gamma_{2j}} \\ \rho_{\gamma_{2j}\alpha_j} & \rho_{\gamma_{2j}\beta_{1j}} & \rho_{\gamma_{2j}\gamma_{1j}} & \sigma^2_{\gamma_{2j}} \end{pmatrix}\right), \text{ for participant j} = 1, \ldots, \text{J} \\
\begin{pmatrix} \alpha_k \\ \beta_{1k} \end{pmatrix} &\sim N\left(\begin{pmatrix} \gamma^\alpha_0 + \gamma^\alpha_1(\text{saliency}_{\text{non-salient}}) + \gamma^\alpha_2(\text{saliency}_{\text{salient}}) \\ \mu_{\beta_{1k}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_{1k}} \\ \rho_{\beta_{1k}\alpha_k} & \sigma^2_{\beta_{1k}} \end{pmatrix}\right), \text{ for choicePoint k} = 1, \ldots, \text{K}
\end{aligned}
$$

Model 4: Interaction Growth Model

$$\text{correct}_i \sim \text{Bernoulli}(p_i)$$
$$\text{logit}(p_i) = \alpha_{j[i],k[i]} + \beta_{1j[i],k[i]}(\text{trial})$$

$$\begin{pmatrix} \alpha_j \\ \beta_{1j} \\ \gamma_{1j} \\ \gamma_{2j} \\ \gamma^{\beta_1}_{1j} \\ \gamma^{\beta_1}_{2j} \end{pmatrix} \sim N \left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \\ \mu_{\gamma_{1j}} \\ \mu_{\gamma_{2j}} \\ \mu_{\gamma^{\beta_1}_{1j}} \\ \mu_{\gamma^{\beta_1}_{2j}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} & \rho_{\alpha_j\gamma_{1j}} & \rho_{\alpha_j\gamma_{2j}} & \rho_{\alpha_j\gamma^{\beta_1}_{1j}} & \rho_{\alpha_j\gamma^{\beta_1}_{2j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}} & \rho_{\beta_{1j}\gamma_{1j}} & \rho_{\beta_{1j}\gamma_{2j}} & \rho_{\beta_{1j}\gamma^{\beta_1}_{1j}} & \rho_{\beta_{1j}\gamma^{\beta_1}_{2j}} \\ \rho_{\gamma_{1j}\alpha_j} & \rho_{\gamma_{1j}\beta_{1j}} & \sigma^2_{\gamma_{1j}} & \rho_{\gamma_{1j}\gamma_{2j}} & \rho_{\gamma_{1j}\gamma^{\beta_1}_{1j}} & \rho_{\gamma_{1j}\gamma^{\beta_1}_{2j}} \\ \rho_{\gamma_{2j}\alpha_j} & \rho_{\gamma_{2j}\beta_{1j}} & \rho_{\gamma_{2j}\gamma_{1j}} & \sigma^2_{\gamma_{2j}} & \rho_{\gamma_{2j}\gamma^{\beta_1}_{1j}} & \rho_{\gamma_{2j}\gamma^{\beta_1}_{2j}} \\ \rho_{\gamma^{\beta_1}_{1j}\alpha_j} & \rho_{\gamma^{\beta_1}_{1j}\beta_{1j}} & \rho_{\gamma^{\beta_1}_{1j}\gamma_{1j}} & \rho_{\gamma^{\beta_1}_{1j}\gamma_{2j}} & \sigma^2_{\gamma^{\beta_1}_{1j}} & \rho_{\gamma^{\beta_1}_{1j}\gamma^{\beta_1}_{2j}} \\ \rho_{\gamma^{\beta_1}_{2j}\alpha_j} & \rho_{\gamma^{\beta_1}_{2j}\beta_{1j}} & \rho_{\gamma^{\beta_1}_{2j}\gamma_{1j}} & \rho_{\gamma^{\beta_1}_{2j}\gamma_{2j}} & \rho_{\gamma^{\beta_1}_{2j}\gamma^{\beta_1}_{1j}} & \sigma^2_{\gamma^{\beta_1}_{2j}} \end{pmatrix} \right), \text{ for participant j} = 1, \ldots, \text{J}$$

$$\begin{pmatrix} \alpha_k \\ \beta_{1k} \end{pmatrix} \sim N \left(\begin{pmatrix} \gamma^{\alpha}_0 + \gamma^{\alpha}_1(\text{saliency}_{\text{non-salient}}) + \gamma^{\alpha}_2(\text{saliency}_{\text{salient}}) \\ \gamma^{\beta_1}_0 + \gamma^{\beta_1}_1(\text{saliency}_{\text{non-salient}}) + \gamma^{\beta_1}_2(\text{saliency}_{\text{salient}}) \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_{1k}} \\ \rho_{\beta_{1k}\alpha_k} & \sigma^2_{\beta_{1k}} \end{pmatrix} \right), \text{ for choicePoint k} = 1, \ldots, \text{K}$$

Choice of Priors and Prior Predictive Checks

In Model 4, we will have three types of effects. For all of them, I chose weakly informative priors that pull the effects toward zero, which can help reduce over-fitting. First, we have what are sometimes referred to as fixed effects (one for trial and one for each of the three landmark salience conditions). I chose a standard normal prior, $N(0, 1)$. Second, we have the standard deviation for what are sometimes referred to as random effects (these are constrained to be positive). I chose an exponential prior, Exp(1), based on the recommendation in McElreath (2020). Third, I have a correlation matrix that specifies the relationship between the random effects. I chose a Lewandowski-Kurowicka-Joe (LKJ) correlation distribution distribution, LKJ(2), also based on the recommendation in McElreath.

See the Appendix for a sensitivity analysis on the impact of these priors.

A prior predictive check is an important way to make sure that the priors give sensible results. I only conduct this with Model 4 because the other models are nested within it.

```
my_priors <- c(set_prior("normal(0,1)", class = "b"),
               set_prior("exponential(1)", class = "sd"),
               set_prior("lkj(2)", class = "cor"))


prior_predictive_model <- brm(correct ~ 1 + trial * saliency + (1 + trial | choicePoint) + (1 + trial *
                                prior = my_priors,
                                data = project.data,
                                family = "bernoulli",
                                chains = 4,
                                warmup = 1000,
                                iter = 3500,
                                sample_prior = "only")
```

I am going to focus on checking the model parameters. Because the data is binary, it is more difficult to evaluate the appropriateness of the model predictions.

```
print(prior_predictive_model)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + trial * saliency + (1 + trial | choicePoint) + (1 + trial * saliency | partic
##    Data: project.data (Number of observations: 6450)
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
##          total post-warmup samples = 10000
```

```
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)             1.01      1.00     0.03     3.72 1.00    13136
## sd(trial)                 1.00      0.98     0.03     3.60 1.00    12875
## cor(Intercept,trial)     -0.00      0.44    -0.81     0.80 1.00    16922
##                       Tail_ESS
## sd(Intercept)             5760
## sd(trial)                 5697
## cor(Intercept,trial)      6389
##
## ~participant (Number of levels: 42)
##                                                    Estimate Est.Error
## sd(Intercept)                                          1.00      1.00
## sd(trial)                                              0.99      1.02
## sd(saliencynonMsalient)                                1.01      1.01
## sd(saliencysalient)                                    1.01      1.03
## sd(trial:saliencynonMsalient)                          1.01      1.05
## sd(trial:saliencysalient)                              1.00      1.03
## cor(Intercept,trial)                                  -0.00      0.33
## cor(Intercept,saliencynonMsalient)                    -0.00      0.33
## cor(trial,saliencynonMsalient)                         0.00      0.34
## cor(Intercept,saliencysalient)                        -0.00      0.33
## cor(trial,saliencysalient)                             0.00      0.33
## cor(saliencynonMsalient,saliencysalient)               0.01      0.33
## cor(Intercept,trial:saliencynonMsalient)              -0.00      0.33
## cor(trial,trial:saliencynonMsalient)                   0.00      0.33
## cor(saliencynonMsalient,trial:saliencynonMsalient)    -0.00      0.33
## cor(saliencysalient,trial:saliencynonMsalient)        -0.00      0.34
## cor(Intercept,trial:saliencysalient)                   0.00      0.33
## cor(trial,trial:saliencysalient)                       0.00      0.34
## cor(saliencynonMsalient,trial:saliencysalient)        -0.00      0.33
## cor(saliencysalient,trial:saliencysalient)            -0.00      0.34
## cor(trial:saliencynonMsalient,trial:saliencysalient)  -0.00      0.34
##                                                    l-95% CI u-95% CI Rhat
## sd(Intercept)                                          0.03     3.68 1.00
## sd(trial)                                              0.02     3.76 1.00
## sd(saliencynonMsalient)                                0.03     3.72 1.00
## sd(saliencysalient)                                    0.02     3.79 1.00
## sd(trial:saliencynonMsalient)                          0.02     3.79 1.00
## sd(trial:saliencysalient)                              0.02     3.75 1.00
## cor(Intercept,trial)                                  -0.63     0.63 1.00
## cor(Intercept,saliencynonMsalient)                    -0.63     0.63 1.00
## cor(trial,saliencynonMsalient)                        -0.64     0.64 1.00
## cor(Intercept,saliencysalient)                        -0.62     0.63 1.00
## cor(trial,saliencysalient)                            -0.62     0.64 1.00
## cor(saliencynonMsalient,saliencysalient)              -0.63     0.64 1.00
## cor(Intercept,trial:saliencynonMsalient)              -0.64     0.63 1.00
## cor(trial,trial:saliencynonMsalient)                  -0.63     0.62 1.00
## cor(saliencynonMsalient,trial:saliencynonMsalient)    -0.64     0.63 1.00
## cor(saliencysalient,trial:saliencynonMsalient)        -0.63     0.63 1.00
## cor(Intercept,trial:saliencysalient)                  -0.62     0.64 1.00
## cor(trial,trial:saliencysalient)                      -0.64     0.63 1.00
```

```
## cor(saliencynonMsalient,trial:saliencysalient)          -0.62     0.63 1.00
## cor(saliencysalient,trial:saliencysalient)              -0.64     0.65 1.00
## cor(trial:saliencynonMsalient,trial:saliencysalient)    -0.63     0.63 1.00
##                                                      Bulk_ESS Tail_ESS
## sd(Intercept)                                           12264     5470
## sd(trial)                                               12323     5584
## sd(saliencynonMsalient)                                 11599     4892
## sd(saliencysalient)                                     12246     5508
## sd(trial:saliencynonMsalient)                           13484     4655
## sd(trial:saliencysalient)                               10991     5320
## cor(Intercept,trial)                                    19867     6807
## cor(Intercept,saliencynonMsalient)                      19094     7046
## cor(trial,saliencynonMsalient)                          14202     6863
## cor(Intercept,saliencysalient)                          19383     6599
## cor(trial,saliencysalient)                              13470     6816
## cor(saliencynonMsalient,saliencysalient)                 9079     6637
## cor(Intercept,trial:saliencynonMsalient)                18956     6807
## cor(trial,trial:saliencynonMsalient)                    11723     7524
## cor(saliencynonMsalient,trial:saliencynonMsalient)       9969     6868
## cor(saliencysalient,trial:saliencynonMsalient)           8119     7977
## cor(Intercept,trial:saliencysalient)                    17031     6851
## cor(trial,trial:saliencysalient)                        15453     6313
## cor(saliencynonMsalient,trial:saliencysalient)          10533     7220
## cor(saliencysalient,trial:saliencysalient)               7949     7536
## cor(trial:saliencynonMsalient,trial:saliencysalient)     7513     7759
##
## Population-Level Effects:
##                           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                     0.08      6.39   -12.32    12.68 1.00    15224
## trial                        -0.00      1.03    -2.04     2.01 1.00    20754
## saliencynonMsalient          -0.00      0.99    -1.96     1.95 1.00    22405
## saliencysalient               0.01      1.00    -1.95     2.02 1.00    17916
## trial:saliencynonMsalient    -0.00      0.98    -1.93     1.89 1.00    19489
## trial:saliencysalient         0.00      1.01    -2.03     2.03 1.00    19131
##                           Tail_ESS
## Intercept                     6322
## trial                         6704
## saliencynonMsalient           6923
## saliencysalient               6562
## trial:saliencynonMsalient     7213
## trial:saliencysalient         6684
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

When looking at the summary of the output, the model (before "seeing" the data) contains lots of uncertainty.

The group-level effects are split into two categories: standard deviations and correlations. The standard deviations are typically referred to as "random" effects. These are all about 1 and have a 95% credible interval from 0 to about 3.75. A value close to zero indicates little variability among decision points or participants. Values above one are very unlikely to occur in the data. The large number of observations will overwhelm that prior, so it is not much of an issue if the values are allowed to be high. Additionally, it is theoretically possible to have very large differences between people and, hence, random effects greater than 1. The correlations are all close to 0 and have a 95% credible interval between about -.65 and .65. I'm not

sure what to expect from the data, though my experience tells me that strong correlations (in this field of research) are about .5, so these values seem reasonable.

The population-level effects are typically referred to as fixed effects. These are all about zero. The intercept has significantly more variability, but all of the other coefficients have a 95% credible interval between about -2 and 2. These are relatively diffuse estimates and are reasonable as weakly informative priors. Overall, I think these priors are reasonable, though they contain no real substantive knowledge about the problem.

brms code

I had trouble creating the models myself in Stan. In particular, Model 3 and Model 4 proved too complex for me. I will keep working to learn Stan better!

For consistency, I decided to run all four models using the **brms** package (Bürkner, 2017), though the corresponding Stan code (also created using the **brms** package) is presented in the appendix.

For all models, I used four MCMC chains. Each chain consisted of 1000 warm-up and 2500 usable iterations. Thus, for each model, I had 10000 MCMC samples.

```
# set priors for each model
priors_one <- c(
  set_prior("normal(0,1)", class = "Intercept"),
  set_prior("exponential(1)", class = "sd")
  )

priors_two <- c(
  set_prior("normal(0,1)", class = "Intercept"),
  set_prior("normal(0,1)", class = "b"),
  set_prior("exponential(1)", class = "sd"),
  set_prior("lkj(2)", class = "cor")
  )

priors_three <- c(
  set_prior("normal(0,1)", class = "Intercept"),
  set_prior("normal(0,1)", class = "b"),
  set_prior("exponential(1)", class = "sd"),
  set_prior("lkj(2)", class = "cor")
  )

mod_one <- brm(correct ~ 1 + (1 | choicePoint) + (1 | participant),
               data = project.data,
               prior = priors_one,
               family = "bernoulli",
               chains = 4,
               warmup = 1000,
               iter = 3500)

mod_two <- brm(correct ~ 1 + trial + (1 + trial | choicePoint) + (1 + trial | participant),
               data = project.data,
               prior = priors_two,
               family = "bernoulli",
               chains = 4,
               warmup = 1000,
               iter = 3500)
```

```
mod_three <- brm(correct ~ 1 + trial + saliency +
                     (1 + trial | choicePoint) +
                     (1 + trial + saliency | participant),
                 data = project.data,
                 prior = priors_three,
                 family = "bernoulli",
                 chains = 4,
                 warmup = 1000,
                 iter = 3500)

mod_four <- brm(correct ~ 1 + trial * saliency +
                     (1 + trial | choicePoint) +
                     (1 + trial * saliency | participant),
                 data = project.data,
                 prior = priors_three,
                 family = "bernoulli",
                 chains = 4,
                 warmup = 1000,
                 iter = 3500)
```

Diagnostic Checks

To check that the MCMC chains converged and that the estimation procedure was stable, I looked at the $\widehat{R}$ values, effective sample size (ESS), and divergences. The code and full output are available in the appendix.

The trace plots looked like they converged, all of the $\widehat{R}$ values are close to 1.00 and the ESS for each parameter is high. Thus, there was no evidence that the chains failed to converge.

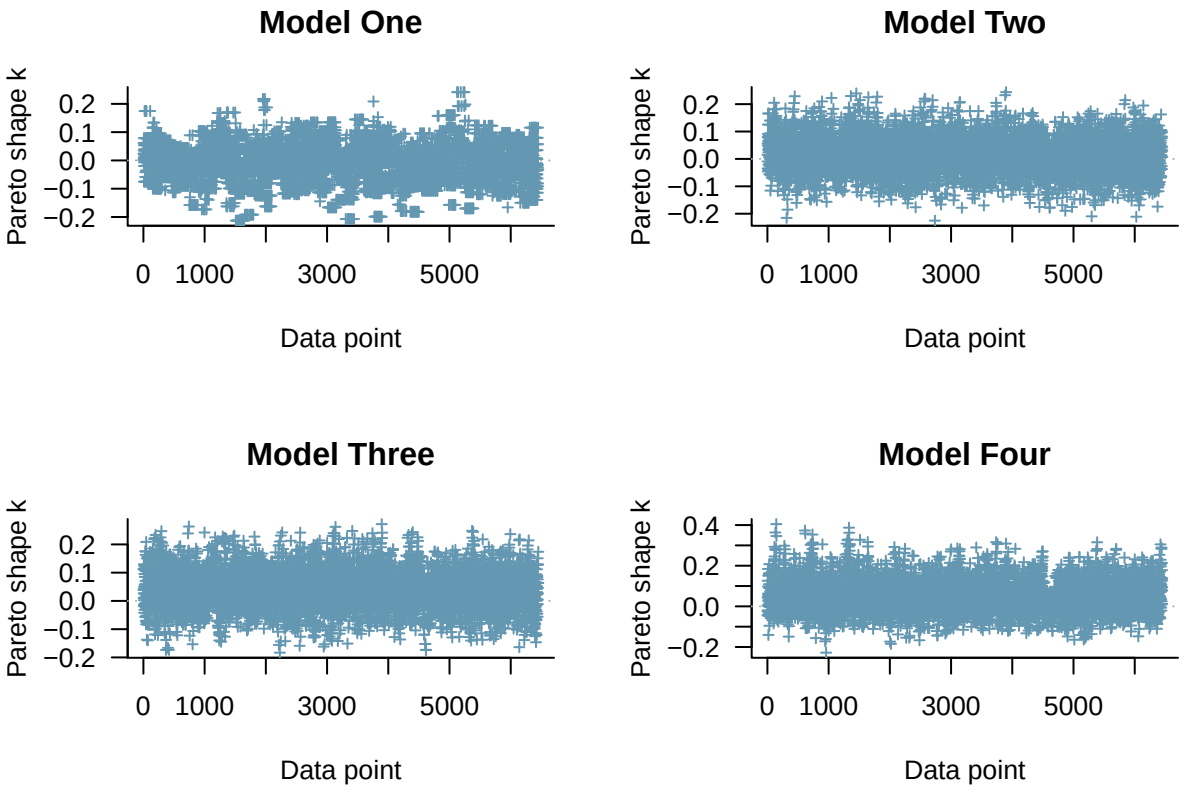
Model Comparison

I will use Pareto-smoothed importance sampling leave-one-out cross validation (LOO-CV) to compare the four models. Of primary interest is the comparison between Model 2 and Models 3 and 4, as the latter two contain an effect of landmark salience. If either Model 3 or Model 4 provide substantially more out-of-sample predictive power than Model 2, it will indicate that the presence and salience of landmarks aids in navigation. Another comparison of interest is between Model 3 and Model 4. This comparison gives insight into the interaction between landmark salience and trial.

```
loo_mod_one <- loo(mod_one)
loo_mod_two <- loo(mod_two)
loo_mod_three <- loo(mod_three)
loo_mod_four <- loo(mod_four)
```

Before comparing the models, we should do a diagnostic check on the LOO-CV estimation procedure.

```
par(mfrow=c(2, 2))
plot(loo_mod_one, main = "Model One")
plot(loo_mod_two, main = "Model Two")
plot(loo_mod_three, main = "Model Three")
plot(loo_mod_four, main = "Model Four")
```



There was only one $\hat{k}$ value above .5, which occurred in model four. This is evidence that the estimates are stable.

```
loo_compare(loo_mod_one, loo_mod_two, loo_mod_three, loo_mod_four)
```

```
##           elpd_diff se_diff
## mod_four     0.0       0.0
## mod_three   -4.8       3.4
## mod_two    -24.8       7.4
## mod_one   -283.4      23.0
```

As stated above, there were two main interests in the model comparison. First, we can see that Model 3 and Model 4 have substantially better out-of-sample predictive power than Model 1 and Model 2. This provides strong evidence that landmark salience has an effect on route learning. Second, the comparison between Model 3 and Model 4 is of interest. Although Model 4 had better predictive power, it also contains more parameters. We would expect the uncertainty around the out-of-sample predictive power to be approximately normally distributed. Thus, we must decide whether we can be confident in Model 4's superior accuracy. I will focus on Model 4 in this project, but would also report Model 3 in a publication, as it is not convincingly worse than Model 4.

Interpretation of Results

```
summary(mod_four)
```

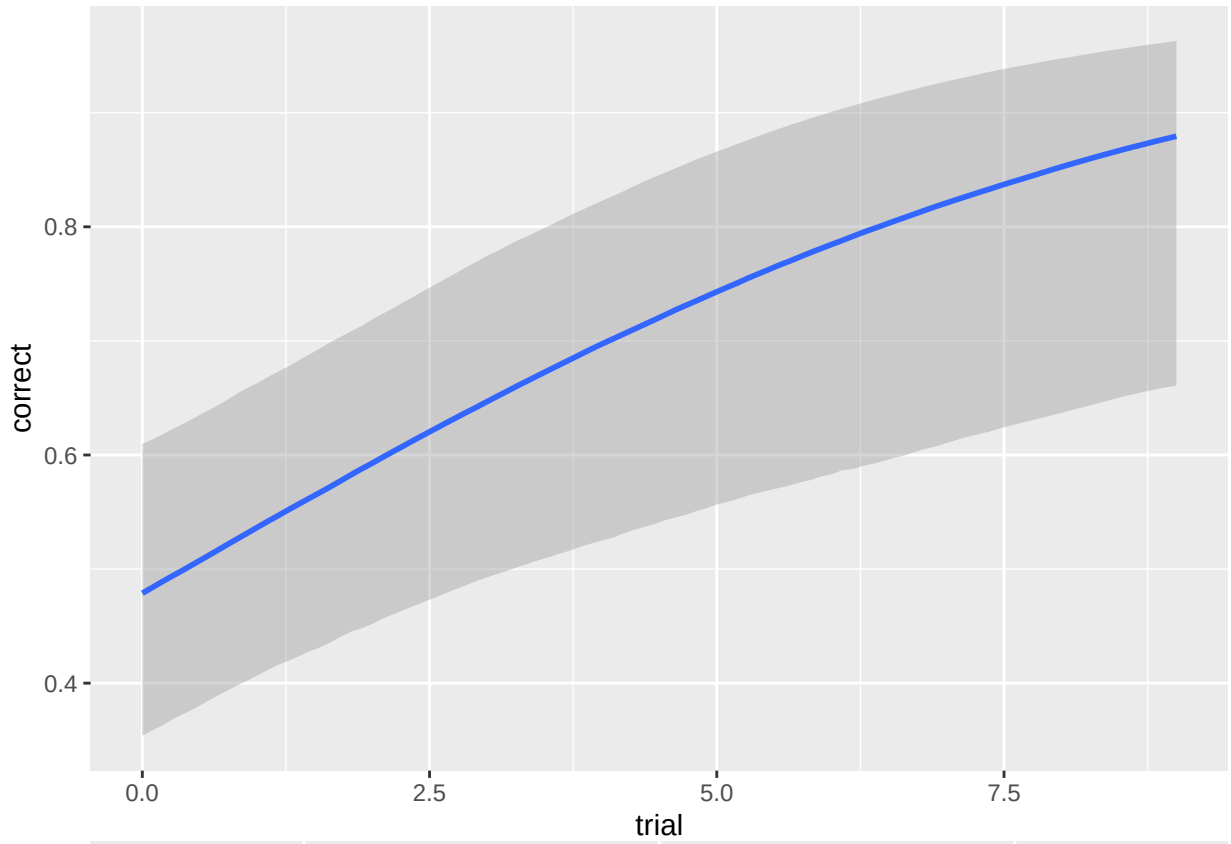
```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + trial * saliency + (1 + trial | choicePoint) + (1 + trial * saliency | partic
##    Data: project.data (Number of observations: 6450)
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
```

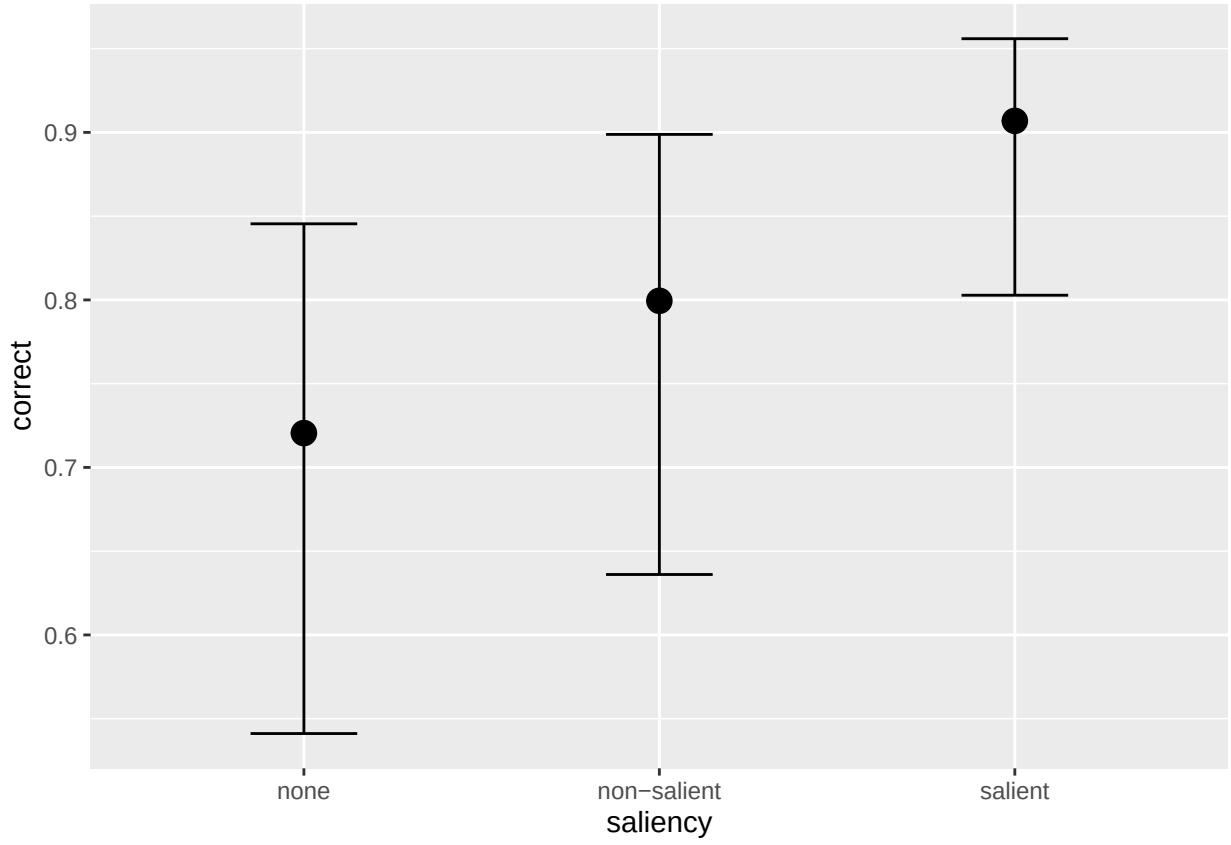
```
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)             0.56      0.15     0.34     0.91 1.00     5648
## sd(trial)                 0.14      0.04     0.08     0.24 1.00     4856
## cor(Intercept,trial)     -0.05      0.27    -0.56     0.48 1.00     4363
##                       Tail_ESS
## sd(Intercept)             6830
## sd(trial)                 5863
## cor(Intercept,trial)      5941
##
## ~participant (Number of levels: 42)
##                                                    Estimate Est.Error
## sd(Intercept)                                          0.45      0.10
## sd(trial)                                              0.09      0.02
## sd(saliencynonMsalient)                                0.26      0.14
## sd(saliencysalient)                                    0.52      0.18
## sd(trial:saliencynonMsalient)                          0.07      0.03
## sd(trial:saliencysalient)                              0.14      0.04
## cor(Intercept,trial)                                   0.24      0.24
## cor(Intercept,saliencynonMsalient)                    -0.01      0.30
## cor(trial,saliencynonMsalient)                         0.03      0.29
## cor(Intercept,saliencysalient)                         0.31      0.25
## cor(trial,saliencysalient)                             0.29      0.25
## cor(saliencynonMsalient,saliencysalient)               0.27      0.31
## cor(Intercept,trial:saliencynonMsalient)               0.17      0.28
## cor(trial,trial:saliencynonMsalient)                   0.04      0.28
## cor(saliencynonMsalient,trial:saliencynonMsalient)     0.06      0.31
## cor(saliencysalient,trial:saliencynonMsalient)         0.32      0.28
## cor(Intercept,trial:saliencysalient)                  -0.08      0.27
## cor(trial,trial:saliencysalient)                      -0.12      0.26
## cor(saliencynonMsalient,trial:saliencysalient)         0.31      0.30
## cor(saliencysalient,trial:saliencysalient)             0.03      0.28
## cor(trial:saliencynonMsalient,trial:saliencysalient)   0.28      0.29
##                                                    l-95% CI u-95% CI Rhat
## sd(Intercept)                                          0.27     0.67 1.00
## sd(trial)                                              0.05     0.14 1.00
## sd(saliencynonMsalient)                                0.02     0.54 1.00
## sd(saliencysalient)                                    0.17     0.88 1.00
## sd(trial:saliencynonMsalient)                          0.01     0.14 1.00
## sd(trial:saliencysalient)                              0.05     0.23 1.00
## cor(Intercept,trial)                                  -0.23     0.70 1.00
## cor(Intercept,saliencynonMsalient)                    -0.56     0.58 1.00
## cor(trial,saliencynonMsalient)                        -0.55     0.58 1.00
## cor(Intercept,saliencysalient)                        -0.21     0.75 1.00
## cor(trial,saliencysalient)                            -0.25     0.72 1.00
## cor(saliencynonMsalient,saliencysalient)              -0.43     0.78 1.00
## cor(Intercept,trial:saliencynonMsalient)              -0.42     0.66 1.00
## cor(trial,trial:saliencynonMsalient)                  -0.49     0.59 1.00
## cor(saliencynonMsalient,trial:saliencynonMsalient)    -0.55     0.65 1.00
## cor(saliencysalient,trial:saliencynonMsalient)        -0.29     0.78 1.00
## cor(Intercept,trial:saliencysalient)                  -0.60     0.44 1.00
```

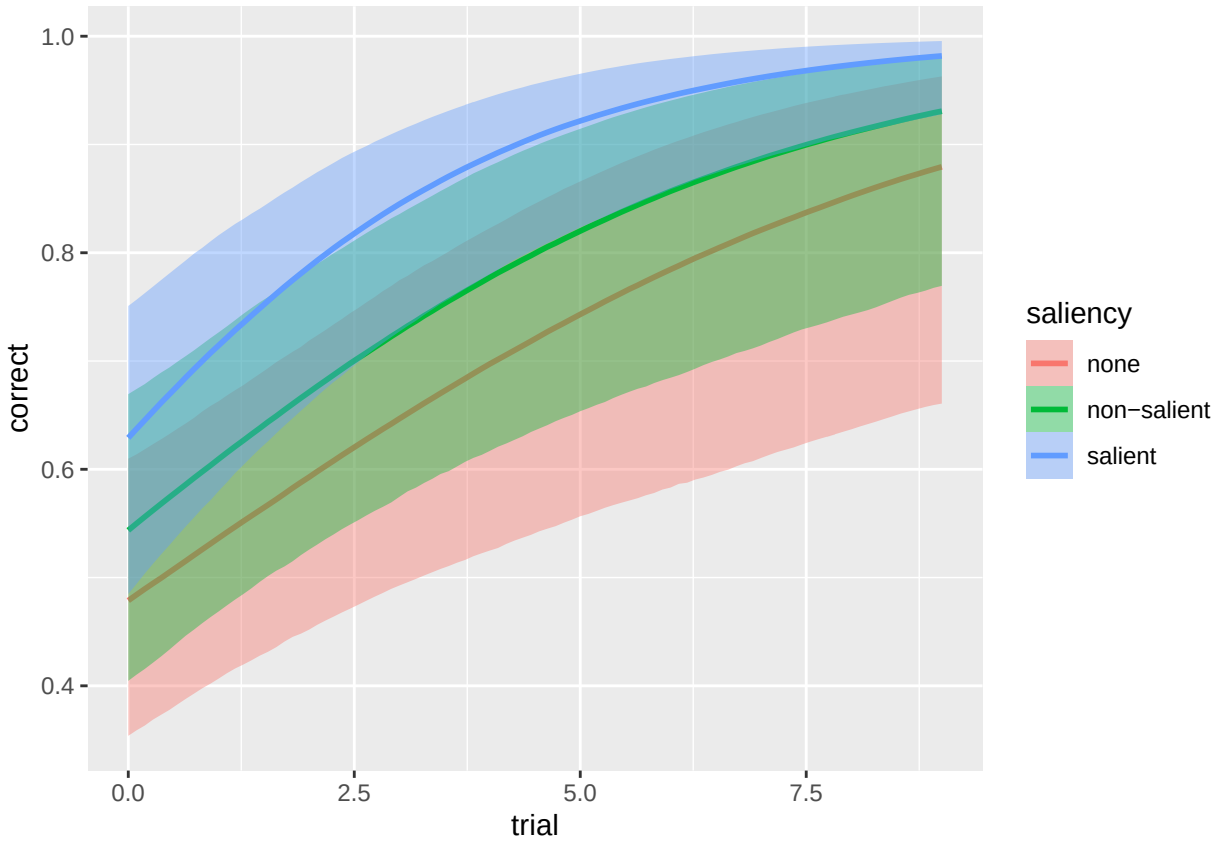
```
## cor(trial,trial:saliencysalient)                          -0.58     0.41 1.00
## cor(saliencynonMsalient,trial:saliencysalient)            -0.36     0.79 1.00
## cor(saliencysalient,trial:saliencysalient)                -0.49     0.60 1.00
## cor(trial:saliencynonMsalient,trial:saliencysalient)      -0.37     0.76 1.00
##                                                       Bulk_ESS Tail_ESS
## sd(Intercept)                                             6110     6544
## sd(trial)                                                 3718     5408
## sd(saliencynonMsalient)                                   1460     2528
## sd(saliencysalient)                                       3001     3261
## sd(trial:saliencynonMsalient)                             1670     2584
## sd(trial:saliencysalient)                                 3100     3582
## cor(Intercept,trial)                                      4515     6737
## cor(Intercept,saliencynonMsalient)                       10172     7807
## cor(trial,saliencynonMsalient)                            5990     7492
## cor(Intercept,saliencysalient)                            5183     6798
## cor(trial,saliencysalient)                                5388     6145
## cor(saliencynonMsalient,saliencysalient)                  2226     3858
## cor(Intercept,trial:saliencynonMsalient)                  5742     6430
## cor(trial,trial:saliencynonMsalient)                      6743     7206
## cor(saliencynonMsalient,trial:saliencynonMsalient)        4612     6467
## cor(saliencysalient,trial:saliencynonMsalient)            4970     5913
## cor(Intercept,trial:saliencysalient)                      3408     6885
## cor(trial,trial:saliencysalient)                          6280     6726
## cor(saliencynonMsalient,trial:saliencysalient)            2129     3110
## cor(saliencysalient,trial:saliencysalient)                5743     6712
## cor(trial:saliencynonMsalient,trial:saliencysalient)      3136     5839
##
## Population-Level Effects:
##                           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                    -0.08      0.27    -0.60     0.45 1.00     4313
## trial                         0.23      0.07     0.09     0.37 1.00     3513
## saliencynonMsalient           0.25      0.36    -0.48     0.94 1.00     4551
## saliencysalient               0.61      0.37    -0.13     1.31 1.00     5121
## trial:saliencynonMsalient     0.04      0.10    -0.16     0.24 1.00     3942
## trial:saliencysalient         0.16      0.10    -0.04     0.36 1.00     4433
##                           Tail_ESS
## Intercept                     6227
## trial                         4466
## saliencynonMsalient           6297
## saliencysalient               6429
## trial:saliencynonMsalient     5356
## trial:saliencysalient         4942
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
posterior_predictions_mod_four <- posterior_predict(mod_four)

plot(conditional_effects(mod_four), ask = FALSE)
```


correct
0.8
0.6
0.4
0.0
2.5
5.0
7.5
trial


correct
0.9
0.8
0.7
0.6
none
non−salient
salient
saliency



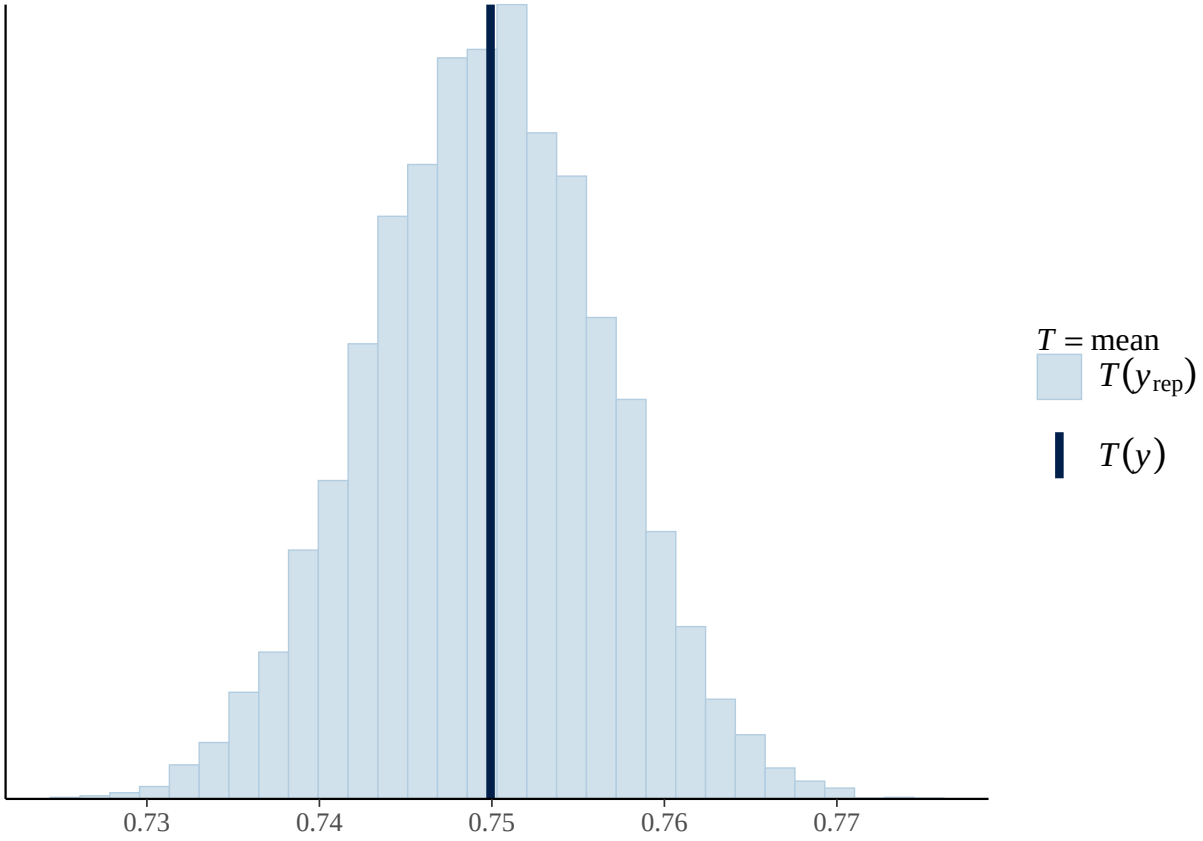
Because there are many parameters, I will focus on the most important results. The effect of trial was positive (even after accounting for the interaction terms). This indicates that, as expected, participants were learning the maze across repeated exposures to the environment. Landmarks that were salient tended to promote successful navigation. However, as participants continued to gain experience in the environment, the salience of landmarks mattered less. Hence the interaction effect.

Finally, as a posterior predictive check, we can examine a histogram of the predicted values.

```
pp_check(mod_four, type = "stat", stat = "mean")
```

```
## Using all posterior samples for ppc type 'stat' by default.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The model appears to be well fit to the data. Posterior predictions are all reasonable.

Appendix

Sensitivity Analysis

To understand the sensitivity of my choice of prior, I conducted the analysis again but with the non-informative priors that **brms** uses by default. Specifically, the default priors are a Student's $t(3, 0, 2.5)$ on the intercept and random effects; an LKJ(1) correlation distribution on the correlated effects, and an improper flat prior on the fixed effects.

```
flat_prior_mod_four <- brm(correct ~ 1 + trial * saliency +
                     (1 + trial | choicePoint) +
                     (1 + trial * saliency | participant),
                 data = project.data,
                 family = "bernoulli",
                 chains = 4,
                 warmup = 1000,
                 iter = 3500)

summary(flat_prior_mod_four)
```

As can be seen from the summary above, the results are very similar to the results obtained with weakly informative priors. There are two minors differences. First, almost all of the parameter estimates are slightly larger in the model that uses the default, flat priors. This makes intuitive sense. My choice of priors regularized the estimates (i.e., brought them toward zero). This should reduce over-fitting. Second, the estimates using the flat priors contained more uncertainty. The posterior intervals tended to be slightly larger. This is likely occurring because my priors have less density in the regions far from zero. This too should help

with over-fitting. Overall, I think my choice of priors proved to be smart in that they regularize the estimates a bit, though the overall effect is very small.

Stan code (created by brms)

Model One

```
make_stancode(correct ~ 1 + (1 | choicePoint) + (1 | participant),
              data = project.data,
              prior = priors_one,
              family = "bernoulli",
              chains = 4,
              warmup = 1000,
              iter = 3500)
```

```
## // generated with brms 2.13.0
## functions {
## }
## data {
##   int<lower=1> N;  // number of observations
##   int Y[N];  // response variable
##   // data for group-level effects of ID 1
##   int<lower=1> N_1;  // number of grouping levels
##   int<lower=1> M_1;  // number of coefficients per level
##   int<lower=1> J_1[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_1_1;
##   // data for group-level effects of ID 2
##   int<lower=1> N_2;  // number of grouping levels
##   int<lower=1> M_2;  // number of coefficients per level
##   int<lower=1> J_2[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_2_1;
##   int prior_only;  // should the likelihood be ignored?
## }
## transformed data {
## }
## parameters {
##   real Intercept;  // temporary intercept for centered predictors
##   vector<lower=0>[M_1] sd_1;  // group-level standard deviations
##   vector[N_1] z_1[M_1];  // standardized group-level effects
##   vector<lower=0>[M_2] sd_2;  // group-level standard deviations
##   vector[N_2] z_2[M_2];  // standardized group-level effects
## }
## transformed parameters {
##   vector[N_1] r_1_1;  // actual group-level effects
##   vector[N_2] r_2_1;  // actual group-level effects
##   r_1_1 = (sd_1[1] * (z_1[1]));
##   r_2_1 = (sd_2[1] * (z_2[1]));
## }
## model {
##   // initialize linear predictor term
##   vector[N] mu = Intercept + rep_vector(0, N);
##   for (n in 1:N) {
##     // add more terms to the linear predictor
```

```
##     mu[n] += r_1_1[J_1[n]] * Z_1_1[n] + r_2_1[J_2[n]] * Z_2_1[n];
##   }
##   // priors including all constants
##   target += normal_lpdf(Intercept | 0,1);
##   target += exponential_lpdf(sd_1 | 1);
##   target += std_normal_lpdf(z_1[1]);
##   target += exponential_lpdf(sd_2 | 1);
##   target += std_normal_lpdf(z_2[1]);
##   // likelihood including all constants
##   if (!prior_only) {
##     target += bernoulli_logit_lpmf(Y | mu);
##   }
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept;
## }
```

Model 2

```
make_stancode(correct ~ 1 + trial + (1 + trial | choicePoint) + (1 + trial | participant),
              data = project.data,
              prior = priors_two,
              family = "bernoulli",
              chains = 4,
              warmup = 1000,
              iter = 3500)
```

```
## // generated with brms 2.13.0
## functions {
## }
## data {
##   int<lower=1> N;  // number of observations
##   int Y[N];  // response variable
##   int<lower=1> K;  // number of population-level effects
##   matrix[N, K] X;  // population-level design matrix
##   // data for group-level effects of ID 1
##   int<lower=1> N_1;  // number of grouping levels
##   int<lower=1> M_1;  // number of coefficients per level
##   int<lower=1> J_1[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_1_1;
##   vector[N] Z_1_2;
##   int<lower=1> NC_1;  // number of group-level correlations
##   // data for group-level effects of ID 2
##   int<lower=1> N_2;  // number of grouping levels
##   int<lower=1> M_2;  // number of coefficients per level
##   int<lower=1> J_2[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_2_1;
##   vector[N] Z_2_2;
##   int<lower=1> NC_2;  // number of group-level correlations
##   int prior_only;  // should the likelihood be ignored?
## }
## transformed data {
```

```
##   int Kc = K - 1;
##   matrix[N, Kc] Xc;  // centered version of X without an intercept
##   vector[Kc] means_X;  // column means of X before centering
##   for (i in 2:K) {
##     means_X[i - 1] = mean(X[, i]);
##     Xc[, i - 1] = X[, i] - means_X[i - 1];
##   }
## }
## parameters {
##   vector[Kc] b;  // population-level effects
##   real Intercept;  // temporary intercept for centered predictors
##   vector<lower=0>[M_1] sd_1;  // group-level standard deviations
##   matrix[M_1, N_1] z_1;  // standardized group-level effects
##   cholesky_factor_corr[M_1] L_1;  // cholesky factor of correlation matrix
##   vector<lower=0>[M_2] sd_2;  // group-level standard deviations
##   matrix[M_2, N_2] z_2;  // standardized group-level effects
##   cholesky_factor_corr[M_2] L_2;  // cholesky factor of correlation matrix
## }
## transformed parameters {
##   matrix[N_1, M_1] r_1;  // actual group-level effects
##   // using vectors speeds up indexing in loops
##   vector[N_1] r_1_1;
##   vector[N_1] r_1_2;
##   matrix[N_2, M_2] r_2;  // actual group-level effects
##   // using vectors speeds up indexing in loops
##   vector[N_2] r_2_1;
##   vector[N_2] r_2_2;
##   // compute actual group-level effects
##   r_1 = (diag_pre_multiply(sd_1, L_1) * z_1)';
##   r_1_1 = r_1[, 1];
##   r_1_2 = r_1[, 2];
##   // compute actual group-level effects
##   r_2 = (diag_pre_multiply(sd_2, L_2) * z_2)';
##   r_2_1 = r_2[, 1];
##   r_2_2 = r_2[, 2];
## }
## model {
##   // initialize linear predictor term
##   vector[N] mu = Intercept + Xc * b;
##   for (n in 1:N) {
##     // add more terms to the linear predictor
##     mu[n] += r_1_1[J_1[n]] * Z_1_1[n] + r_1_2[J_1[n]] * Z_1_2[n] + r_2_1[J_2[n]] * Z_2_1[n] + r_2_2[
##   }
##   // priors including all constants
##   target += normal_lpdf(b | 0,1);
##   target += normal_lpdf(Intercept | 0,1);
##   target += exponential_lpdf(sd_1 | 1);
##   target += std_normal_lpdf(to_vector(z_1));
##   target += lkj_corr_cholesky_lpdf(L_1 | 2);
##   target += exponential_lpdf(sd_2 | 1);
##   target += std_normal_lpdf(to_vector(z_2));
##   target += lkj_corr_cholesky_lpdf(L_2 | 2);
##   // likelihood including all constants
##   if (!prior_only) {
```

```
##     target += bernoulli_logit_lpmf(Y | mu);
##   }
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept - dot_product(means_X, b);
##   // compute group-level correlations
##   corr_matrix[M_1] Cor_1 = multiply_lower_tri_self_transpose(L_1);
##   vector<lower=-1,upper=1>[NC_1] cor_1;
##   // compute group-level correlations
##   corr_matrix[M_2] Cor_2 = multiply_lower_tri_self_transpose(L_2);
##   vector<lower=-1,upper=1>[NC_2] cor_2;
##   // extract upper diagonal of correlation matrix
##   for (k in 1:M_1) {
##     for (j in 1:(k - 1)) {
##       cor_1[choose(k - 1, 2) + j] = Cor_1[j, k];
##     }
##   }
##   // extract upper diagonal of correlation matrix
##   for (k in 1:M_2) {
##     for (j in 1:(k - 1)) {
##       cor_2[choose(k - 1, 2) + j] = Cor_2[j, k];
##     }
##   }
## }
```

Model 3

```
make_stancode(correct ~ 1 + trial + saliency +
                    (1 + trial | choicePoint) +
                    (1 + trial + saliency | participant),
                data = project.data,
                prior = priors_three,
                family = "bernoulli",
                chains = 4,
                warmup = 1000,
                iter = 3500)
```

```
## // generated with brms 2.13.0
## functions {
## }
## data {
##   int<lower=1> N;  // number of observations
##   int Y[N];  // response variable
##   int<lower=1> K;  // number of population-level effects
##   matrix[N, K] X;  // population-level design matrix
##   // data for group-level effects of ID 1
##   int<lower=1> N_1;  // number of grouping levels
##   int<lower=1> M_1;  // number of coefficients per level
##   int<lower=1> J_1[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_1_1;
##   vector[N] Z_1_2;
##   int<lower=1> NC_1;  // number of group-level correlations
##   // data for group-level effects of ID 2
```

```
##   int<lower=1> N_2;  // number of grouping levels
##   int<lower=1> M_2;  // number of coefficients per level
##   int<lower=1> J_2[N];  // grouping indicator per observation
##   // group-level predictor values
##   vector[N] Z_2_1;
##   vector[N] Z_2_2;
##   vector[N] Z_2_3;
##   vector[N] Z_2_4;
##   int<lower=1> NC_2;  // number of group-level correlations
##   int prior_only;  // should the likelihood be ignored?
## }
## transformed data {
##   int Kc = K - 1;
##   matrix[N, Kc] Xc;  // centered version of X without an intercept
##   vector[Kc] means_X;  // column means of X before centering
##   for (i in 2:K) {
##     means_X[i - 1] = mean(X[, i]);
##     Xc[, i - 1] = X[, i] - means_X[i - 1];
##   }
## }
## parameters {
##   vector[Kc] b;  // population-level effects
##   real Intercept;  // temporary intercept for centered predictors
##   vector<lower=0>[M_1] sd_1;  // group-level standard deviations
##   matrix[M_1, N_1] z_1;  // standardized group-level effects
##   cholesky_factor_corr[M_1] L_1;  // cholesky factor of correlation matrix
##   vector<lower=0>[M_2] sd_2;  // group-level standard deviations
##   matrix[M_2, N_2] z_2;  // standardized group-level effects
##   cholesky_factor_corr[M_2] L_2;  // cholesky factor of correlation matrix
## }
## transformed parameters {
##   matrix[N_1, M_1] r_1;  // actual group-level effects
##   // using vectors speeds up indexing in loops
##   vector[N_1] r_1_1;
##   vector[N_1] r_1_2;
##   matrix[N_2, M_2] r_2;  // actual group-level effects
##   // using vectors speeds up indexing in loops
##   vector[N_2] r_2_1;
##   vector[N_2] r_2_2;
##   vector[N_2] r_2_3;
##   vector[N_2] r_2_4;
##   // compute actual group-level effects
##   r_1 = (diag_pre_multiply(sd_1, L_1) * z_1)';
##   r_1_1 = r_1[, 1];
##   r_1_2 = r_1[, 2];
##   // compute actual group-level effects
##   r_2 = (diag_pre_multiply(sd_2, L_2) * z_2)';
##   r_2_1 = r_2[, 1];
##   r_2_2 = r_2[, 2];
##   r_2_3 = r_2[, 3];
##   r_2_4 = r_2[, 4];
## }
## model {
##   // initialize linear predictor term
```

```
##   vector[N] mu = Intercept + Xc * b;
##   for (n in 1:N) {
##     // add more terms to the linear predictor
##     mu[n] += r_1_1[J_1[n]] * Z_1_1[n] + r_1_2[J_1[n]] * Z_1_2[n] + r_2_1[J_2[n]] * Z_2_1[n] + r_2_2[
##   }
##   // priors including all constants
##   target += normal_lpdf(b | 0,1);
##   target += normal_lpdf(Intercept | 0,1);
##   target += exponential_lpdf(sd_1 | 1);
##   target += std_normal_lpdf(to_vector(z_1));
##   target += lkj_corr_cholesky_lpdf(L_1 | 2);
##   target += exponential_lpdf(sd_2 | 1);
##   target += std_normal_lpdf(to_vector(z_2));
##   target += lkj_corr_cholesky_lpdf(L_2 | 2);
##   // likelihood including all constants
##   if (!prior_only) {
##     target += bernoulli_logit_lpmf(Y | mu);
##   }
## }
## generated quantities {
##   // actual population-level intercept
##   real b_Intercept = Intercept - dot_product(means_X, b);
##   // compute group-level correlations
##   corr_matrix[M_1] Cor_1 = multiply_lower_tri_self_transpose(L_1);
##   vector<lower=-1,upper=1>[NC_1] cor_1;
##   // compute group-level correlations
##   corr_matrix[M_2] Cor_2 = multiply_lower_tri_self_transpose(L_2);
##   vector<lower=-1,upper=1>[NC_2] cor_2;
##   // extract upper diagonal of correlation matrix
##   for (k in 1:M_1) {
##     for (j in 1:(k - 1)) {
##       cor_1[choose(k - 1, 2) + j] = Cor_1[j, k];
##     }
##   }
##   // extract upper diagonal of correlation matrix
##   for (k in 1:M_2) {
##     for (j in 1:(k - 1)) {
##       cor_2[choose(k - 1, 2) + j] = Cor_2[j, k];
##     }
##   }
## }
```

Diagnostic Checks

For each model, I show the trace plots and $\widehat{R}$ values to gauge chain convergence, as well as the ESS. There was no evidence that the chains failed to converge.

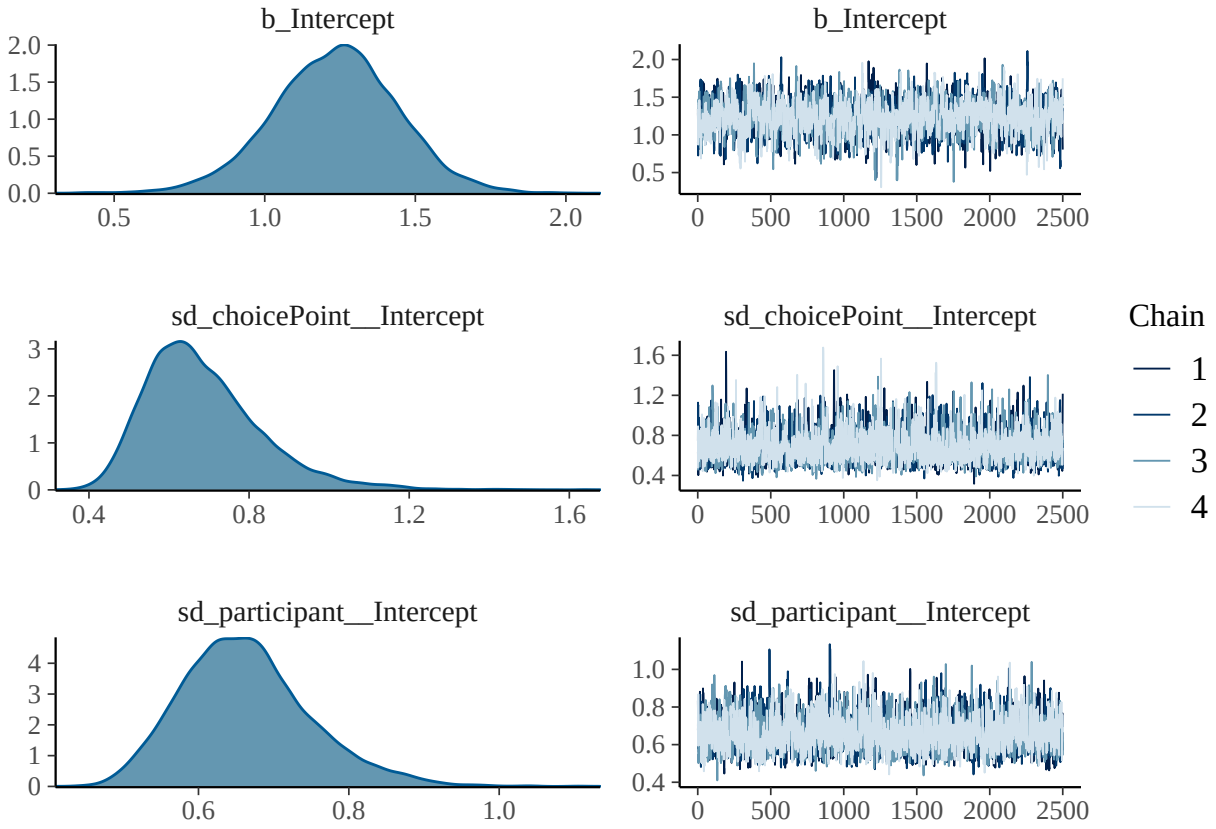
Model One

```
summary(mod_one)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + (1 | choicePoint) + (1 | participant)
##    Data: project.data (Number of observations: 6450)
```

```
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.69      0.15     0.47     1.03 1.00     2225     4179
##
## ~participant (Number of levels: 42)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.66      0.09     0.52     0.86 1.00     1986     3441
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.24      0.21     0.81     1.64 1.00     1530     3015
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(mod_one)
```



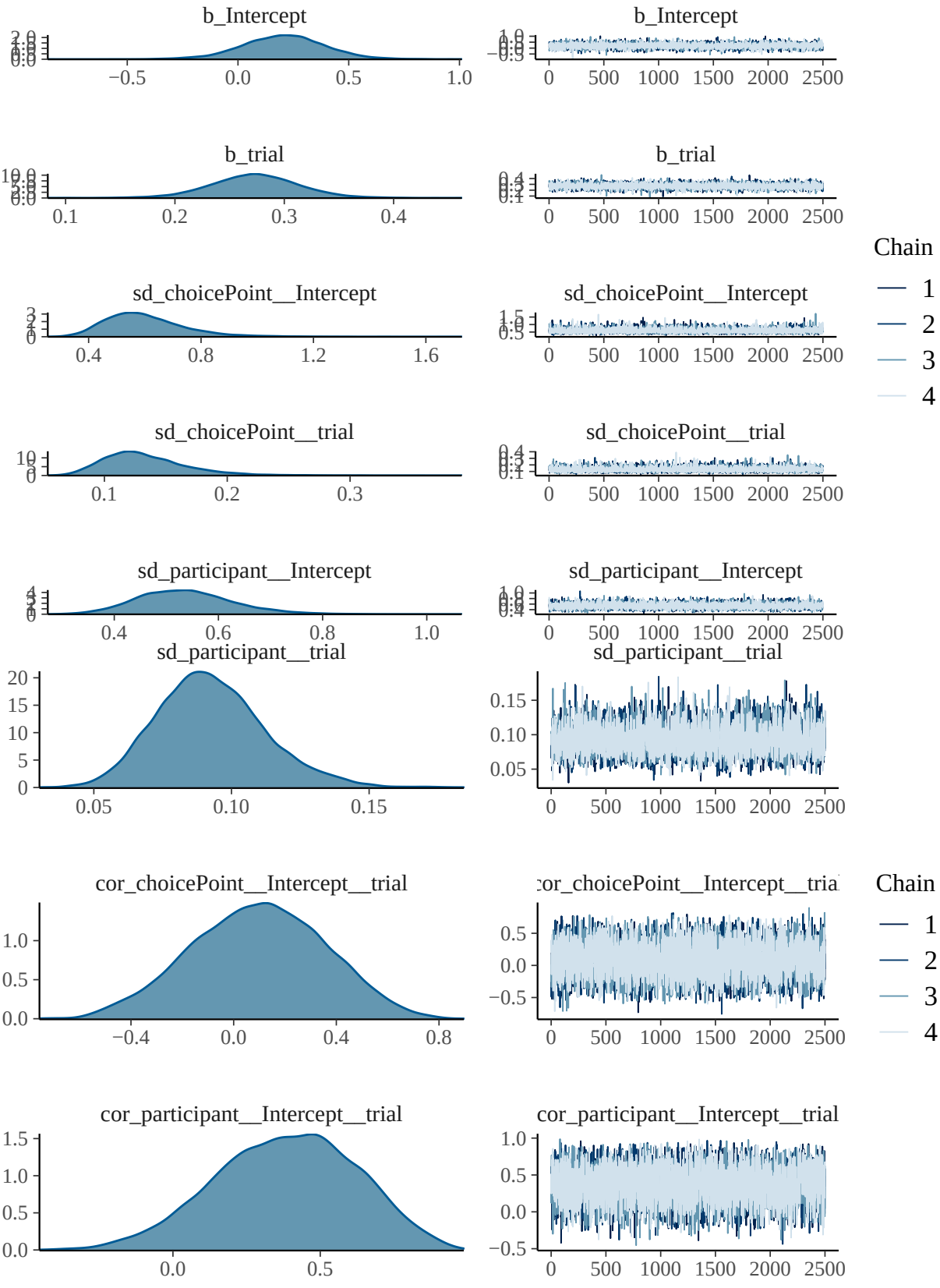
Model Two

```
summary(mod_two)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + trial + (1 + trial | choicePoint) + (1 + trial | participant)
```

```
##    Data: project.data (Number of observations: 6450)
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)             0.60      0.14     0.38     0.93 1.00     6289
## sd(trial)                 0.13      0.03     0.08     0.21 1.00     6203
## cor(Intercept,trial)      0.10      0.26    -0.42     0.58 1.00     6401
##                       Tail_ESS
## sd(Intercept)             7527
## sd(trial)                 7027
## cor(Intercept,trial)      7112
##
## ~participant (Number of levels: 42)
##                       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)             0.54      0.09     0.37     0.73 1.00     7266
## sd(trial)                 0.09      0.02     0.06     0.14 1.00     4919
## cor(Intercept,trial)      0.38      0.24    -0.10     0.82 1.00     5022
##                       Tail_ESS
## sd(Intercept)             7767
## sd(trial)                 6402
## cor(Intercept,trial)      5079
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.21      0.19    -0.17     0.57 1.00     5782     6871
## trial         0.27      0.04     0.19     0.35 1.00     6471     6606
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(mod_two)
```

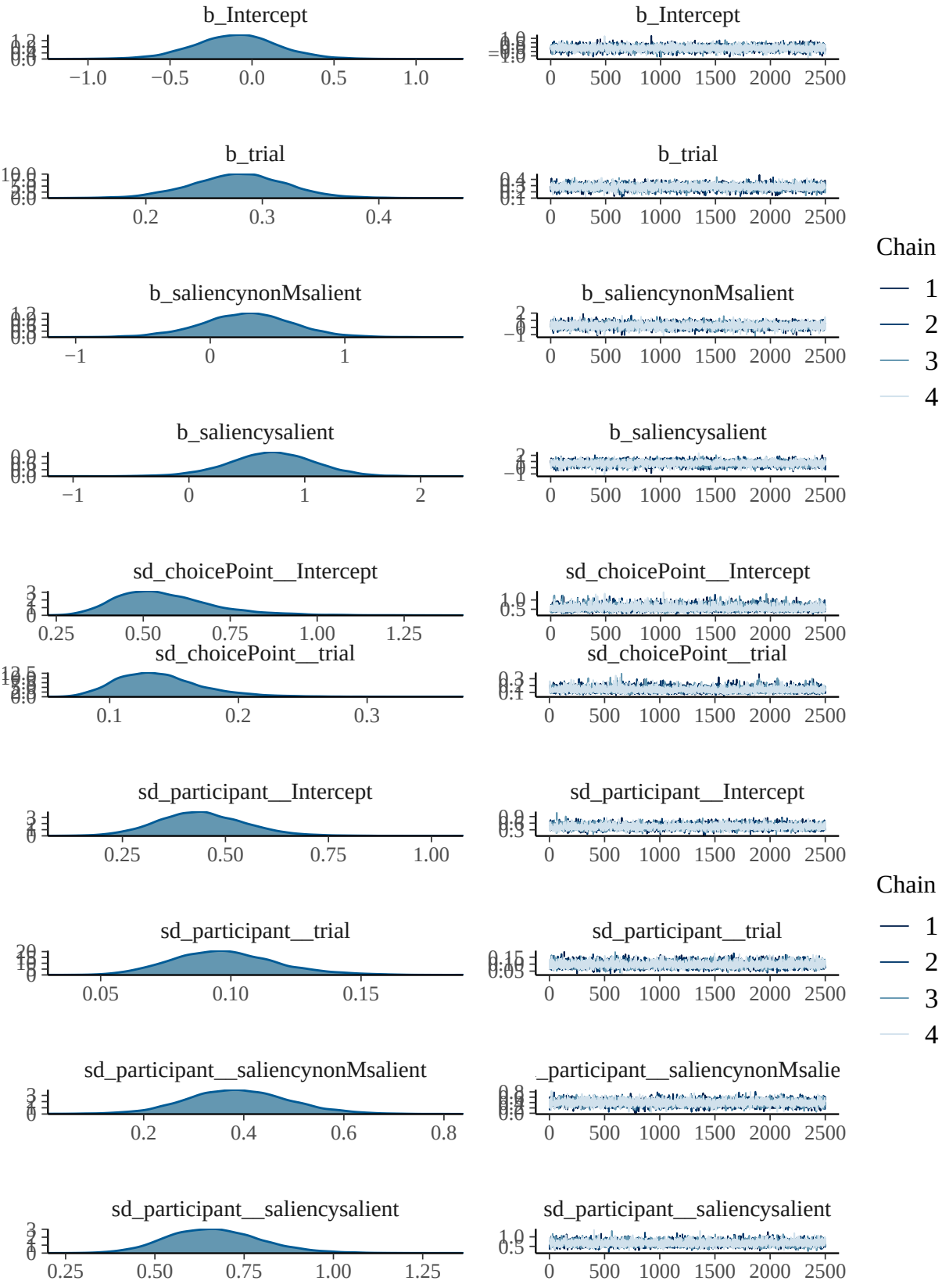

b_Intercept
b_Intercept
b_trial
b_trial
Chain
1
2
3
4
sd_choicePoint__Intercept
sd_choicePoint__Intercept
sd_choicePoint__trial
sd_choicePoint__trial
sd_participant__Intercept
sd_participant__Intercept
sd_participant__trial
sd_participant__trial
cor_choicePoint__Intercept__trial
Chain
1
2
3
4
cor_participant__Intercept__trial
cor_participant__Intercept__trial

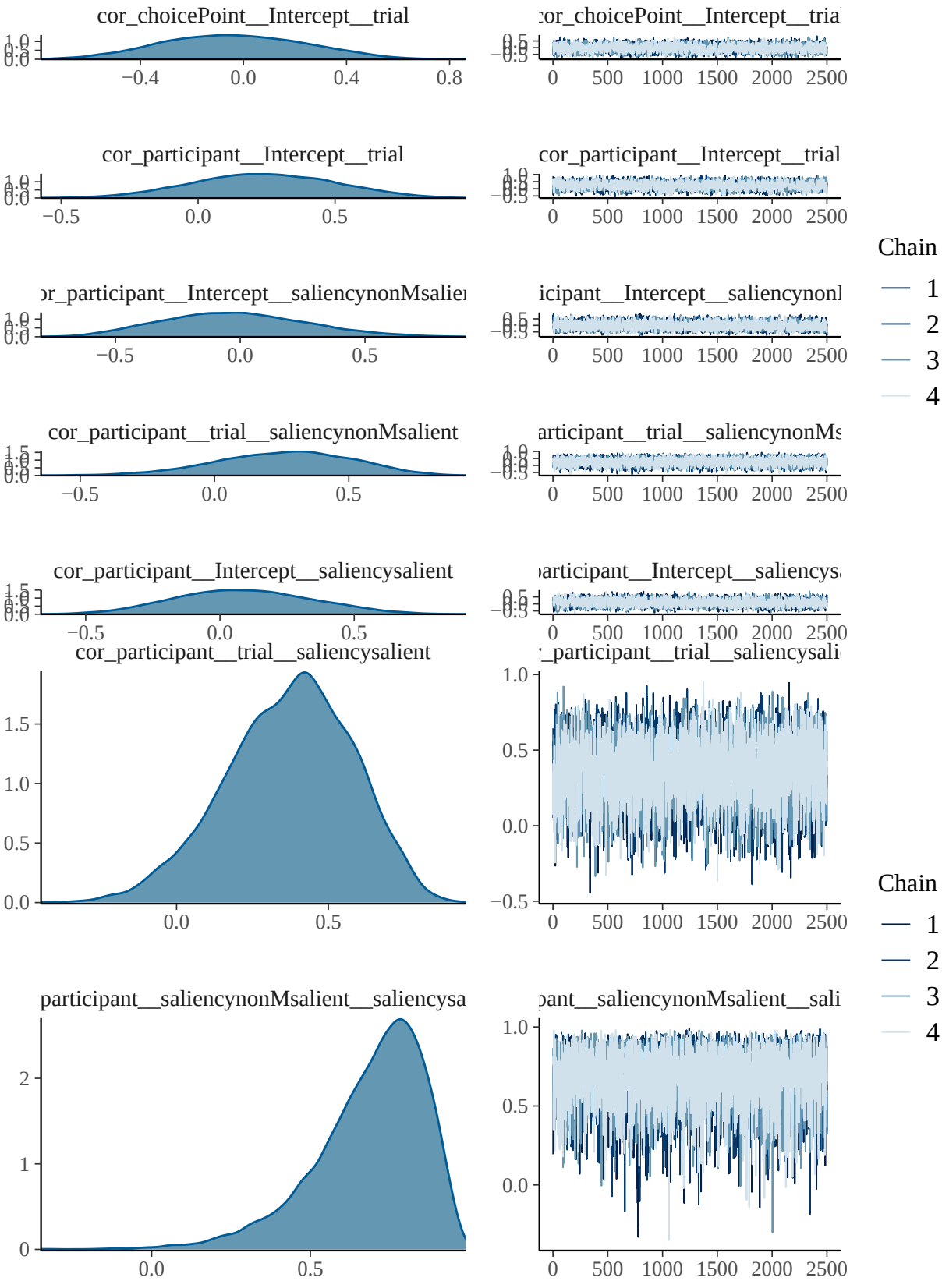
Model Three

```
summary(mod_three)
```

```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + trial + saliency + (1 + trial | choicePoint) + (1 + trial + saliency | partici
##    Data: project.data (Number of observations: 6450)
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)           0.56      0.14     0.35     0.90 1.00     7159
## sd(trial)               0.14      0.03     0.08     0.22 1.00     6585
## cor(Intercept,trial)   -0.04      0.27    -0.55     0.49 1.00     4331
##                     Tail_ESS
## sd(Intercept)           7621
## sd(trial)               7712
## cor(Intercept,trial)    6102
##
## ~participant (Number of levels: 42)
##                                       Estimate Est.Error l-95% CI u-95% CI
## sd(Intercept)                             0.44      0.11     0.24     0.66
## sd(trial)                                 0.10      0.02     0.06     0.14
## sd(saliencynonMsalient)                   0.39      0.11     0.18     0.60
## sd(saliencysalient)                       0.67      0.13     0.43     0.95
## cor(Intercept,trial)                      0.24      0.25    -0.25     0.73
## cor(Intercept,saliencynonMsalient)       -0.02      0.28    -0.53     0.56
## cor(trial,saliencynonMsalient)            0.26      0.25    -0.26     0.71
## cor(Intercept,saliencysalient)            0.09      0.25    -0.38     0.59
## cor(trial,saliencysalient)                0.37      0.21    -0.06     0.74
## cor(saliencynonMsalient,saliencysalient)  0.69      0.17     0.26     0.93
##                                       Rhat Bulk_ESS Tail_ESS
## sd(Intercept)                         1.00     5840     7140
## sd(trial)                             1.00     6186     6962
## sd(saliencynonMsalient)               1.00     4772     4173
## sd(saliencysalient)                   1.00     6432     7269
## cor(Intercept,trial)                  1.00     3888     5121
## cor(Intercept,saliencynonMsalient)    1.00     5482     6789
## cor(trial,saliencynonMsalient)        1.00     6422     8346
## cor(Intercept,saliencysalient)        1.00     3913     5664
## cor(trial,saliencysalient)            1.00     4898     6359
## cor(saliencynonMsalient,saliencysalient) 1.00     4860     6347
##
## Population-Level Effects:
##                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept            -0.10      0.26    -0.63     0.41 1.00     7746     7459
## trial                 0.28      0.04     0.20     0.36 1.00     5962     6838
## saliencynonMsalient   0.27      0.35    -0.42     0.95 1.00     7628     6964
## saliencysalient       0.72      0.37    -0.03     1.42 1.00     6737     7364
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(mod_three)
```




cor_choicePoint__Intercept__trial
cor_participant__Intercept__trial
cor_participant__trial__saliencynonMsalient
cor_participant__Intercept__saliencysalient
cor_participant__trial__saliencysalient
participant__saliencynonMsalient__saliencysa
Chain
1
2
3
4

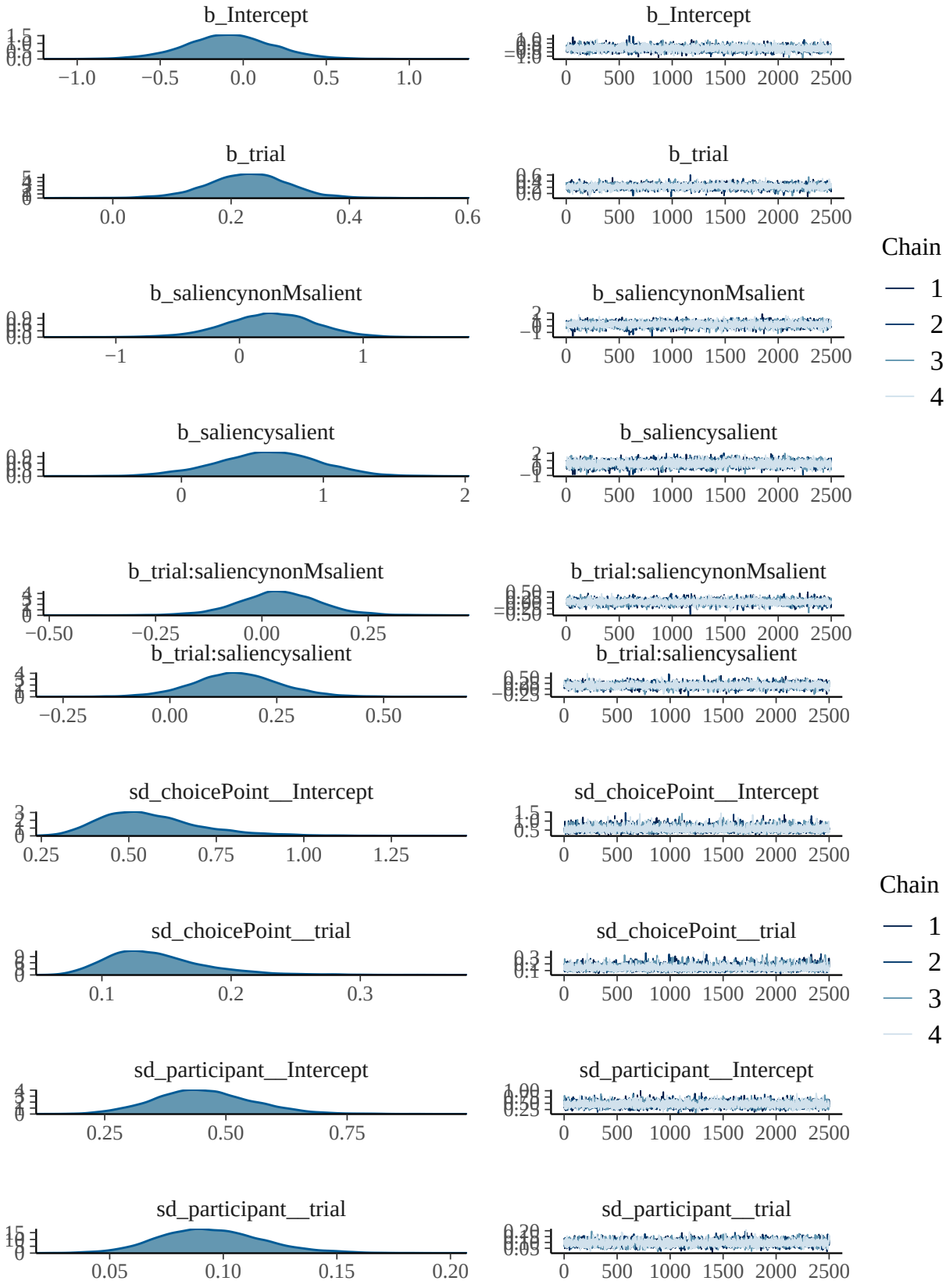
Model Four

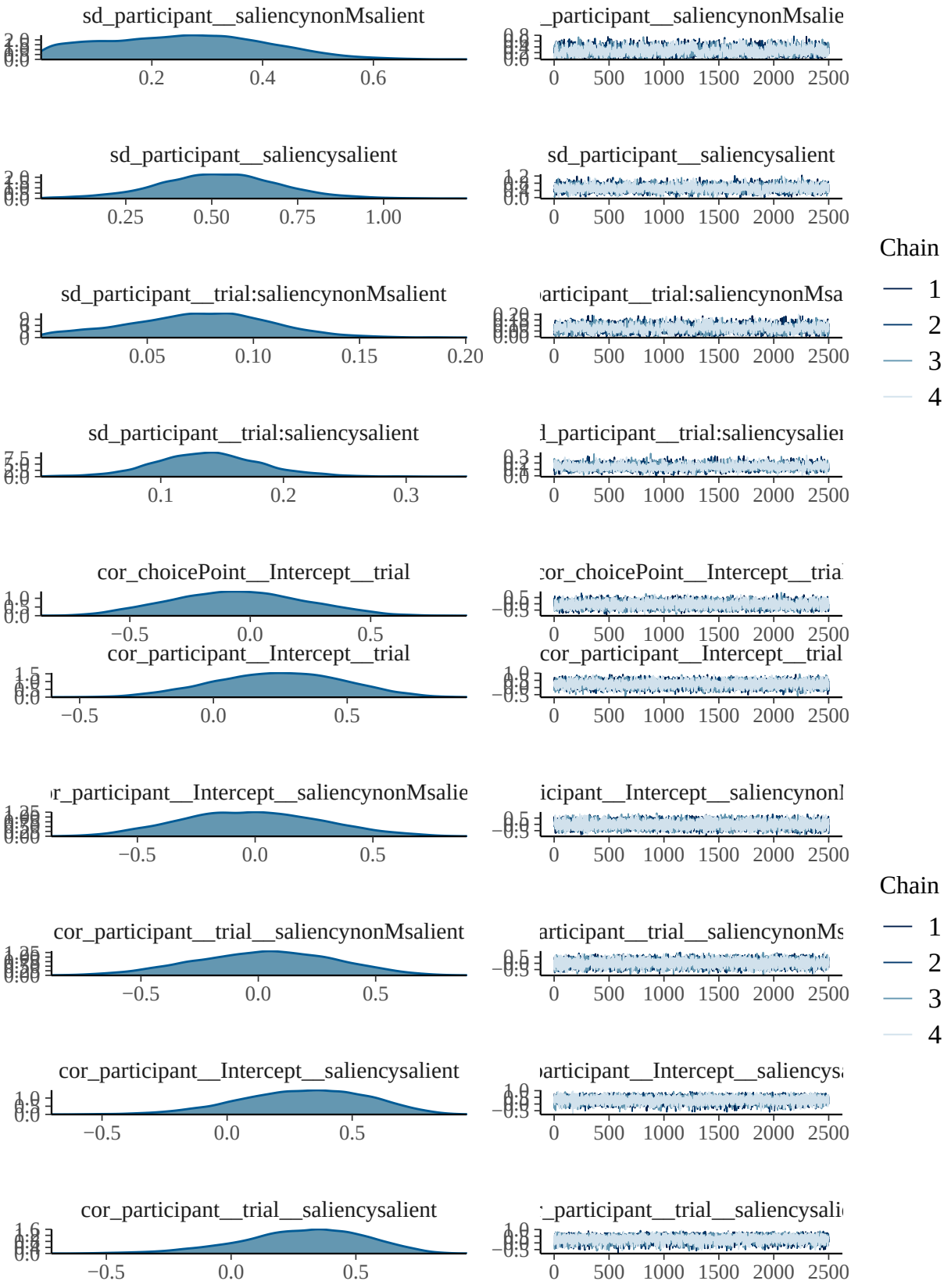
```
summary(mod_four)
```

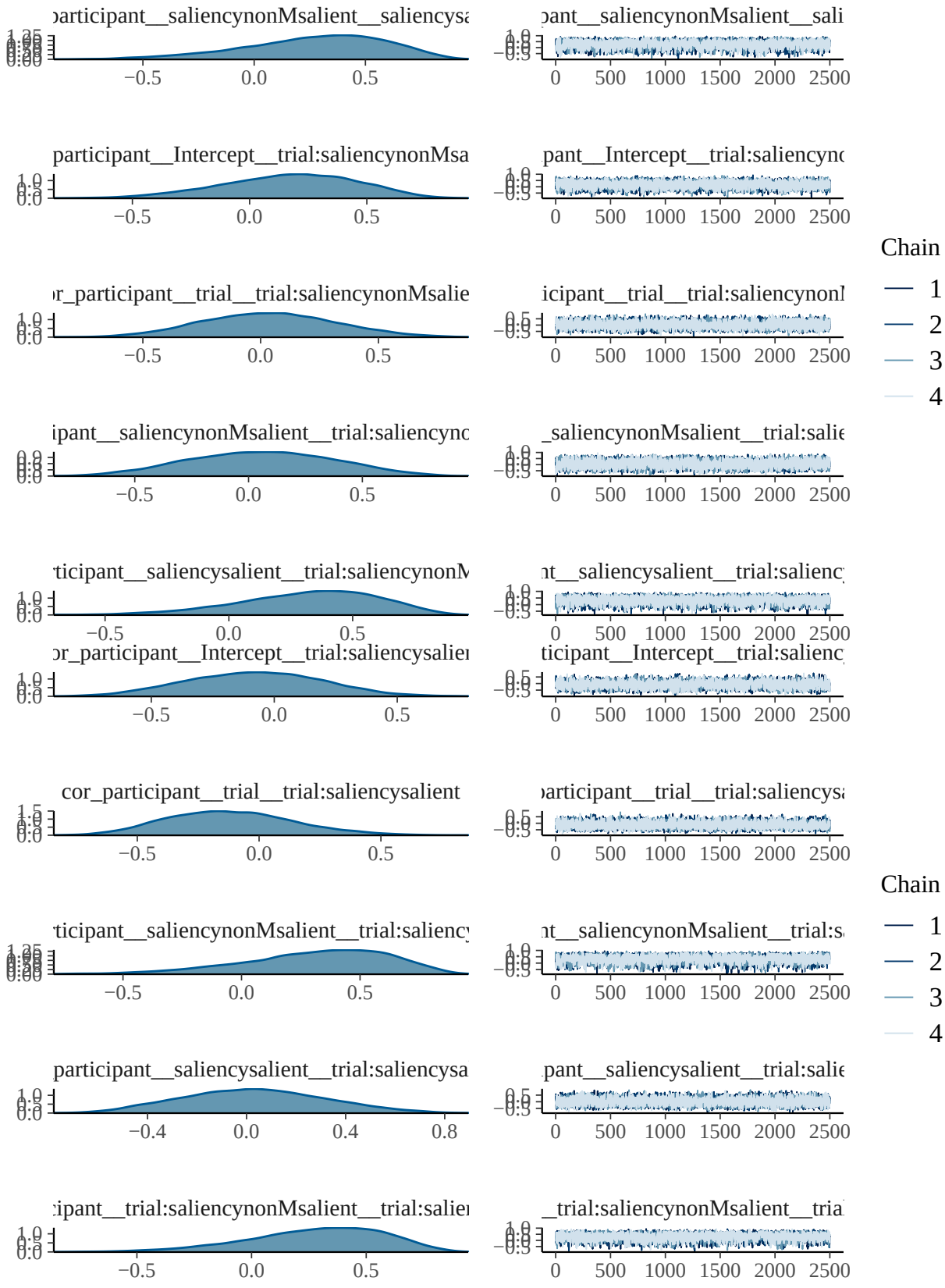
```
##  Family: bernoulli
##   Links: mu = logit
## Formula: correct ~ 1 + trial * saliency + (1 + trial | choicePoint) + (1 + trial * saliency | partici
##    Data: project.data (Number of observations: 6450)
## Samples: 4 chains, each with iter = 3500; warmup = 1000; thin = 1;
##          total post-warmup samples = 10000
##
## Group-Level Effects:
## ~choicePoint (Number of levels: 15)
##                     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)           0.56      0.15     0.34     0.91 1.00     5648
## sd(trial)               0.14      0.04     0.08     0.24 1.00     4856
## cor(Intercept,trial)   -0.05      0.27    -0.56     0.48 1.00     4363
##                     Tail_ESS
## sd(Intercept)           6830
## sd(trial)               5863
## cor(Intercept,trial)    5941
##
## ~participant (Number of levels: 42)
##                                                   Estimate Est.Error
## sd(Intercept)                                         0.45      0.10
## sd(trial)                                             0.09      0.02
## sd(saliencynonMsalient)                               0.26      0.14
## sd(saliencysalient)                                   0.52      0.18
## sd(trial:saliencynonMsalient)                         0.07      0.03
## sd(trial:saliencysalient)                             0.14      0.04
## cor(Intercept,trial)                                  0.24      0.24
## cor(Intercept,saliencynonMsalient)                   -0.01      0.30
## cor(trial,saliencynonMsalient)                        0.03      0.29
## cor(Intercept,saliencysalient)                        0.31      0.25
## cor(trial,saliencysalient)                            0.29      0.25
## cor(saliencynonMsalient,saliencysalient)              0.27      0.31
## cor(Intercept,trial:saliencynonMsalient)              0.17      0.28
## cor(trial,trial:saliencynonMsalient)                  0.04      0.28
## cor(saliencynonMsalient,trial:saliencynonMsalient)    0.06      0.31
## cor(saliencysalient,trial:saliencynonMsalient)        0.32      0.28
## cor(Intercept,trial:saliencysalient)                 -0.08      0.27
## cor(trial,trial:saliencysalient)                     -0.12      0.26
## cor(saliencynonMsalient,trial:saliencysalient)        0.31      0.30
## cor(saliencysalient,trial:saliencysalient)            0.03      0.28
## cor(trial:saliencynonMsalient,trial:saliencysalient)  0.28      0.29
##                                                   l-95% CI u-95% CI Rhat
## sd(Intercept)                                         0.27     0.67 1.00
## sd(trial)                                             0.05     0.14 1.00
## sd(saliencynonMsalient)                               0.02     0.54 1.00
## sd(saliencysalient)                                   0.17     0.88 1.00
## sd(trial:saliencynonMsalient)                         0.01     0.14 1.00
## sd(trial:saliencysalient)                             0.05     0.23 1.00
## cor(Intercept,trial)                                 -0.23     0.70 1.00
## cor(Intercept,saliencynonMsalient)                   -0.56     0.58 1.00
## cor(trial,saliencynonMsalient)                       -0.55     0.58 1.00
## cor(Intercept,saliencysalient)                       -0.21     0.75 1.00
```

```
## cor(trial,saliencysalient)                                -0.25     0.72 1.00
## cor(saliencynonMsalient,saliencysalient)                  -0.43     0.78 1.00
## cor(Intercept,trial:saliencynonMsalient)                  -0.42     0.66 1.00
## cor(trial,trial:saliencynonMsalient)                      -0.49     0.59 1.00
## cor(saliencynonMsalient,trial:saliencynonMsalient)        -0.55     0.65 1.00
## cor(saliencysalient,trial:saliencynonMsalient)            -0.29     0.78 1.00
## cor(Intercept,trial:saliencysalient)                      -0.60     0.44 1.00
## cor(trial,trial:saliencysalient)                          -0.58     0.41 1.00
## cor(saliencynonMsalient,trial:saliencysalient)            -0.36     0.79 1.00
## cor(saliencysalient,trial:saliencysalient)                -0.49     0.60 1.00
## cor(trial:saliencynonMsalient,trial:saliencysalient)      -0.37     0.76 1.00
##                                                       Bulk_ESS Tail_ESS
## sd(Intercept)                                             6110     6544
## sd(trial)                                                 3718     5408
## sd(saliencynonMsalient)                                   1460     2528
## sd(saliencysalient)                                       3001     3261
## sd(trial:saliencynonMsalient)                             1670     2584
## sd(trial:saliencysalient)                                 3100     3582
## cor(Intercept,trial)                                      4515     6737
## cor(Intercept,saliencynonMsalient)                       10172     7807
## cor(trial,saliencynonMsalient)                            5990     7492
## cor(Intercept,saliencysalient)                            5183     6798
## cor(trial,saliencysalient)                                5388     6145
## cor(saliencynonMsalient,saliencysalient)                  2226     3858
## cor(Intercept,trial:saliencynonMsalient)                  5742     6430
## cor(trial,trial:saliencynonMsalient)                      6743     7206
## cor(saliencynonMsalient,trial:saliencynonMsalient)        4612     6467
## cor(saliencysalient,trial:saliencynonMsalient)            4970     5913
## cor(Intercept,trial:saliencysalient)                      3408     6885
## cor(trial,trial:saliencysalient)                          6280     6726
## cor(saliencynonMsalient,trial:saliencysalient)            2129     3110
## cor(saliencysalient,trial:saliencysalient)                5743     6712
## cor(trial:saliencynonMsalient,trial:saliencysalient)      3136     5839
##
## Population-Level Effects:
##                           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## Intercept                    -0.08      0.27    -0.60     0.45 1.00     4313
## trial                         0.23      0.07     0.09     0.37 1.00     3513
## saliencynonMsalient           0.25      0.36    -0.48     0.94 1.00     4551
## saliencysalient               0.61      0.37    -0.13     1.31 1.00     5121
## trial:saliencynonMsalient     0.04      0.10    -0.16     0.24 1.00     3942
## trial:saliencysalient         0.16      0.10    -0.04     0.36 1.00     4433
##                           Tail_ESS
## Intercept                     6227
## trial                         4466
## saliencynonMsalient           6297
## saliencysalient               6429
## trial:saliencynonMsalient     5356
## trial:saliencysalient         4942
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(mod_four)
```




sd_participant__saliencynonMsalient
sd_participant__saliencysalient
sd_participant__trial:saliencynonMsalient
sd_participant__trial:saliencysalient
cor_choicePoint__Intercept__trial
cor_participant__Intercept__trial
cor_participant__trial__saliencynonMsalient
cor_participant__Intercept__saliencysalient
cor_participant__trial__saliencysalient
Chain
1
2
3
4


participant__saliencynonMsalient__saliencysa
pant__saliencynonMsalient__sali
participant__Intercept__trial:saliencynonMsa
pant__Intercept__trial:saliencync
r_participant__trial__trial:saliencynonMsalie
icipant__trial__trial:saliencynonI
ipant__saliencynonMsalient__trial:saliencync
_saliencynonMsalient__trial:salie
ticipant__saliencysalient__trial:saliencynonM
nt__saliencysalient__trial:salienc
or_participant__Intercept__trial:saliencysalien
ticipant__Intercept__trial:salienc
cor_participant__trial__trial:saliencysalient
participant__trial__trial:saliencysa
ticipant__saliencynonMsalient__trial:saliency
nt__saliencynonMsalient__trial:s
participant__saliencysalient__trial:saliencysa
pant__saliencysalient__trial:salie
ipant__trial:saliencynonMsalient__trial:salien
_trial:saliencynonMsalient__trial
Chain
1
2
3
4
Chain
1
2
3
4