Direct effect of human feedback

Nele Albers

09 August, 2024

Contents

Introduction

Here we fit multilevel models to assess the direct impact of sending human feedback messages on the effort people spend on their preparatory activities and the likelihood that they would have returned to the next session if it was part of an unpaid intervention.

Required files:

- Data/data_rl_samples.csv

Setup

First, we load the "rethinking"-package, which we need to fit and sample from models. We also load the "formatR"-package for formatting.

```
library(formatR)  # For formatting
library(rethinking)  # For Bayesian models
```

Also, we set the number of chains and number of iterations used for fitting the models.

```
NUM_CHAINS = 4  # our value: 4
NUM_ITERATIONS = 3000  # our value: 3000
```

Data file

We load the pre-processed data.

```
df = read.csv(file = "Data/data_rl_samples.csv")
```

Fit model for effort

Here we fit a model for the effort. We fit a t-distribution for the dependent variable.

```
# Create a data list to be used for the model
dat_list_effort <- list(effort = df$effort, humansupport = df$a,
    id = df$cons_id + 1  # Needs to start at 1 and not 0
)

set.seed(18)

ml.effort <- ulam(alist(effort ~ dstudent(v, mu, sigma), mu <- a_bar +
    z[id] * sigma_a + b_hs * humansupport, v ~ gamma(2, 0.1),
    z[id] ~ dnorm(0, 1), sigma_a ~ dexp(1), a_bar ~ dnorm(5,
        10), sigma ~ dexp(1), b_hs ~ dnorm(0, 10)), data = dat_list_effort,
    chains = NUM_CHAINS, log_lik = TRUE, cores = NUM_CHAINS,
    iter = NUM_ITERATIONS)
```

```
## Running MCMC with 4 parallel chains, with 1 thread(s) per chain...
##
## Chain 1 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 1 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 1 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 1
## Chain 1 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 1 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 1
## Chain 1 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 1 Exception: gamma_lpdf: Random variable is inf, but must be positive finite! (in '/tmp/Rtmpr8u
## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 1
## Chain 2 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 2 Exception: gamma_lpdf: Random variable is inf, but must be positive finite! (in '/tmp/Rtmpr8u
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 2
## Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of the
```

```
## Chain 2 Exception: gamma_lpdf: Random variable is inf, but must be positive finite! (in '/tmp/Rtmpr8w
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 2
## Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 2 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 2
## Chain 3 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 3 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 3
## Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 3 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 3
## Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 3 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 3
## Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 3 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 3
## Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 3 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 3
## Chain 4 Iteration:    1 / 3000 [  0%]  (Warmup)
```

```
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/Rtmpr8w84
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 4
## Chain 1 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 1 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 3 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 4 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 2 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 1 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 3 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 4 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 2 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 1 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 4 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 3 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 2 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 1 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 4 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 3 Iteration:  400 / 3000 [ 13%]  (Warmup)
```

```
## Chain 2 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 1 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 4 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 3 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 2 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 1 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 4 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 3 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 2 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 1 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 4 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 3 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 2 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 1 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 4 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 3 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 2 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 1 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 4 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 3 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 2 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 1 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 4 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 3 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 2 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 1 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 4 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 3 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 2 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 1 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 4 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 3 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 2 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 1 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 4 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 3 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 2 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 4 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 1 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 3 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 2 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 1 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 3 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 4 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 4 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 2 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 2 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 3 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 4 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 2 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3 Iteration: 1700 / 3000 [ 56%]  (Sampling)
```

```
## Chain 4 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 1 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 2 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 3 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 4 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 2 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 3 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 1 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 4 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 2 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 3 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 4 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 1 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 2 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 3 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 4 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 1 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 3 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 2 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 1 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 4 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 3 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 2 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 1 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 2 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 3 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 1 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 4 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 2 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 3 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 4 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 2 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 3 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 1 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 4 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 2 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 3 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 1 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 4 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 2 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 3 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 1 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 4 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 2 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 3 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 finished in 39.2 seconds.
## Chain 2 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 3 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3 finished in 39.9 seconds.
```

```
## Chain 4 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4 finished in 40.1 seconds.
## Chain 2 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 2 finished in 40.7 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 40.0 seconds.
## Total execution time: 40.9 seconds.
```

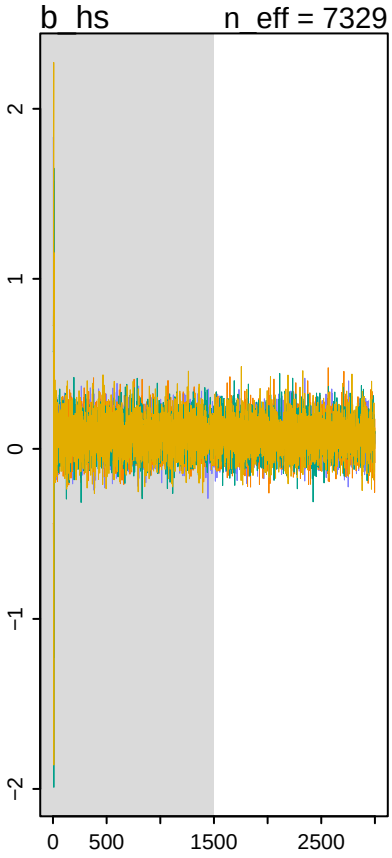
```
output_effort = precis(ml.effort, prob = 0.95)
```

```
## 679 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
output_effort
```

```
##               mean         sd       2.5%     97.5%      rhat ess_bulk
## v       4.18202505 0.42067474  3.4203162 5.0873885 1.0002125 4022.357
## sigma_a 1.97956407 0.07093367  1.8435090 2.1200412 1.0020812 1502.944
## a_bar   5.81358427 0.09080373  5.6332778 5.9911817 1.0006572 2056.130
## sigma   1.56934878 0.05364061  1.4630683 1.6742608 1.0013226 2823.304
## b_hs    0.07975835 0.10929487 -0.1273857 0.2935971 0.9998777 7329.398
```

```
traceplot(ml.effort, pars = c("b_hs"))
```



And we compute the posterior probability that the parameter `b_hs` is greater than 0.

```
set.seed(18)  # For reproducibility
samples.ml.effort <- extract.samples(ml.effort)
Heffort_post <- samples.ml.effort$b_hs[which(samples.ml.effort$b_hs >
```

```
    0)]
Heffort_post_p <- round(length(Heffort_post)/length(samples.ml.effort$b_hs),
    2)
Heffort_post_p
```

```
## [1] 0.76
```

And let's also compute an effect size.

```
b_hs = output_effort$mean[5]
sd = output_effort$mean[4]
effect_size = b_hs/sd
round(effect_size, 2)
```

```
## [1] 0.05
```

Fit model for return likelihood

Next we fit a model for the return likelihood. We fit a normal distribution for the dependent variable.

```
# Create a data list to be used for the model
dat_list_dropout <- list(dropout = df$dropout_response, humansupport = df$a,
    id = df$cons_id + 1  # Needs to start at 1 and not 0
)

set.seed(18)

ml.dropout <- ulam(alist(dropout ~ dnorm(mu, sigma), mu <- a_bar +
    z[id] + b_hs * humansupport, z[id] ~ dnorm(0, sigma_a), sigma_a ~
    dexp(1), a_bar ~ dnorm(0, 10), sigma ~ dexp(1), b_hs ~ dnorm(0,
    10)), data = dat_list_dropout, chains = NUM_CHAINS, log_lik = TRUE,
    cores = NUM_CHAINS, iter = NUM_ITERATIONS, control = list(adapt_delta = 0.999))
```

```
## Running MCMC with 4 parallel chains, with 1 thread(s) per chain...
##
## Chain 1 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 1 Informational Message: The current Metropolis proposal is about to be rejected because of the
## Chain 1 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/tmp/Rtmpr8w84L/mode
## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like covar
## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or mi
## Chain 1
## Chain 2 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 3 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 4 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 3 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 4 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 2 Iteration:  100 / 3000 [  3%]  (Warmup)
## Chain 3 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 1 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 4 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 2 Iteration:  200 / 3000 [  6%]  (Warmup)
```

```
## Chain 3 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 4 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 2 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 3 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 4 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 1 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 2 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 3 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 1 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 4 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 2 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 3 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 1 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 4 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 2 Iteration:  600 / 3000 [ 20%]  (Warmup)
## Chain 3 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 1 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 4 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 2 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 1 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 3 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 4 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 2 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 1 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 3 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 4 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 2 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 4 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 1 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 3 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 2 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 4 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 1 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 3 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 2 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 4 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 1 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 3 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 2 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 4 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 1 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 3 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 2 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 4 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 3 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 1 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 2 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 3 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 3 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 4 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 4 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1501 / 3000 [ 50%]  (Sampling)
```

```
## Chain 3 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 2 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 2 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 3 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 4 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 1 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 2 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 4 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 1 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 3 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 2 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 3 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 4 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 3 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 2 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 3 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 1 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 2 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 3 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 4 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 1 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 3 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 2 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 3 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 4 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 1 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 3 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 2 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 3 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 1 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 2 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 3 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 1 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 3 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 4 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 2 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 3 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3 finished in 64.6 seconds.
## Chain 1 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 4 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 2 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 1 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 4 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 2 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 4 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 2 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 1 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 4 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 2 Iteration: 2700 / 3000 [ 90%]  (Sampling)
```

```
## Chain 1 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 4 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 2 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 1 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 4 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 2 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 1 finished in 79.6 seconds.
## Chain 4 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4 finished in 79.5 seconds.
## Chain 2 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 2 finished in 80.1 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 76.0 seconds.
## Total execution time: 80.2 seconds.
```

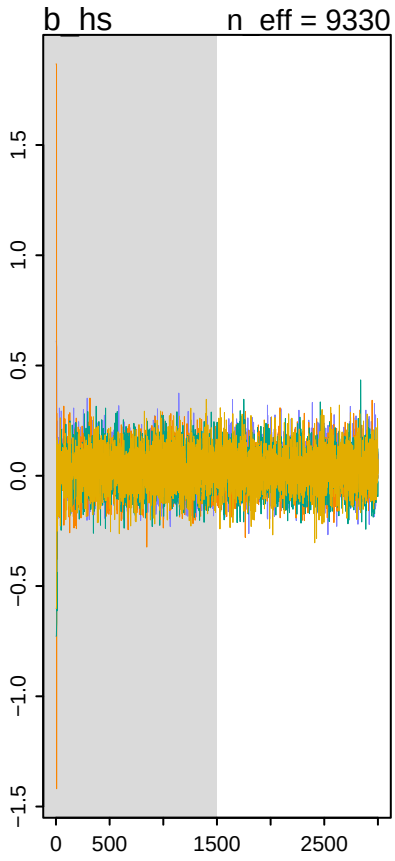
```
output_dropout = precis(ml.dropout, prob = 0.95)
```

```
## 679 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
output_dropout
```

```
##               mean         sd       2.5%     97.5%      rhat ess_bulk
## sigma_a 2.18716573 0.06978142  2.0549652 2.3240120 0.9997913 8027.242
## a_bar   1.76257051 0.09512971  1.5674073 1.9407960 1.0025981 1423.737
## sigma   1.59572066 0.02798415  1.5429992 1.6524715 0.9996661 6376.027
## b_hs    0.03341997 0.09329156 -0.1481035 0.2175969 1.0009374 9329.619
```

```
traceplot(ml.dropout, pars = c("b_hs"))
```



And we compute the posterior probability that the parameter `b_hs` is greater than 0.

```
set.seed(18)  # For reproducibility
samples.ml.dropout <- extract.samples(ml.dropout)
print(paste0("Number of samples extracted: ", length(samples.ml.dropout$b_hs)))
```

```
## [1] "Number of samples extracted: 6000"
```

```
Hdrop_post <- samples.ml.dropout$b_hs[which(samples.ml.dropout$b_hs >
    0)]
Hdrop_post_p <- round(length(Hdrop_post)/length(samples.ml.dropout$b_hs),
    2)
Hdrop_post_p
```

```
## [1] 0.64
```

And let's also compute an effect size.

```
b_hs = output_dropout$mean[4]
sd = output_dropout$mean[3]
effect_size = b_hs/sd
round(effect_size, 2)
```

```
## [1] 0.02
```