Delayed effect of single human feedback

Nele Albers

09 August, 2024

Contents

Introduction

Here we fit multilevel models to assess the delayed 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. Specifically, we include a factor that captures whether a person has received feedback so far.

Setup

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

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

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

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

Data file

We load the pre-processed data.

```
df = read.csv(file = "Data/data_rl_samples.csv")
df$Prev_Feedback_Binary = as.integer(df$Prev_Feedback_Count > 0)
```

Fit model for effort

First we fit a model for the effort.

```r
# Create a data list to be used for the model
dat_list_effort <- list(
  effort = df$effort,
  humansupport = df$a,
  id = df$cons_id + 1,  # needs to start at 1
  prev_humansupport = df$Prev_Feedback_Binary
)

set.seed(18)

ml.effort <- ulam(
  alist(
    effort ~ dstudent(v, mu, sigma),
    mu <- a_bar + z[id] * sigma_a + b_prevhs * prev_humansupport,
    v ~ gamma(2,0.1),
    z[id] ~ dnorm(0, 1),
    sigma_a ~ dexp(1),
    a_bar ~ dnorm(5, 10),
    sigma ~ dexp(1),
    b_prevhs ~ dnorm(0, 10)
  ), data = dat_list_effort, chains=NUM_CHAINS, log_lik = TRUE, cores=NUM_CHAINS, iter = 3000
)
```

```
## 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/RtmpPsXV

## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like cova

## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or m

## 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/RtmpPsXV

## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like cova

## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or m

## 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/RtmpPsXV

## Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like cova

## Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or m

## 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/RtmpPs
```

```
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPs
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
## Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
## 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/RtmpPsXV
```

```
## Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 3
## Chain 4 Iteration:    1 / 3000 [  0%]  (Warmup)
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/RtmpPsXV
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/RtmpPsXV
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/RtmpPsXV
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/RtmpPsXV
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 4
## Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
## Chain 4 Exception: gamma_lpdf: Random variable is 0, but must be positive finite! (in '/tmp/RtmpPsXV
## Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
## Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
## Chain 4
## 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 4 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 1 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 3 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 2 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 1 Iteration:  300 / 3000 [ 10%]  (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:  400 / 3000 [ 13%]  (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:  500 / 3000 [ 16%]  (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:  600 / 3000 [ 20%]  (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:  700 / 3000 [ 23%]  (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:  800 / 3000 [ 26%]  (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:  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 1 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 3 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 4 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 2 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 1 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 3 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 4 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 2 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 3 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 1 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 4 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 2 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 3 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 4 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 1 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 2 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 3 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 1 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 4 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 2 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 2 Iteration: 1501 / 3000 [ 50%]  (Sampling)
```

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

```
## Chain 2 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 3 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3 finished in 41.5 seconds.
## Chain 4 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4 finished in 41.5 seconds.
## Chain 2 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 2 finished in 41.6 seconds.
## Chain 1 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 1 finished in 41.9 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 41.6 seconds.
## Total execution time: 42.2 seconds.
```

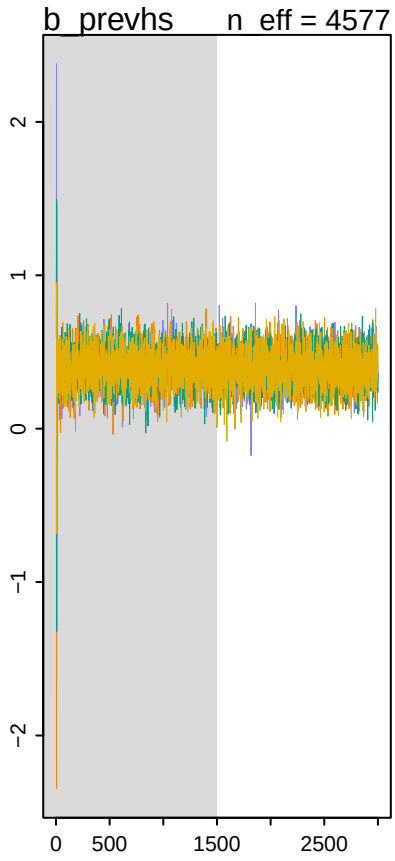
```
output_effort = precis(ml.effort, prob = 0.95)
```

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

```
output_effort
```

```
##               mean         sd      2.5%    97.5%     rhat ess_bulk
## v        4.1518407 0.41402516 3.4151708 5.047034 1.000075 5035.841
## sigma_a  1.9779697 0.07115322 1.8412082 2.118003 1.000699 1990.774
## a_bar    5.6811836 0.09816456 5.4882932 5.873021 1.000724 2341.391
## sigma    1.5599923 0.05245779 1.4556682 1.660070 1.000429 3277.486
## b_prevhs 0.3937635 0.11693118 0.1683984 0.624800 1.001478 4576.658
```

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



Let's compute the posterior probability that `b_prevhs` is greater than 0.

```
set.seed(18)  # For reproducibility
samples.ml.effort <- extract.samples(ml.effort)
Heffort_post <- samples.ml.effort$b_prevhs[which(samples.ml.effort$b_prevhs >
    0)]
Heffort_post_p <- round(length(Heffort_post)/length(samples.ml.effort$b_prevhs),
    5)
Heffort_post_p
```

```
## [1] 0.9995
```

And let's also compute an effect size.

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

```
## [1] 0.25
```

```
rm(ml.effort, output_effort, samples.ml.effort)
```

Fit model for return likelihood

Now we fit a model for the return likelihood.

```
# Create a data list to be used for the model
dat_list_dropout <- list(
```

```
  dropout = df$dropout_response,
  humansupport = df$a,
  id = df$cons_id + 1,  # needs to start at 1
  prev_humansupport = df$Prev_Feedback_Binary
)

set.seed(18)

ml.dropout <- ulam(
  alist(
    dropout ~ dnorm(mu, sigma),
    mu <- a_bar + z[id] + b_prevhs * prev_humansupport,
    z[id] ~ dnorm(0, sigma_a),
    sigma_a ~ dexp(1),
    a_bar ~ dnorm(0, 10),
    sigma ~ dexp(1),
    b_prevhs ~ dnorm(0, 10)
  ), data = dat_list_dropout, chains=NUM_CHAINS, log_lik = TRUE, cores=NUM_CHAINS, iter = 3000, control=
)
```

```
## 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/RtmpPsXVch/model
## 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 2 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 4 Iteration:  200 / 3000 [  6%]  (Warmup)
## Chain 3 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 1 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 2 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 4 Iteration:  300 / 3000 [ 10%]  (Warmup)
## Chain 3 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 1 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 2 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 4 Iteration:  400 / 3000 [ 13%]  (Warmup)
## Chain 3 Iteration:  500 / 3000 [ 16%]  (Warmup)
## Chain 1 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:  500 / 3000 [ 16%]  (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:  600 / 3000 [ 20%]  (Warmup)
## Chain 2 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 3 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 1 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 4 Iteration:  700 / 3000 [ 23%]  (Warmup)
## Chain 2 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 3 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 1 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 4 Iteration:  800 / 3000 [ 26%]  (Warmup)
## Chain 3 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 2 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 1 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 4 Iteration:  900 / 3000 [ 30%]  (Warmup)
## Chain 2 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 3 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 1 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 4 Iteration: 1000 / 3000 [ 33%]  (Warmup)
## Chain 2 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 1 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 3 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 4 Iteration: 1100 / 3000 [ 36%]  (Warmup)
## Chain 2 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 1 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 4 Iteration: 1200 / 3000 [ 40%]  (Warmup)
## Chain 2 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 3 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 1 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 4 Iteration: 1300 / 3000 [ 43%]  (Warmup)
## Chain 2 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 3 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 1 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 1 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 4 Iteration: 1400 / 3000 [ 46%]  (Warmup)
## Chain 2 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 2 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 4 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 4 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 1 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3 Iteration: 1500 / 3000 [ 50%]  (Warmup)
## Chain 3 Iteration: 1501 / 3000 [ 50%]  (Sampling)
## Chain 2 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 4 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 3 Iteration: 1600 / 3000 [ 53%]  (Sampling)
## Chain 1 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 4 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 2 Iteration: 1700 / 3000 [ 56%]  (Sampling)
## Chain 4 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 1 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 3 Iteration: 1700 / 3000 [ 56%]  (Sampling)
```

```
## Chain 4 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 2 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 4 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 1 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 3 Iteration: 1800 / 3000 [ 60%]  (Sampling)
## Chain 4 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 2 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 4 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 1 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 3 Iteration: 1900 / 3000 [ 63%]  (Sampling)
## Chain 4 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 2 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 1 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 4 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 3 Iteration: 2000 / 3000 [ 66%]  (Sampling)
## Chain 2 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 4 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 3 Iteration: 2100 / 3000 [ 70%]  (Sampling)
## Chain 4 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 2 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 1 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 3 Iteration: 2200 / 3000 [ 73%]  (Sampling)
## Chain 4 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 2 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 4 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 3 Iteration: 2300 / 3000 [ 76%]  (Sampling)
## Chain 4 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 4 finished in 61.7 seconds.
## Chain 2 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 1 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 3 Iteration: 2400 / 3000 [ 80%]  (Sampling)
## Chain 2 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 1 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 3 Iteration: 2500 / 3000 [ 83%]  (Sampling)
## Chain 2 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 1 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 3 Iteration: 2600 / 3000 [ 86%]  (Sampling)
## Chain 2 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 1 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 3 Iteration: 2700 / 3000 [ 90%]  (Sampling)
## Chain 2 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 1 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 3 Iteration: 2800 / 3000 [ 93%]  (Sampling)
## Chain 2 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3 Iteration: 2900 / 3000 [ 96%]  (Sampling)
## Chain 1 finished in 74.3 seconds.
## Chain 2 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 2 finished in 75.1 seconds.
## Chain 3 Iteration: 3000 / 3000 [100%]  (Sampling)
## Chain 3 finished in 75.7 seconds.
```

```
##
## All 4 chains finished successfully.
## Mean chain execution time: 71.7 seconds.
## Total execution time: 76.0 seconds.
```

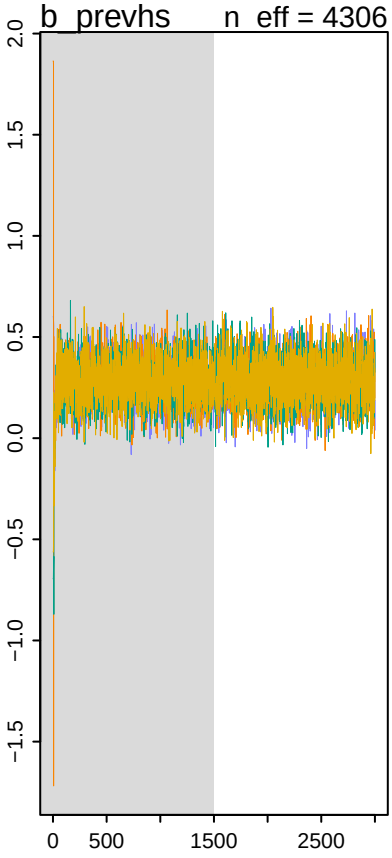
```
output_dropout = precis(ml.dropout, prob = 0.95)
```

```
## 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  2.1961023 0.07086758 2.06103400 2.3407505 1.000286 7691.238
## a_bar    1.6530914 0.10022664 1.45822150 1.8510733 1.001344 1736.542
## sigma    1.5912287 0.02795542 1.53637950 1.6481470 1.000059 6495.600
## b_prevhs 0.2869409 0.10647136 0.08115047 0.5014545 1.000986 4306.200
```

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



Let's compute the posterior probability that `b_prevhs` 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_prevhs)))
```

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

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

```
    5)
Hdrop_post_p
```

```
## [1] 0.99683
```

And let's also compute an effect size.

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

```
## [1] 0.18
```