

# Statistical Analysis

Ramya Ghantasala, Nele Albers

18 April, 2023

This file reproduces Figure 3a and Figure 3b, the sample mean motivational impact per condition reported in the “Results”-section of the paper, and the results from our Bayesian analysis reported in the “Results”-section of the paper.

Required files: `scripts/preprocessing.r`

Output files: `Statistical-Analysis.pdf`, `Figures/comparison_motivational_impact.pdf`, `Figures/post_fixed_effect.pdf`

## Setup

Let's first import the packages that we need.

```
library(formatR)
library(ggplot2)
library(reshape2)
library(rethinking)
```

Now we run the pre-processing.

```
source("./scripts/preprocessing.r")
```

## Sample Motivational Impact of Message Types

Let's first compute the sample mean and standard deviation of the motivational impact per condition. These values are reported in the “Results”-section of the paper.

```
mean_gen = mean(combinedMotivation[combinedMotivation$messageType == 0,
                                   ]$motivation)
sd_gen = sd(combinedMotivation[combinedMotivation$messageType == 0, ]$motivation)
paste("Sample mean motivational impact generic messages:", round(mean_gen,
  2), "(SD = ", round(sd_gen, 2), ")")
```

```
## [1] "Sample mean motivational impact generic messages: 1.32 (SD = 2.29 )"
```

```
mean_tailored = mean(combinedMotivation[combinedMotivation$messageType ==
  1, ]$motivation)
sd_tailored = sd(combinedMotivation[combinedMotivation$messageType == 1,
                                   ]$motivation)
paste("Sample mean motivational impact tailored messages:", round(mean_tailored,
  2), "(SD = ", round(sd_tailored, 2), ")")
```

```
## [1] "Sample mean motivational impact tailored messages: 2.33 (SD = 2.11 )"
```

The below figure compares the motivational impact of generic and tailored messages. This reproduces Figure 3a from the paper.

```
comparison_motivation <- ggplot(combinedMotivation, aes(x = factor(messageType,
  labels = c("Generic", "Tailored")), y = motivation, fill = factor(messageType,
  labels = c("Generic", "Tailored")))) + geom_boxplot() + labs(x = "Message Type",
  y = "Perceived Motivational Impact", fill = "Message Type") + theme_light()

# Save image
pdf_file <- "Figures/comparison_motivational_impact.pdf"
ggsave(pdf_file, dpi = 1500)
knitr::plot_crop(pdf_file)
```

```
## [1] "Figures/comparison_motivational_impact.pdf"
```

This figure corresponds to Figure 3a in the paper.

## Statistical Modelling

### Fit Model

First we fit our model. This model contains a general intercept, random intercepts for the participants, and tailoring as fixed effect. We fit a t-distribution for the outcome variable.

```
set.seed(4) # For reproducibility

m <- ulam(
  alist(
    motivation ~ dstudent(v, mu, sigma),
    # model equation
    mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
    # priors
    a_person[pid] ~ dnorm(0, 1),
    sigma_p ~ dexp(1),
    v ~ gamma(2, 0.1),
    a ~ dnorm(0, 10),
    b ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
  control=list(adapt_delta=.99, max_treedepth = 15)
)
```

```
## Running MCMC with 4 parallel chains, with 1 thread(s) per chain...
```

```
##
## Chain 1 Iteration: 1 / 10000 [ 0%] (Warmup)
## Chain 2 Iteration: 1 / 10000 [ 0%] (Warmup)
## Chain 3 Iteration: 1 / 10000 [ 0%] (Warmup)
## Chain 4 Iteration: 1 / 10000 [ 0%] (Warmup)
## Chain 1 Iteration: 100 / 10000 [ 1%] (Warmup)
## Chain 3 Iteration: 100 / 10000 [ 1%] (Warmup)
## Chain 2 Iteration: 100 / 10000 [ 1%] (Warmup)
## Chain 1 Iteration: 200 / 10000 [ 2%] (Warmup)
## Chain 2 Iteration: 200 / 10000 [ 2%] (Warmup)
## Chain 3 Iteration: 200 / 10000 [ 2%] (Warmup)
## Chain 4 Iteration: 100 / 10000 [ 1%] (Warmup)
## Chain 1 Iteration: 300 / 10000 [ 3%] (Warmup)
## Chain 3 Iteration: 300 / 10000 [ 3%] (Warmup)
## Chain 2 Iteration: 300 / 10000 [ 3%] (Warmup)
```

```

## Chain 1 Iteration: 400 / 10000 [ 4%] (Warmup)
## Chain 3 Iteration: 400 / 10000 [ 4%] (Warmup)
## Chain 4 Iteration: 200 / 10000 [ 2%] (Warmup)
## Chain 2 Iteration: 400 / 10000 [ 4%] (Warmup)
## Chain 1 Iteration: 500 / 10000 [ 5%] (Warmup)
## Chain 3 Iteration: 500 / 10000 [ 5%] (Warmup)
## Chain 4 Iteration: 300 / 10000 [ 3%] (Warmup)
## Chain 2 Iteration: 500 / 10000 [ 5%] (Warmup)
## Chain 1 Iteration: 600 / 10000 [ 6%] (Warmup)
## Chain 3 Iteration: 600 / 10000 [ 6%] (Warmup)
## Chain 2 Iteration: 600 / 10000 [ 6%] (Warmup)
## Chain 4 Iteration: 400 / 10000 [ 4%] (Warmup)
## Chain 1 Iteration: 700 / 10000 [ 7%] (Warmup)
## Chain 3 Iteration: 700 / 10000 [ 7%] (Warmup)
## Chain 2 Iteration: 700 / 10000 [ 7%] (Warmup)
## Chain 4 Iteration: 500 / 10000 [ 5%] (Warmup)
## Chain 3 Iteration: 800 / 10000 [ 8%] (Warmup)
## Chain 1 Iteration: 800 / 10000 [ 8%] (Warmup)
## Chain 4 Iteration: 600 / 10000 [ 6%] (Warmup)
## Chain 2 Iteration: 800 / 10000 [ 8%] (Warmup)
## Chain 3 Iteration: 900 / 10000 [ 9%] (Warmup)
## Chain 1 Iteration: 900 / 10000 [ 9%] (Warmup)
## Chain 4 Iteration: 700 / 10000 [ 7%] (Warmup)
## Chain 2 Iteration: 900 / 10000 [ 9%] (Warmup)
## Chain 3 Iteration: 1000 / 10000 [ 10%] (Warmup)
## Chain 1 Iteration: 1000 / 10000 [ 10%] (Warmup)
## Chain 4 Iteration: 800 / 10000 [ 8%] (Warmup)
## Chain 2 Iteration: 1000 / 10000 [ 10%] (Warmup)
## Chain 3 Iteration: 1100 / 10000 [ 11%] (Warmup)
## Chain 1 Iteration: 1100 / 10000 [ 11%] (Warmup)
## Chain 4 Iteration: 900 / 10000 [ 9%] (Warmup)
## Chain 2 Iteration: 1100 / 10000 [ 11%] (Warmup)
## Chain 3 Iteration: 1200 / 10000 [ 12%] (Warmup)
## Chain 1 Iteration: 1200 / 10000 [ 12%] (Warmup)
## Chain 4 Iteration: 1000 / 10000 [ 10%] (Warmup)
## Chain 2 Iteration: 1200 / 10000 [ 12%] (Warmup)
## Chain 3 Iteration: 1300 / 10000 [ 13%] (Warmup)
## Chain 4 Iteration: 1100 / 10000 [ 11%] (Warmup)
## Chain 1 Iteration: 1300 / 10000 [ 13%] (Warmup)
## Chain 2 Iteration: 1300 / 10000 [ 13%] (Warmup)
## Chain 3 Iteration: 1400 / 10000 [ 14%] (Warmup)
## Chain 1 Iteration: 1400 / 10000 [ 14%] (Warmup)
## Chain 4 Iteration: 1200 / 10000 [ 12%] (Warmup)
## Chain 2 Iteration: 1400 / 10000 [ 14%] (Warmup)
## Chain 3 Iteration: 1500 / 10000 [ 15%] (Warmup)
## Chain 4 Iteration: 1300 / 10000 [ 13%] (Warmup)
## Chain 1 Iteration: 1500 / 10000 [ 15%] (Warmup)
## Chain 2 Iteration: 1500 / 10000 [ 15%] (Warmup)
## Chain 3 Iteration: 1600 / 10000 [ 16%] (Warmup)
## Chain 1 Iteration: 1600 / 10000 [ 16%] (Warmup)
## Chain 4 Iteration: 1400 / 10000 [ 14%] (Warmup)
## Chain 2 Iteration: 1600 / 10000 [ 16%] (Warmup)
## Chain 1 Iteration: 1700 / 10000 [ 17%] (Warmup)
## Chain 2 Iteration: 1700 / 10000 [ 17%] (Warmup)

```

```

## Chain 3 Iteration: 1700 / 10000 [ 17%] (Warmup)
## Chain 4 Iteration: 1500 / 10000 [ 15%] (Warmup)
## Chain 1 Iteration: 1800 / 10000 [ 18%] (Warmup)
## Chain 3 Iteration: 1800 / 10000 [ 18%] (Warmup)
## Chain 2 Iteration: 1800 / 10000 [ 18%] (Warmup)
## Chain 4 Iteration: 1600 / 10000 [ 16%] (Warmup)
## Chain 3 Iteration: 1900 / 10000 [ 19%] (Warmup)
## Chain 1 Iteration: 1900 / 10000 [ 19%] (Warmup)
## Chain 2 Iteration: 1900 / 10000 [ 19%] (Warmup)
## Chain 4 Iteration: 1700 / 10000 [ 17%] (Warmup)
## Chain 1 Iteration: 2000 / 10000 [ 20%] (Warmup)
## Chain 3 Iteration: 2000 / 10000 [ 20%] (Warmup)
## Chain 2 Iteration: 2000 / 10000 [ 20%] (Warmup)
## Chain 4 Iteration: 1800 / 10000 [ 18%] (Warmup)
## Chain 3 Iteration: 2100 / 10000 [ 21%] (Warmup)
## Chain 1 Iteration: 2100 / 10000 [ 21%] (Warmup)
## Chain 2 Iteration: 2100 / 10000 [ 21%] (Warmup)
## Chain 4 Iteration: 1900 / 10000 [ 19%] (Warmup)
## Chain 1 Iteration: 2200 / 10000 [ 22%] (Warmup)
## Chain 3 Iteration: 2200 / 10000 [ 22%] (Warmup)
## Chain 2 Iteration: 2200 / 10000 [ 22%] (Warmup)
## Chain 4 Iteration: 2000 / 10000 [ 20%] (Warmup)
## Chain 1 Iteration: 2300 / 10000 [ 23%] (Warmup)
## Chain 3 Iteration: 2300 / 10000 [ 23%] (Warmup)
## Chain 2 Iteration: 2300 / 10000 [ 23%] (Warmup)
## Chain 4 Iteration: 2100 / 10000 [ 21%] (Warmup)
## Chain 1 Iteration: 2400 / 10000 [ 24%] (Warmup)
## Chain 2 Iteration: 2400 / 10000 [ 24%] (Warmup)
## Chain 3 Iteration: 2400 / 10000 [ 24%] (Warmup)
## Chain 4 Iteration: 2200 / 10000 [ 22%] (Warmup)
## Chain 3 Iteration: 2500 / 10000 [ 25%] (Warmup)
## Chain 1 Iteration: 2500 / 10000 [ 25%] (Warmup)
## Chain 2 Iteration: 2500 / 10000 [ 25%] (Warmup)
## Chain 4 Iteration: 2300 / 10000 [ 23%] (Warmup)
## Chain 3 Iteration: 2600 / 10000 [ 26%] (Warmup)
## Chain 1 Iteration: 2600 / 10000 [ 26%] (Warmup)
## Chain 2 Iteration: 2600 / 10000 [ 26%] (Warmup)
## Chain 4 Iteration: 2400 / 10000 [ 24%] (Warmup)
## Chain 3 Iteration: 2700 / 10000 [ 27%] (Warmup)
## Chain 1 Iteration: 2700 / 10000 [ 27%] (Warmup)
## Chain 2 Iteration: 2700 / 10000 [ 27%] (Warmup)
## Chain 3 Iteration: 2800 / 10000 [ 28%] (Warmup)
## Chain 4 Iteration: 2500 / 10000 [ 25%] (Warmup)
## Chain 1 Iteration: 2800 / 10000 [ 28%] (Warmup)
## Chain 2 Iteration: 2800 / 10000 [ 28%] (Warmup)
## Chain 4 Iteration: 2600 / 10000 [ 26%] (Warmup)
## Chain 3 Iteration: 2900 / 10000 [ 29%] (Warmup)
## Chain 2 Iteration: 2900 / 10000 [ 29%] (Warmup)
## Chain 4 Iteration: 2700 / 10000 [ 27%] (Warmup)
## Chain 1 Iteration: 2900 / 10000 [ 29%] (Warmup)
## Chain 3 Iteration: 3000 / 10000 [ 30%] (Warmup)
## Chain 1 Iteration: 3000 / 10000 [ 30%] (Warmup)
## Chain 2 Iteration: 3000 / 10000 [ 30%] (Warmup)
## Chain 3 Iteration: 3100 / 10000 [ 31%] (Warmup)

```

```

## Chain 4 Iteration: 2800 / 10000 [ 28%] (Warmup)
## Chain 2 Iteration: 3100 / 10000 [ 31%] (Warmup)
## Chain 4 Iteration: 2900 / 10000 [ 29%] (Warmup)
## Chain 1 Iteration: 3100 / 10000 [ 31%] (Warmup)
## Chain 3 Iteration: 3200 / 10000 [ 32%] (Warmup)
## Chain 1 Iteration: 3200 / 10000 [ 32%] (Warmup)
## Chain 2 Iteration: 3200 / 10000 [ 32%] (Warmup)
## Chain 3 Iteration: 3300 / 10000 [ 33%] (Warmup)
## Chain 4 Iteration: 3000 / 10000 [ 30%] (Warmup)
## Chain 1 Iteration: 3300 / 10000 [ 33%] (Warmup)
## Chain 2 Iteration: 3300 / 10000 [ 33%] (Warmup)
## Chain 4 Iteration: 3100 / 10000 [ 31%] (Warmup)
## Chain 3 Iteration: 3400 / 10000 [ 34%] (Warmup)
## Chain 4 Iteration: 3200 / 10000 [ 32%] (Warmup)
## Chain 1 Iteration: 3400 / 10000 [ 34%] (Warmup)
## Chain 2 Iteration: 3400 / 10000 [ 34%] (Warmup)
## Chain 3 Iteration: 3500 / 10000 [ 35%] (Warmup)
## Chain 4 Iteration: 3300 / 10000 [ 33%] (Warmup)
## Chain 1 Iteration: 3500 / 10000 [ 35%] (Warmup)
## Chain 2 Iteration: 3500 / 10000 [ 35%] (Warmup)
## Chain 3 Iteration: 3600 / 10000 [ 36%] (Warmup)
## Chain 1 Iteration: 3600 / 10000 [ 36%] (Warmup)
## Chain 3 Iteration: 3700 / 10000 [ 37%] (Warmup)
## Chain 4 Iteration: 3400 / 10000 [ 34%] (Warmup)
## Chain 2 Iteration: 3600 / 10000 [ 36%] (Warmup)
## Chain 4 Iteration: 3500 / 10000 [ 35%] (Warmup)
## Chain 1 Iteration: 3700 / 10000 [ 37%] (Warmup)
## Chain 3 Iteration: 3800 / 10000 [ 38%] (Warmup)
## Chain 2 Iteration: 3700 / 10000 [ 37%] (Warmup)
## Chain 1 Iteration: 3800 / 10000 [ 38%] (Warmup)
## Chain 3 Iteration: 3900 / 10000 [ 39%] (Warmup)
## Chain 4 Iteration: 3600 / 10000 [ 36%] (Warmup)
## Chain 2 Iteration: 3800 / 10000 [ 38%] (Warmup)
## Chain 4 Iteration: 3700 / 10000 [ 37%] (Warmup)
## Chain 1 Iteration: 3900 / 10000 [ 39%] (Warmup)
## Chain 3 Iteration: 4000 / 10000 [ 40%] (Warmup)
## Chain 2 Iteration: 3900 / 10000 [ 39%] (Warmup)
## Chain 1 Iteration: 4000 / 10000 [ 40%] (Warmup)
## Chain 3 Iteration: 4100 / 10000 [ 41%] (Warmup)
## Chain 4 Iteration: 3800 / 10000 [ 38%] (Warmup)
## Chain 2 Iteration: 4000 / 10000 [ 40%] (Warmup)
## Chain 3 Iteration: 4200 / 10000 [ 42%] (Warmup)
## Chain 1 Iteration: 4100 / 10000 [ 41%] (Warmup)
## Chain 2 Iteration: 4100 / 10000 [ 41%] (Warmup)
## Chain 4 Iteration: 3900 / 10000 [ 39%] (Warmup)
## Chain 3 Iteration: 4300 / 10000 [ 43%] (Warmup)
## Chain 1 Iteration: 4200 / 10000 [ 42%] (Warmup)
## Chain 4 Iteration: 4000 / 10000 [ 40%] (Warmup)
## Chain 2 Iteration: 4200 / 10000 [ 42%] (Warmup)
## Chain 3 Iteration: 4400 / 10000 [ 44%] (Warmup)
## Chain 1 Iteration: 4300 / 10000 [ 43%] (Warmup)
## Chain 4 Iteration: 4100 / 10000 [ 41%] (Warmup)
## Chain 2 Iteration: 4300 / 10000 [ 43%] (Warmup)
## Chain 3 Iteration: 4500 / 10000 [ 45%] (Warmup)

```

```

## Chain 1 Iteration: 4400 / 10000 [ 44%] (Warmup)
## Chain 4 Iteration: 4200 / 10000 [ 42%] (Warmup)
## Chain 2 Iteration: 4400 / 10000 [ 44%] (Warmup)
## Chain 3 Iteration: 4600 / 10000 [ 46%] (Warmup)
## Chain 1 Iteration: 4500 / 10000 [ 45%] (Warmup)
## Chain 4 Iteration: 4300 / 10000 [ 43%] (Warmup)
## Chain 2 Iteration: 4500 / 10000 [ 45%] (Warmup)
## Chain 1 Iteration: 4600 / 10000 [ 46%] (Warmup)
## Chain 3 Iteration: 4700 / 10000 [ 47%] (Warmup)
## Chain 4 Iteration: 4400 / 10000 [ 44%] (Warmup)
## Chain 1 Iteration: 4700 / 10000 [ 47%] (Warmup)
## Chain 2 Iteration: 4600 / 10000 [ 46%] (Warmup)
## Chain 3 Iteration: 4800 / 10000 [ 48%] (Warmup)
## Chain 4 Iteration: 4500 / 10000 [ 45%] (Warmup)
## Chain 2 Iteration: 4700 / 10000 [ 47%] (Warmup)
## Chain 4 Iteration: 4600 / 10000 [ 46%] (Warmup)
## Chain 1 Iteration: 4800 / 10000 [ 48%] (Warmup)
## Chain 3 Iteration: 4900 / 10000 [ 49%] (Warmup)
## Chain 2 Iteration: 4800 / 10000 [ 48%] (Warmup)
## Chain 4 Iteration: 4700 / 10000 [ 47%] (Warmup)
## Chain 1 Iteration: 4900 / 10000 [ 49%] (Warmup)
## Chain 3 Iteration: 5000 / 10000 [ 50%] (Warmup)
## Chain 3 Iteration: 5001 / 10000 [ 50%] (Sampling)
## Chain 2 Iteration: 4900 / 10000 [ 49%] (Warmup)
## Chain 4 Iteration: 4800 / 10000 [ 48%] (Warmup)
## Chain 1 Iteration: 5000 / 10000 [ 50%] (Warmup)
## Chain 1 Iteration: 5001 / 10000 [ 50%] (Sampling)
## Chain 3 Iteration: 5100 / 10000 [ 51%] (Sampling)
## Chain 4 Iteration: 4900 / 10000 [ 49%] (Warmup)
## Chain 1 Iteration: 5100 / 10000 [ 51%] (Sampling)
## Chain 2 Iteration: 5000 / 10000 [ 50%] (Warmup)
## Chain 2 Iteration: 5001 / 10000 [ 50%] (Sampling)
## Chain 3 Iteration: 5200 / 10000 [ 52%] (Sampling)
## Chain 4 Iteration: 5000 / 10000 [ 50%] (Warmup)
## Chain 4 Iteration: 5001 / 10000 [ 50%] (Sampling)
## Chain 1 Iteration: 5200 / 10000 [ 52%] (Sampling)
## Chain 2 Iteration: 5100 / 10000 [ 51%] (Sampling)
## Chain 3 Iteration: 5300 / 10000 [ 53%] (Sampling)
## Chain 4 Iteration: 5100 / 10000 [ 51%] (Sampling)
## Chain 1 Iteration: 5300 / 10000 [ 53%] (Sampling)
## Chain 2 Iteration: 5200 / 10000 [ 52%] (Sampling)
## Chain 1 Iteration: 5400 / 10000 [ 54%] (Sampling)
## Chain 3 Iteration: 5400 / 10000 [ 54%] (Sampling)
## Chain 4 Iteration: 5200 / 10000 [ 52%] (Sampling)
## Chain 4 Iteration: 5300 / 10000 [ 53%] (Sampling)
## Chain 1 Iteration: 5500 / 10000 [ 55%] (Sampling)
## Chain 2 Iteration: 5300 / 10000 [ 53%] (Sampling)
## Chain 3 Iteration: 5500 / 10000 [ 55%] (Sampling)
## Chain 4 Iteration: 5400 / 10000 [ 54%] (Sampling)
## Chain 3 Iteration: 5600 / 10000 [ 56%] (Sampling)
## Chain 1 Iteration: 5600 / 10000 [ 56%] (Sampling)
## Chain 2 Iteration: 5400 / 10000 [ 54%] (Sampling)
## Chain 4 Iteration: 5500 / 10000 [ 55%] (Sampling)
## Chain 1 Iteration: 5700 / 10000 [ 57%] (Sampling)

```

## Chain 3 Iteration: 5700 / 10000 [ 57%] (Sampling)  
## Chain 2 Iteration: 5500 / 10000 [ 55%] (Sampling)  
## Chain 4 Iteration: 5600 / 10000 [ 56%] (Sampling)  
## Chain 1 Iteration: 5800 / 10000 [ 58%] (Sampling)  
## Chain 3 Iteration: 5800 / 10000 [ 58%] (Sampling)  
## Chain 4 Iteration: 5700 / 10000 [ 57%] (Sampling)  
## Chain 2 Iteration: 5600 / 10000 [ 56%] (Sampling)  
## Chain 1 Iteration: 5900 / 10000 [ 59%] (Sampling)  
## Chain 4 Iteration: 5800 / 10000 [ 58%] (Sampling)  
## Chain 3 Iteration: 5900 / 10000 [ 59%] (Sampling)  
## Chain 1 Iteration: 6000 / 10000 [ 60%] (Sampling)  
## Chain 2 Iteration: 5700 / 10000 [ 57%] (Sampling)  
## Chain 4 Iteration: 5900 / 10000 [ 59%] (Sampling)  
## Chain 3 Iteration: 6000 / 10000 [ 60%] (Sampling)  
## Chain 2 Iteration: 5800 / 10000 [ 58%] (Sampling)  
## Chain 1 Iteration: 6100 / 10000 [ 61%] (Sampling)  
## Chain 4 Iteration: 6000 / 10000 [ 60%] (Sampling)  
## Chain 3 Iteration: 6100 / 10000 [ 61%] (Sampling)  
## Chain 2 Iteration: 5900 / 10000 [ 59%] (Sampling)  
## Chain 4 Iteration: 6100 / 10000 [ 61%] (Sampling)  
## Chain 1 Iteration: 6200 / 10000 [ 62%] (Sampling)  
## Chain 3 Iteration: 6200 / 10000 [ 62%] (Sampling)  
## Chain 4 Iteration: 6200 / 10000 [ 62%] (Sampling)  
## Chain 1 Iteration: 6300 / 10000 [ 63%] (Sampling)  
## Chain 2 Iteration: 6000 / 10000 [ 60%] (Sampling)  
## Chain 3 Iteration: 6300 / 10000 [ 63%] (Sampling)  
## Chain 1 Iteration: 6400 / 10000 [ 64%] (Sampling)  
## Chain 4 Iteration: 6300 / 10000 [ 63%] (Sampling)  
## Chain 2 Iteration: 6100 / 10000 [ 61%] (Sampling)  
## Chain 3 Iteration: 6400 / 10000 [ 64%] (Sampling)  
## Chain 4 Iteration: 6400 / 10000 [ 64%] (Sampling)  
## Chain 1 Iteration: 6500 / 10000 [ 65%] (Sampling)  
## Chain 1 Iteration: 6600 / 10000 [ 66%] (Sampling)  
## Chain 2 Iteration: 6200 / 10000 [ 62%] (Sampling)  
## Chain 3 Iteration: 6500 / 10000 [ 65%] (Sampling)  
## Chain 4 Iteration: 6500 / 10000 [ 65%] (Sampling)  
## Chain 1 Iteration: 6700 / 10000 [ 67%] (Sampling)  
## Chain 2 Iteration: 6300 / 10000 [ 63%] (Sampling)  
## Chain 3 Iteration: 6600 / 10000 [ 66%] (Sampling)  
## Chain 4 Iteration: 6600 / 10000 [ 66%] (Sampling)  
## Chain 4 Iteration: 6700 / 10000 [ 67%] (Sampling)  
## Chain 1 Iteration: 6800 / 10000 [ 68%] (Sampling)  
## Chain 2 Iteration: 6400 / 10000 [ 64%] (Sampling)  
## Chain 3 Iteration: 6700 / 10000 [ 67%] (Sampling)  
## Chain 4 Iteration: 6800 / 10000 [ 68%] (Sampling)  
## Chain 1 Iteration: 6900 / 10000 [ 69%] (Sampling)  
## Chain 3 Iteration: 6800 / 10000 [ 68%] (Sampling)  
## Chain 4 Iteration: 6900 / 10000 [ 69%] (Sampling)  
## Chain 2 Iteration: 6500 / 10000 [ 65%] (Sampling)  
## Chain 3 Iteration: 6900 / 10000 [ 69%] (Sampling)  
## Chain 1 Iteration: 7000 / 10000 [ 70%] (Sampling)  
## Chain 4 Iteration: 7000 / 10000 [ 70%] (Sampling)  
## Chain 2 Iteration: 6600 / 10000 [ 66%] (Sampling)  
## Chain 1 Iteration: 7100 / 10000 [ 71%] (Sampling)

## Chain 3 Iteration: 7000 / 10000 [ 70%] (Sampling)  
## Chain 4 Iteration: 7100 / 10000 [ 71%] (Sampling)  
## Chain 1 Iteration: 7200 / 10000 [ 72%] (Sampling)  
## Chain 2 Iteration: 6700 / 10000 [ 67%] (Sampling)  
## Chain 3 Iteration: 7100 / 10000 [ 71%] (Sampling)  
## Chain 4 Iteration: 7200 / 10000 [ 72%] (Sampling)  
## Chain 1 Iteration: 7300 / 10000 [ 73%] (Sampling)  
## Chain 4 Iteration: 7300 / 10000 [ 73%] (Sampling)  
## Chain 2 Iteration: 6800 / 10000 [ 68%] (Sampling)  
## Chain 3 Iteration: 7200 / 10000 [ 72%] (Sampling)  
## Chain 4 Iteration: 7400 / 10000 [ 74%] (Sampling)  
## Chain 1 Iteration: 7400 / 10000 [ 74%] (Sampling)  
## Chain 2 Iteration: 6900 / 10000 [ 69%] (Sampling)  
## Chain 1 Iteration: 7500 / 10000 [ 75%] (Sampling)  
## Chain 3 Iteration: 7300 / 10000 [ 73%] (Sampling)  
## Chain 4 Iteration: 7500 / 10000 [ 75%] (Sampling)  
## Chain 2 Iteration: 7000 / 10000 [ 70%] (Sampling)  
## Chain 1 Iteration: 7600 / 10000 [ 76%] (Sampling)  
## Chain 4 Iteration: 7600 / 10000 [ 76%] (Sampling)  
## Chain 3 Iteration: 7400 / 10000 [ 74%] (Sampling)  
## Chain 2 Iteration: 7100 / 10000 [ 71%] (Sampling)  
## Chain 4 Iteration: 7700 / 10000 [ 77%] (Sampling)  
## Chain 1 Iteration: 7700 / 10000 [ 77%] (Sampling)  
## Chain 3 Iteration: 7500 / 10000 [ 75%] (Sampling)  
## Chain 1 Iteration: 7800 / 10000 [ 78%] (Sampling)  
## Chain 4 Iteration: 7800 / 10000 [ 78%] (Sampling)  
## Chain 2 Iteration: 7200 / 10000 [ 72%] (Sampling)  
## Chain 3 Iteration: 7600 / 10000 [ 76%] (Sampling)  
## Chain 4 Iteration: 7900 / 10000 [ 79%] (Sampling)  
## Chain 1 Iteration: 7900 / 10000 [ 79%] (Sampling)  
## Chain 2 Iteration: 7300 / 10000 [ 73%] (Sampling)  
## Chain 3 Iteration: 7700 / 10000 [ 77%] (Sampling)  
## Chain 4 Iteration: 8000 / 10000 [ 80%] (Sampling)  
## Chain 1 Iteration: 8000 / 10000 [ 80%] (Sampling)  
## Chain 2 Iteration: 7400 / 10000 [ 74%] (Sampling)  
## Chain 4 Iteration: 8100 / 10000 [ 81%] (Sampling)  
## Chain 1 Iteration: 8100 / 10000 [ 81%] (Sampling)  
## Chain 3 Iteration: 7800 / 10000 [ 78%] (Sampling)  
## Chain 2 Iteration: 7500 / 10000 [ 75%] (Sampling)  
## Chain 4 Iteration: 8200 / 10000 [ 82%] (Sampling)  
## Chain 1 Iteration: 8200 / 10000 [ 82%] (Sampling)  
## Chain 3 Iteration: 7900 / 10000 [ 79%] (Sampling)  
## Chain 4 Iteration: 8300 / 10000 [ 83%] (Sampling)  
## Chain 1 Iteration: 8300 / 10000 [ 83%] (Sampling)  
## Chain 2 Iteration: 7600 / 10000 [ 76%] (Sampling)  
## Chain 3 Iteration: 8000 / 10000 [ 80%] (Sampling)  
## Chain 4 Iteration: 8400 / 10000 [ 84%] (Sampling)  
## Chain 1 Iteration: 8400 / 10000 [ 84%] (Sampling)  
## Chain 2 Iteration: 7700 / 10000 [ 77%] (Sampling)  
## Chain 4 Iteration: 8500 / 10000 [ 85%] (Sampling)  
## Chain 3 Iteration: 8100 / 10000 [ 81%] (Sampling)  
## Chain 1 Iteration: 8500 / 10000 [ 85%] (Sampling)  
## Chain 2 Iteration: 7800 / 10000 [ 78%] (Sampling)  
## Chain 4 Iteration: 8600 / 10000 [ 86%] (Sampling)



```

## Chain 3 Iteration: 8200 / 10000 [ 82%] (Sampling)
## Chain 1 Iteration: 8600 / 10000 [ 86%] (Sampling)
## Chain 2 Iteration: 7900 / 10000 [ 79%] (Sampling)
## Chain 4 Iteration: 8700 / 10000 [ 87%] (Sampling)
## Chain 1 Iteration: 8700 / 10000 [ 87%] (Sampling)
## Chain 3 Iteration: 8300 / 10000 [ 83%] (Sampling)
## Chain 2 Iteration: 8000 / 10000 [ 80%] (Sampling)
## Chain 4 Iteration: 8800 / 10000 [ 88%] (Sampling)
## Chain 1 Iteration: 8800 / 10000 [ 88%] (Sampling)
## Chain 3 Iteration: 8400 / 10000 [ 84%] (Sampling)
## Chain 4 Iteration: 8900 / 10000 [ 89%] (Sampling)
## Chain 2 Iteration: 8100 / 10000 [ 81%] (Sampling)
## Chain 1 Iteration: 8900 / 10000 [ 89%] (Sampling)
## Chain 4 Iteration: 9000 / 10000 [ 90%] (Sampling)
## Chain 2 Iteration: 8200 / 10000 [ 82%] (Sampling)
## Chain 3 Iteration: 8500 / 10000 [ 85%] (Sampling)
## Chain 1 Iteration: 9000 / 10000 [ 90%] (Sampling)
## Chain 4 Iteration: 9100 / 10000 [ 91%] (Sampling)
## Chain 2 Iteration: 8300 / 10000 [ 83%] (Sampling)
## Chain 3 Iteration: 8600 / 10000 [ 86%] (Sampling)
## Chain 1 Iteration: 9100 / 10000 [ 91%] (Sampling)
## Chain 4 Iteration: 9200 / 10000 [ 92%] (Sampling)
## Chain 2 Iteration: 8400 / 10000 [ 84%] (Sampling)
## Chain 3 Iteration: 8700 / 10000 [ 87%] (Sampling)
## Chain 1 Iteration: 9200 / 10000 [ 92%] (Sampling)
## Chain 4 Iteration: 9300 / 10000 [ 93%] (Sampling)
## Chain 1 Iteration: 9300 / 10000 [ 93%] (Sampling)
## Chain 2 Iteration: 8500 / 10000 [ 85%] (Sampling)
## Chain 3 Iteration: 8800 / 10000 [ 88%] (Sampling)
## Chain 4 Iteration: 9400 / 10000 [ 94%] (Sampling)
## Chain 4 Iteration: 9500 / 10000 [ 95%] (Sampling)
## Chain 1 Iteration: 9400 / 10000 [ 94%] (Sampling)
## Chain 3 Iteration: 8900 / 10000 [ 89%] (Sampling)
## Chain 2 Iteration: 8600 / 10000 [ 86%] (Sampling)
## Chain 4 Iteration: 9600 / 10000 [ 96%] (Sampling)
## Chain 1 Iteration: 9500 / 10000 [ 95%] (Sampling)
## Chain 3 Iteration: 9000 / 10000 [ 90%] (Sampling)
## Chain 2 Iteration: 8700 / 10000 [ 87%] (Sampling)
## Chain 4 Iteration: 9700 / 10000 [ 97%] (Sampling)
## Chain 1 Iteration: 9600 / 10000 [ 96%] (Sampling)
## Chain 3 Iteration: 9100 / 10000 [ 91%] (Sampling)
## Chain 4 Iteration: 9800 / 10000 [ 98%] (Sampling)
## Chain 2 Iteration: 8800 / 10000 [ 88%] (Sampling)
## Chain 1 Iteration: 9700 / 10000 [ 97%] (Sampling)
## Chain 3 Iteration: 9200 / 10000 [ 92%] (Sampling)
## Chain 4 Iteration: 9900 / 10000 [ 99%] (Sampling)
## Chain 2 Iteration: 8900 / 10000 [ 89%] (Sampling)
## Chain 1 Iteration: 9800 / 10000 [ 98%] (Sampling)
## Chain 3 Iteration: 9300 / 10000 [ 93%] (Sampling)
## Chain 4 Iteration: 10000 / 10000 [100%] (Sampling)
## Chain 4 finished in 27.2 seconds.
## Chain 2 Iteration: 9000 / 10000 [ 90%] (Sampling)
## Chain 1 Iteration: 9900 / 10000 [ 99%] (Sampling)
## Chain 2 Iteration: 9100 / 10000 [ 91%] (Sampling)

```

```
## Chain 3 Iteration: 9400 / 10000 [ 94%] (Sampling)
## Chain 1 Iteration: 10000 / 10000 [100%] (Sampling)
## Chain 1 finished in 27.7 seconds.
## Chain 2 Iteration: 9200 / 10000 [ 92%] (Sampling)
## Chain 3 Iteration: 9500 / 10000 [ 95%] (Sampling)
## Chain 2 Iteration: 9300 / 10000 [ 93%] (Sampling)
## Chain 3 Iteration: 9600 / 10000 [ 96%] (Sampling)
## Chain 3 Iteration: 9700 / 10000 [ 97%] (Sampling)
## Chain 2 Iteration: 9400 / 10000 [ 94%] (Sampling)
## Chain 3 Iteration: 9800 / 10000 [ 98%] (Sampling)
## Chain 2 Iteration: 9500 / 10000 [ 95%] (Sampling)
## Chain 3 Iteration: 9900 / 10000 [ 99%] (Sampling)
## Chain 2 Iteration: 9600 / 10000 [ 96%] (Sampling)
## Chain 3 Iteration: 10000 / 10000 [100%] (Sampling)
## Chain 3 finished in 29.0 seconds.
## Chain 2 Iteration: 9700 / 10000 [ 97%] (Sampling)
## Chain 2 Iteration: 9800 / 10000 [ 98%] (Sampling)
## Chain 2 Iteration: 9900 / 10000 [ 99%] (Sampling)
## Chain 2 Iteration: 10000 / 10000 [100%] (Sampling)
## Chain 2 finished in 29.6 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 28.4 seconds.
## Total execution time: 29.8 seconds.
```

```
# Print parameter estimate information for fit model
precis(m, prob = 0.95)
```

```
##           mean          sd      2.5%    97.5%    n_eff    Rhat4
## sigma_p 1.114933 0.12762111 0.8902280 1.392680 4650.092 1.0005290
## v       3.521057 0.42257984 2.7755287 4.419436 16742.100 1.0001565
## a       1.534538 0.17455886 1.1924290 1.875954 4769.161 1.0002530
## b       1.015475 0.13087637 0.7598671 1.273652 21932.120 1.0000222
## sigma   1.431725 0.06572531 1.3026598 1.561391 15784.588 0.9999592
```

## Posterior Probability

We are interested in the posterior value of b, as this is the estimator for the fixed effect of tailoring. We first extract samples from the fit model.

```
set.seed(4) # For reproducibility
post <- extract.samples(m, 10000)
```

Based on the extracted samples, we now calculate the posterior probability that b is greater than 0.

```
posterior_probability <- (length(post[which(post$b>0)])/length(post$b))
cat("Calculated posterior probability value is ", posterior_probability)
```

```
## Calculated posterior probability value is 1
```

We are interested in the parameter b, which is the estimator for the fixed effect of tailoring. The posterior probability that b is greater than 0 calculated by extracting samples from model is 1, supporting the hypothesis that tailored messages are more motivating than generic messages. The posterior probability can be interpreted according to Chechile's guidelines (Chechile 2020) as Virtually certain that  $b > 0$ .

## Calculate 95%-HPDI

In the paper we also report the 95%-HPDI for b. Let's reproduce this.

```
hpdi_lower = HPDI(post$b, prob = 0.95)[1]
hpdi_higher = HPDI(post$b, prob = 0.95)[2]

paste("95%-HPDI: [", round(hpdi_lower, 2), ",", round(hpdi_higher, 2), "]")

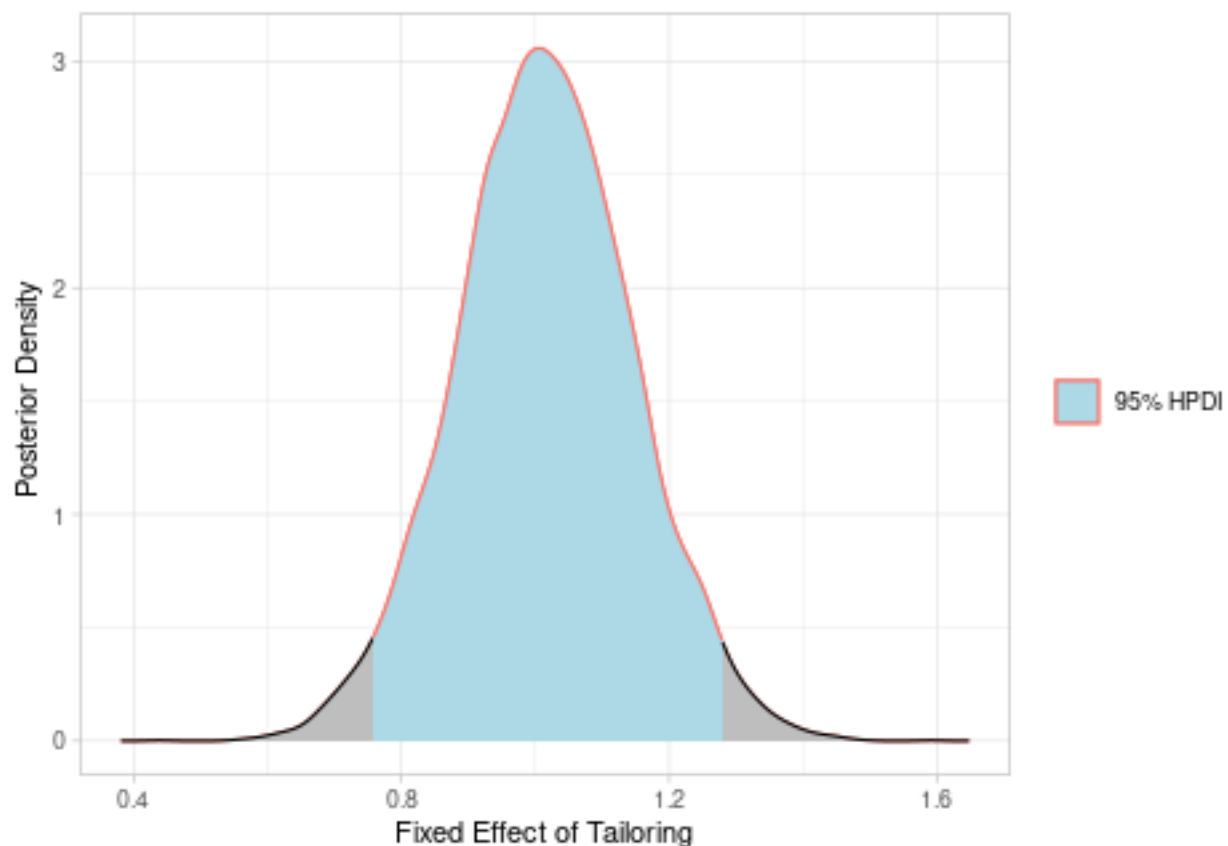
## [1] "95%-HPDI: [ 0.76 , 1.28 ]"
```

## Plot Posterior Density

Let's plot the posterior density for the fixed effect of tailoring. This figure corresponds to Figure 3b in the paper.

```
post_df = data.frame(post)
dens_df <- data.frame(x = as.numeric(density(post$b)$x), y = as.numeric(density(post$b)$y))

ggplot(post_df, aes(b, color = "95% HPDI")) + geom_density() + geom_area(data = subset(dens_df,
  x >= hpdi_lower & x <= hpdi_higher), aes(x = x, y = y), fill = "lightblue") +
  geom_area(data = subset(dens_df, x < hpdi_lower), aes(x = x, y = y),
    fill = "grey", color = "black") + geom_area(data = subset(dens_df,
  x > hpdi_higher), aes(x = x, y = y), fill = "grey", color = "black") +
  labs(x = "Fixed Effect of Tailoring", y = "Posterior Density") + theme_light() +
  theme(legend.title = element_blank())
```



```
# Save image  
pdf_file <- "Figures/post_fixed_effect_tailoring.pdf"  
ggsave(pdf_file, dpi = 1500)
```

```
## Saving 6.5 x 4.5 in image
```

```
knitr::plot_crop(pdf_file)
```

```
## [1] "Figures/post_fixed_effect_tailoring.pdf"
```

## References

Chechile, Richard A. 2020. *Bayesian Statistics for Experimental Scientists*.