Bayesian Analyses

Nele Albers, Beyza Hizli

22 September, 2022

Contents

This file is to reproduce our Bayesian analyses for our four hypotheses from Table 3, as well as the Cronbach's alpha value for the acceptance mentioned in the "Data Preparation and Analysis Strategies"-section.

Authored by Nele Albers, Beyza Hizli, Bouke L. Scheltinga, Eline Meijer, and Willem-Paul Brinkman.

Required and output files

The following files are required: Data/df_se.csv, Data/df_se_plots.csv, Data/df_ratings.csv, and Data/df_acceptance.csv.

And these files are created: Figures/self_efficacy_pre_post_grouped.pdf.

Load packages

First, we load the packages that we need.

```
library(BayesianFirstAid)  # For Bayesian t-test
library(formatR)  # To wrap lines
library(ggplot2)  # For plot
library(ltm)  # For cronbach's alpha
library(pander)  # For table
```

Load data

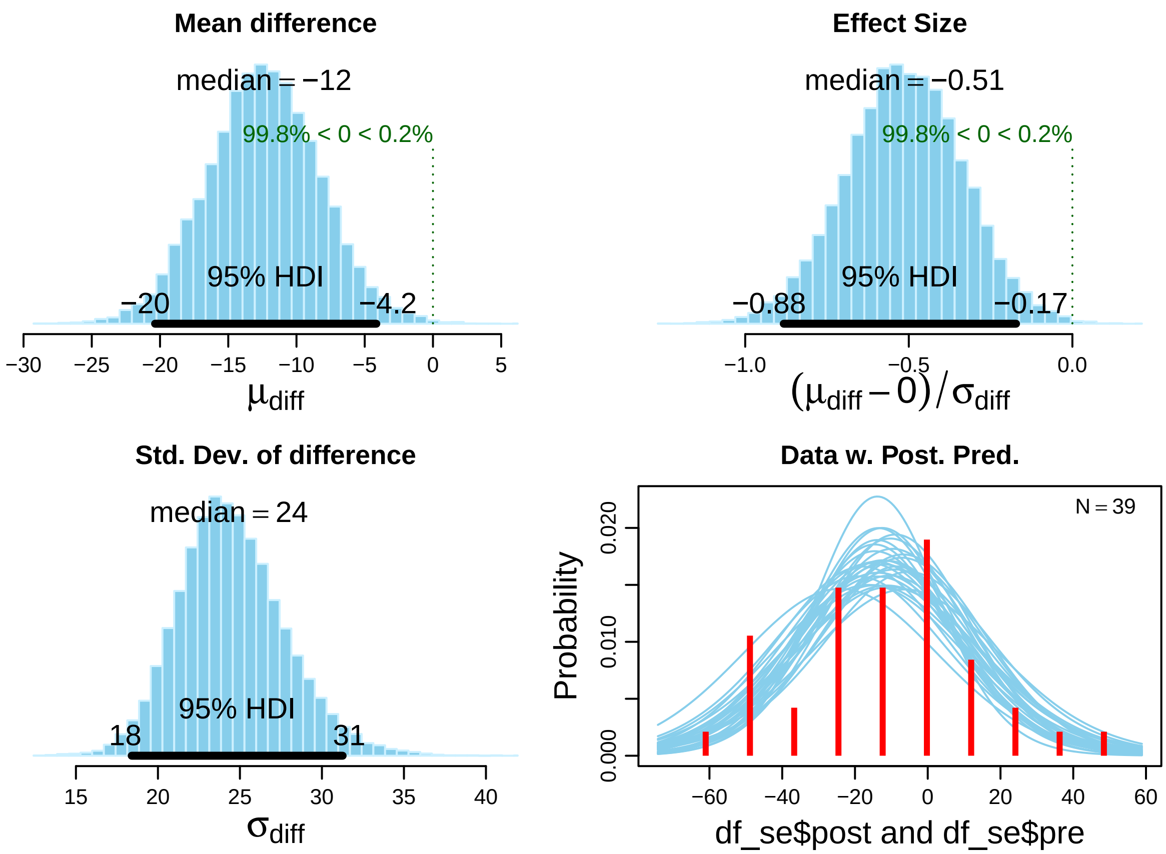
Now we load the pre-processed data.

```
# self-efficacy data
df_se <- read.csv(file = 'Data/df_se.csv')
df_se_plots <- read.csv(file = 'Data/df_se_plots.csv')
# motivation ratings data
df_ratings <- read.csv(file = 'Data/df_ratings.csv')
# acceptance questions data
df_acceptance <- read.csv(file = 'Data/df_acceptance.csv')
```

Hypothesis 1

Now we conduct a Bayesian t-test for the hypothesis H1 that people's self-efficacy is higher after the dialog with the virtual coach than before.

```
set.seed(22)  # For reproducibility

fit1 <- bayes.t.test(df_se$post, df_se$pre, paired = TRUE)
plot(fit1)
```




```
summary(fit1)
```

```
##   Data
## df_se$post, n = 39
## df_se$pre, n = 39
##
```

```
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between df_se$post and df_se$pre
## sigma_diff: the scale of the pairwise difference, a consistent
##   estimate of SD when nu is large.
## nu: the degrees-of-freedom for the t distribution fitted to the pairwise difference
## eff_size: the effect size calculated as (mu_diff - 0) / sigma_diff
## diff_pred: predicted distribution for a new datapoint generated
##   as the pairwise difference between df_se$post and df_se$pre
##
##   Measures
##                mean     sd   HDIlo   HDIup %<comp %>comp
## mu_diff    -12.385  4.129 -20.362 -4.151  0.998  0.002
## sigma_diff  24.541  3.277  18.401 31.261  0.000  1.000
## nu          34.275 28.845   2.122 92.432  0.000  1.000
## eff_size    -0.513  0.181  -0.882 -0.172  0.998  0.002
## diff_pred  -12.279 26.720 -65.945 39.829  0.687  0.313
##
## 'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
## '%<comp' and '%>comp' are the probabilities of the respective parameter being
## smaller or larger than 0.
##
##   Quantiles
##              q2.5%    q25%  median    q75%   q97.5%
## mu_diff    -20.408 -15.110 -12.392 -9.635  -4.188
## sigma_diff  18.642  22.315  24.324 26.562  31.619
## nu           4.320  13.882  26.065 45.814 111.287
## eff_size    -0.873  -0.632  -0.513 -0.390  -0.160
## diff_pred  -65.128 -29.053 -12.383  4.760  40.879
```

```r
# Mean, SD, CI
H1_mean <- fit1$stats[1, 1]
H1_SD <- fit1$stats[2, 1]
H1_ci_low <- fit1$stats[1, 5]
H1_ci_high <- fit1$stats[1, 6]

# Posterior probability that H1 is true
H1_post_p <- fit1$stats[1, 7]

print(paste("Posterior probability that H1 is true:", round(H1_post_p, 3)))
```

```
## [1] "Posterior probability that H1 is true: 0.002"
```

This posterior probability can be evaluated based on the guidelines from (Chechile (2020)) and their extension to posterior probabilities below 0.5 by (Andraszewicz et al. (2015)).

```r
if (H1_post_p < 0.0005){
  evaluation_H1 = "Nearing certainty against"
}else if (H1_post_p < 0.005){
  evaluation_H1 = "Very strong bet against"
}else if (H1_post_p < 0.01){
  evaluation_H1 = "Strong bet against - irresponsible to avoid"
}else if (H1_post_p < 0.1){
  evaluation_H1 = "A promising but risky bet against"
}else if (H1_post_p < 0.25){
  evaluation_H1 = "Only a casual bet against"
}else if (H1_post_p < 0.5){
```

```r
  evaluation_H1 = "Not worth betting against"
}else if (H1_post_p < 0.75){
  evaluation_H1 = "Not worth betting on"
}else if (H1_post_p < 0.9){
  evaluation_H1 = "Only a casual bet"
}else if (H1_post_p < 0.95){
  evaluation_H1 = "A promising but risky bet"
}else if (H1_post_p < 0.99){
  evaluation_H1 = "Good bet - too good to disregard"
}else if (H1_post_p < 0.995){
  evaluation_H1 = "Strong bet - irresponsible to avoid"
}else if (H1_post_p < 0.9995){
  evaluation_H1 = "Very strong bet"
}else if (H1_post_p < 0.99995){
  evaluation_H1 = "Nearing certainty"
}else{
  evaluation_H1 = "Virtually certain"
}

evaluation_H1
```

```
## [1] "Very strong bet against"
```

Hypothesis 2

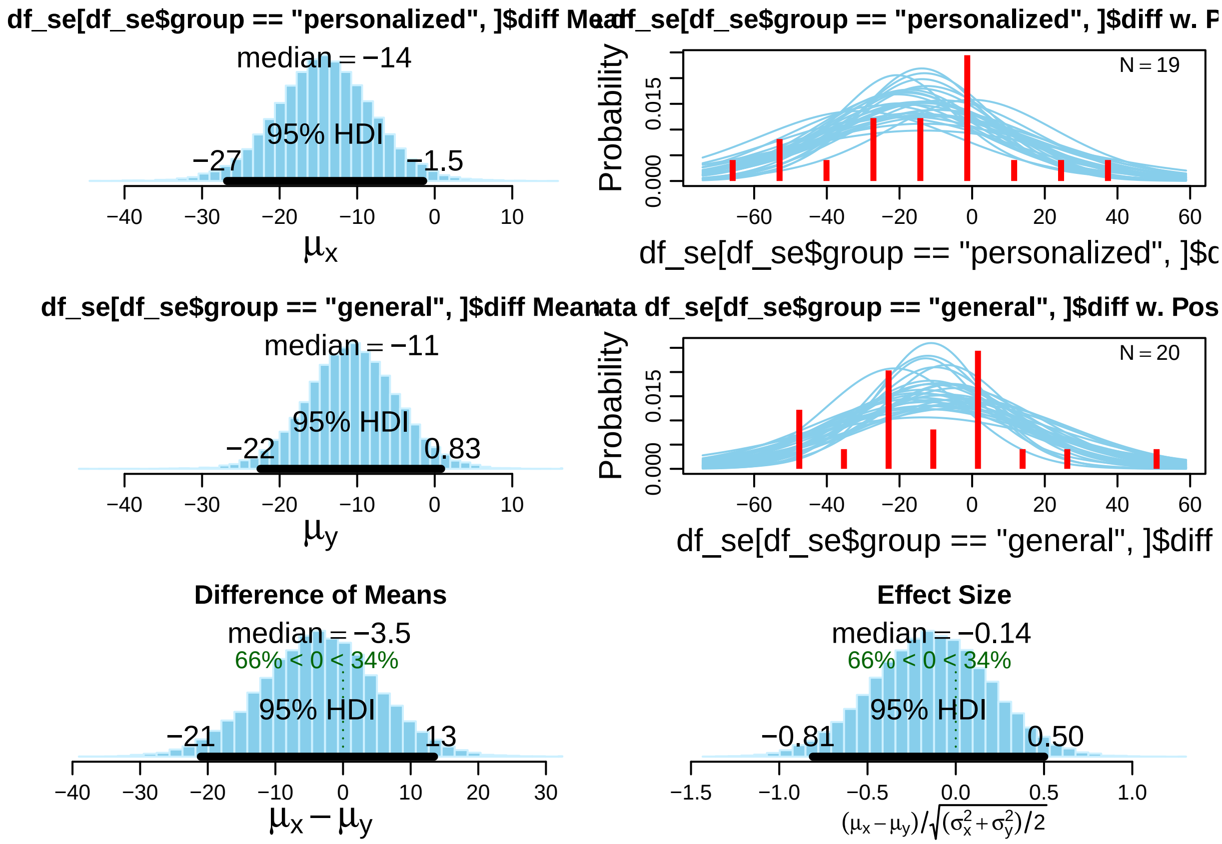
Here we conduct a Bayesian t-test for the hypothesis H2 that people's self-efficacy is higher when receiving personalized examples than when receiving generic examples. For this, we look at the change in self-efficacy between the pre- and post-measurement.

```r
set.seed(22)  # For reproducibility

fit2 <- bayes.t.test(df_se[df_se$group == "personalized", ]$diff, df_se[df_se$group ==
    "general", ]$diff)
summary(fit2)
```

```
##   Data
## df_se[df_se$group == "personalized", ]$diff, n = 19
## df_se[df_se$group == "general", ]$diff, n = 20
##
##   Model parameters and generated quantities
## mu_x: the mean of df_se[df_se$group == "personalized", ]$diff
## sigma_x: the scale of df_se[df_se$group == "personalized", ]$diff , a consistent
##   estimate of SD when nu is large.
## mu_y: the mean of df_se[df_se$group == "general", ]$diff
## sigma_y: the scale of df_se[df_se$group == "general", ]$diff
## mu_diff: the difference in means (mu_x - mu_y)
## sigma_diff: the difference in scale (sigma_x - sigma_y)
## nu: the degrees-of-freedom for the t distribution
##   fitted to df_se[df_se$group == "personalized", ]$diff and df_se[df_se$group == "general", ]$diff
## eff_size: the effect size calculated as
##   (mu_x - mu_y) / sqrt((sigma_x^2 + sigma_y^2) / 2)
## x_pred: predicted distribution for a new datapoint
##   generated as df_se[df_se$group == "personalized", ]$diff
## y_pred: predicted distribution for a new datapoint
```

```
##   generated as df_se[df_se$group == "general", ]$diff
##
##   Measures
##                mean     sd   HDIlo   HDIup %<comp %>comp
## mu_x       -14.269  6.404 -26.782 -1.479  0.986  0.014
## sigma_x     26.485  5.308  17.010 36.928  0.000  1.000
## mu_y       -10.701  5.918 -22.466  0.832  0.965  0.035
## sigma_y     25.127  4.778  16.592 34.776  0.000  1.000
## mu_diff     -3.568  8.727 -21.033 13.456  0.660  0.340
## sigma_diff   1.359  6.987 -12.333 15.417  0.426  0.574
## nu          34.713 29.378   1.714 93.146  0.000  1.000
## eff_size    -0.139  0.334  -0.810  0.501  0.660  0.340
## x_pred     -14.308 29.862 -71.893 45.815  0.700  0.300
## y_pred     -10.798 28.246 -66.747 45.537  0.665  0.335
##
## 'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
## '%<comp' and '%>comp' are the probabilities of the respective parameter being
## smaller or larger than 0.
##
##   Quantiles
##               q2.5%    q25%  median    q75%  q97.5%
## mu_x       -26.995 -18.400 -14.256 -10.091  -1.645
## sigma_x     18.008  22.785  25.826  29.499  38.570
## mu_y       -22.401 -14.545 -10.698  -6.825   0.904
## sigma_y     17.408  21.792  24.588  27.851  36.020
## mu_diff    -20.973  -9.261  -3.516   2.217  13.524
## sigma_diff -12.225  -3.109   1.211   5.694  15.576
## nu           4.374  13.811  26.299  46.402 112.881
## eff_size    -0.804  -0.364  -0.138   0.086   0.510
## x_pred     -73.203 -32.487 -14.445   4.029  44.744
## y_pred     -67.158 -28.173 -10.599   6.461  45.247
```

```
plot(fit2)
```




```r
# Mean, SD, CI
H2_mean <- fit2$stats[5, 1]
H2_SD <- fit2$stats[6, 1]
H2_ci_low <- fit2$stats[5, 5]
H2_ci_high <- fit2$stats[5, 6]

# Posterior probability that H2 is true
H2_post_p <- fit2$stats[5, 7]

print(paste("Posterior probability that H2 is true:", round(H2_post_p,
    2)))
```

```
## [1] "Posterior probability that H2 is true: 0.34"
```

This posterior probability can be evaluated based on the guidelines from (Chechile (2020)) and their extension to posterior probabilities below 0.5 by (Andraszewicz et al. (2015)).

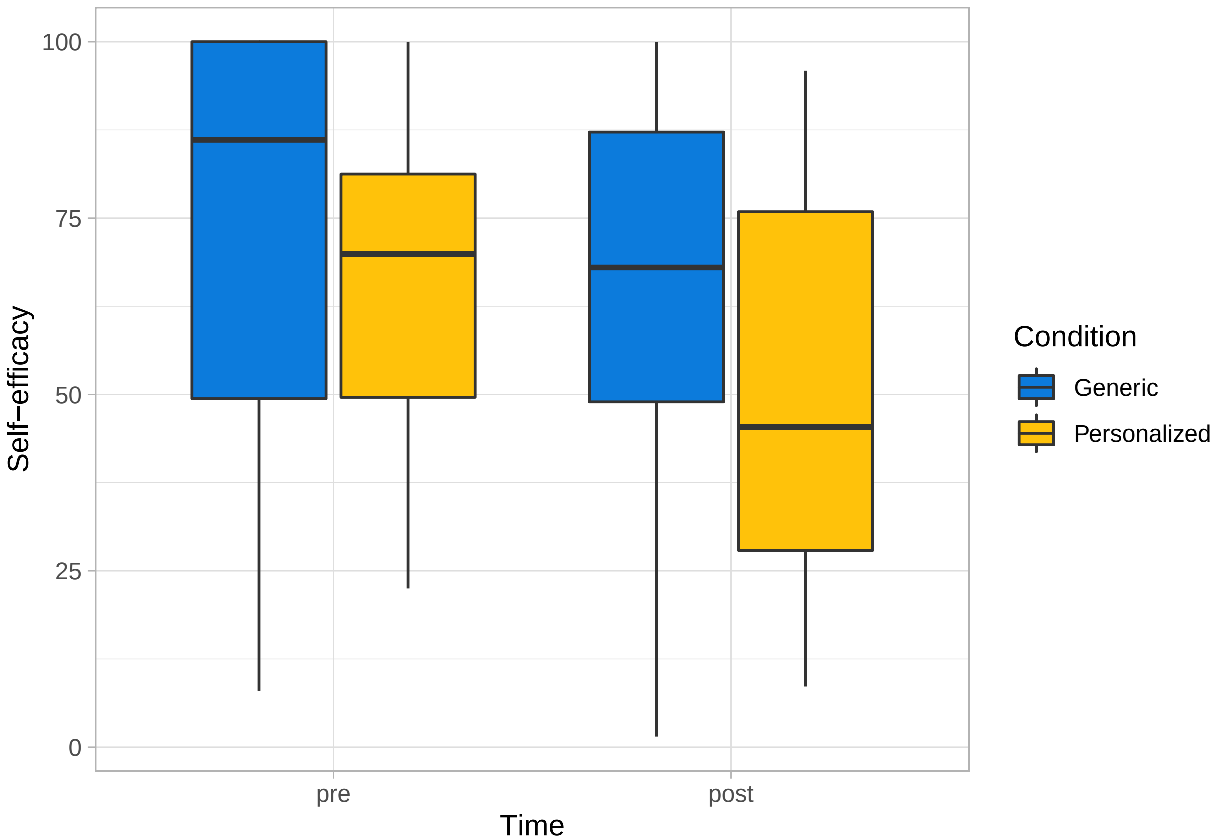
```r
if (H2_post_p < 0.0005){
  evaluation_H2 = "Nearing certainty against"
}else if (H2_post_p < 0.005){
  evaluation_H2 = "Very strong bet against"
}else if (H2_post_p < 0.01){
  evaluation_H2 = "Strong bet against - irresponsible to avoid"
}else if (H2_post_p < 0.1){
  evaluation_H2 = "A promising but risky bet against"
}else if (H2_post_p < 0.25){
  evaluation_H2 = "Only a casual bet against"
}else if (H2_post_p < 0.5){
```

```
  evaluation_H2 = "Not worth betting against"
}else if (H1_post_p < 0.75){
  evaluation_H2 = "Not worth betting on"
}else if (H2_post_p < 0.9){
  evaluation_H2 = "Only a casual bet"
}else if (H2_post_p < 0.95){
  evaluation_H2 = "A promising but risky bet"
}else if (H2_post_p < 0.99){
  evaluation_H2 = "Good bet - too good to disregard"
}else if (H2_post_p < 0.995){
  evaluation_H2 = "Strong bet - irresponsible to avoid"
}else if (H2_post_p < 0.9995){
  evaluation_H2 = "Very strong bet"
}else if (H2_post_p < 0.99995){
  evaluation_H2 = "Nearing certainty"
}else{
  evaluation_H2 = "Virtually certain"
}

evaluation_H2
```

```
## [1] "Not worth betting against"
```

We also plot the self-efficacy before and after the dialog for both conditions.

```
df_se_plots$time <- factor(df_se_plots$time, levels = c("pre", "post"))

ggplot(df_se_plots, aes(x = time, y = se, fill = factor(group))) + geom_boxplot() +
    theme_light() + xlab("Time") + ylab("Self-efficacy") + labs(fill = "Condition") +
    scale_fill_manual(values = c("#0C7BDC", "#FFC20A"), labels = c("Generic",
        "Personalized")) + scale_color_manual(values = c("#0C7BDC", "#FFC20A"),
    labels = c("Generic", "Personalized"))
```




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

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

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

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

Hypothesis 3

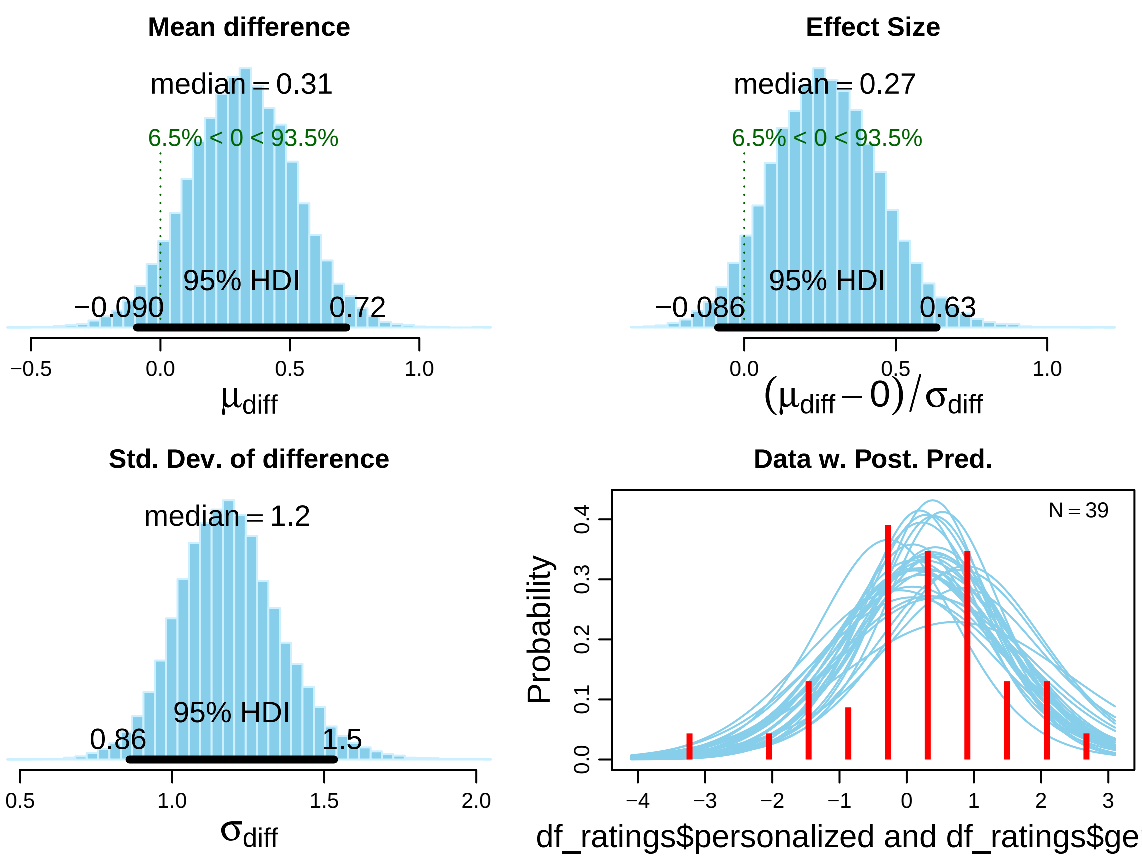
Now we conduct a Bayesian t-test for our third hypothesis that the personalized examples are perceived as more motivating than generic examples.

```r
set.seed(22)  # for reproducibility

fit3 <- bayes.t.test(df_ratings$personalized, df_ratings$general, paired = TRUE)
summary(fit3)
```

```
##   Data
## df_ratings$personalized, n = 39
## df_ratings$general, n = 39
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between df_ratings$personalized and df_ratings$general
## sigma_diff: the scale of the pairwise difference, a consistent
##   estimate of SD when nu is large.
```

```
## nu: the degrees-of-freedom for the t distribution fitted to the pairwise difference
## eff_size: the effect size calculated as (mu_diff - 0) / sigma_diff
## diff_pred: predicted distribution for a new datapoint generated
##   as the pairwise difference between df_ratings$personalized and df_ratings$general
##
##   Measures
##               mean     sd  HDIlo  HDIup %<comp %>comp
## mu_diff      0.312  0.206 -0.090  0.718  0.065  0.935
## sigma_diff   1.189  0.171  0.860  1.532  0.000  1.000
## nu          30.293 27.978  1.488 86.785  0.000  1.000
## eff_size     0.271  0.185 -0.086  0.635  0.065  0.935
## diff_pred    0.303  1.345 -2.322  2.889  0.399  0.601
##
## 'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
## '%<comp' and '%>comp' are the probabilities of the respective parameter being
## smaller or larger than 0.
##
##   Quantiles
##              q2.5%   q25% median    q75%  q97.5%
## mu_diff     -0.096  0.174  0.313  0.452   0.715
## sigma_diff   0.871  1.074  1.181  1.294   1.548
## nu           3.343 10.773 21.719 40.190 106.452
## eff_size    -0.076  0.144  0.266  0.391   0.651
## diff_pred   -2.318 -0.529  0.305  1.126   2.897
```

```
plot(fit3)
```



```r
# Mean, SD, CI
H3_mean <- fit3$stats[1, 1]
H3_SD <- fit3$stats[2, 1]
H3_ci_low <- fit3$stats[1, 5]
H3_ci_high <- fit3$stats[1, 6]

# Posterior probability
H3_post_p <- fit3$stats[1, 7]

print(paste("Posterior probability that H3 is true:", round(H3_post_p,
    2)))
```

```
## [1] "Posterior probability that H3 is true: 0.93"
```

This posterior probability can be evaluated based on the guidelines from (Chechile (2020)) and their extension to posterior probabilities below 0.5 by (Andraszewicz et al. (2015)).

```r
if (H3_post_p < 0.0005){
  evaluation_H3 = "Nearing certainty against"
}else if (H3_post_p < 0.005){
  evaluation_H3 = "Very strong bet against"
}else if (H3_post_p < 0.01){
  evaluation_H3 = "Strong bet against - irresponsible to avoid"
}else if (H3_post_p < 0.1){
  evaluation_H3 = "A promising but risky bet against"
}else if (H3_post_p < 0.25){
  evaluation_H3 = "Only a casual bet against"
}else if (H3_post_p < 0.5){
  evaluation_H3 = "Not worth betting against"
}else if (H3_post_p < 0.75){
  evaluation_H3 = "Not worth betting on"
}else if (H3_post_p < 0.9){
  evaluation_H3 = "Only a casual bet"
}else if (H3_post_p < 0.95){
  evaluation_H3 = "A promising but risky bet"
}else if (H3_post_p < 0.99){
  evaluation_H3 = "Good bet - too good to disregard"
}else if (H3_post_p < 0.995){
  evaluation_H3 = "Strong bet - irresponsible to avoid"
}else if (H3_post_p < 0.9995){
  evaluation_H3 = "Very strong bet"
}else if (H3_post_p < 0.99995){
  evaluation_H3 = "Nearing certainty"
}else{
  evaluation_H3 = "Virtually certain"
}

evaluation_H3
```

```
## [1] "A promising but risky bet"
```

Hypothesis 4

For our fourth hypothesis, we first compute Cronbach's alpha based on the six acceptance items and then calculate the index measure.

```r
data <- data.frame(df_acceptance$acceptance_1, df_acceptance$acceptance_2,
    df_acceptance$acceptance_3, df_acceptance$acceptance_4, df_acceptance$acceptance_5,
    df_acceptance$acceptance_6)

# calculate Cronbach's Alpha
alpha <- cronbach.alpha(data)

print(paste("Cronbach's alpha:", round(alpha$alpha, 2)))
```

```
## [1] "Cronbach's alpha: 0.87"
```

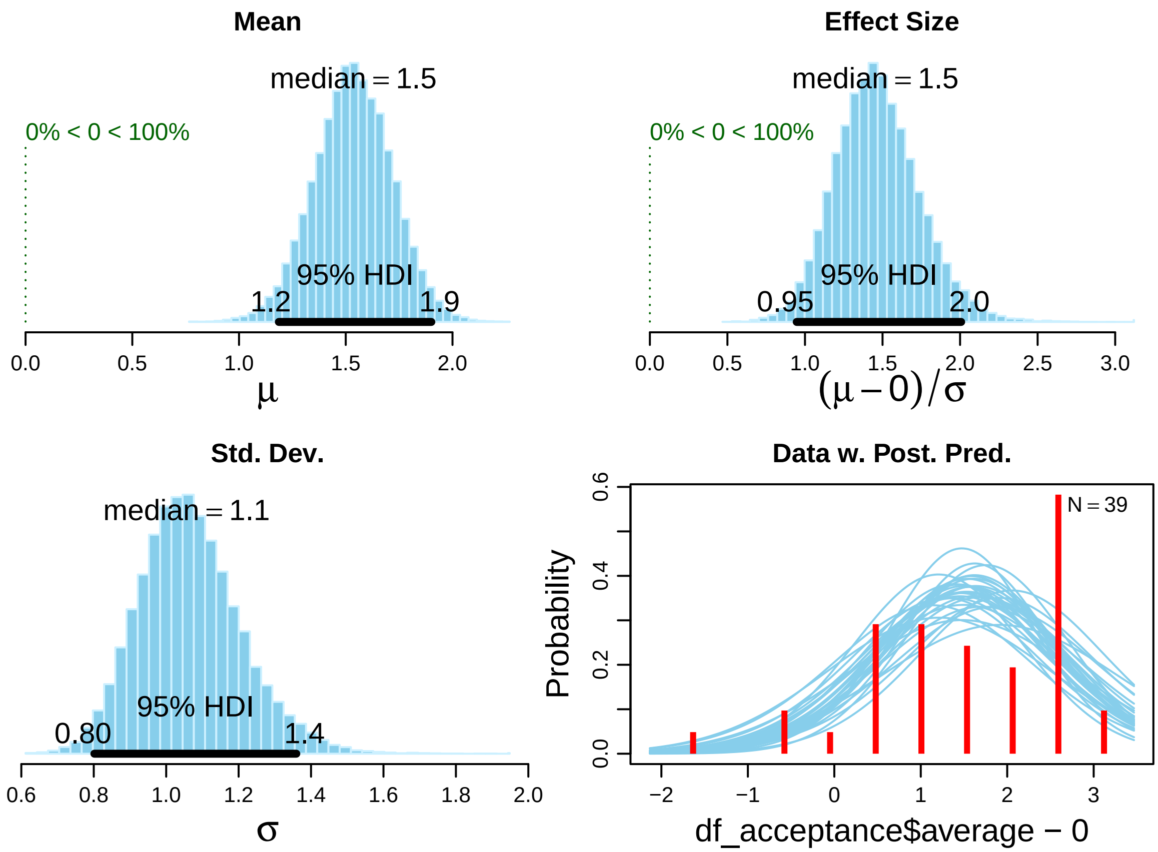
```r
# Now compute index measure as the mean of the 6 items
df_acceptance$average <- (df_acceptance$acceptance_1 + df_acceptance$acceptance_2 +
    df_acceptance$acceptance_3 + df_acceptance$acceptance_4 + df_acceptance$acceptance_5 +
    df_acceptance$acceptance_6)/6
```

Now we conduct a Bayesian t-test for whether the acceptance index measure is greater than 0.

```r
set.seed(22)  # for reproducibility

fit4 <- bayes.t.test(df_acceptance$average - 0)
summary(fit4)
```

```
##   Data
## df_acceptance$average - 0, n = 39
##
##   Model parameters and generated quantities
## mu: the mean of df_acceptance$average - 0
## sigma: the scale of df_acceptance$average - 0 , a consistent
##   estimate of SD when nu is large.
## nu: the degrees-of-freedom for the t distribution fitted to df_acceptance$average - 0
## eff_size: the effect size calculated as (mu - 0) / sigma
## x_pred: predicted distribution for a new datapoint generated as df_acceptance$average - 0
##
##   Measures
##             mean     sd  HDIlo  HDIup %<comp %>comp
## mu         1.535  0.183  1.187  1.901   0.00   1.00
## sigma      1.066  0.142  0.802  1.359   0.00   1.00
## nu        33.320 28.854  1.884 89.967   0.00   1.00
## eff_size   1.468  0.272  0.947  2.007   0.00   1.00
## x_pred     1.528  1.170 -0.873  3.753   0.09   0.91
##
## 'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
## '%<comp' and '%>comp' are the probabilities of the respective parameter being
## smaller or larger than 0.
##
##   Quantiles
##            q2.5%   q25% median   q75%  q97.5%
## mu         1.175  1.415  1.536  1.658   1.892
## sigma      0.815  0.969  1.056  1.152   1.377
## nu         4.477 13.214 24.891 44.404 110.695
## eff_size   0.976  1.282  1.453  1.637   2.040
## x_pred    -0.804  0.785  1.536  2.267   3.831
```

```r
plot(fit4)
```




```r
# Mean, SD, CI
H4_mean <- fit4$stats[1, 1]
H4_SD <- fit4$stats[2, 1]
H4_ci_low <- fit4$stats[1, 5]
H4_ci_high <- fit4$stats[1, 6]

# Posterior probability that H4 is true
H4_post_p <- fit4$stats[1, 7]

print(paste("Posterior probability that H4 is true:", H4_post_p))
```

```
## [1] "Posterior probability that H4 is true: 0.999966668888741"
```

This posterior probability can be evaluated based on the guidelines from (Chechile (2020)) and their extension to posterior probabilities below 0.5 by (Andraszewicz et al. (2015)).

```r
if (H4_post_p < 0.0005){
  evaluation_H4 = "Nearing certainty against"
}else if (H4_post_p < 0.005){
  evaluation_H4 = "Very strong bet against"
}else if (H4_post_p < 0.01){
  evaluation_H4 = "Strong bet against - irresponsible to avoid"
}else if (H4_post_p < 0.1){
  evaluation_H4 = "A promising but risky bet against"
}else if (H4_post_p < 0.25){
  evaluation_H4 = "Only a casual bet against"
```

```r
}else if (H4_post_p < 0.5){
  evaluation_H4 = "Not worth betting against"
}else if (H4_post_p < 0.75){
  evaluation_H4 = "Not worth betting on"
}else if (H4_post_p < 0.9){
  evaluation_H4 = "Only a casual bet"
}else if (H4_post_p < 0.95){
  evaluation_H4 = "A promising but risky bet"
}else if (H4_post_p < 0.99){
  evaluation_H4 = "Good bet - too good to disregard"
}else if (H4_post_p < 0.995){
  evaluation_H4 = "Strong bet - irresponsible to avoid"
}else if (H4_post_p < 0.9995){
  evaluation_H4 = "Very strong bet"
}else if (H4_post_p < 0.99995){
  evaluation_H4 = "Nearing certainty"
}else{
  evaluation_H4 = "Virtually certain"
}

evaluation_H4
```

```
## [1] "Virtually certain"
```

Summary

Below we print a summary, which reproduces Table 3 from the paper.

```r
tab <- rbind(c("H1: SELF-EFFICACY", "", "", "", ""))
tab <- rbind(tab, c("Post - Pre", paste(round(H1_mean, 2), "(", round(H1_SD,
    2), ")"), paste("[", round(H1_ci_low, 2), ",", round(H1_ci_high, 2),
    "]"), round(H1_post_p, 3), evaluation_H1))

tab <- rbind(tab, c("H2: CHANGE IN SELF-EFFICACY", "", "", "", ""))
tab <- rbind(tab, c("Personalized - Generic", paste(round(H2_mean, 2),
    "(", round(H2_SD, 2), ")"), paste("[", round(H2_ci_low, 2), ",", round(H2_ci_high,
    2), "]"), round(H2_post_p, 2), evaluation_H2))

tab <- rbind(tab, c("H3: PERCEIVED MOTIVATIONAL IMPACT", "", "", "", ""))
tab <- rbind(tab, c("Personalized - Generic", paste(round(H3_mean, 2),
    "(", round(H3_SD, 2), ")"), paste("[", round(H3_ci_low, 2), ",", round(H3_ci_high,
    2), "]"), round(H3_post_p, 2), evaluation_H3))

tab <- rbind(tab, c("H4: ACCEPTANCE", "", "", "", ""))
tab <- rbind(tab, c("Mean", paste(round(H4_mean, 2), "(", round(H4_SD,
    2), ")"), paste("[", round(H4_ci_low, 2), ",", round(H4_ci_high, 2),
    "]"), round(H4_post_p, 5), evaluation_H4))

colnames(tab) = c("Parameter", "Mean (SD)", "95% CI", "Post", "Evaluation")

pander(tab, caption = "Results of Bayesian analyses for the four hypotheses.")
```

Table 1: Results of Bayesian analyses for the four hypotheses. (continued below)

Parameter	Mean (SD)	95% CI	Post
H1: SELF-EFFICACY			
Post - Pre	-12.38 (24.54)	[-20.36 , -4.15]	0.002
H2: CHANGE IN SELF-EFFICACY			
Personalized - Generic	-3.57 (1.36)	[-21.03 , 13.46]	0.34
H3: PERCEIVED MOTIVATIONAL IMPACT			
Personalized - Generic	0.31 (1.19)	[-0.09 , 0.72]	0.93
H4: ACCEPTANCE			
Mean	1.54 (1.07)	[1.19 , 1.9]	0.99997

Evaluation

Very strong bet against

Not worth betting against

A promising but risky bet

Virtually certain

References

Andraszewicz, Sandra, Benjamin Scheibehenne, Jörg Rieskamp, Raoul Grasman, Josine Verhagen, and Eric-Jan Wagenmakers. 2015. "An Introduction to Bayesian Hypothesis Testing for Management Research." *Journal of Management* 41 (2): 521–43.

Chechile, Richard A. 2020. *Bayesian Statistics for Experimental Scientists: A General Introduction Using Distribution-Free Methods.* MIT Press.