Effects of individual actions on confidence

Andrei Stefan

09 November, 2023

This file contains the code for the analysis of the effects of individual actions on confidence. Note that, due to the nature of the Bayesian t-tests, results will not always be the same for the means and standard deviations. From the output of the `print` of the result, we use the probability of the mean difference being larger and smaller than 0. From the output of the `summary` of the result, we use the mean and standard deviation values, and their 95% CIs.

```
## Loading required package: rjags

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs
```

Run a Bayesian t-test on the raw data before and after the first change to the plan made, and calculate Cohen's d for these values.

```
# get the values for confidence before making the first change to the plan as a list
c_changes_1_b = as.numeric(t(read.csv(
  "~/analysis/data/c_changes_to_plan_1_before.csv", header = FALSE)))

# get the values for confidence after making the first change to the plan as a list
c_changes_1_a = as.numeric(t(read.csv(
  "~/analysis/data/c_changes_to_plan_1_after.csv", header = FALSE)))

# run the Bayesian t-test
c_changes_to_plan_1 = bayes.t.test(c_changes_1_b, c_changes_1_a, paired = TRUE)

# print the results of the test, and the summary
print(c_changes_to_plan_1)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
##
## data: c_changes_1_b and c_changes_1_a, n = 114
##
##   Estimates [95% credible interval]
## mean paired difference: -4e-06 [-0.00071, 0.00069]
## sd of the paired differences: 0.0020 [0.0014, 0.0077]
##
## The mean difference is more than 0 by a probability of 0.492
## and less than 0 by a probability of 0.508
```

```
summary(c_changes_to_plan_1)
```

```
##   Data
```

```
## c_changes_1_b, n = 114
## c_changes_1_a, n = 114
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_changes_1_b and c_changes_1_a
## 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 c_changes_1_b and c_changes_1_a
##
##   Measures
##               mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff    -0.002 0.029 -0.001 0.001  0.508  0.492
## sigma_diff  0.007 0.049  0.001 0.008  0.000  1.000
## nu          1.012 0.131  1.000 1.011  0.000  1.000
## eff_size   -0.005 0.103 -0.191 0.186  0.508  0.492
## diff_pred   0.006 0.868 -0.042 0.044  0.501  0.499
##
## '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    -0.001  0.000  0.000 0.000  0.001
## sigma_diff  0.001  0.002  0.002 0.003  0.012
## nu          1.000  1.001  1.002 1.005  1.015
## eff_size   -0.195 -0.067 -0.002 0.061  0.182
## diff_pred  -0.045 -0.002  0.000 0.002  0.041
```

```
# calculate Cohen's d
cohen.d(c_changes_1_a, c_changes_1_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: 0.3242204 (small)
## 95 percent confidence interval:
##     lower     upper
## 0.2111578 0.4372830
```

Run a Bayesian t-test on the raw data before and after the second change to the plan made, and calculate Cohen's d for these values.

```
c_changes_2_b = as.numeric(t(read.csv(
  "~/analysis/data/c_changes_to_plan_2_before.csv", header = FALSE)))
c_changes_2_a = as.numeric(t(read.csv(
  "~/analysis/data/c_changes_to_plan_2_after.csv", header = FALSE)))
c_changes_to_plan_2 = bayes.t.test(c_changes_2_b, c_changes_2_a, paired = TRUE)

print(c_changes_to_plan_2)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
```

```
##
## data: c_changes_2_b and c_changes_2_a, n = 59
##
##   Estimates [95% credible interval]
## mean paired difference: -5.6e-07 [-0.00014, 0.00014]
## sd of the paired differences: 0.00064 [0.00062, 7e-04]
##
## The mean difference is more than 0 by a probability of 0.497
## and less than 0 by a probability of 0.503
```

```r
summary(c_changes_to_plan_2)
```

```
##   Data
## c_changes_2_b, n = 59
## c_changes_2_a, n = 59
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_changes_2_b and c_changes_2_a
## 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 c_changes_2_b and c_changes_2_a
##
##   Measures
##               mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff      0.000 0.000  0.000 0.000  0.503  0.497
## sigma_diff   0.001 0.000  0.001 0.001  0.000  1.000
## nu           1.011 0.010  1.000 1.031  0.000  1.000
## eff_size    -0.001 0.111 -0.220 0.214  0.503  0.497
## diff_pred    0.001 0.196 -0.008 0.008  0.498  0.502
##
## '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      0.000  0.000  0.000 0.000  0.000
## sigma_diff   0.001  0.001  0.001 0.001  0.001
## nu           1.000  1.003  1.007 1.015  1.038
## eff_size    -0.220 -0.076 -0.001 0.074  0.216
## diff_pred   -0.008 -0.001  0.000 0.001  0.009
```

```r
cohen.d(c_changes_2_a, c_changes_2_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: 0.1084785 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.02982207 0.18713500
```

Run a Bayesian *t*-test on the raw data before and after explaining the importance of planning, and calculate

Cohen's d for these values.

```r
c_planning_b = as.numeric(t(read.csv(
  "~/analysis/data/c_explain_planning_before.csv", header = FALSE)))
c_planning_a = as.numeric(t(read.csv(
  "~/analysis/data/c_explain_planning_after.csv", header = FALSE)))
c_explain_planning = bayes.t.test(c_planning_b, c_planning_a, paired = TRUE)

print(c_explain_planning)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
##
## data: c_planning_b and c_planning_a, n = 86
##
##   Estimates [95% credible interval]
## mean paired difference: 1.1e-07 [-0.00016, 0.00016]
## sd of the paired differences: 9e-04 [0.00089, 0.00097]
##
## The mean difference is more than 0 by a probability of 0.501
## and less than 0 by a probability of 0.499
```

```r
summary(c_explain_planning)
```

```
##   Data
## c_planning_b, n = 86
## c_planning_a, n = 86
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_planning_b and c_planning_a
## 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 c_planning_b and c_planning_a
##
##   Measures
##             mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff    0.000 0.000  0.000 0.000  0.499  0.501
## sigma_diff 0.001 0.000  0.001 0.001  0.000  1.000
## nu         1.007 0.007  1.000 1.022  0.000  1.000
## eff_size   0.000 0.091 -0.180 0.173  0.499  0.501
## diff_pred  0.001 0.187 -0.011 0.011  0.496  0.504
##
## '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     0.000  0.000  0.000 0.000  0.000
## sigma_diff  0.001  0.001  0.001 0.001  0.001
## nu          1.000  1.002  1.005 1.010  1.027
## eff_size   -0.177 -0.061  0.000 0.061  0.176
```

```
## diff_pred  -0.011 -0.001  0.000 0.001  0.011
```

```
cohen.d(c_planning_a, c_planning_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: 0.1481568 (negligible)
## 95 percent confidence interval:
##      lower      upper
## 0.06176261 0.23455108
```

Run a Bayesian t-test on the raw data before and after identifying barriers, and calculate Cohen's d for these values.

```
c_i_barriers_b = as.numeric(t(read.csv(
  "~/analysis/data/c_identify_barriers_before.csv", header = FALSE)))
c_i_barriers_a = as.numeric(t(read.csv(
  "~/analysis/data/c_identify_barriers_after.csv", header = FALSE)))
c_identify_barriers = bayes.t.test(c_i_barriers_b, c_i_barriers_a, paired = TRUE)


print(c_identify_barriers)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
##
## data: c_i_barriers_b and c_i_barriers_a, n = 85
##
##   Estimates [95% credible interval]
## mean paired difference: 1.3e-06 [-0.00019, 0.00018]
## sd of the paired differences: 0.00095 [0.00091, 0.0011]
##
## The mean difference is more than 0 by a probability of 0.506
## and less than 0 by a probability of 0.494
```

```
summary(c_identify_barriers)
```

```
##   Data
## c_i_barriers_b, n = 85
## c_i_barriers_a, n = 85
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_i_barriers_b and c_i_barriers_a
## 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 c_i_barriers_b and c_i_barriers_a
##
##   Measures
##             mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff    0.000 0.000  0.000 0.000  0.494  0.506
## sigma_diff 0.001 0.000  0.001 0.001  0.000  1.000
## nu         1.006 0.006  1.000 1.017  0.000  1.000
## eff_size   0.001 0.098 -0.191 0.190  0.494  0.506
```

```
## diff_pred  0.002 0.217 -0.012 0.012  0.498  0.502
##
## '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      0.000  0.000  0.000 0.000  0.000
## sigma_diff   0.001  0.001  0.001 0.001  0.001
## nu           1.000  1.002  1.004 1.008  1.021
## eff_size    -0.190 -0.064  0.001 0.067  0.192
## diff_pred   -0.012 -0.001  0.000 0.001  0.012
```

```
cohen.d(c_i_barriers_a, c_i_barriers_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: -0.06919324 (negligible)
## 95 percent confidence interval:
##       lower       upper
## -0.16532241  0.02693592
```

Run a Bayesian t-test on the raw data before and after dealing with barriers, and calculate Cohen's d for these values.

```
c_d_barriers_b = as.numeric(t(read.csv(
  "~/analysis/data/c_deal_with_barriers_before.csv", header = FALSE)))
c_d_barriers_a = as.numeric(t(read.csv(
  "~/analysis/data/c_deal_with_barriers_after.csv", header = FALSE)))
c_deal_with_barriers = bayes.t.test(c_d_barriers_b, c_d_barriers_a, paired = TRUE)

print(c_deal_with_barriers)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
##
## data: c_d_barriers_b and c_d_barriers_a, n = 85
##
##   Estimates [95% credible interval]
## mean paired difference: 2e-06 [-0.00024, 0.00024]
## sd of the paired differences: 0.0013 [0.0012, 0.0014]
##
## The mean difference is more than 0 by a probability of 0.507
## and less than 0 by a probability of 0.493
```

```
summary(c_deal_with_barriers)
```

```
##   Data
## c_d_barriers_b, n = 85
## c_d_barriers_a, n = 85
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_d_barriers_b and c_d_barriers_a
## 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 c_d_barriers_b and c_d_barriers_a
##
##   Measures
##              mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff     0.000 0.000  0.000 0.000  0.493  0.507
## sigma_diff  0.001 0.000  0.001 0.001  0.000  1.000
## nu          1.006 0.006  1.000 1.019  0.000  1.000
## eff_size    0.001 0.095 -0.187 0.187  0.493  0.507
## diff_pred  -0.001 0.287 -0.017 0.015  0.496  0.504
##
## '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     0.000  0.000  0.000 0.000  0.000
## sigma_diff  0.001  0.001  0.001 0.001  0.001
## nu          1.000  1.002  1.004 1.009  1.023
## eff_size   -0.187 -0.063  0.002 0.065  0.187
## diff_pred  -0.016 -0.001  0.000 0.001  0.016
```

```r
cohen.d(c_d_barriers_a, c_d_barriers_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: 0.2011579 (small)
## 95 percent confidence interval:
##     lower     upper
## 0.0814886 0.3208271
```

Run a Bayesian t-test on the raw data before and after showing testimonials from other people who created plans, and calculate Cohen's d for these values.

```r
c_testimonials_b = as.numeric(t(read.csv(
  "~/analysis/data/c_show_testimonials_before.csv", header = FALSE)))
c_testimonials_a = as.numeric(t(read.csv(
  "~/analysis/data/c_show_testimonials_after.csv", header = FALSE)))
c_show_testimonials = bayes.t.test(c_testimonials_b, c_testimonials_a, paired = TRUE)

print(c_show_testimonials)
```

```
##
##  Bayesian estimation supersedes the t test (BEST) - paired samples
##
## data: c_testimonials_b and c_testimonials_a, n = 90
##
##   Estimates [95% credible interval]
## mean paired difference: -2.8e-06 [-0.00026, 0.00026]
## sd of the paired differences: 0.0012 [0.0010, 0.0018]
##
## The mean difference is more than 0 by a probability of 0.491
```

```
## and less than 0 by a probability of 0.509
```

```
summary(c_show_testimonials)
```

```
##   Data
## c_testimonials_b, n = 90
## c_testimonials_a, n = 90
##
##   Model parameters and generated quantities
## mu_diff: the mean pairwise difference between c_testimonials_b and c_testimonials_a
## 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 c_testimonials_b and c_testimonials_a
##
##   Measures
##               mean    sd  HDIlo HDIup %<comp %>comp
## mu_diff      0.000 0.000  0.000 0.000  0.509  0.491
## sigma_diff   0.001 0.000  0.001 0.002  0.000  1.000
## nu           1.004 0.004  1.000 1.013  0.000  1.000
## eff_size    -0.002 0.103 -0.205 0.199  0.509  0.491
## diff_pred   -0.032 5.459 -0.016 0.016  0.499  0.501
##
## '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      0.000  0.000  0.000 0.000  0.000
## sigma_diff   0.001  0.001  0.001 0.001  0.002
## nu           1.000  1.001  1.003 1.006  1.016
## eff_size    -0.203 -0.071 -0.002 0.067  0.201
## diff_pred   -0.016 -0.001  0.000 0.001  0.017
```

```
cohen.d(c_testimonials_a, c_testimonials_b, paired=TRUE)
```

```
##
## Cohen's d
##
## d estimate: 0.2148119 (small)
## 95 percent confidence interval:
##     lower     upper
## 0.1140171 0.3156066
```