Mindset Manipulation Experiment 2 Bayes Supplement

RW Hass

8/1/2024

Notes on the Supplement

This was designed to illustrate our compliance with recommendations of van de Schoot, et al (2020). Importantly, that tutorial utilizes the advanced tools made possible by `rjags` and `rstan`. Some of the information is particular to the MCMC methods used by `rstan`, and we tried our best to produce information that was relevant given the `BayesFactor` package's limited capabilities. The data and the underlying RMarkdown for this and the second experiment are available here.

Packages used: `psych`, `car`, `dplyr`, `BayesFactor`

session info:

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.3.2 (2023-10-31)
##  os       macOS Sonoma 14.6
##  system   aarch64, darwin20
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       America/New_York
##  date     2024-08-01
##  pandoc   3.2.1 @ /opt/homebrew/bin/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package      * version    date (UTC) lib source
##  abind          1.4-5      2016-07-21 [1] CRAN (R 4.3.0)
##  BayesFactor  * 0.9.12-4.7 2024-01-24 [1] CRAN (R 4.3.1)
##  cachem         1.1.0      2024-05-16 [1] CRAN (R 4.3.3)
##  car          * 3.1-2      2023-03-30 [1] CRAN (R 4.3.0)
##  carData      * 3.0-5      2022-01-06 [1] CRAN (R 4.3.0)
##  cli            3.6.3      2024-06-21 [1] CRAN (R 4.3.3)
##  coda         * 0.19-4.1   2024-01-31 [1] CRAN (R 4.3.1)
##  colorspace     2.1-0      2023-01-23 [1] CRAN (R 4.3.0)
##  devtools       2.4.5      2022-10-11 [1] CRAN (R 4.3.0)
##  digest         0.6.36     2024-06-23 [1] CRAN (R 4.3.3)
##  dplyr        * 1.1.4      2023-11-17 [1] CRAN (R 4.3.1)
##  ellipsis       0.3.2      2021-04-29 [1] CRAN (R 4.3.0)
##  evaluate       0.24.0     2024-06-10 [1] CRAN (R 4.3.3)
##  fansi          1.0.6      2023-12-08 [1] CRAN (R 4.3.1)
##  fastmap        1.2.0      2024-05-15 [1] CRAN (R 4.3.3)
##  forcats      * 1.0.0      2023-01-29 [1] CRAN (R 4.3.0)
##  fs             1.6.4      2024-04-25 [1] CRAN (R 4.3.1)
```

```
##  generics       0.1.3      2022-07-05 [1] CRAN (R 4.3.0)
##  ggplot2      * 3.5.1      2024-04-23 [1] CRAN (R 4.3.1)
##  glue           1.7.0      2024-01-09 [1] CRAN (R 4.3.1)
##  gtable         0.3.5      2024-04-22 [1] CRAN (R 4.3.1)
##  hms            1.1.3      2023-03-21 [1] CRAN (R 4.3.0)
##  htmltools      0.5.8.1    2024-04-04 [1] CRAN (R 4.3.1)
##  htmlwidgets    1.6.4      2023-12-06 [1] CRAN (R 4.3.1)
##  httpuv         1.6.15     2024-03-26 [1] CRAN (R 4.3.1)
##  knitr          1.48       2024-07-07 [1] CRAN (R 4.3.3)
##  later          1.3.2      2023-12-06 [1] CRAN (R 4.3.1)
##  lattice        0.22-6     2024-03-20 [1] CRAN (R 4.3.1)
##  lifecycle      1.0.4      2023-11-07 [1] CRAN (R 4.3.1)
##  lubridate    * 1.9.3      2023-09-27 [1] CRAN (R 4.3.1)
##  magrittr       2.0.3      2022-03-30 [1] CRAN (R 4.3.0)
##  Matrix       * 1.6-5      2024-01-11 [1] CRAN (R 4.3.1)
##  MatrixModels   0.5-3      2023-11-06 [1] CRAN (R 4.3.1)
##  memoise        2.0.1      2021-11-26 [1] CRAN (R 4.3.0)
##  mime           0.12       2021-09-28 [1] CRAN (R 4.3.0)
##  miniUI         0.1.1.1    2018-05-18 [1] CRAN (R 4.3.0)
##  mnormt         2.1.1      2022-09-26 [1] CRAN (R 4.3.0)
##  munsell        0.5.1      2024-04-01 [1] CRAN (R 4.3.1)
##  mvtnorm        1.2-5      2024-05-21 [1] CRAN (R 4.3.3)
##  nlme           3.1-165    2024-06-06 [1] CRAN (R 4.3.3)
##  pbapply        1.7-2      2023-06-27 [1] CRAN (R 4.3.0)
##  pillar         1.9.0      2023-03-22 [1] CRAN (R 4.3.0)
##  pkgbuild       1.4.4      2024-03-17 [1] CRAN (R 4.3.1)
##  pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 4.3.0)
##  pkgload        1.4.0      2024-06-28 [1] CRAN (R 4.3.3)
##  profvis        0.3.8      2023-05-02 [1] CRAN (R 4.3.0)
##  promises       1.3.0      2024-04-05 [1] CRAN (R 4.3.1)
##  psych        * 2.4.6.26   2024-06-27 [1] CRAN (R 4.3.3)
##  purrr        * 1.0.2      2023-08-10 [1] CRAN (R 4.3.0)
##  R6             2.5.1      2021-08-19 [1] CRAN (R 4.3.0)
##  Rcpp           1.0.12     2024-01-09 [1] CRAN (R 4.3.1)
##  readr        * 2.1.5      2024-01-10 [1] CRAN (R 4.3.1)
##  remotes        2.5.0      2024-03-17 [1] CRAN (R 4.3.1)
##  rlang          1.1.4      2024-06-04 [1] CRAN (R 4.3.3)
##  rmarkdown      2.27       2024-05-17 [1] CRAN (R 4.3.3)
##  rstudioapi     0.16.0     2024-03-24 [1] CRAN (R 4.3.1)
##  scales         1.3.0      2023-11-28 [1] CRAN (R 4.3.1)
##  sessioninfo    1.2.2      2021-12-06 [1] CRAN (R 4.3.0)
##  shiny          1.8.1.1    2024-04-02 [1] CRAN (R 4.3.1)
##  stringi        1.8.4      2024-05-06 [1] CRAN (R 4.3.1)
##  stringr      * 1.5.1      2023-11-14 [1] CRAN (R 4.3.1)
##  tibble       * 3.2.1      2023-03-20 [1] CRAN (R 4.3.0)
##  tidyr        * 1.3.1      2024-01-24 [1] CRAN (R 4.3.1)
##  tidyselect     1.2.1      2024-03-11 [1] CRAN (R 4.3.1)
##  tidyverse    * 2.0.0      2023-02-22 [1] CRAN (R 4.3.0)
##  timechange     0.3.0      2024-01-18 [1] CRAN (R 4.3.1)
##  tzdb           0.4.0      2023-05-12 [1] CRAN (R 4.3.0)
##  urlchecker     1.0.1      2021-11-30 [1] CRAN (R 4.3.0)
##  usethis        2.2.3      2024-02-19 [1] CRAN (R 4.3.1)
##  utf8           1.2.4      2023-10-22 [1] CRAN (R 4.3.1)
##  vctrs          0.6.5      2023-12-01 [1] CRAN (R 4.3.1)
```

```
##  withr          3.0.0      2024-01-16 [1] CRAN (R 4.3.1)
##  xfun           0.45       2024-06-16 [1] CRAN (R 4.3.3)
##  xtable         1.8-4      2019-04-21 [1] CRAN (R 4.3.0)
##  yaml           2.3.9      2024-07-05 [1] CRAN (R 4.3.3)
##
##  [1] /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/library
##
## ------------------------------------------------------------------------------
```

Descriptive statistics and correlation (correlations not used in manuscript)

```
# r.out <- corr.test(d[,c("JoanScore","DTScore","DTFlu","Fixed","Growth","CSE")])
# desc.out <- describe(d[,c("JoanScore","DTScore","DTFlu","Fixed","Growth","CSE")])

# per group
describeBy(d[,c("JoanScore","DTScore","DTFlu","Fixed","Growth","CSE")], group = d$Condition)
```

```
##
##  Descriptive statistics by group
## group: Fixed
##           vars  n mean   sd median trimmed  mad min max range  skew kurtosis
## JoanScore    1 55 2.85 0.73   3.00    2.82 1.48 2.0   4   2.0  0.22    -1.15
## DTScore      2 53 2.82 0.72   2.86    2.82 0.64 1.0   5   4.0  0.04     0.63
## DTFlu        3 55 6.60 3.41   6.00    6.40 2.97 0.0  17  17.0  0.63     0.45
## Fixed        4 55 3.13 1.05   3.25    3.14 1.48 1.0   5   4.0 -0.06    -1.15
## Growth       5 55 3.52 0.87   3.50    3.55 0.74 1.5   5   3.5 -0.32    -0.66
## CSE          6 55 3.37 0.95   3.33    3.39 0.99 1.0   5   4.0 -0.37    -0.19
##             se
## JoanScore 0.10
## DTScore   0.10
## DTFlu     0.46
## Fixed     0.14
## Growth    0.12
## CSE       0.13
## ------------------------------------------------------------
## group: Growth
##           vars  n mean   sd median trimmed  mad min  max range  skew kurtosis
## JoanScore    1 56 2.68 0.90   3.00    2.65 1.48 1.0  5.0   4.0  0.36    -0.55
## DTScore      2 56 2.79 0.59   2.75    2.79 0.46 1.5  4.5   3.0  0.26     0.47
## DTFlu        3 56 7.98 3.76   7.50    7.63 3.71 2.0 17.0  15.0  0.82    -0.02
## Fixed        4 56 2.60 0.92   2.42    2.58 0.86 1.0  5.0   4.0  0.38    -0.59
## Growth       5 56 3.98 0.57   4.00    3.98 0.37 2.5  5.0   2.5 -0.15    -0.44
## CSE          6 56 3.18 1.03   3.33    3.20 0.99 1.0  5.0   4.0 -0.21    -0.90
##             se
## JoanScore 0.12
## DTScore   0.08
## DTFlu     0.50
## Fixed     0.12
## Growth    0.08
## CSE       0.14
```

Bayesian Estimation for Hypotheses 1 and 2

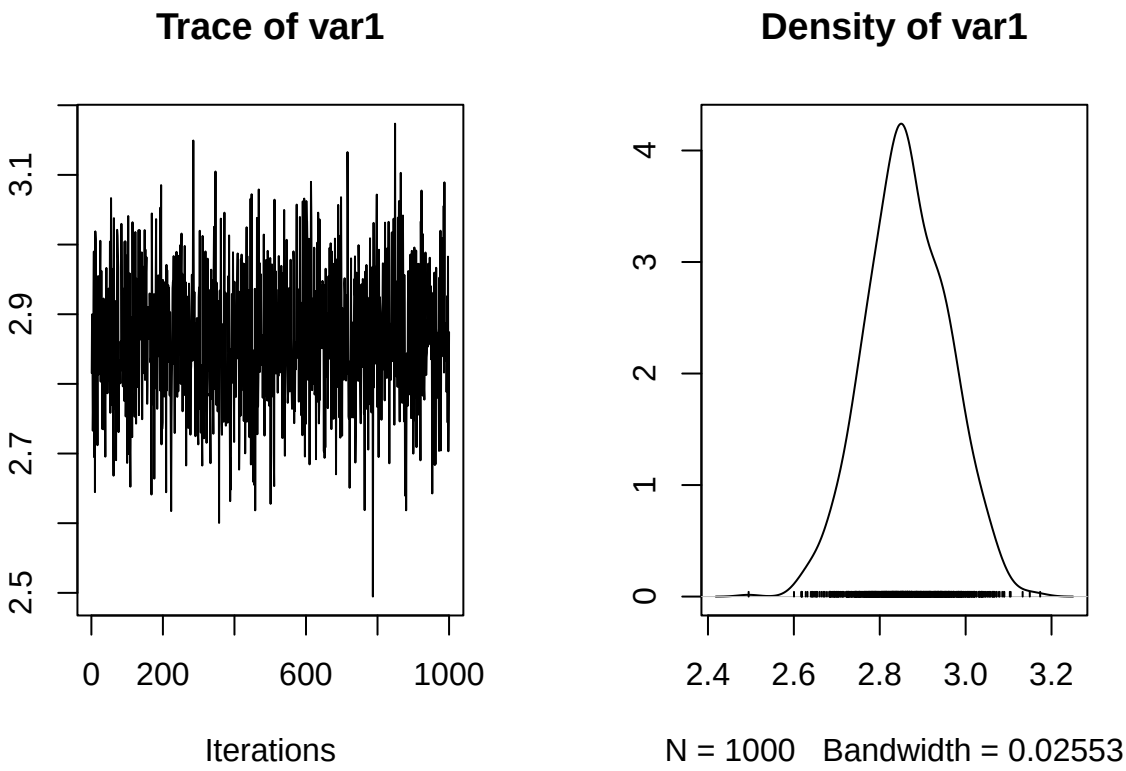
Note, the following sections follow the same layout of Supplement 1, and more info on the checks are available there.

Fixed Mindset

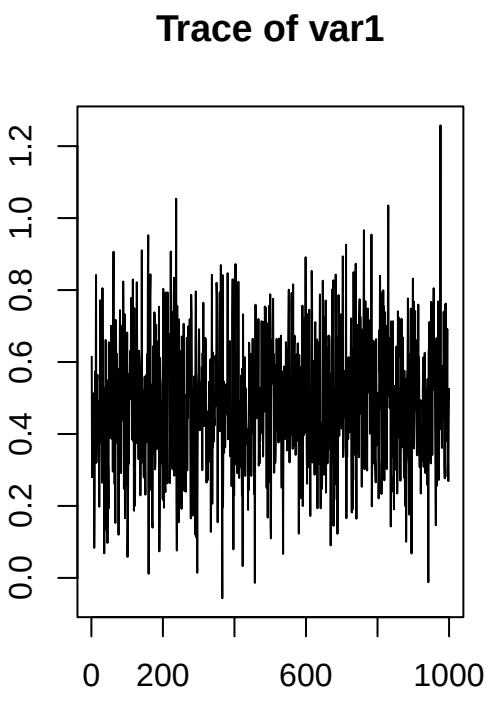
```
# default is rscale = medium or sqrt(2)/2
ttestBF(formula = Fixed ~ Condition, data = d)
```

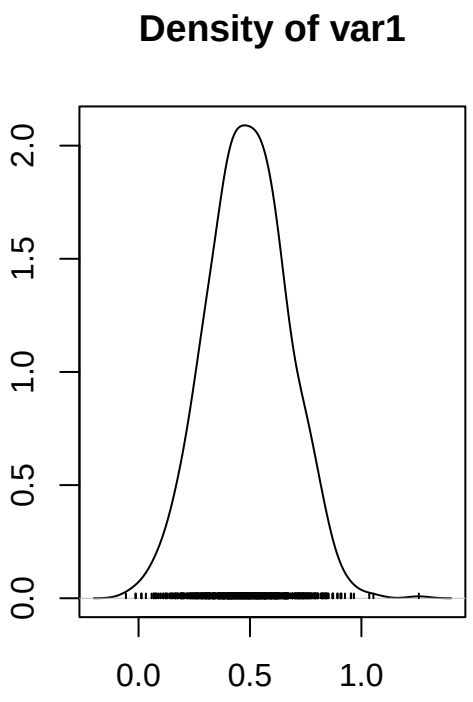
```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 6.569093 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
fixedpost <- ttestBF(formula = Fixed ~ Condition, data = d, posterior = TRUE, iterations = 1000)
plot(fixedpost[,"mu"])
```




```
plot(fixedpost[,"beta (Fixed - Growth)"])
```





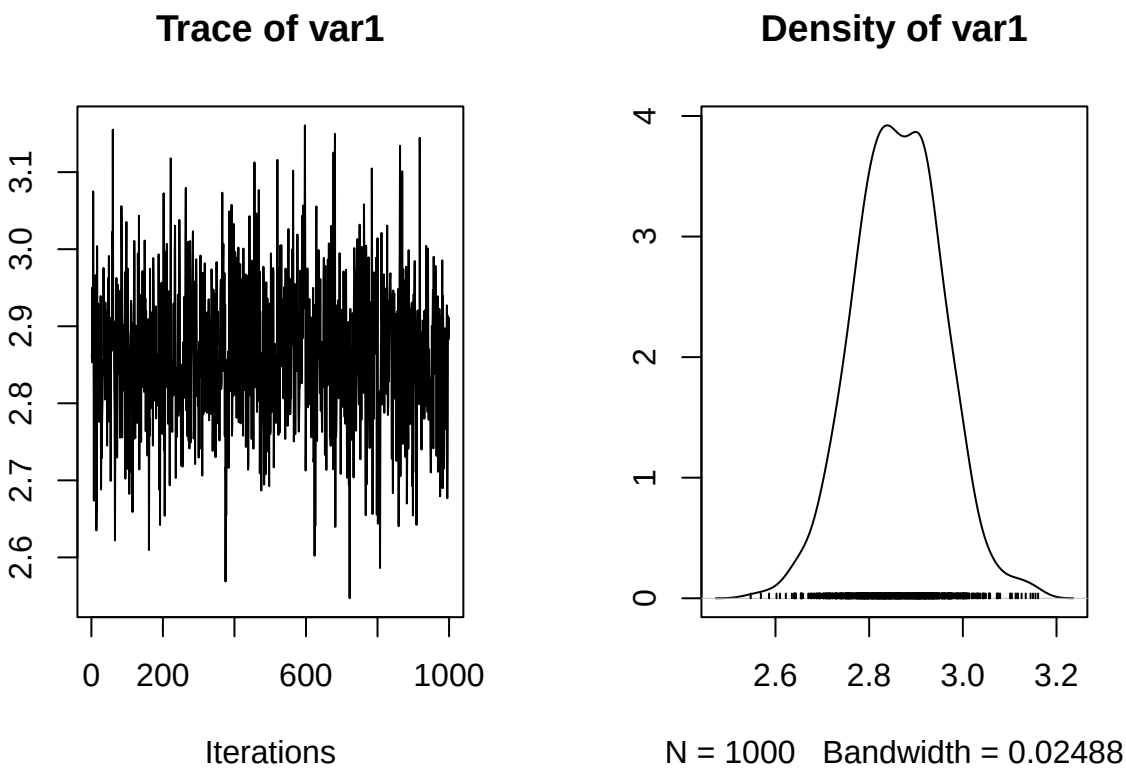

```
quantile(fixedpost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##      2.5%       50%     97.5%
## 0.1394619 0.4874589 0.8344100
```

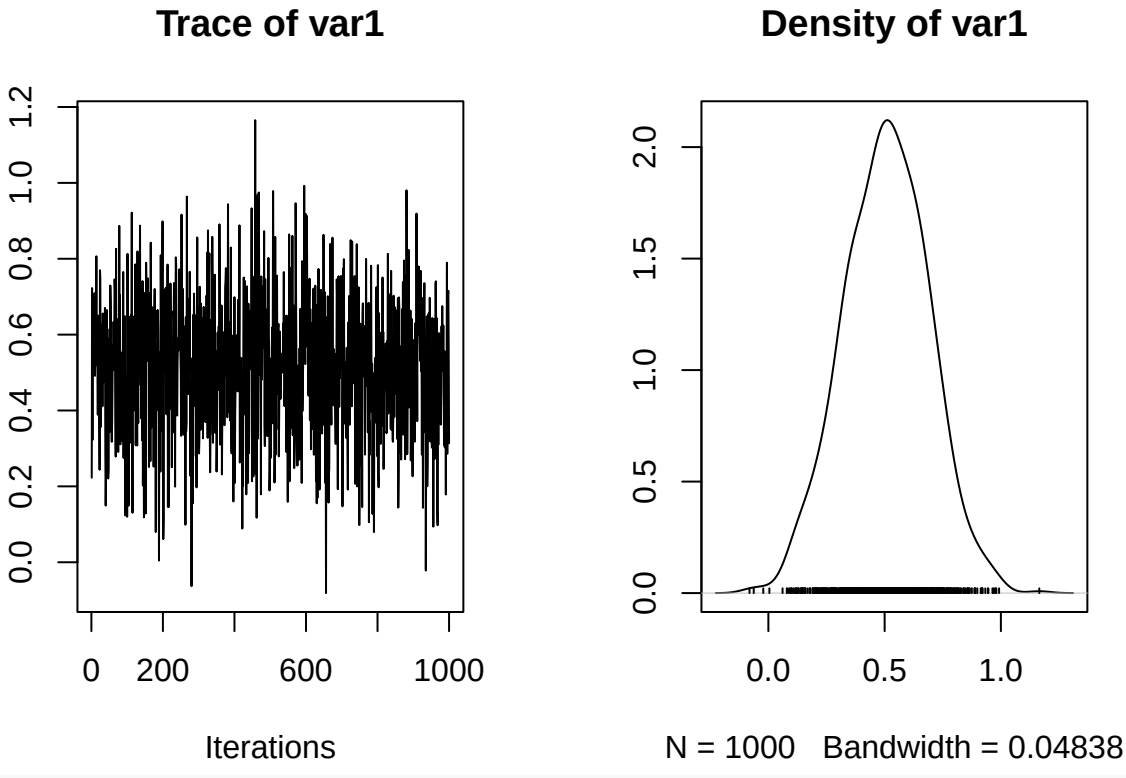
```
### with wide rscale = wide or 1
ttestBF(formula = Fixed ~ Condition, data = d, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 5.574359 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
fixedpost.w <- ttestBF(formula = Fixed ~ Condition, data = d, rscale = "wide", posterior = TRUE, iterati
plot(fixedpost.w[,"mu"])
```




```
plot(fixedpost.w[,"beta (Fixed - Growth)"])
```



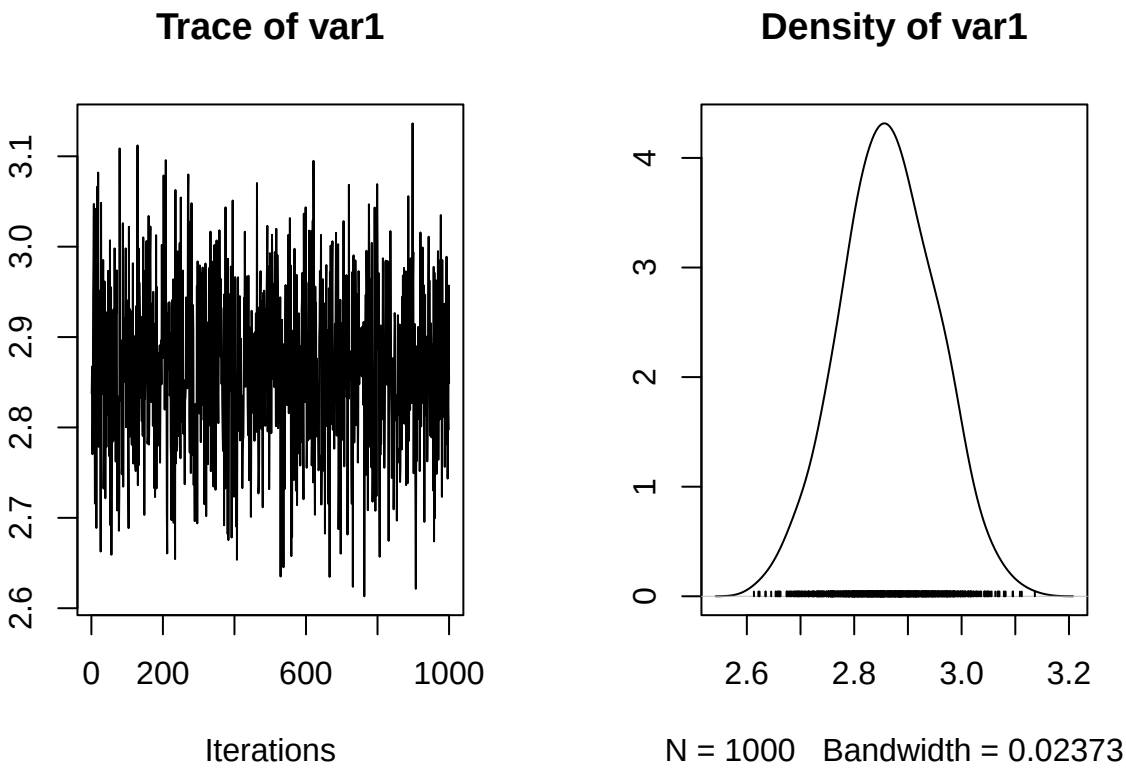

```
quantile(fixedpost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##      2.5%       50%     97.5%
## 0.1440049 0.5086717 0.8577710
```

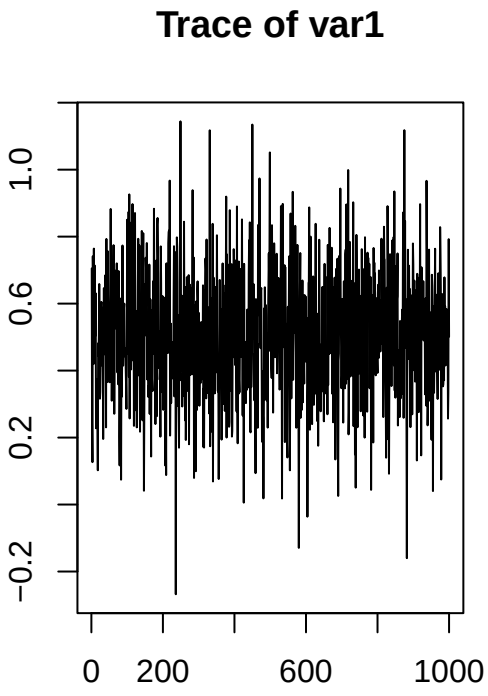
```
### with ultrawide rscale = ultrawide or sqrt(2)
ttestBF(formula = Fixed ~ Condition, data = d, rscale = "ultrawide")
```

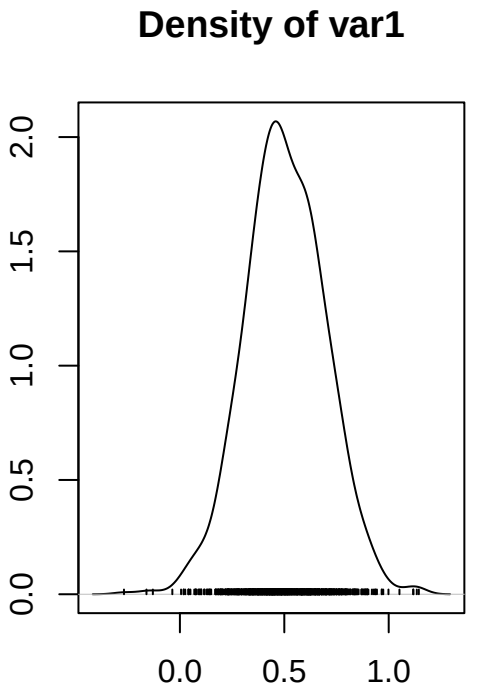
```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 4.42181 ±0.01%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
fixedpost.uw <- ttestBF(formula = Fixed ~ Condition, data = d, rscale = "ultrawide", posterior = TRUE, i
plot(fixedpost.uw[,"mu"])
```




```
plot(fixedpost.uw[,"beta (Fixed - Growth)"])
```

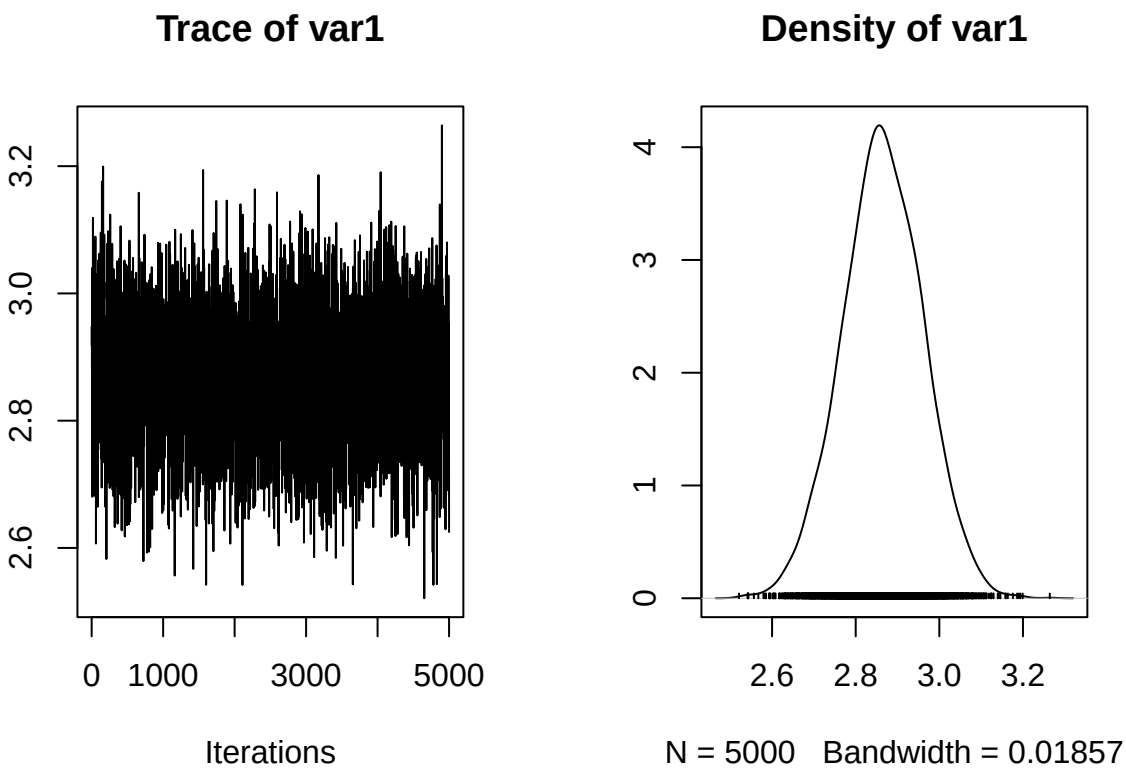




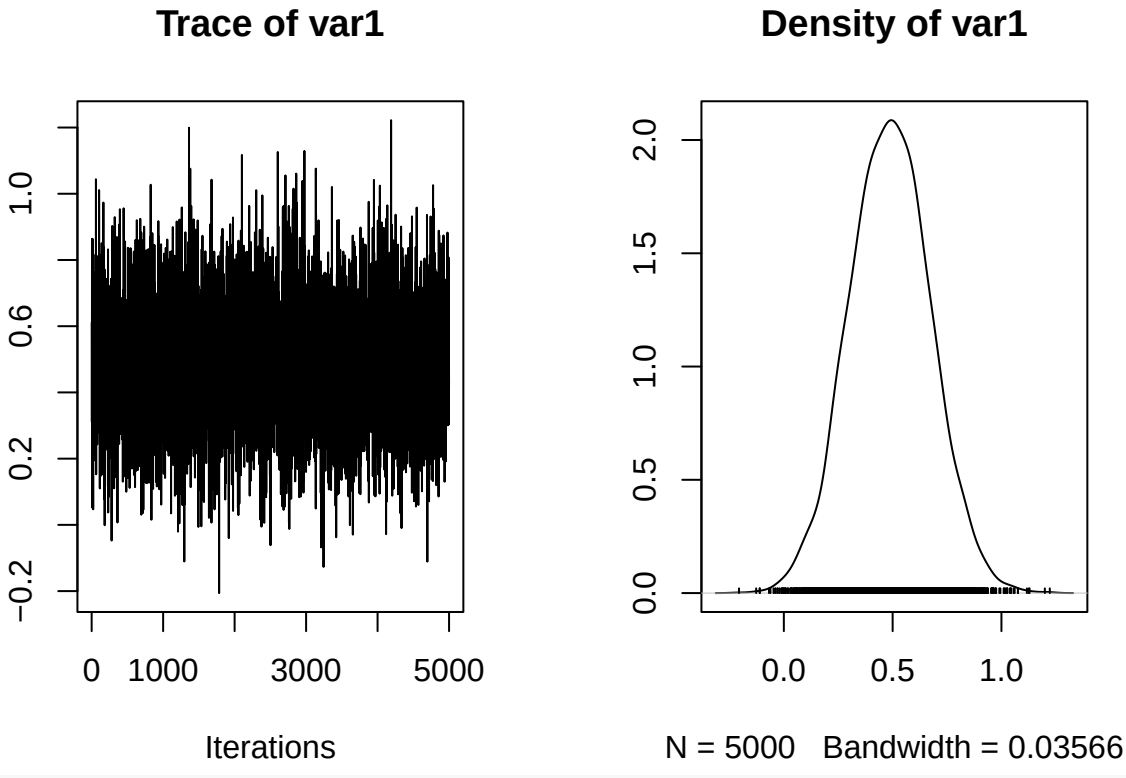

```
quantile(fixedpost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##      2.5%       50%     97.5%
## 0.1263479 0.4986083 0.8826288
```

```
### More iterations
fixedpost.check <- ttestBF(formula = Fixed ~ Condition, data = d, posterior = TRUE, iterations = 5000)
plot(fixedpost.check[,"mu"])
```




```
plot(fixedpost.check[,"beta (Fixed - Growth)"])
```




```
quantile(fixedpost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

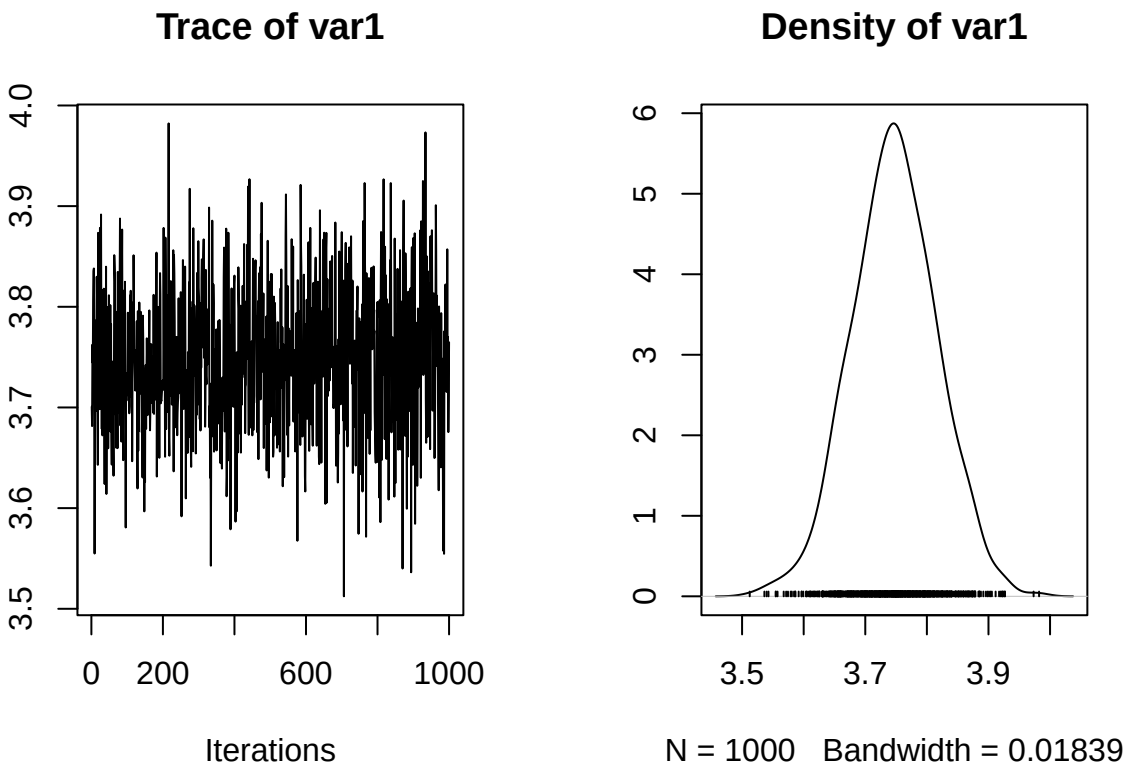
```
##      2.5%       50%     97.5%
## 0.1269584 0.4884059 0.8476794
```

Growth Mindset as outcome

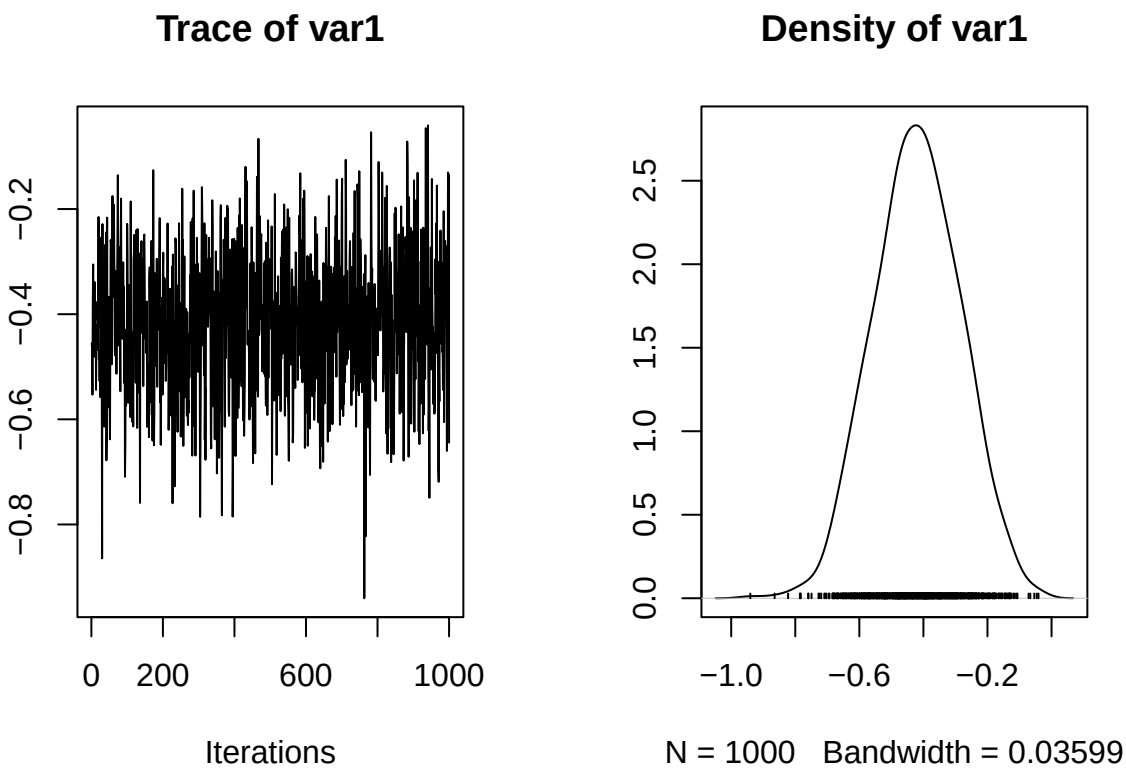
```
## default / medium
ttestBF(formula = Growth ~ Condition, data = d)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 21.44464 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
gpost <- ttestBF(formula = Growth ~ Condition, data = d, posterior = TRUE, iterations = 1000)
plot(gpost[,"mu"])
```




```
plot(gpost[,"beta (Fixed - Growth)"])
```



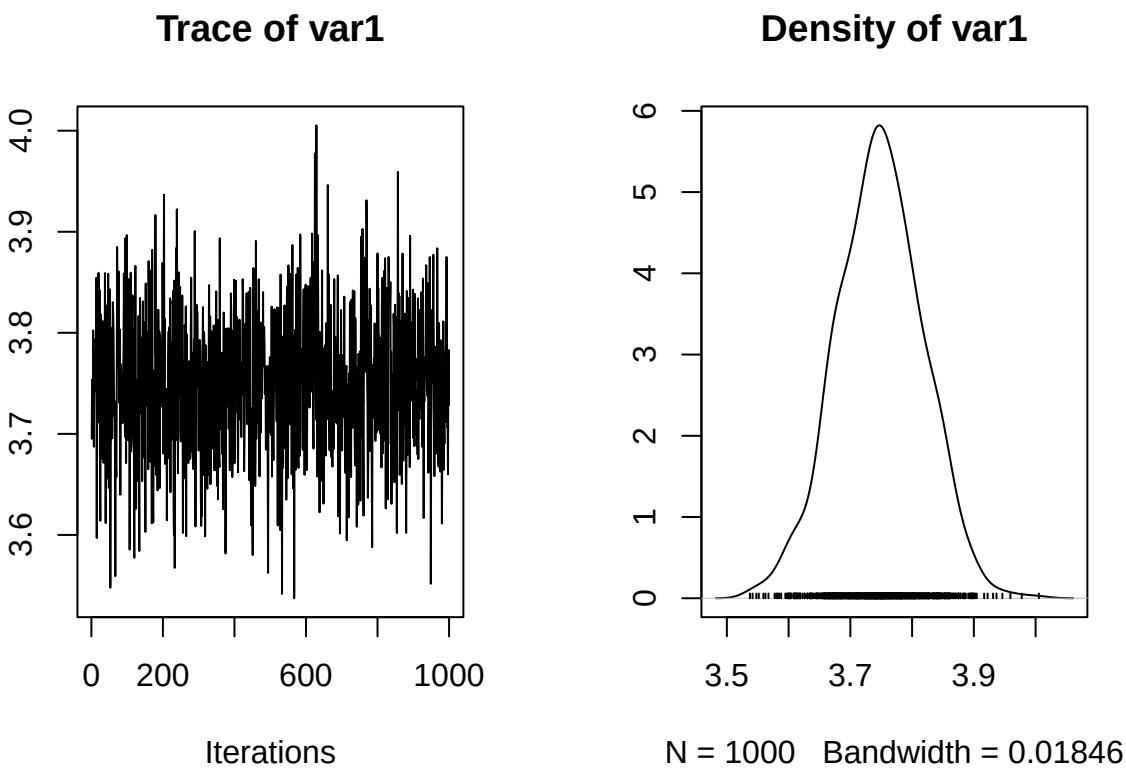

```
quantile(gpost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.6733249 -0.4188110 -0.1560531
```

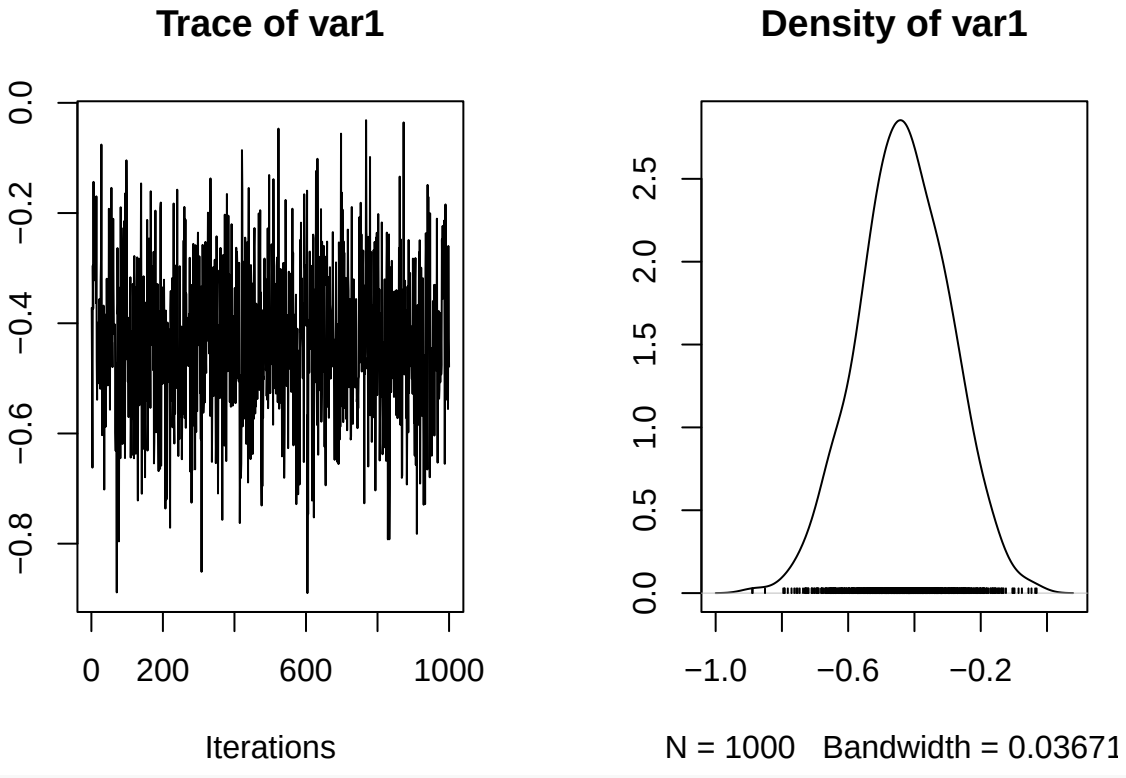
```
# Wide

ttestBF(formula = Growth ~ Condition, data = d, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 18.94251 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
gpost.w <- ttestBF(formula = Growth ~ Condition, data = d, rscale = "wide", posterior = TRUE, iterations
plot(gpost.w[,"mu"])
```




```
plot(gpost.w[,"beta (Fixed - Growth)"])
```



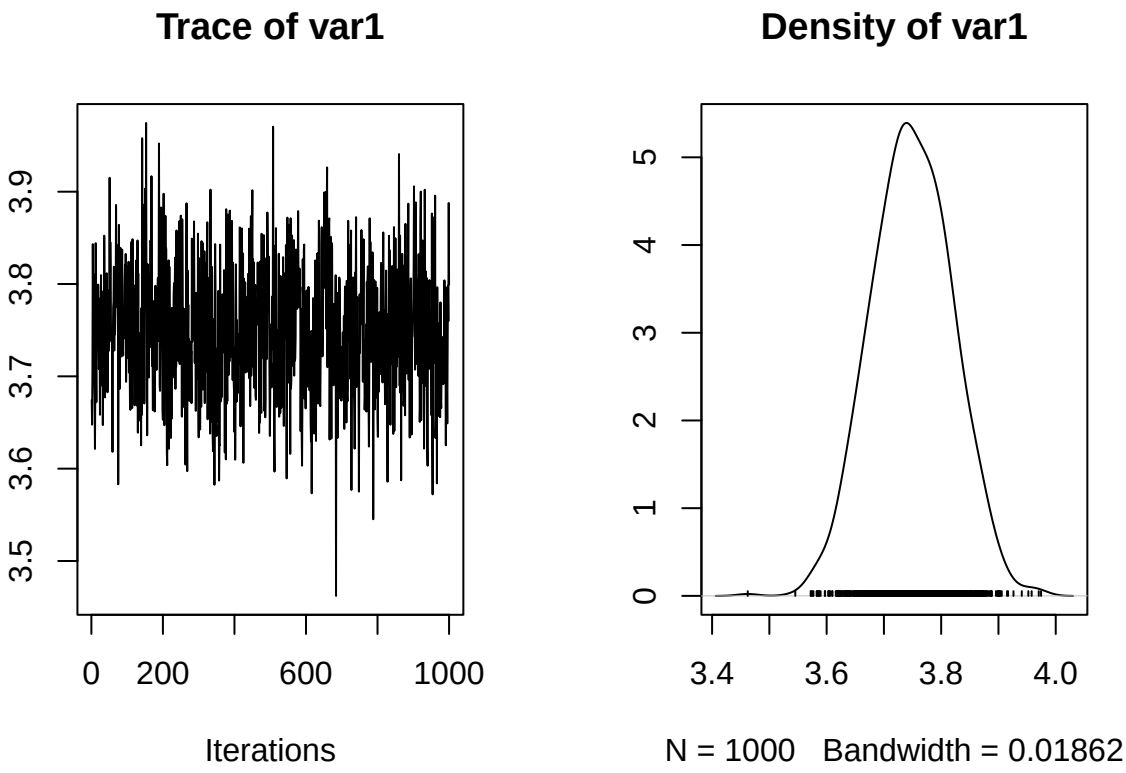

```
quantile(gpost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.7034592 -0.4325637 -0.1653726
```

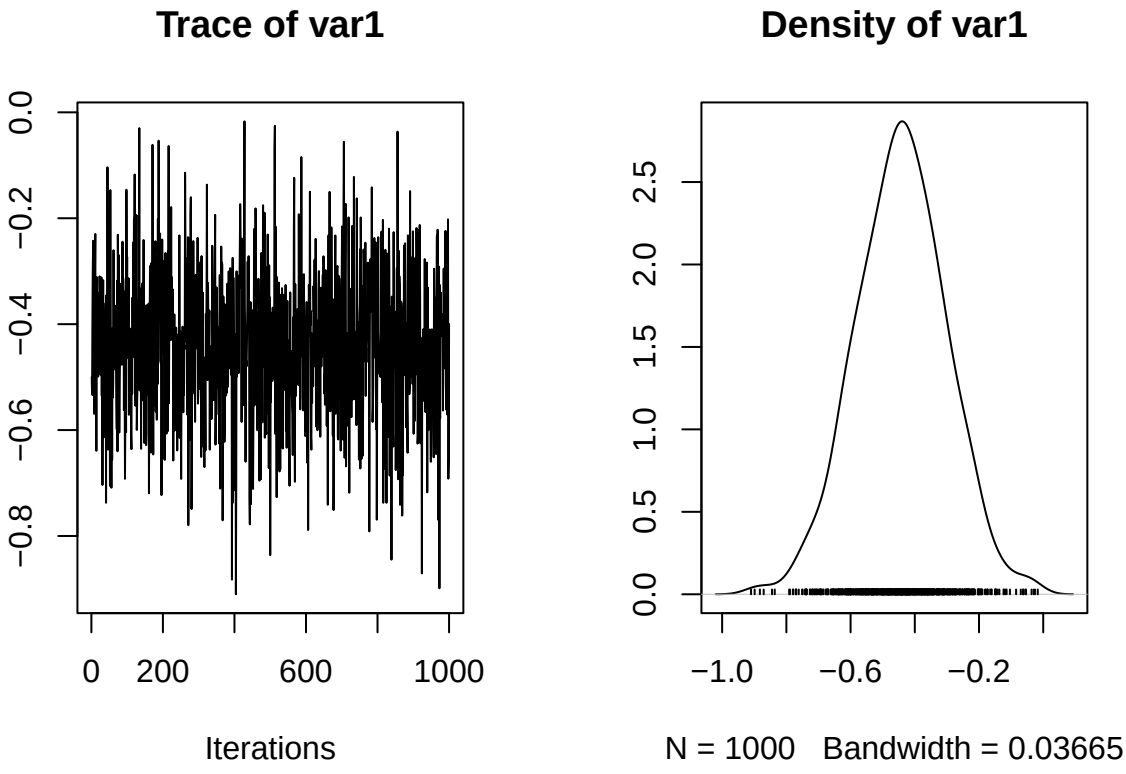
```r
# UltraWide

ttestBF(formula = Growth ~ Condition, data = d, rscale = "ultrawide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 15.46237 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```r
gpost.uw <- ttestBF(formula = Growth ~ Condition, data = d, rscale = "ultrawide", posterior = TRUE, ite
plot(gpost.uw[,"mu"])
```



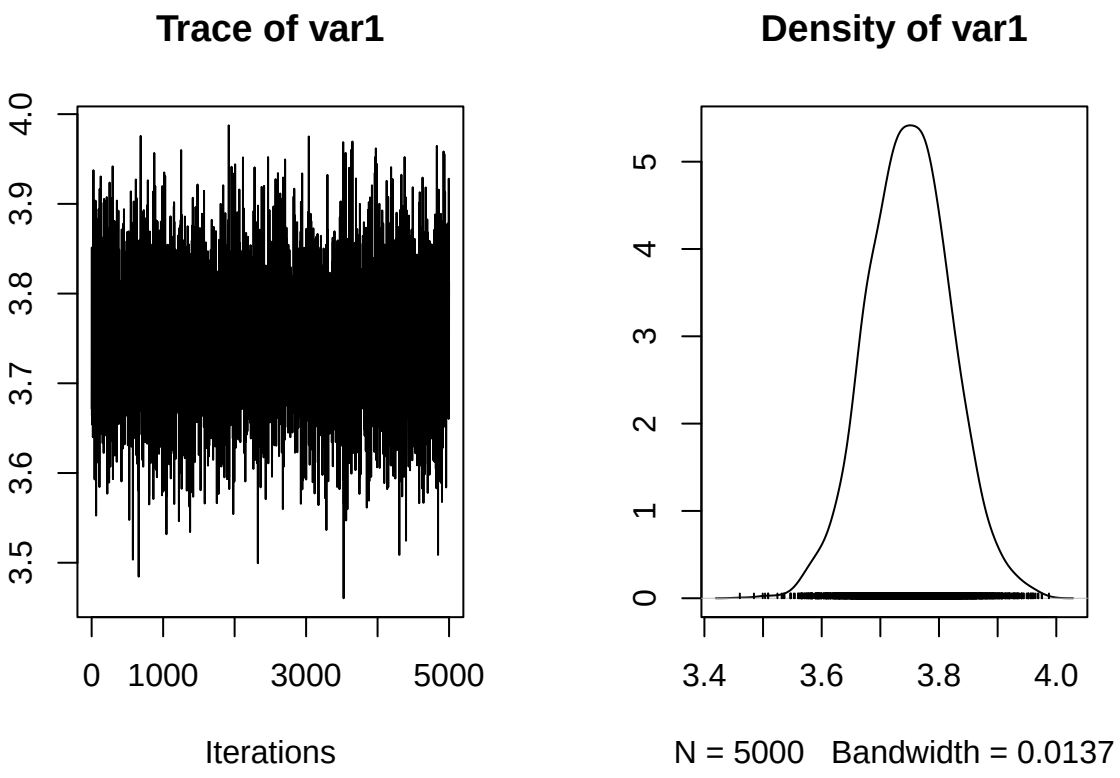

```r
plot(gpost.uw[,"beta (Fixed - Growth)"])
```



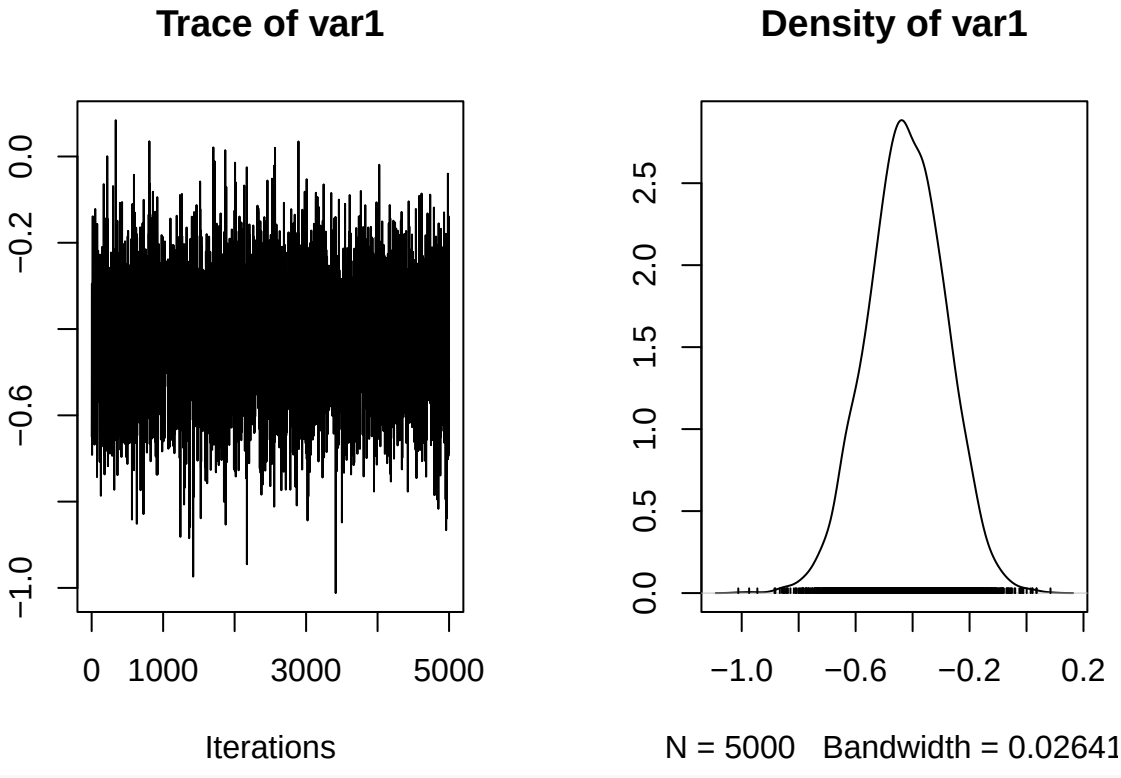

```
quantile(gpost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.7228277 -0.4413185 -0.1727467
```

```
### More iterations
gpost.check <- ttestBF(formula = Growth ~ Condition, data = d, posterior = TRUE, iterations = 5000)
plot(gpost.check[,"mu"])
```




```
plot(gpost.check[,"beta (Fixed - Growth)"])
```




```
quantile(gpost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

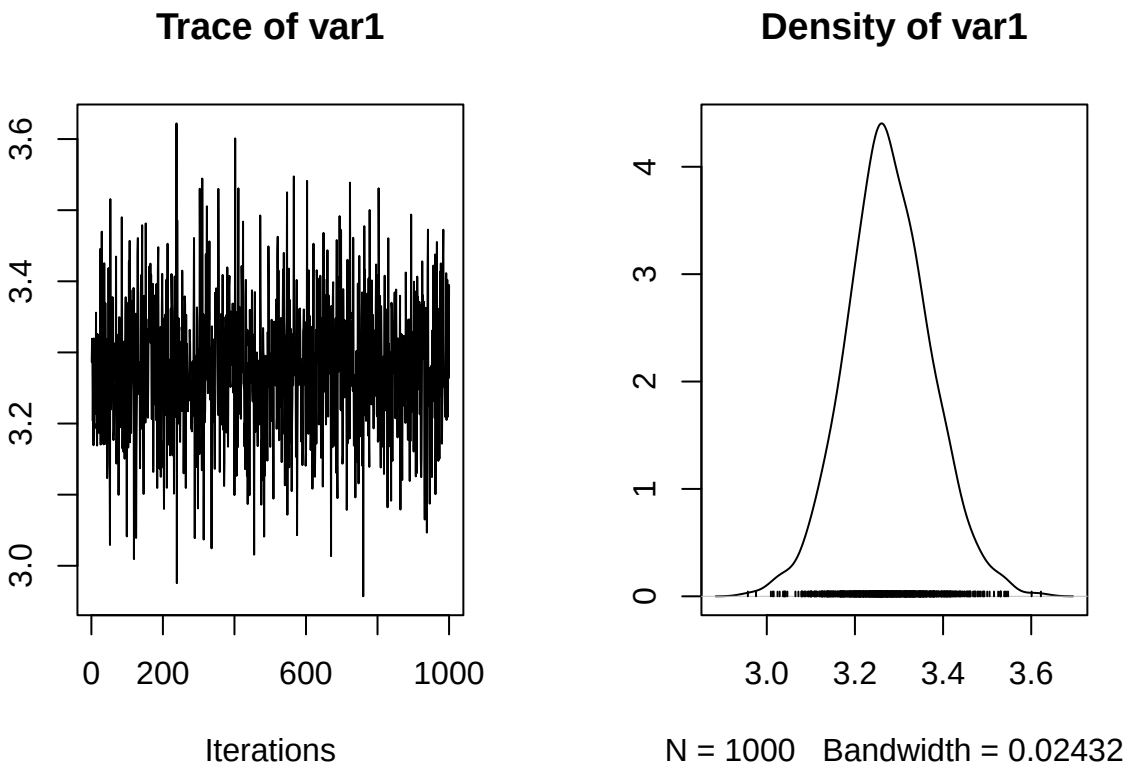
```
##       2.5%        50%      97.5%
## -0.6918381 -0.4225028 -0.1643170
```

Creative Self-efficacy as outcome

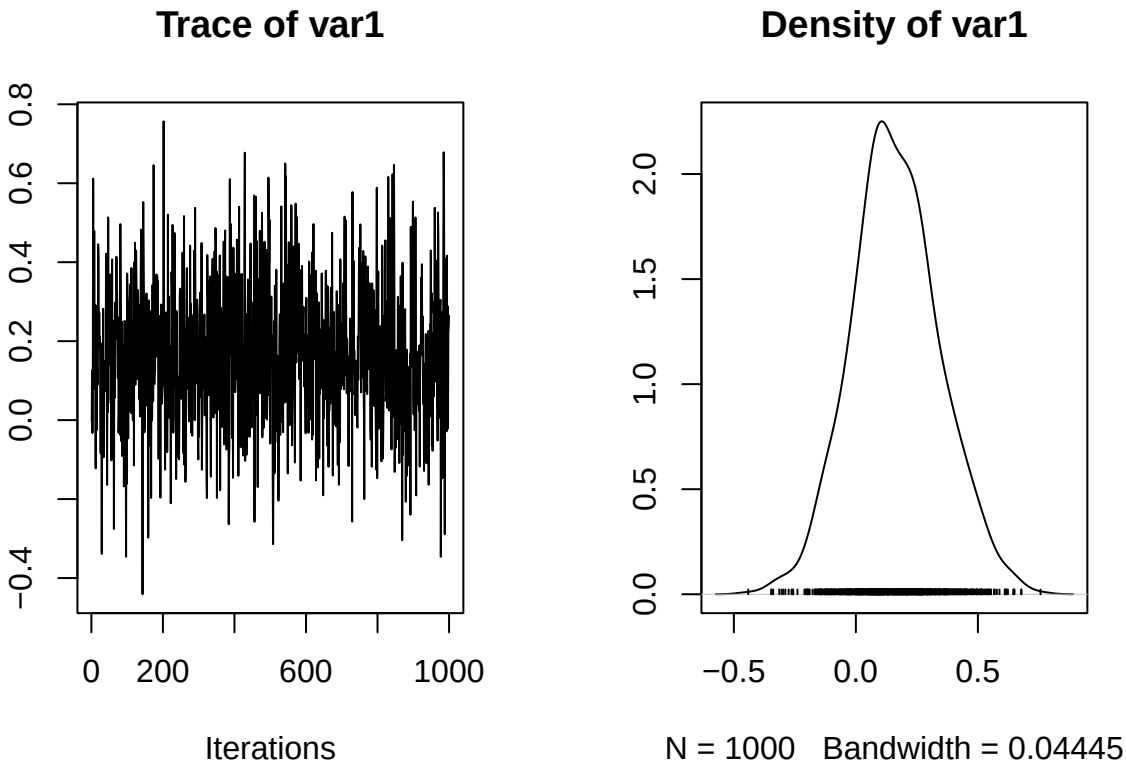
```
## Default / medium
ttestBF(formula = CSE ~ Condition, data = d)
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.3184985 ±0.02%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
csepost <- ttestBF(formula = CSE ~ Condition, data = d, posterior = TRUE, iterations = 1000)
plot(csepost[,"mu"])
```




```
plot(csepost[,"beta (Fixed - Growth)"])
```



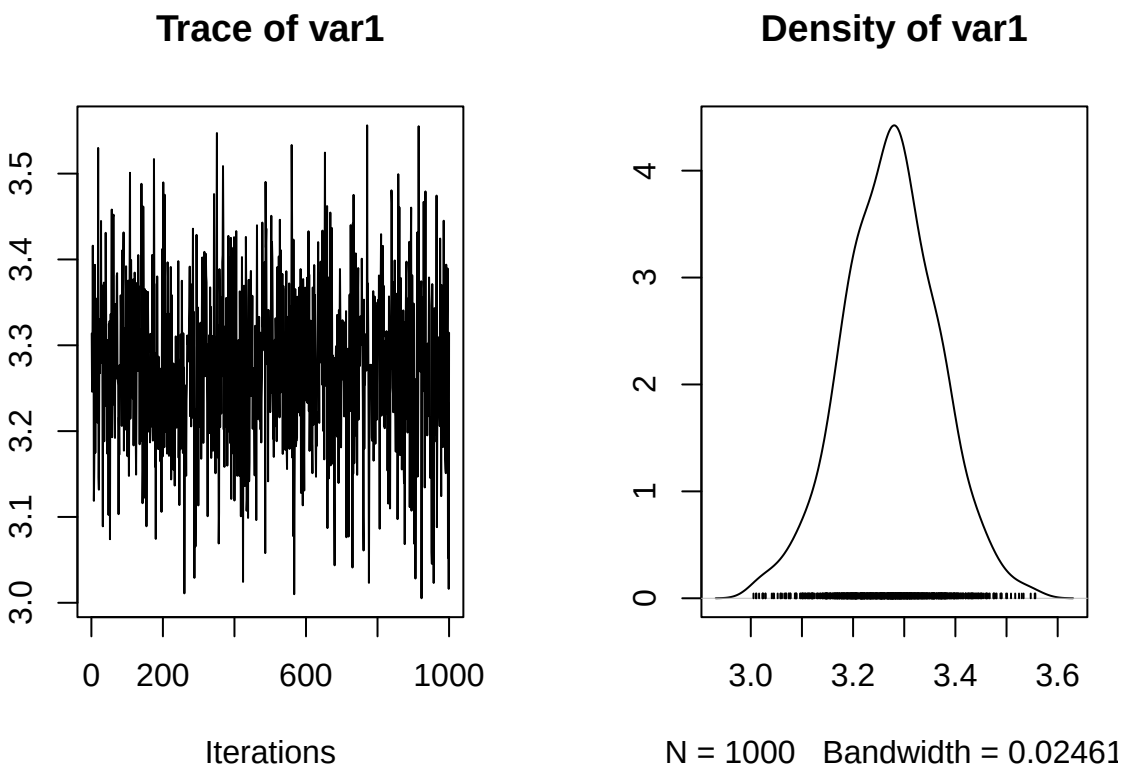

```
quantile(csepost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1679336  0.1584152  0.5259200
```

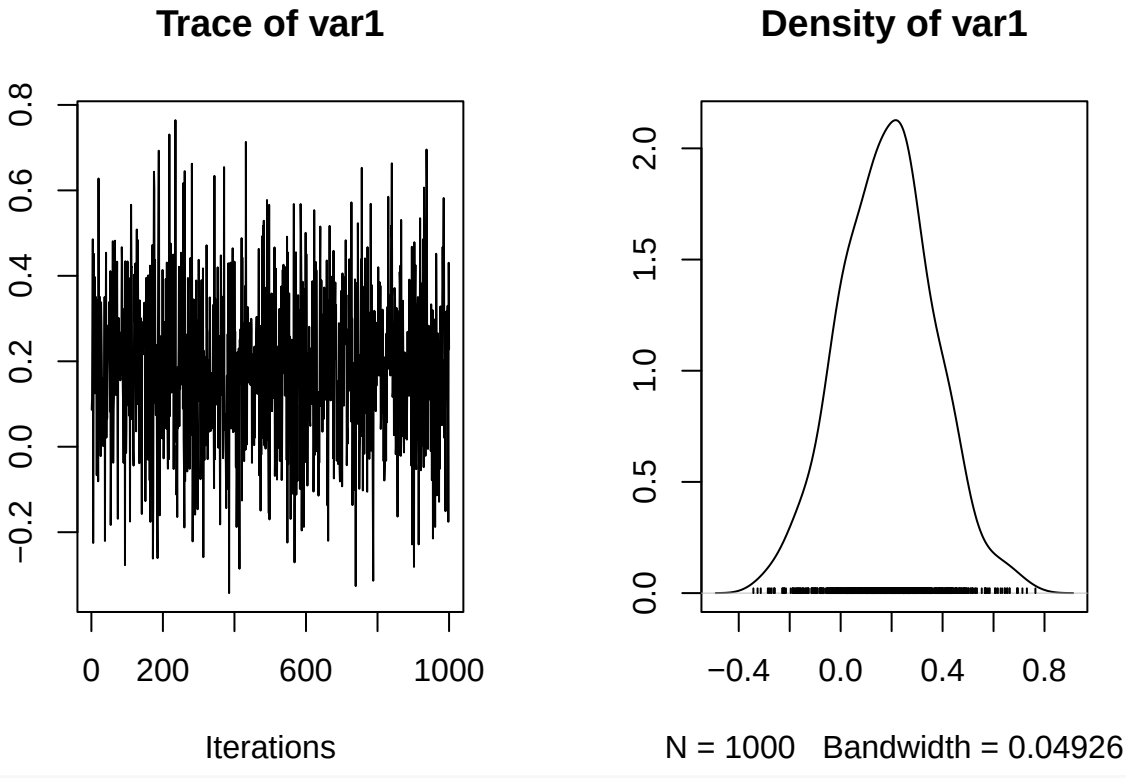
```
# Wide
ttestBF(formula = CSE ~ Condition, data = d, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 0.2374207 ±0.06%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
csepost.w <- ttestBF(formula = CSE ~ Condition, data = d, rscale = "wide", posterior = TRUE, iterations
plot(csepost.w[,"mu"])
```




```
plot(csepost.w[,"beta (Fixed - Growth)"])
```



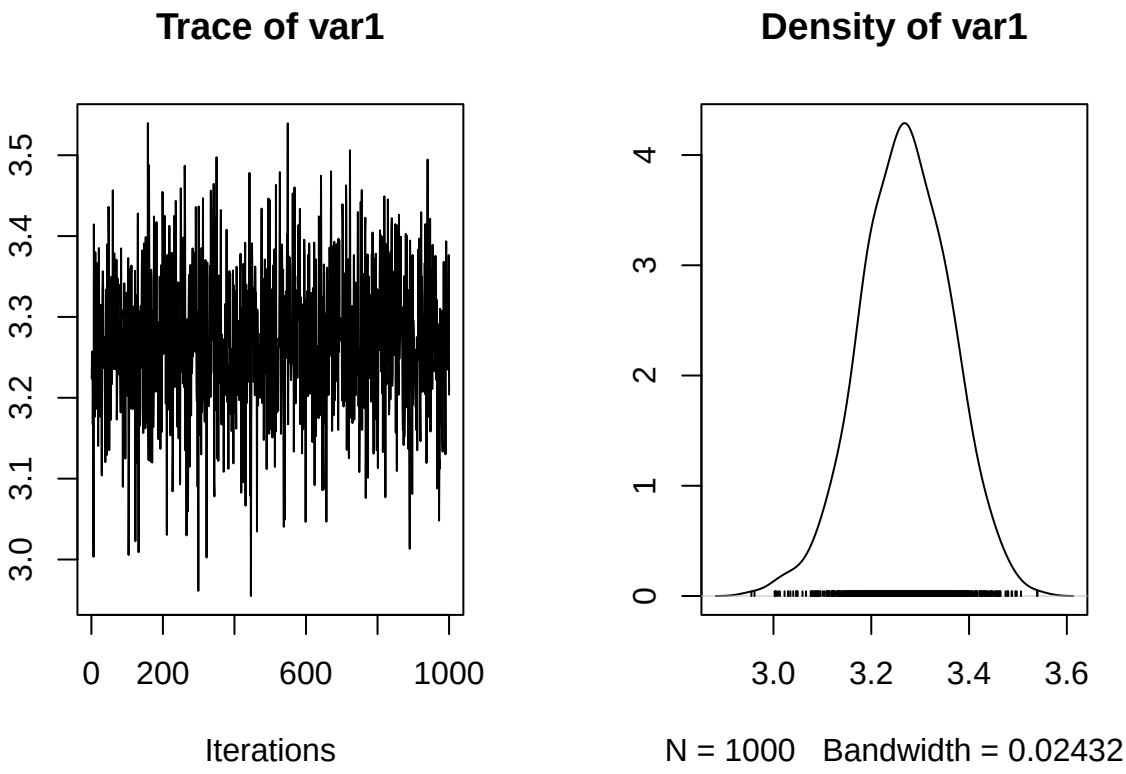

```
quantile(csepost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1760123  0.1823150  0.5662757
```

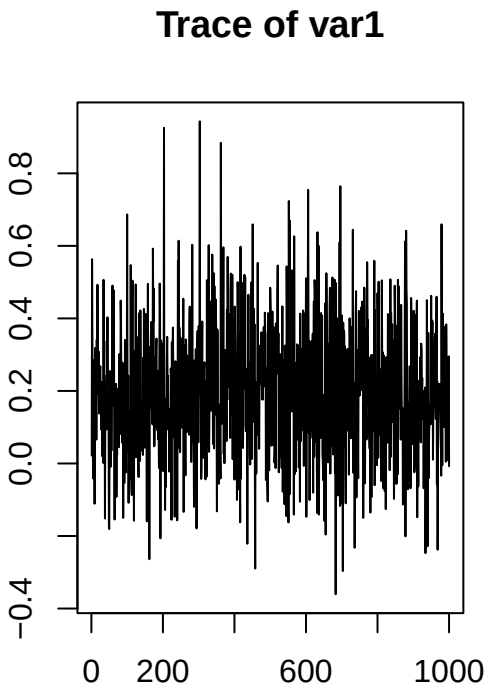
```r
# ultra Wide
ttestBF(formula = CSE ~ Condition, data = d, rscale = "ultrawide")
```

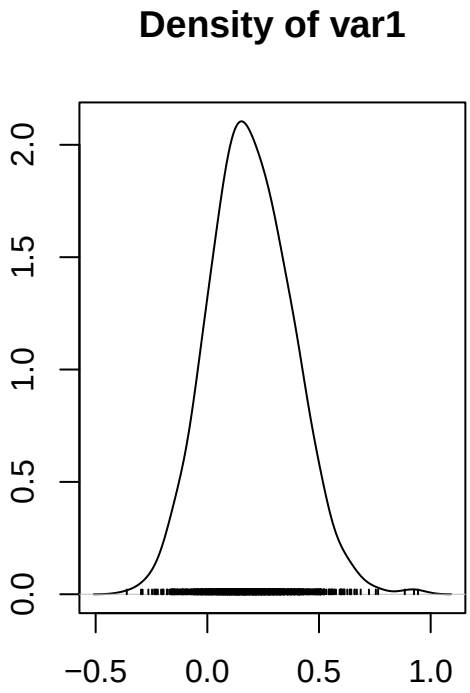
```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 0.173071 ±0.1%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```r
csepost.uw <- ttestBF(formula = CSE ~ Condition, data = d, rscale = "ultrawide", posterior = TRUE, itera
plot(csepost.uw[,"mu"])
```




```r
plot(csepost.uw[,"beta (Fixed - Growth)"])
```

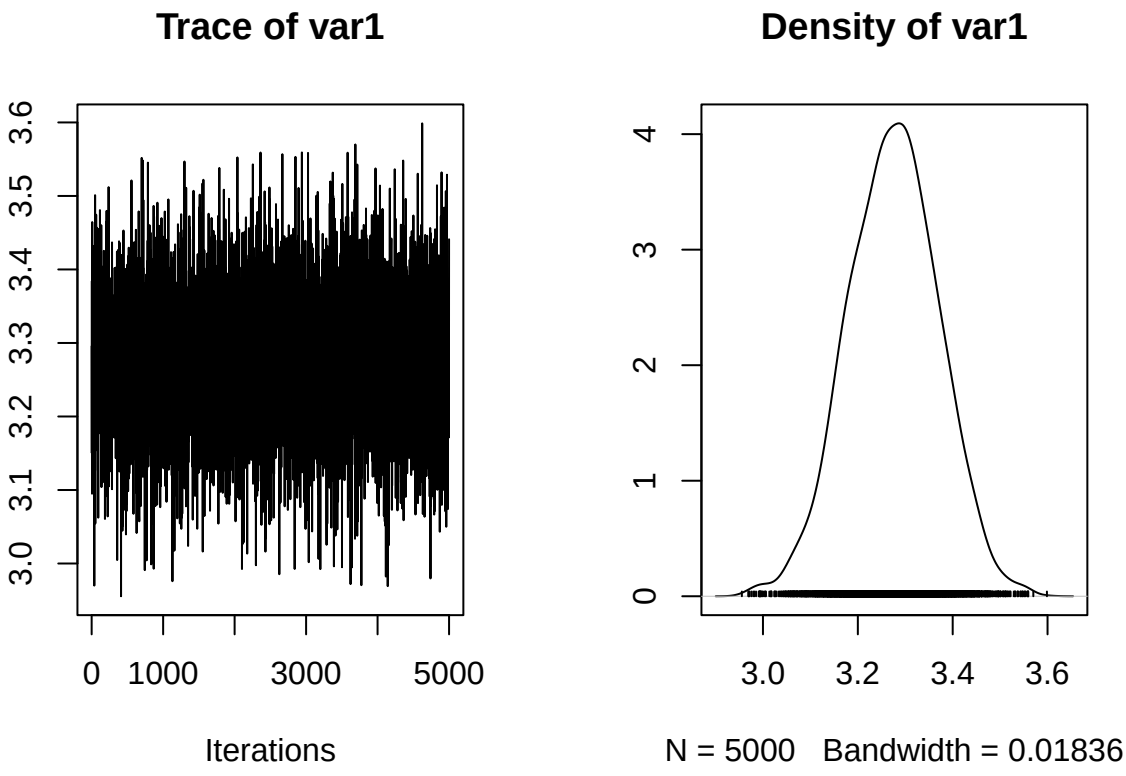




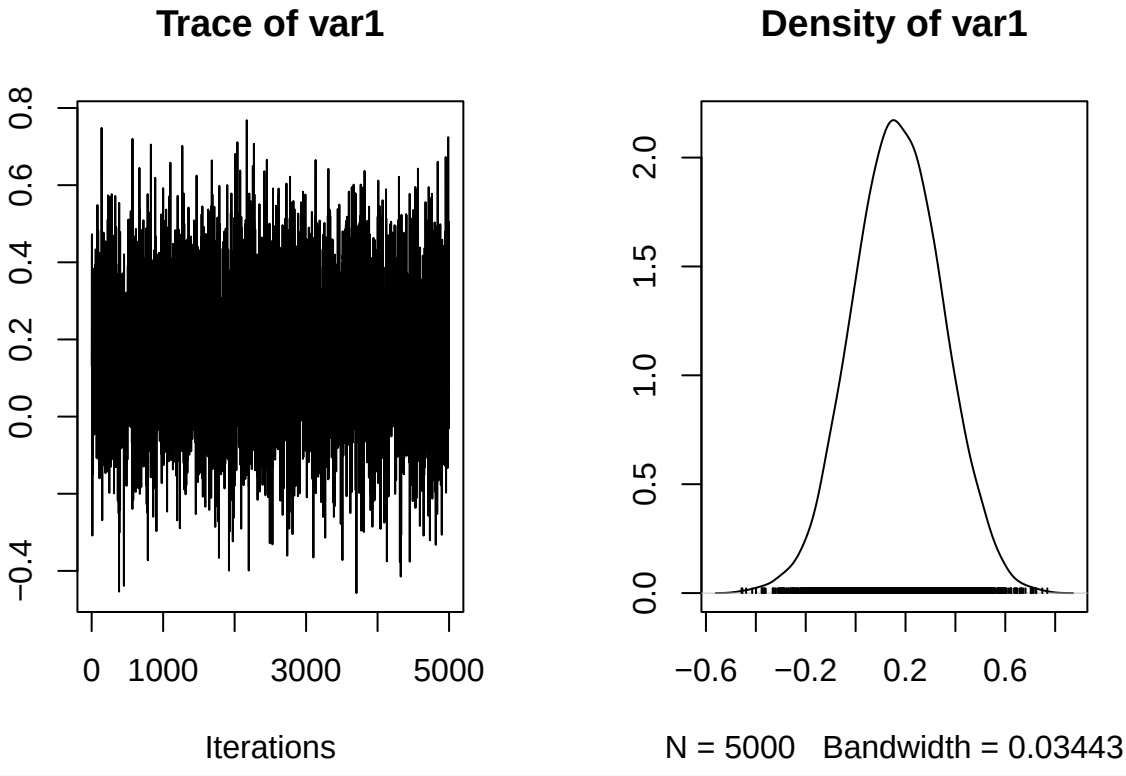

```
quantile(csepost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1499714  0.1867698  0.5634292
```

```
### more iterations
csepost.check <- ttestBF(formula = CSE ~ Condition, data = d, posterior = TRUE, iterations = 5000)
plot(csepost.check[,"mu"])
```




```
plot(csepost.check[,"beta (Fixed - Growth)"])
```




```
quantile(csepost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

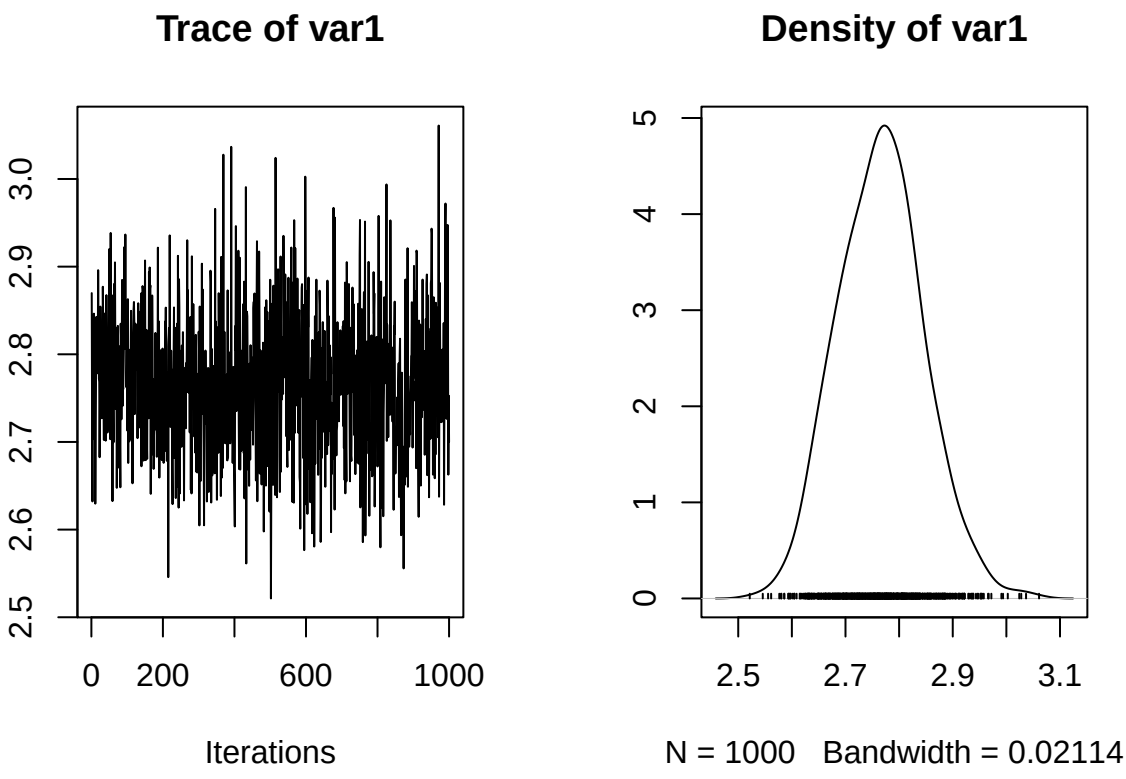
```
##       2.5%        50%      97.5%
## -0.1781354  0.1673873  0.5169461
```

Joan's problem ratings

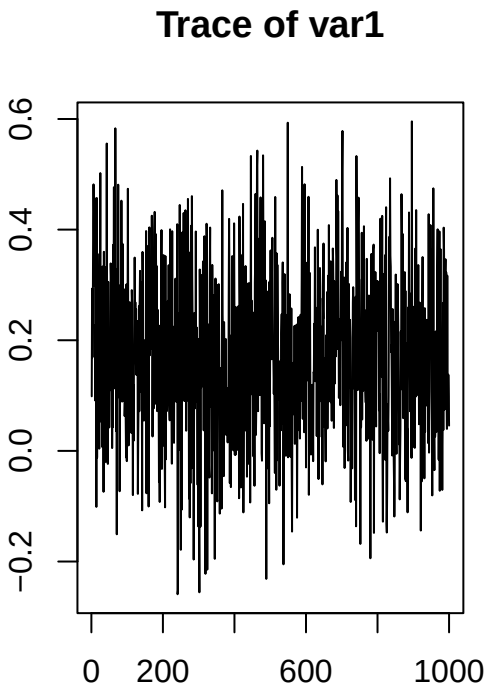
```
# default / medium
ttestBF(formula = JoanScore ~ Condition, data = d)
```

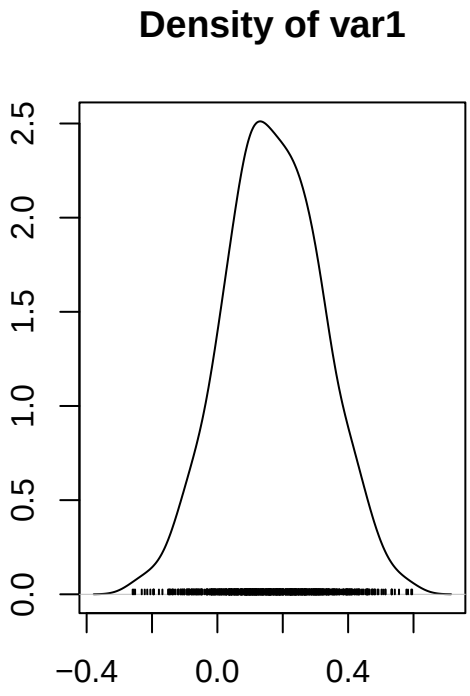
```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.3574592 ±0.02%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
joanpost <- ttestBF(formula = JoanScore ~ Condition, data = d, posterior = TRUE, iterations = 1000)

plot(joanpost[,"mu"])
```




```
plot(joanpost[,"beta (Fixed - Growth)"])
```





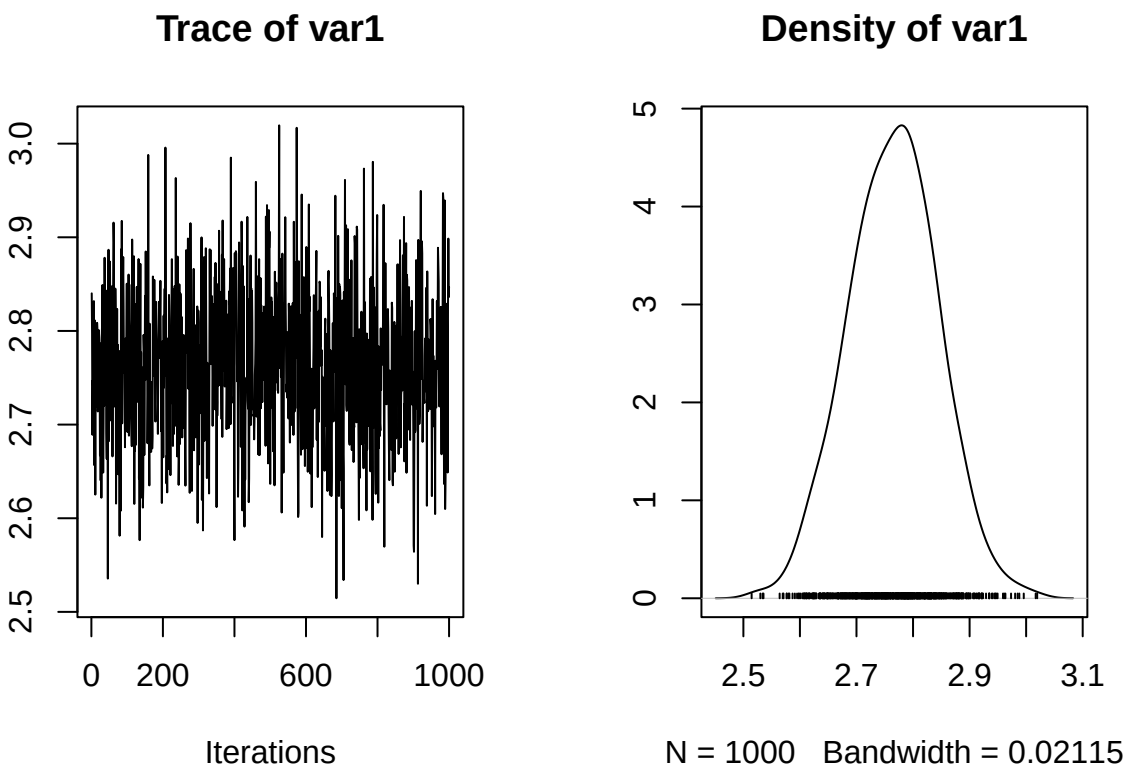

```
quantile(joanpost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1186595  0.1673654  0.4605831
```

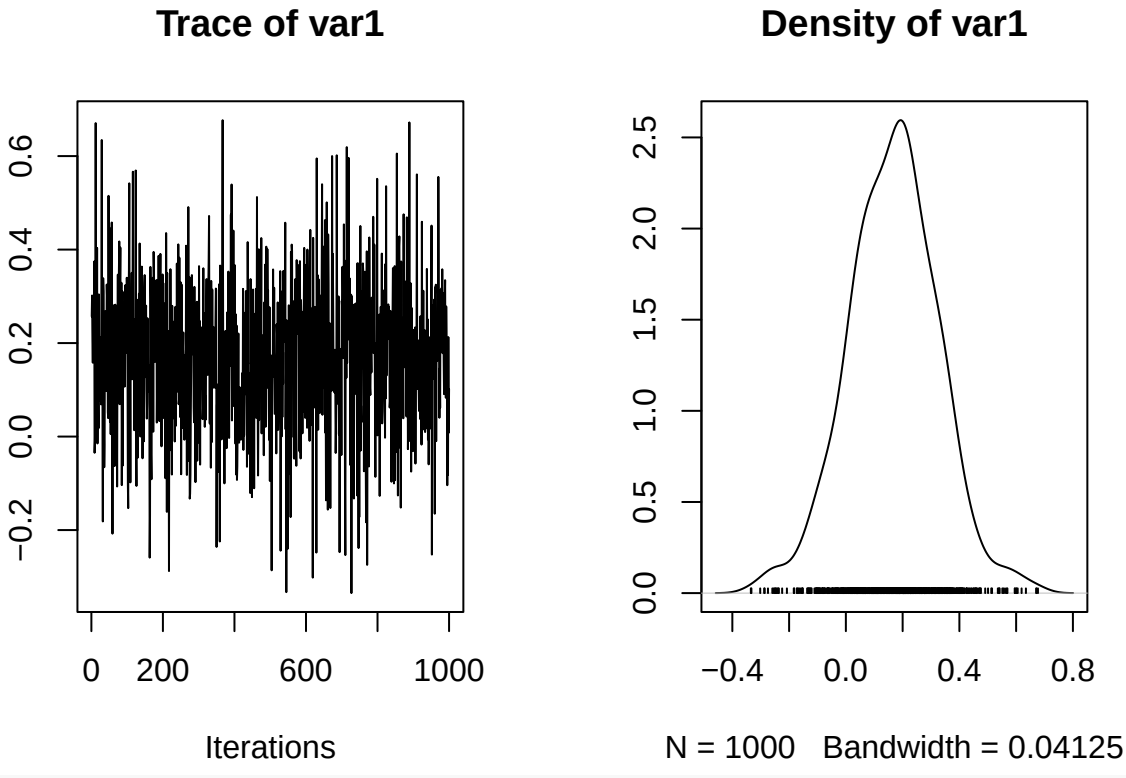
```
# wide
ttestBF(formula = JoanScore ~ Condition, data = d, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 0.2680115 ±0.05%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
joanpost.w <- ttestBF(formula = JoanScore ~ Condition, data = d, rscale = "wide", posterior = TRUE, ite

plot(joanpost.w[,"mu"])
```




```
plot(joanpost.w[,"beta (Fixed - Growth)"])
```



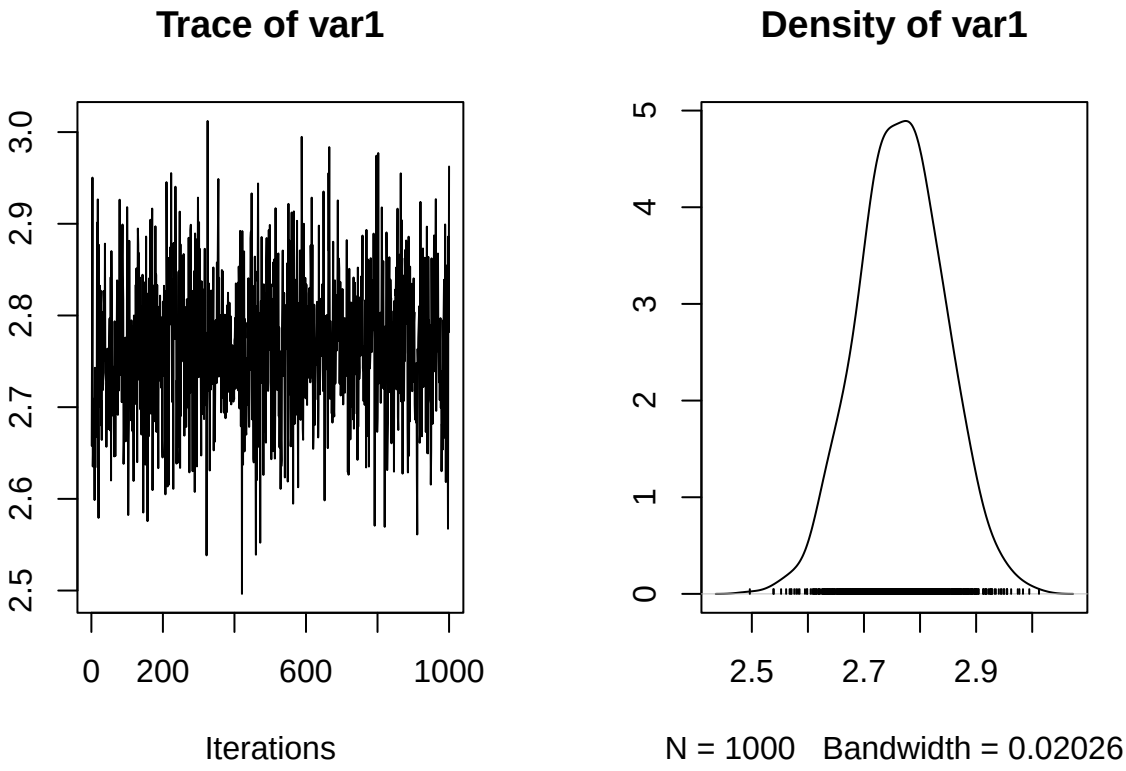

```
quantile(joanpost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1531376  0.1739365  0.4720155
```

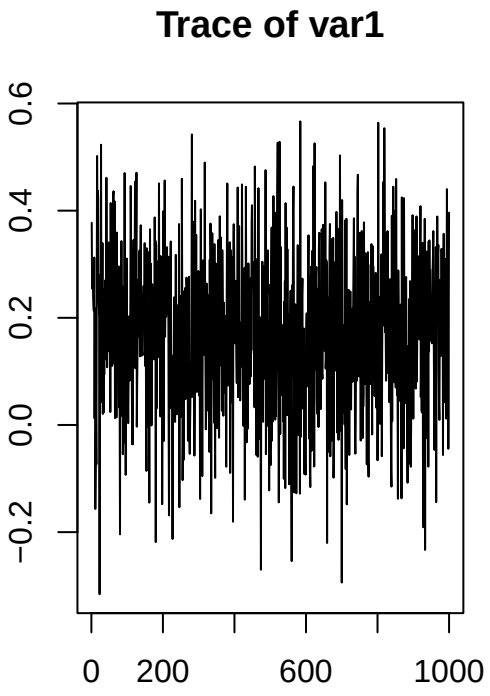
```
# ultrawide
ttestBF(formula = JoanScore ~ Condition, data = d, rscale = "ultrawide")
```

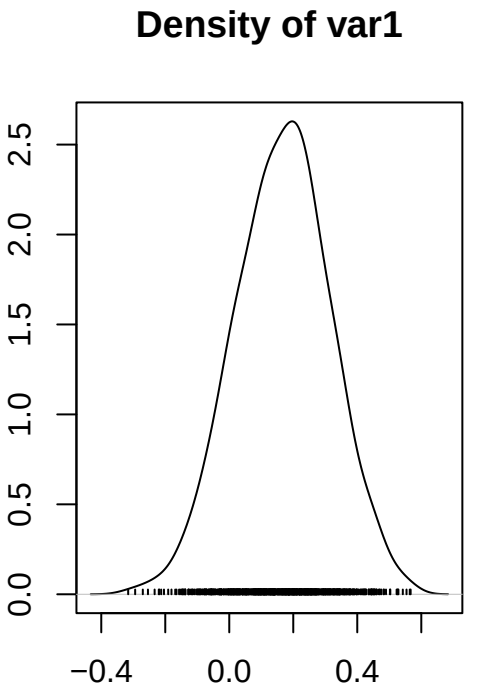
```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 0.1960675 ±0.09%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
joanpost.uw <- ttestBF(formula = JoanScore ~ Condition, data = d, rscale = "ultrawide", posterior = TRU

plot(joanpost.uw[,"mu"])
```




```
plot(joanpost.uw[,"beta (Fixed - Growth)"])
```

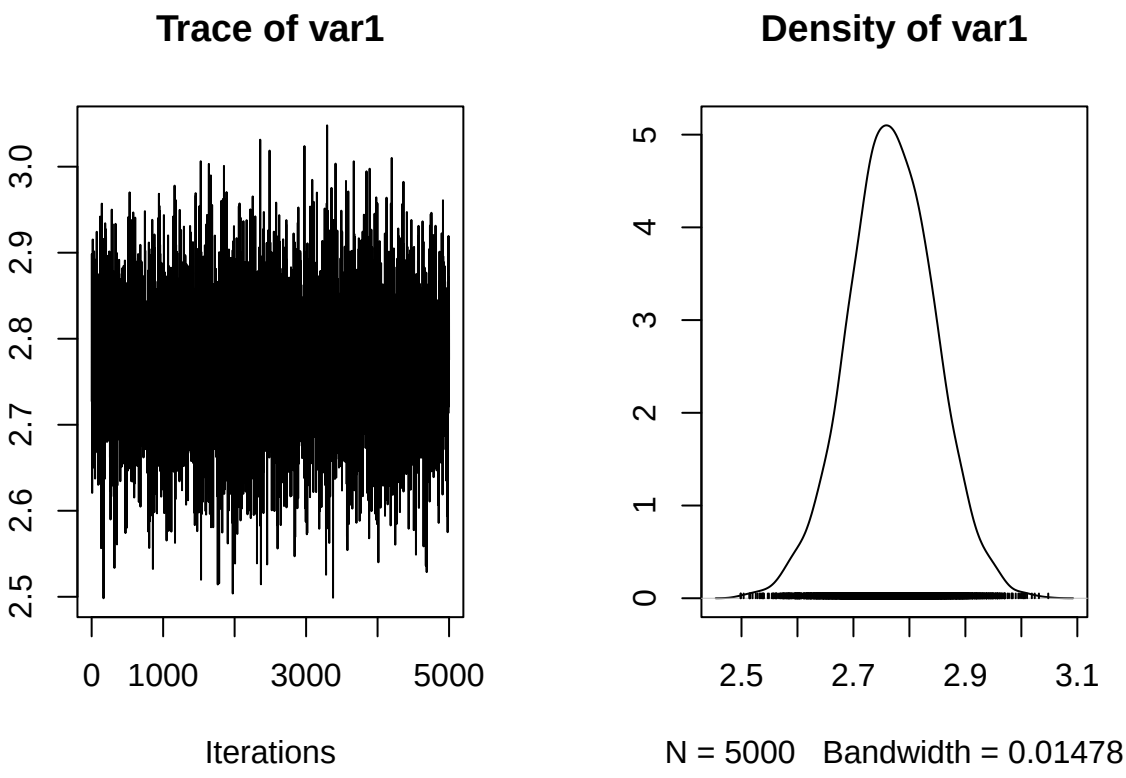




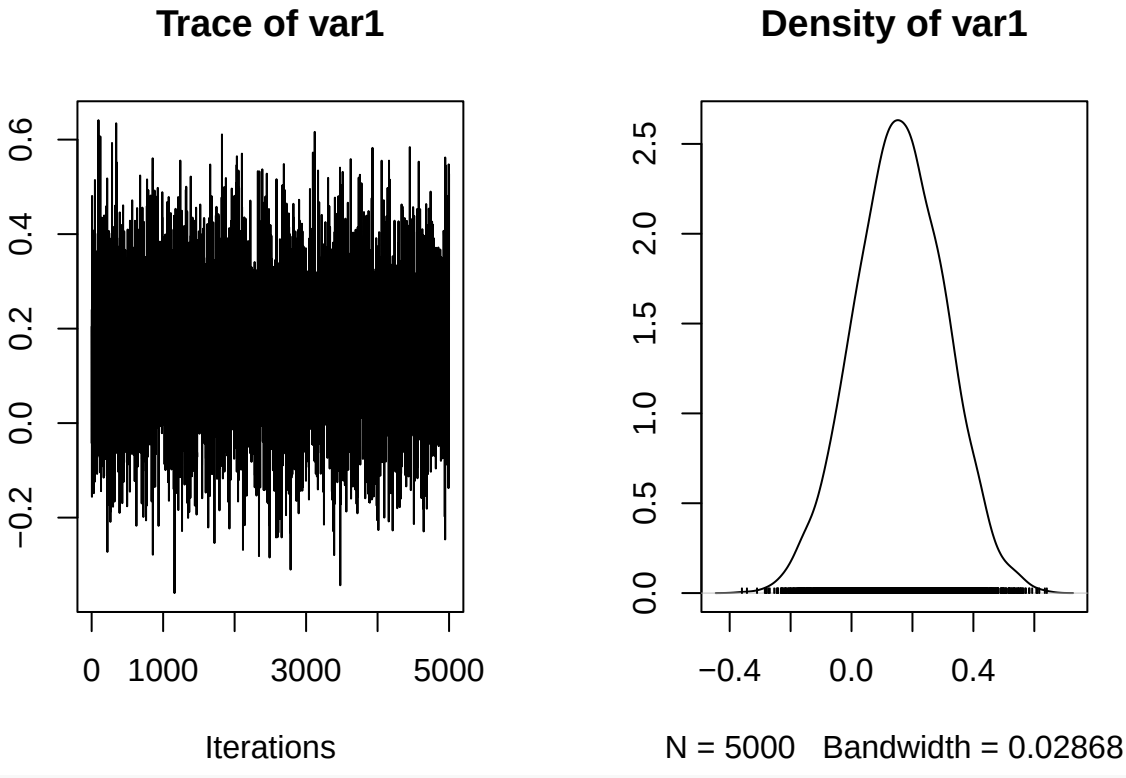

```
quantile(joanpost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1274836  0.1734112  0.4507211
```

```
### more iterations
joanpost.check <- ttestBF(formula = JoanScore ~ Condition, data = d, posterior = TRUE, iterations = 500

plot(joanpost.check[,"mu"])
```




```
plot(joanpost.check[,"beta (Fixed - Growth)"])
```




```
quantile(joanpost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.1395923  0.1578671  0.4416287
```

Divergent thinking

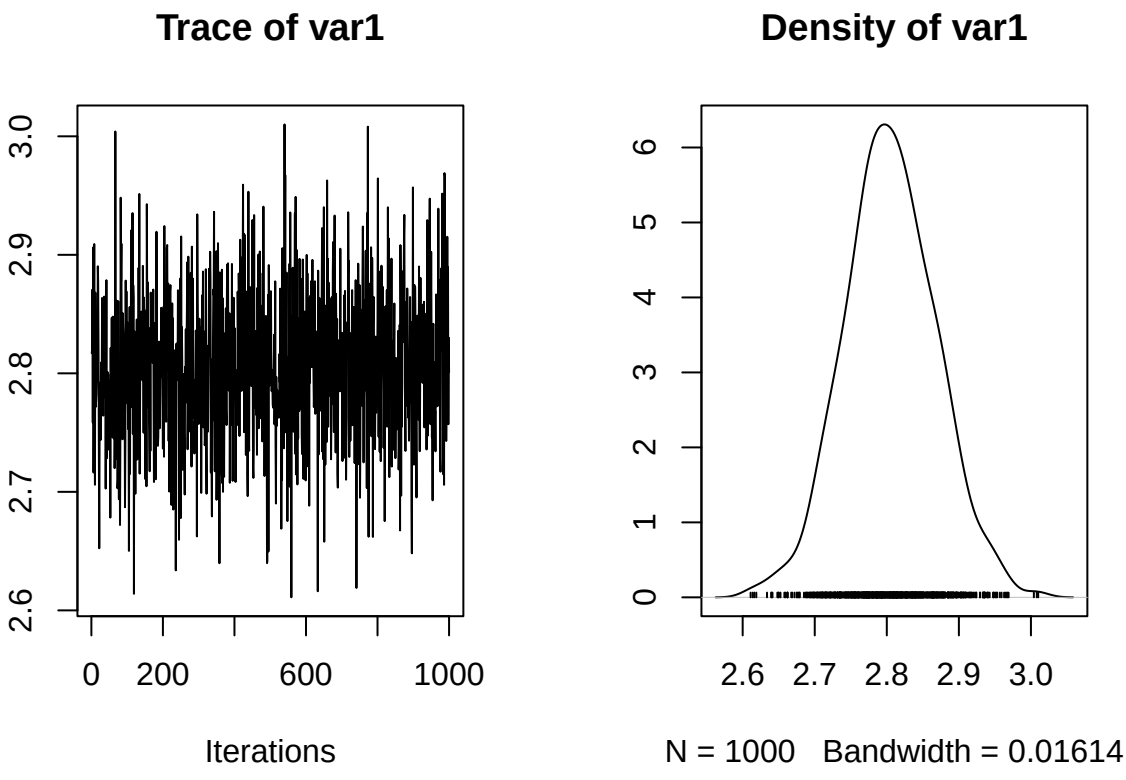
note, two participants failed to respond to the DT prompts (row 17 and 25) and were not included in this analysis

Divergent Thinking Originality

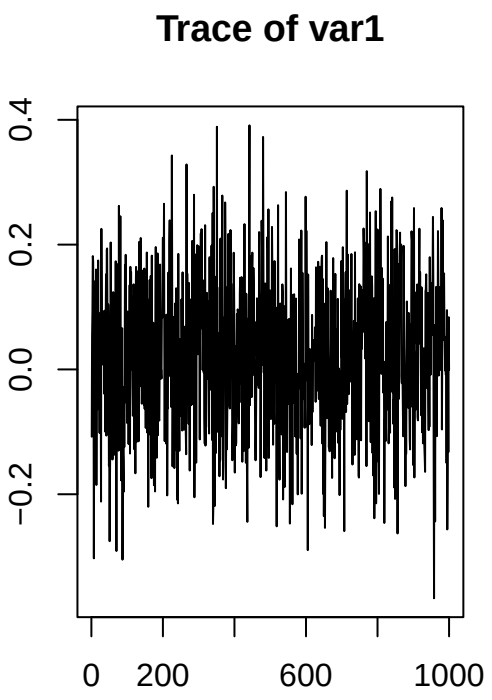
```
## first make the subset data
dt.subset <- na.omit(d[,c("DTScore","DTFlu","Condition")])

## DT originality ratings

# Default / medium
ttestBF(formula = DTScore ~ Condition, data = dt.subset)
```

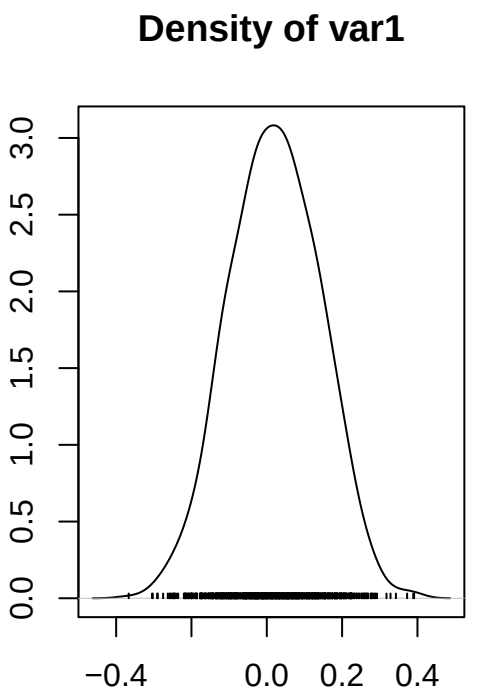
```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.2056849 ±0.03%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
dtpost <- ttestBF(formula = DTScore ~ Condition, data = dt.subset, posterior = TRUE, iterations = 1000)

plot(dtpost[,"mu"])
```




```
plot(dtpost[,"beta (Fixed - Growth)"])
```





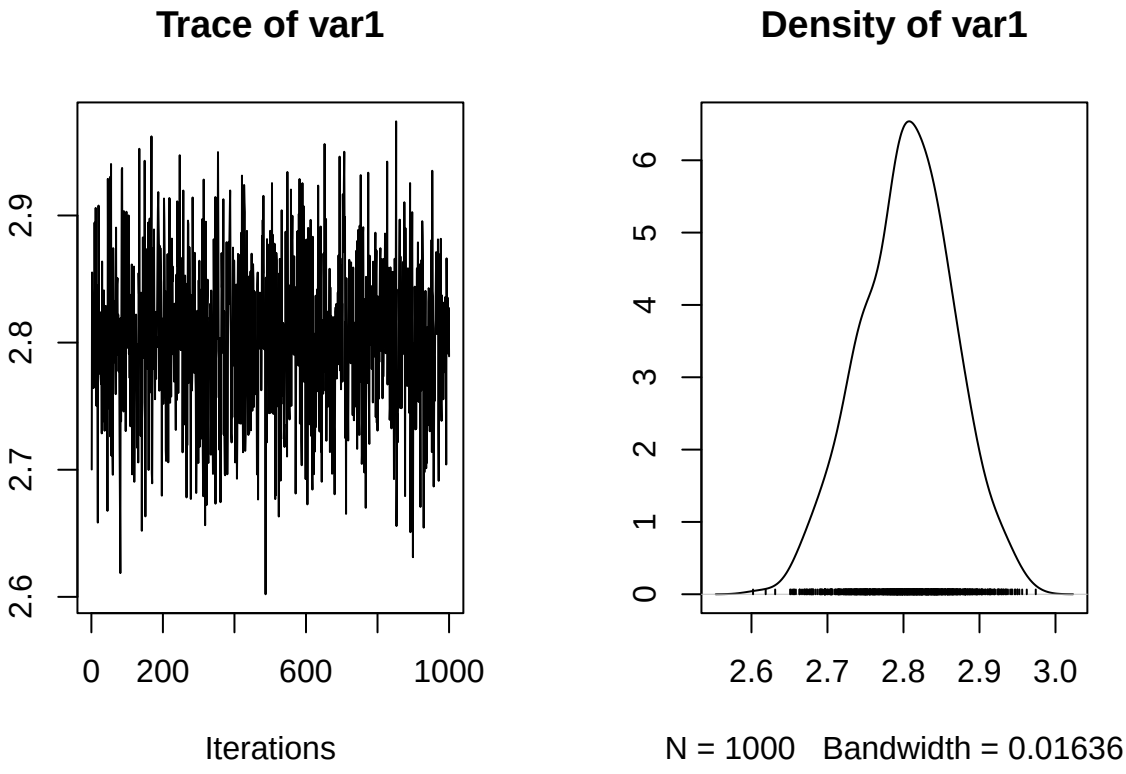

```
quantile(dtpost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##        2.5%         50%       97.5%
## -0.20890209  0.02049619  0.24566789
```

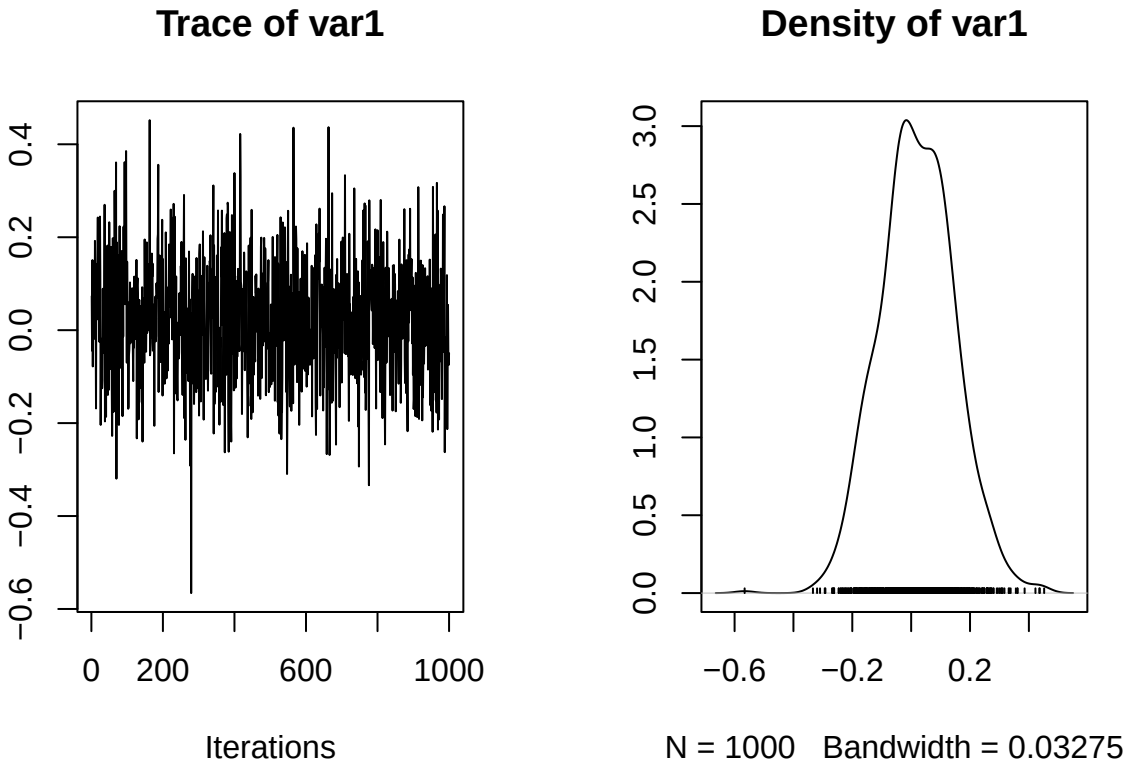
```
# wide
ttestBF(formula = DTScore ~ Condition, data = dt.subset, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 0.1498916 ±0.08%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
dtpost.w <- ttestBF(formula = DTScore ~ Condition, data = dt.subset, rscale = "wide", posterior = TRUE,

plot(dtpost.w[,"mu"])
```




```
plot(dtpost.w[,"beta (Fixed - Growth)"])
```



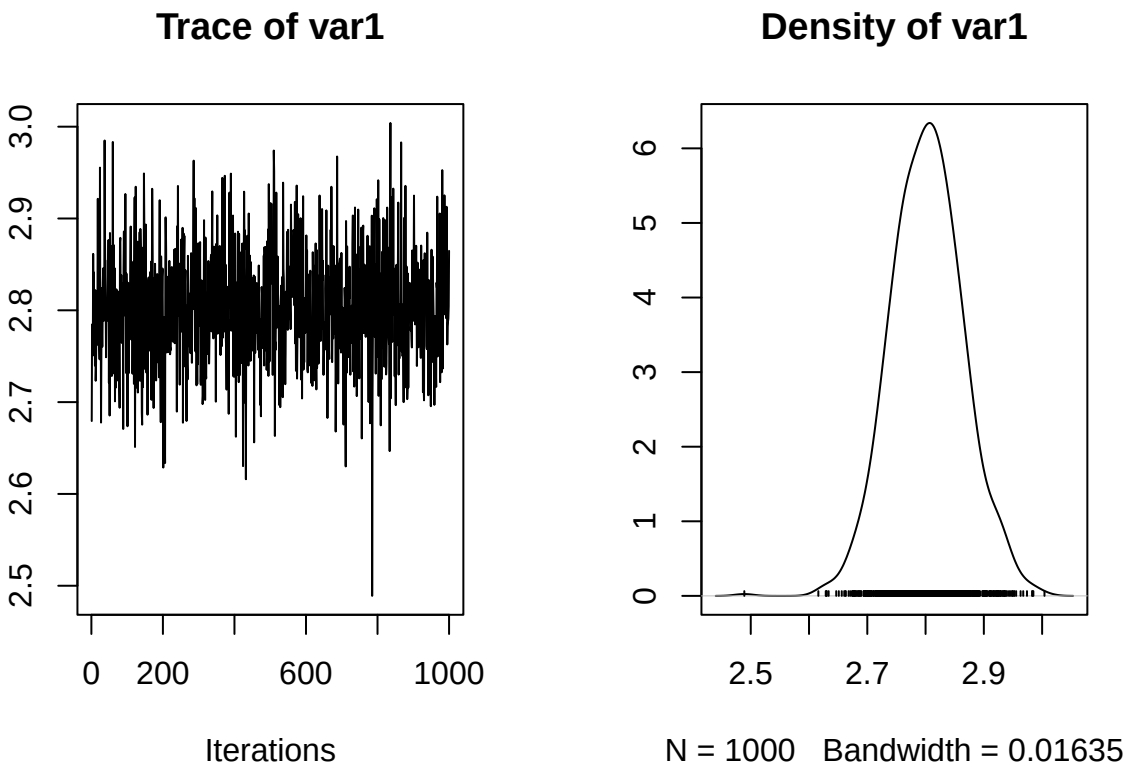

```
quantile(dtpost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -0.2174833  0.0188377  0.2685011
```

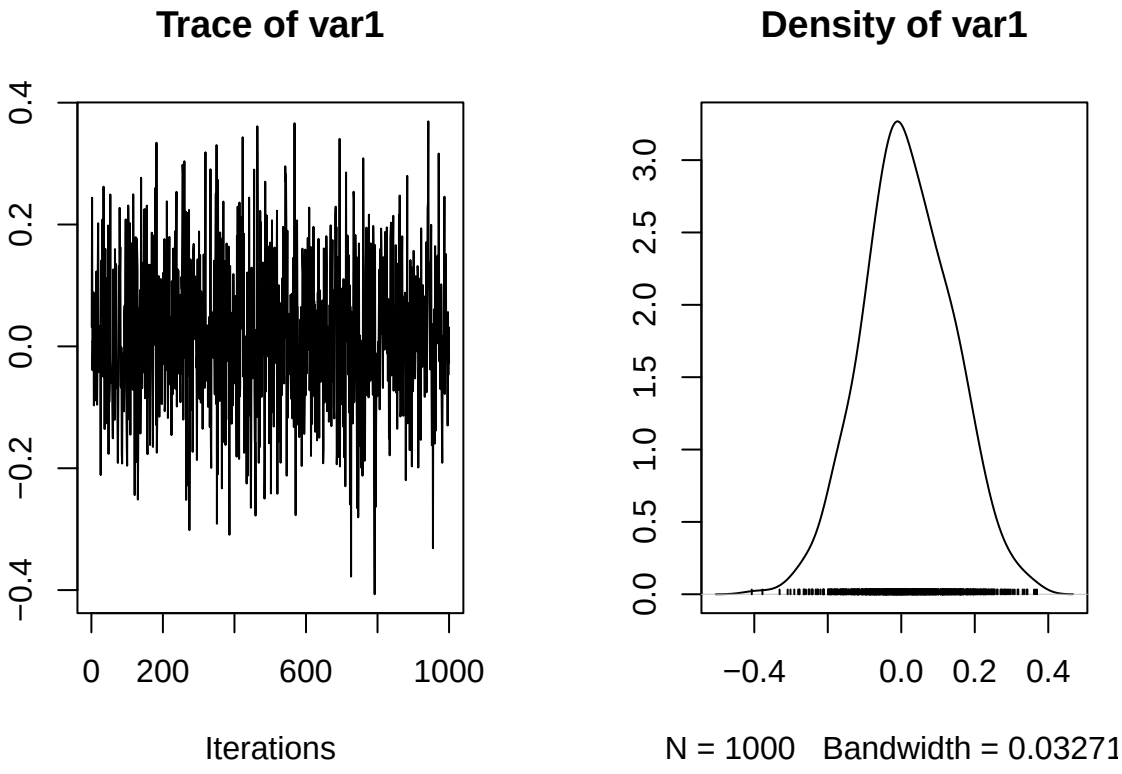
```
# ultra wide
ttestBF(formula = DTScore ~ Condition, data = dt.subset, rscale = "ultrawide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 0.1077728 ±0.15%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
dtpost.uw <- ttestBF(formula = DTScore ~ Condition, data = dt.subset, rscale = "ultrawide", posterior =

plot(dtpost.uw[,"mu"])
```



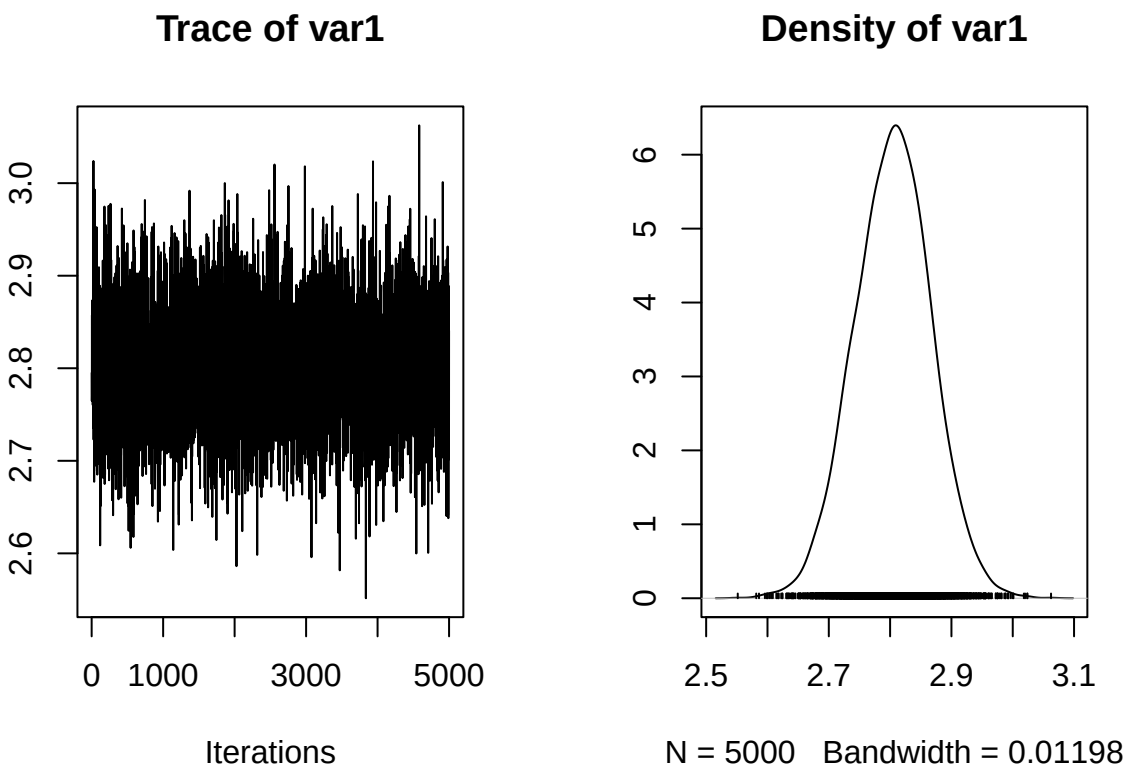

```
plot(dtpost.uw[,"beta (Fixed - Growth)"])
```



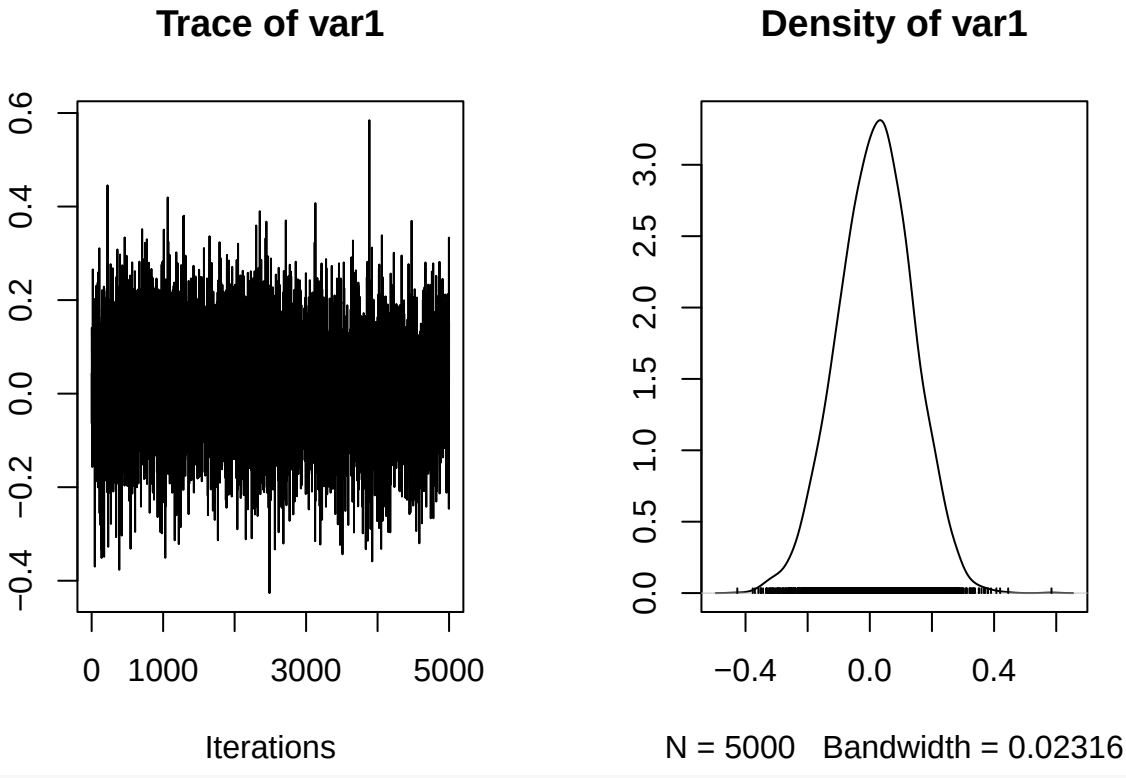

```
quantile(dtpost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##        2.5%         50%       97.5%
## -0.21249611  0.01019647  0.25403206
```

```
# More iterations
dtpost.check <- ttestBF(formula = DTScore ~ Condition, data = dt.subset, posterior = TRUE, iterations =

plot(dtpost.check[,"mu"])
```




```
plot(dtpost.check[,"beta (Fixed - Growth)"])
```




```
quantile(dtpost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

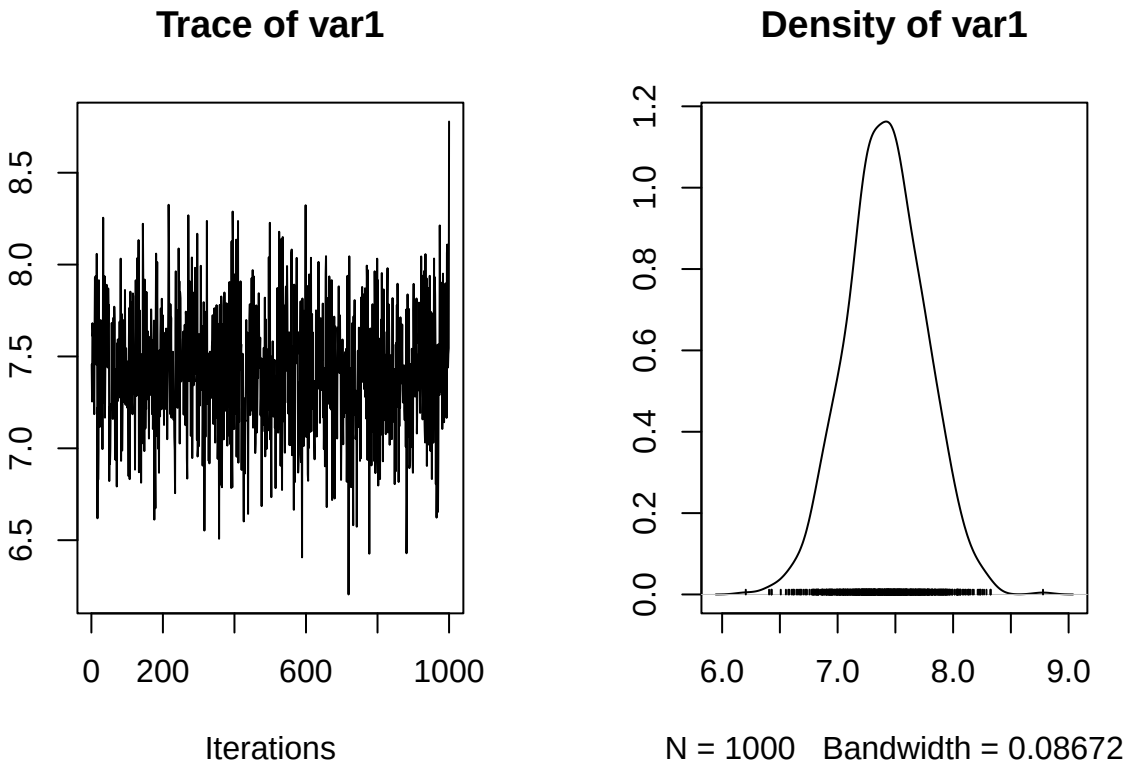
```
##        2.5%         50%       97.5%
## -0.21780014  0.02103769  0.24612178
```

DT fluency

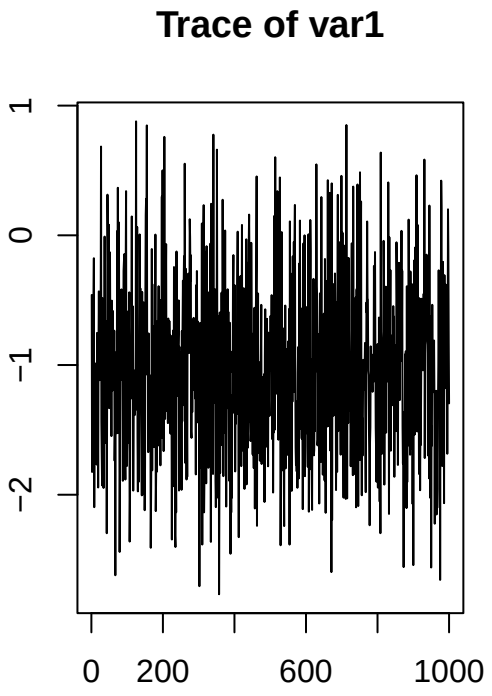
```
# default / medium
ttestBF(formula = DTFlu ~ Condition, data = dt.subset)
```

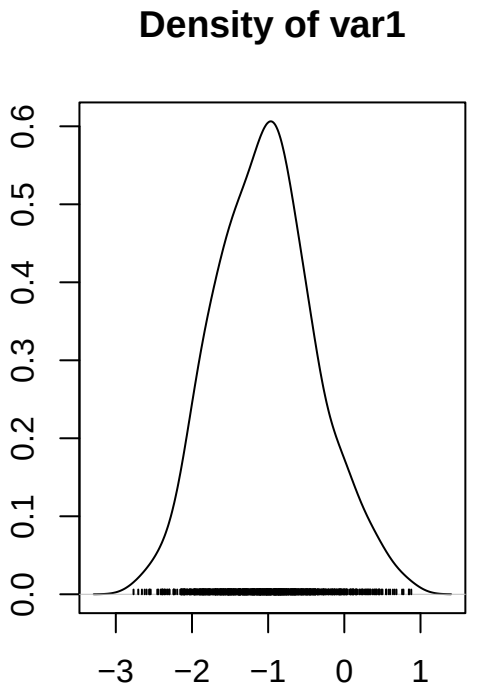
```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.7210424 ±0.02%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
flupost <- ttestBF(formula = DTFlu ~ Condition, data = dt.subset, posterior = TRUE, iterations = 1000)

plot(flupost[,"mu"])
```




```
plot(flupost[,"beta (Fixed - Growth)"])
```





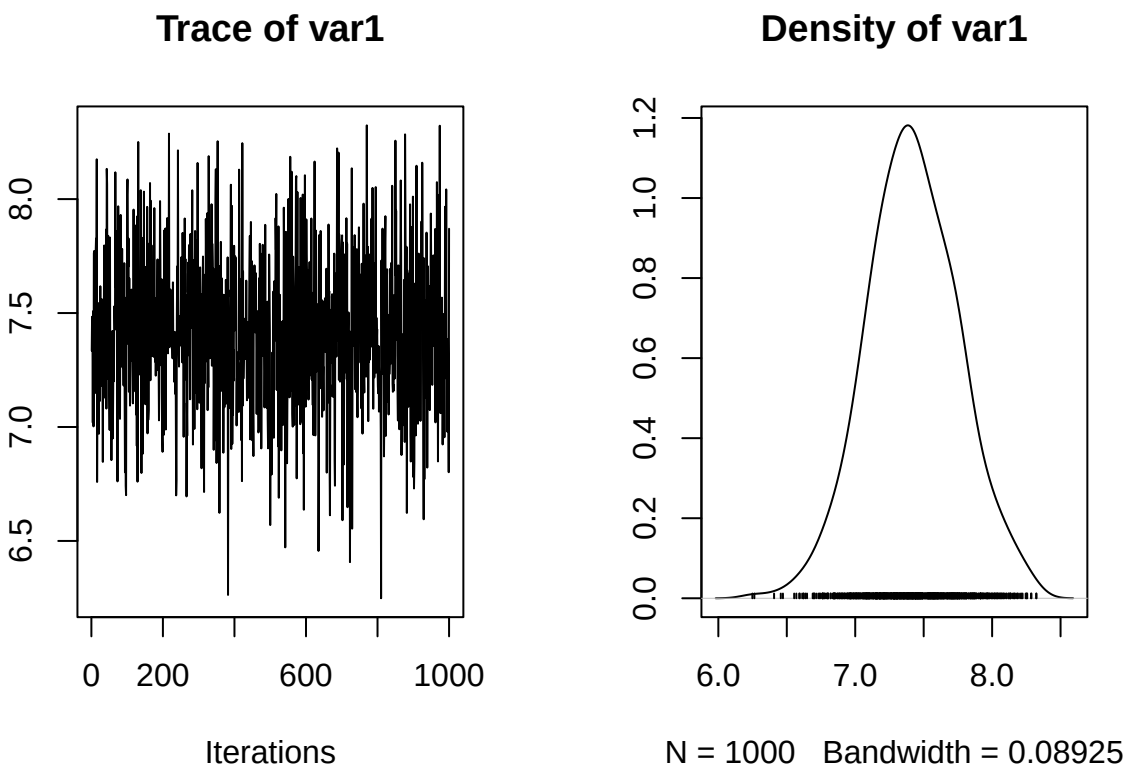

```
quantile(flupost[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -2.2237639 -1.0404673  0.3412296
```

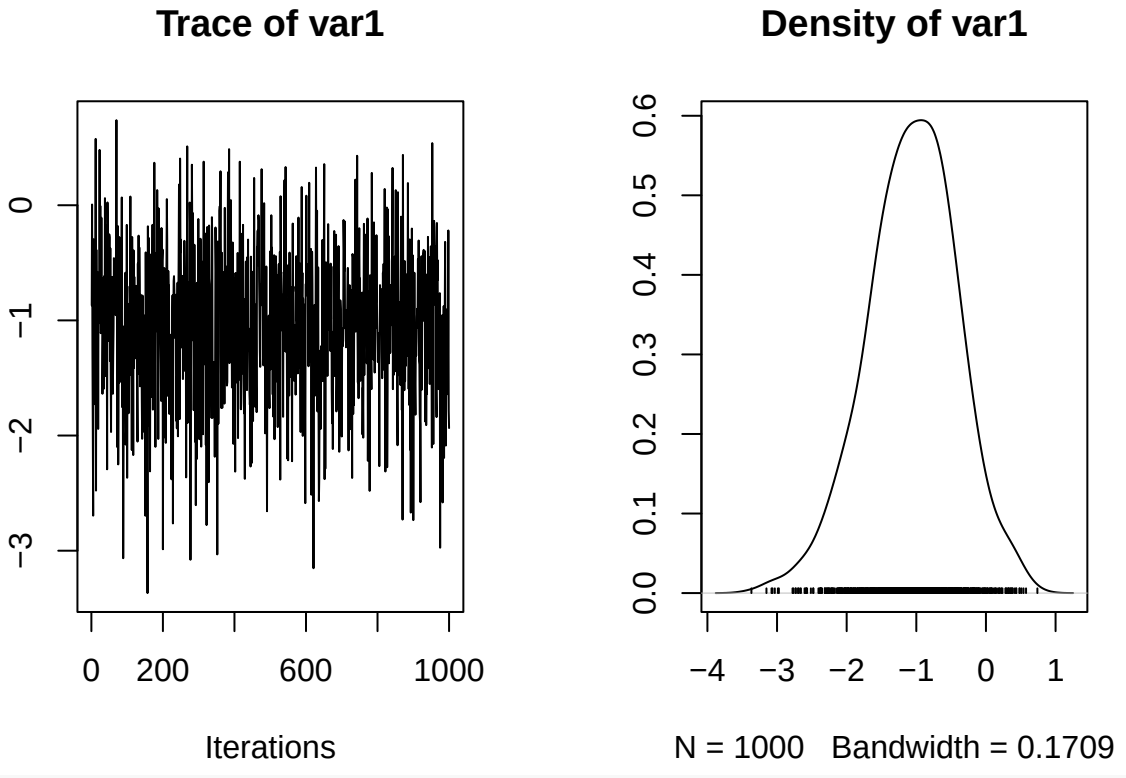
```
# wide

ttestBF(formula = DTFlu ~ Condition, data = dt.subset, rscale = "wide")
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=1 : 0.5596128 ±0.03%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```
flupost.w <- ttestBF(formula = DTFlu ~ Condition, data = dt.subset, rscale = "wide", posterior = TRUE, i

plot(flupost.w[,"mu"])
```




```
plot(flupost.w[,"beta (Fixed - Growth)"])
```



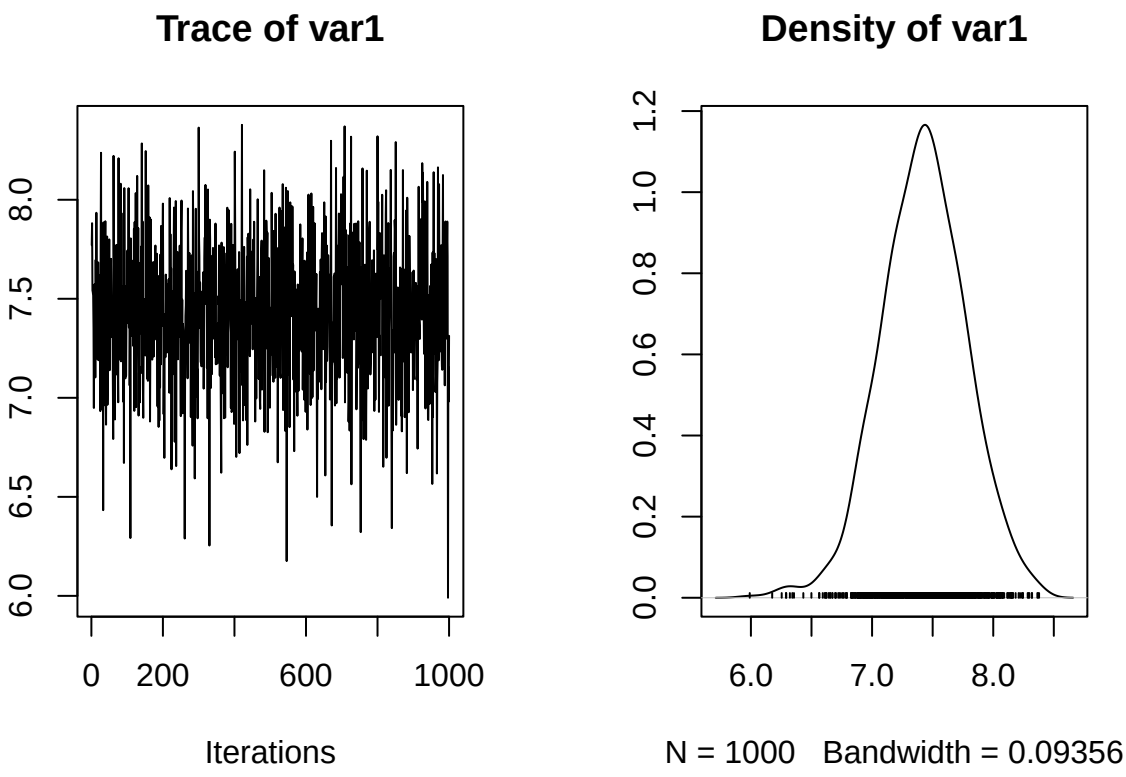

```
quantile(flupost.w[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -2.3803232 -1.0393014  0.1930067
```

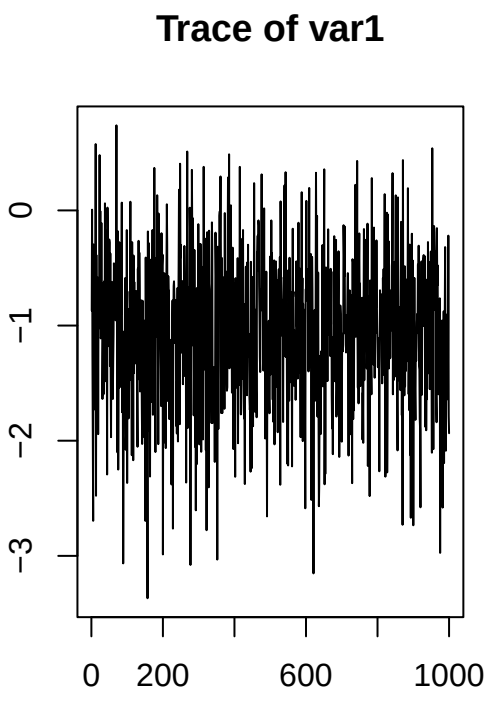
```r
# ultrawide

ttestBF(formula = DTFlu ~ Condition, data = dt.subset, rscale = "ultrawide")
```

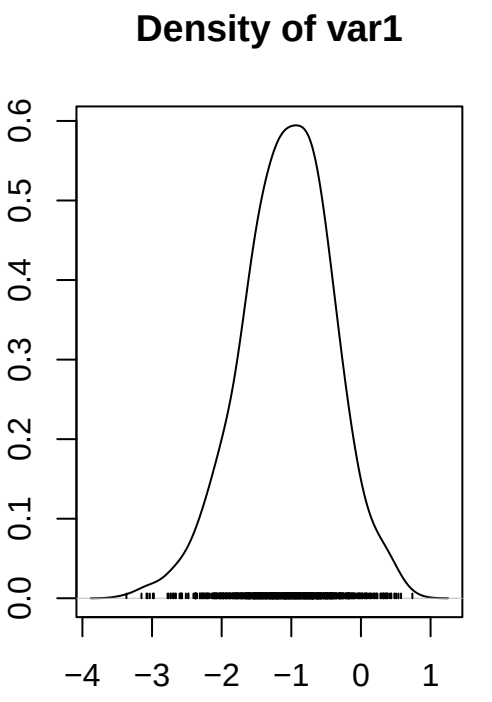
```
## Bayes factor analysis
## --------------
## [1] Alt., r=1.414 : 0.4183677 ±0.05%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

```r
flupost.uw <- ttestBF(formula = DTFlu ~ Condition, data = dt.subset, rscale = "ultrawide", posterior = 

plot(flupost.uw[,"mu"])
```




```r
plot(flupost.w[,"beta (Fixed - Growth)"])
```

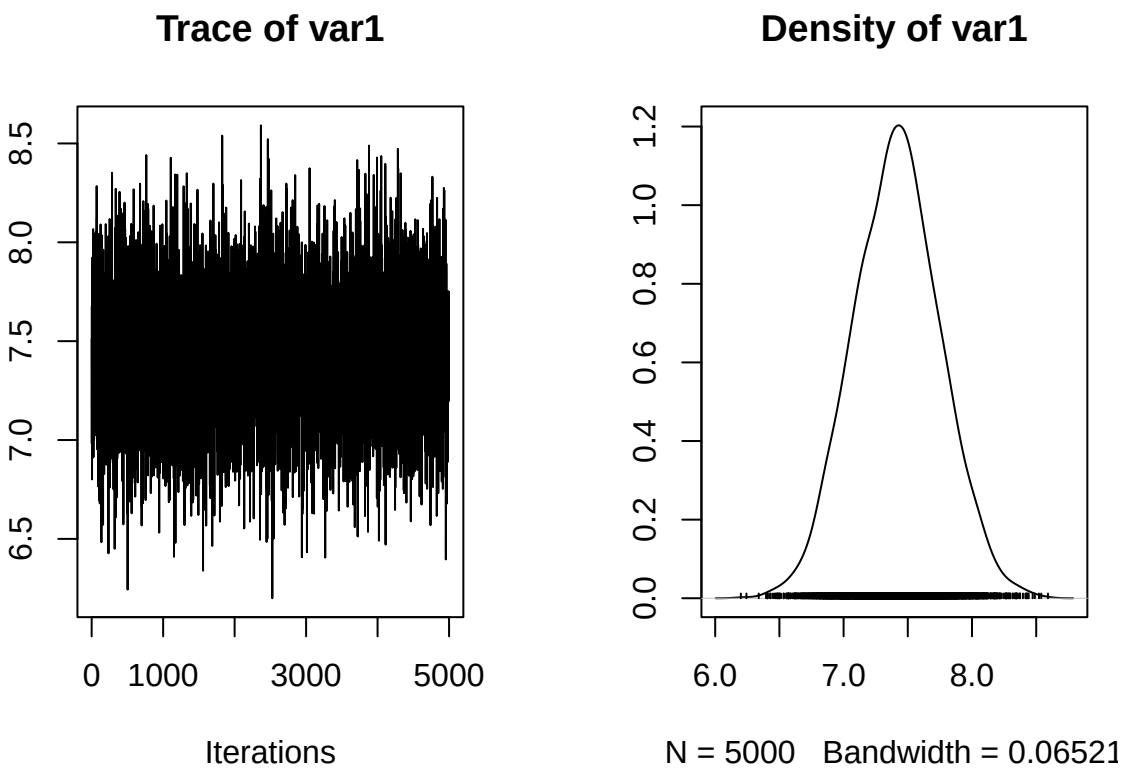




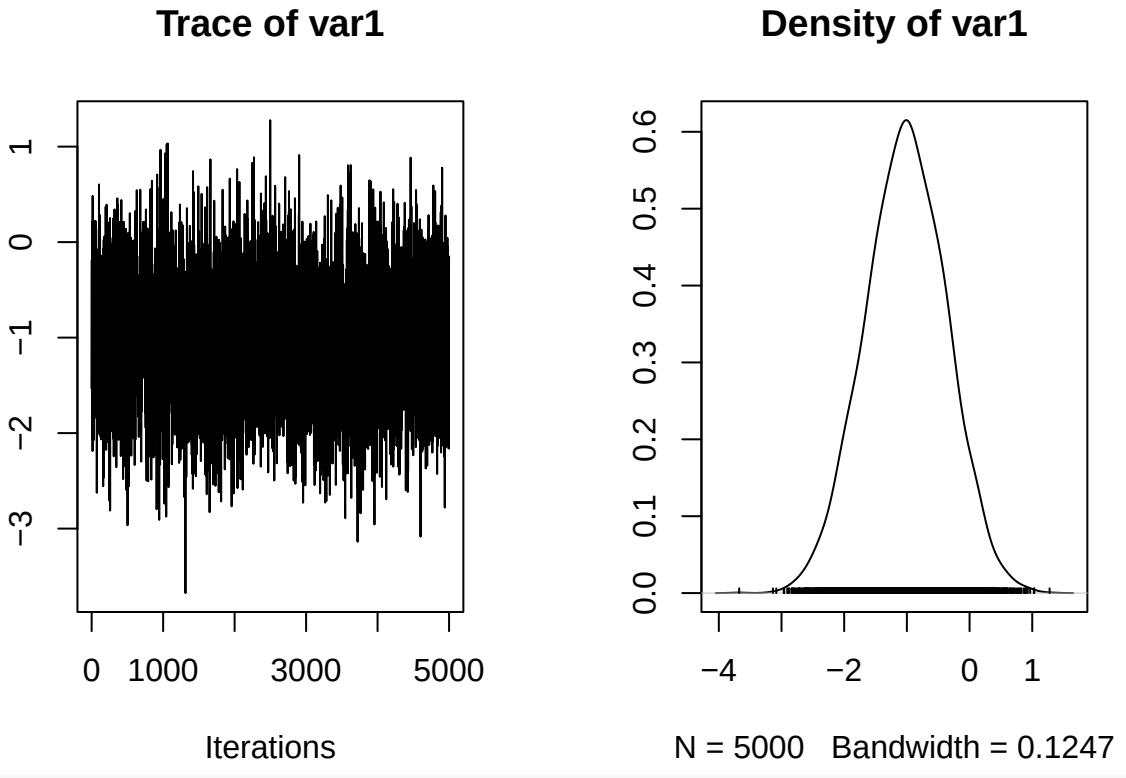

```
quantile(flupost.uw[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##       2.5%        50%      97.5%
## -2.4107607 -1.1280036  0.0965674
```

```
## More iterations
flupost.check <- ttestBF(formula = DTFlu ~ Condition, data = dt.subset, posterior = TRUE, iterations = 5

plot(flupost.check[,"mu"])
```




```
plot(flupost.check[,"beta (Fixed - Growth)"])
```




```
quantile(flupost.check[,"beta (Fixed - Growth)"], probs = c(.025, .5, .975))
```

```
##      2.5%       50%     97.5%
## -2.300783 -1.011913  0.212568
```

Cohen's D for all effects

```
# effect sizes
cohen.d(Fixed ~ Condition, data = d)
```

```
## Call: cohen.d(x = Fixed ~ Condition, data = d)
## Cohen d statistic of difference between two means
##       lower effect upper
## Fixed -0.92  -0.54 -0.16
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.54
## r equivalent of difference between two means
## Fixed
## -0.26
```

```
cohen.d(Growth ~ Condition, data = d)
```

```
## Call: cohen.d(x = Growth ~ Condition, data = d)
## Cohen d statistic of difference between two means
##        lower effect upper
## Growth  0.24   0.63  1.01
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.63
## r equivalent of difference between two means
## Growth
##    0.3
```

```
cohen.d(CSE ~ Condition, data = d)
```

```
## Call: cohen.d(x = CSE ~ Condition, data = d)
## Cohen d statistic of difference between two means
##     lower effect upper
## CSE -0.57  -0.19  0.18
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.19
## r equivalent of difference between two means
##  CSE
## -0.1
```

```
cohen.d(JoanScore ~ Condition, data = d)
```

```
## Call: cohen.d(x = JoanScore ~ Condition, data = d)
## Cohen d statistic of difference between two means
##           lower effect upper
## JoanScore -0.59  -0.22  0.16
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.22
## r equivalent of difference between two means
## JoanScore
##     -0.11
```

```
cohen.d(DTScore ~ Condition, data = dt.subset)
```

```
## Call: cohen.d(x = DTScore ~ Condition, data = dt.subset)
## Cohen d statistic of difference between two means
##         lower effect upper
## DTScore -0.41  -0.03  0.34
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.033
## r equivalent of difference between two means
## DTScore
##   -0.02
```

```
cohen.d(DTFlu ~ Condition, data = dt.subset)
```

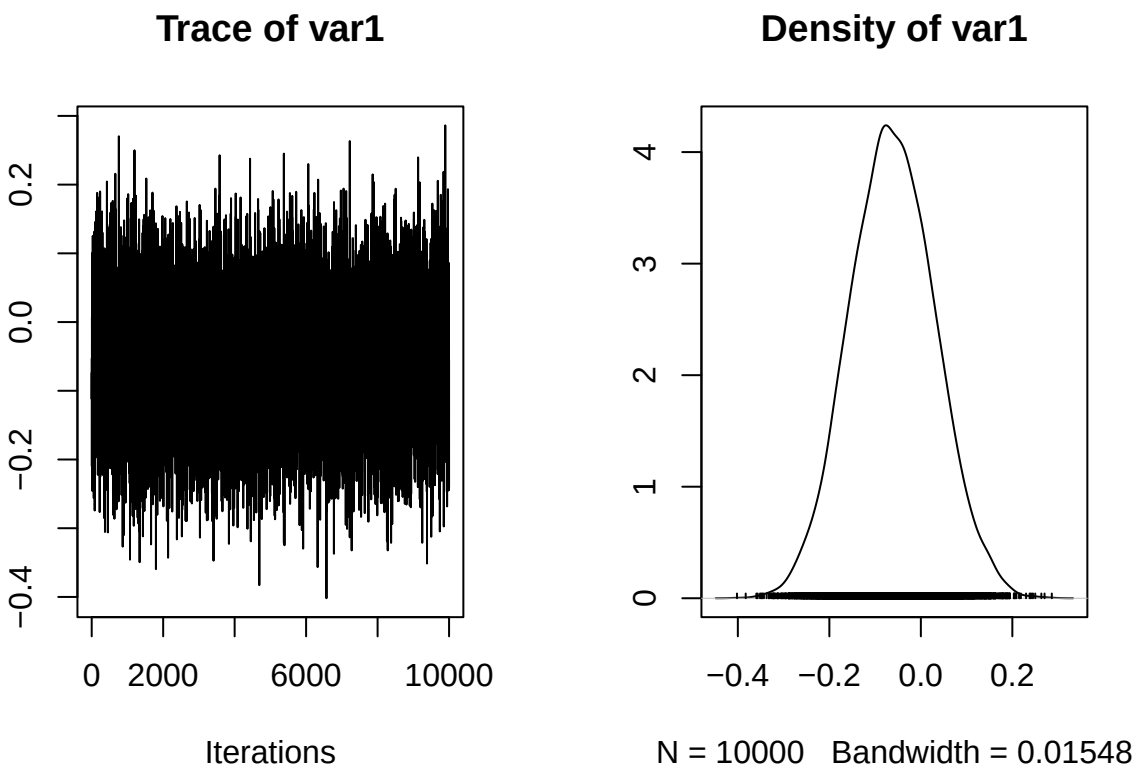
```
## Call: cohen.d(x = DTFlu ~ Condition, data = dt.subset)
## Cohen d statistic of difference between two means
##       lower effect upper
## DTFlu -0.05   0.33   0.7
##
## Multivariate (Mahalanobis) distance between groups
## [1] 0.33
## r equivalent of difference between two means
## DTFlu
##  0.16
```

Hypothesis 3 and 4

```
# correlations from BF including posterior

# hypothesis 3
samp3 <- correlationBF(y = d$JoanScore, x = d$Fixed, posterior = T, iterations = 10000)
```

```
## Independent-candidate M-H acceptance rate: 98%
```

```
plot(samp3[,"rho"])
```



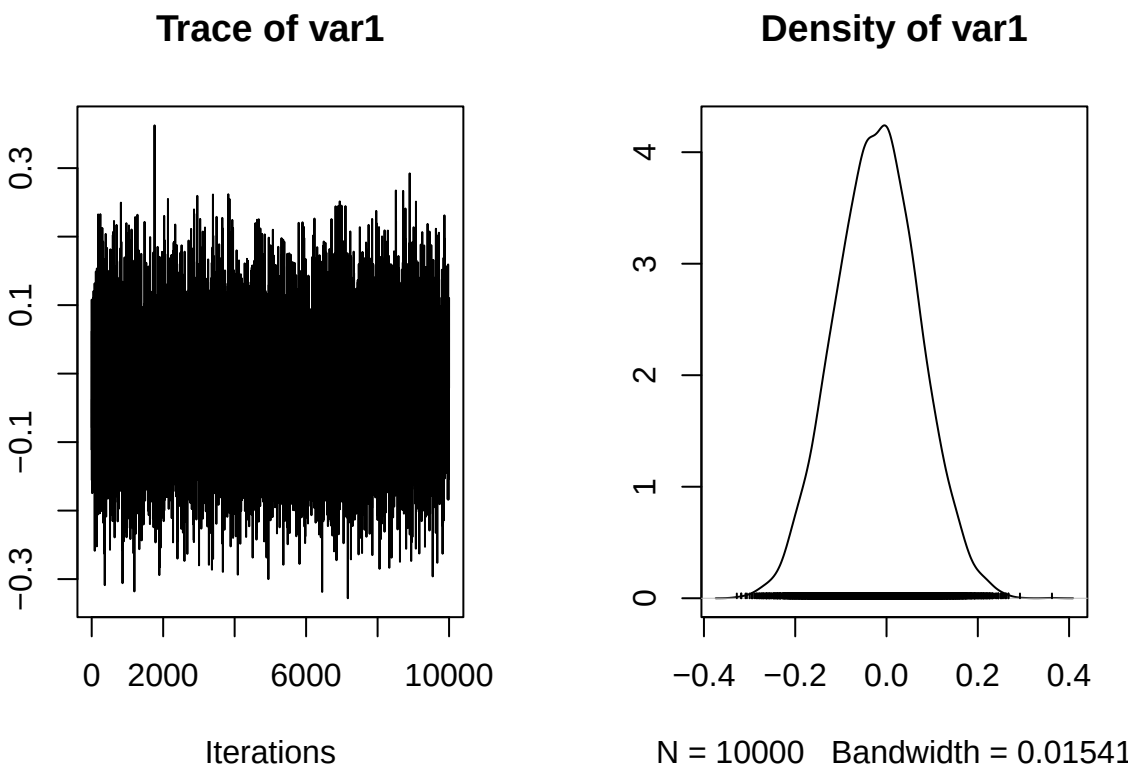

```
summary(samp3)
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean      SD  Naive SE Time-series SE
## rho  -0.06171 0.09215 0.0009215      0.0009824
## zeta -0.06232 0.09328 0.0009328      0.0009943
##
## 2. Quantiles for each variable:
##
##         2.5%     25%      50%       75%  97.5%
## rho  -0.2396 -0.1253 -0.06311 0.0005138 0.1213
## zeta -0.2444 -0.1259 -0.06320 0.0005138 0.1219
```

```
# hypothesis 4
samp4 <- correlationBF(y = d$DTFlu, x = d$CSE, posterior = T, iterations = 10000)
```

```
## Independent-candidate M-H acceptance rate: 98%
```

```
plot(samp4[,"rho"])
```




```
summary(samp4)
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##          Mean      SD  Naive SE Time-series SE
## rho  -0.02098 0.09175 0.0009175      0.0009560
## zeta -0.02116 0.09255 0.0009255      0.0009644
##
## 2. Quantiles for each variable:
##
##         2.5%      25%      50%     75%  97.5%
## rho  -0.1999 -0.08383 -0.02064 0.04139 0.1587
## zeta -0.2026 -0.08403 -0.02064 0.04142 0.1601
```

For checking multiple choice answers

```
McheckF <- d[d$Condition == "Fixed",c("Condition","Q12","Q20")]
table(McheckF$Q12) / nrow(McheckF)
```

```
##
##         1         2         3
## 0.2727273 0.6181818 0.1090909
```

```r
table(McheckF$Q20) / nrow(McheckF)
```

```
##
##          1          2          3
## 0.05454545 0.76363636 0.18181818
```

```r
McheckG <- d[d$Condition == "Growth",c("Condition","Q21","Q11")]
table(McheckG$Q21) / nrow(McheckG)
```

```
##
##          1          2          3
## 0.89285714 0.07142857 0.03571429
```

```r
table(McheckG$Q11) / nrow(McheckG)
```

```
##
##          1          2          3
## 0.92857143 0.01785714 0.05357143
```