

Package ‘SSRtcalc’

July 6, 2026

Type Package

Title Monte Carlo and Bayesian Stop-Signal Reaction Time Estimation

Version 2.1.1

Description Estimates stop-signal reaction time (SSRT) in the stop-signal task using the integration and mean methods described by Verbruggen and colleagues (2019) <[doi:10.7554/eLife.46323](https://doi.org/10.7554/eLife.46323)>. In addition to point estimates, the package provides Monte Carlo tools (nonparametric bootstrap confidence intervals, parametric ex-Gaussian simulation, minimum-trial-count and power analysis, and sensitivity analysis under violations of the horse-race assumptions) and Bayesian estimation via 'Stan', including single-subject and hierarchical ex-Gaussian horse-race models with an optional trigger-failure parameter following Matzke and colleagues (2013) <[doi:10.1037/a0030543](https://doi.org/10.1037/a0030543)>, posterior inhibition functions, and posterior predictive checks. The Bayesian layer works with either the 'cmdstanr' or 'rstan' backend.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.1

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices

Suggests MASS, cmdstanr (>= 0.5.0), rstan (>= 2.26.0), posterior, bayesplot, loo, parallel, testthat (>= 3.0.0)

Additional_repositories <https://mc-stan.org/r-packages/>

Config/testthat/edition 3

URL <https://github.com/agleontyev/SSRtcalc>

BugReports <https://github.com/agleontyev/SSRtcalc/issues>

NeedsCompilation no

Author Anton Leontyev [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8880-6161>)

Maintainer Anton Leontyev <antonleontyev@missouristate.edu>

Repository CRAN

Date/Publication 2026-07-06 06:10:20 UTC

Contents

adaptive	2
fixed	3
integration_adaptiveSSD	4
integration_fixedSSD	5
mean_adaptiveSSD	6
mean_fixedSSD	7
ranef	8
ranef.ssrt_stan	9
run_all_mc	9
ssrt_boot	10
ssrt_power	11
ssrt_robustness	12
ssrt_simulate	13
ssrt_stan	14
ssrt_stan_compare	16
ssrt_stan_inhibition_fn	16
ssrt_stan_loo	17
ssrt_stan_pp_check	17
Index	19

adaptive	<i>Example adaptive stop-signal task data</i>
----------	---

Description

A long-format dataset from a stop-signal task using an adaptive (staircase-tracking) stop-signal delay, suitable for demonstrating every function in SSRTcalc.

Usage

adaptive

Format

A data frame with 4000 rows (20 subjects x 200 trials) and 6 columns:

SubjID Integer subject identifier (1-20).

trial Integer trial number within subject (1-200).

vol Trial type: 0 = go trial, 1 = stop trial.

RT_exp Observed reaction time in milliseconds. NA on successfully inhibited stop trials.

correct Accuracy indicator (1 = correct response on go trials, or successful inhibition on stop trials; 0 otherwise).

soa Stop-signal delay (SOA) in milliseconds for stop trials; NA on go trials. Adjusted trial-by-trial by a staircase algorithm to track ~50% inhibition.

Source

Simulated data distributed with the SSRTcalc package, generated to resemble a typical adaptive stop-signal task following Verbruggen et al. (2019) <doi:10.7554/eLife.46323>.

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
integration_adaptiveSSD(d)
```

fixed

Fixed-SSD stop-signal task data (motion discrimination)

Description

A long-format dataset from a stop-signal variant of a random-dot motion-discrimination task in which the stop-signal delay (SOA) was set to one of six *fixed* values rather than being tracked by a staircase. It complements the [adaptive](#) dataset and is intended for demonstrating the fixed-SSD estimators [integration_fixedSSD](#) and [mean_fixedSSD](#).

Usage

fixed

Format

A data frame with 28,799 rows (50 subjects) and 8 columns:

SubjID Integer subject identifier. Subject 2 from the original recording is excluded (see Details), so the IDs run 1 and 3-51.

trial Trial number within subject (1-576; a subject with one removed trial has 575).

vol Trial type: 0 = go trial, 1 = stop trial.

RT_exp Observed reaction time in milliseconds. NA when no response was made (successfully inhibited stop trials and go-trial omissions).

correct Accuracy indicator (1 = correct go response or successful inhibition; 0 otherwise).

soa Stop-signal delay (SOA) in milliseconds for stop trials, one of six fixed values (100, 200, 300, 400, 500, 600 ms); NA on go trials, matching the adaptive dataset.

coh Motion coherence for the go discrimination (0.1, 0.5, or 0.8), an experimental difficulty manipulation.

response Response direction ("left", "right", or NA when no response was made).

Details

All timing variables are in **milliseconds**, matching the rest of the package and the adaptive dataset.

During preparation, subject 2 (who had no recorded responses on any trial) and a single stop trial with a recorded reaction time of 0 ms were removed. Subject identifiers otherwise follow the original recording, so identifier 2 is absent.

Source

Experimental data distributed with the SSRTcalc package, from a fixed-SSD motion-discrimination stop-signal task.

Examples

```
data(fixed)
d <- fixed[fixed$SubjID == 1, ]
integration_fixedSSD(d)
mean_fixedSSD(d)
```

integration_adaptiveSSD

Estimate SSRT via the integration method (adaptive / staircase SSD design)

Description

Implements the recommended integration method from Verbruggen et al. (2019). For each dataset:

1. Compute $p(\text{respond}|\text{signal})$ from all stop trials.
2. Find the n th percentile of the go-RT distribution ($n = p_{\text{respond}}$).
3. Subtract the mean SSD: $\text{SSRT} = \text{nth_percentile_RT} - \text{mean}(\text{SSD})$.

Usage

```
integration_adaptiveSSD(
  data,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  min_rt = 50
)
```

Arguments

data	A data.frame with one row per trial.
stop_col	Column name for the stop-trial indicator (1 = stop, 0 = go). Default "vol".
rt_col	Column name for reaction time in ms. Default "RT_exp".
acc_col	Column name for accuracy (1 = correct). Default "correct".
ssd_col	Column name for stop-signal delay in ms. Default "soa".
min_rt	Minimum valid RT in ms; shorter responses are excluded as anticipations. Default 50.

Value

A single numeric value: the estimated SSRT in milliseconds.

References

Verbruggen, F., et al. (2019). A consensus guide to capturing the ability to inhibit actions and impulses: the stop-signal task. *eLife*, 8, e46323. doi:10.7554/eLife.46323

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
integration_adaptiveSSD(d)
```

integration_fixedSSD *Estimate SSRT via the integration method (fixed SSD design)*

Description

Identical to [integration_adaptiveSSD](#) in computation, but intended for experiments using a fixed (constant) stop-signal delay. When multiple fixed SSD values are used, SSRT is computed separately for each SSD and the results are averaged (Verbruggen et al., 2019, Appendix).

Usage

```
integration_fixedSSD(
  data,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  min_rt = 50
)
```

Arguments

data	A data.frame with one row per trial.
stop_col	Column name for the stop-trial indicator (1 = stop, 0 = go). Default "vol".
rt_col	Column name for reaction time in ms. Default "RT_exp".
acc_col	Column name for accuracy (1 = correct). Default "correct".
ssd_col	Column name for stop-signal delay in ms. Default "soa".
min_rt	Minimum valid RT in ms; shorter responses are excluded as anticipations. Default 50.

Value

A single numeric value: the estimated SSRT in milliseconds.

References

Verbruggen, F., et al. (2019). A consensus guide to capturing the ability to inhibit actions and impulses: the stop-signal task. *eLife*, 8, e46323. doi:10.7554/eLife.46323

Examples

```
data(fixed)
d <- fixed[fixed$SubjID == 1, ]
integration_fixedSSD(d)
```

mean_adaptiveSSD	<i>Estimate SSRT via the mean method (adaptive SSD design)</i>
------------------	--

Description

Computes SSRT as the difference between mean go RT and mean SSD:

$$SSRT = \bar{RT}_{go} - \bar{SSD}$$

This method is less accurate than the integration method but is included for comparison and historical compatibility.

Usage

```
mean_adaptiveSSD(
  data,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  min_rt = 50
)
```

Arguments

data	A data.frame with one row per trial.
stop_col	Column name for the stop-trial indicator (1 = stop, 0 = go). Default "vol".
rt_col	Column name for reaction time in ms. Default "RT_exp".
acc_col	Column name for accuracy (1 = correct). Default "correct".
ssd_col	Column name for stop-signal delay in ms. Default "soa".
min_rt	Minimum valid RT in ms; shorter responses are excluded as anticipations. Default 50.

Value

A single numeric value: the estimated SSRT in milliseconds.

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
mean_adaptiveSSD(d)
```

 mean_fixedSSD

Estimate SSRT via the mean method (fixed SSD design)

Description

Computes SSRT as $\bar{RT}_{go} - \bar{SSD}$. For multiple fixed SSD values the mean is taken across all stop trials.

Usage

```
mean_fixedSSD(
  data,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
```

```

  ssd_col = "soa",
  min_rt = 50
)
```

Arguments

<code>data</code>	A data.frame with one row per trial.
<code>stop_col</code>	Column name for the stop-trial indicator (1 = stop, 0 = go). Default "vol".
<code>rt_col</code>	Column name for reaction time in ms. Default "RT_exp".
<code>acc_col</code>	Column name for accuracy (1 = correct). Default "correct".
<code>ssd_col</code>	Column name for stop-signal delay in ms. Default "soa".
<code>min_rt</code>	Minimum valid RT in ms; shorter responses are excluded as anticipations. Default 50.

Value

A single numeric value: the estimated SSRT in milliseconds.

Examples

```

data(fixed)
d <- fixed[fixed$SubjID == 1, ]
mean_fixedSSD(d)
```

ranef	<i>Extract per-group random-effect summaries</i>
-------	--

Description

A generic function for extracting per-group (e.g. per-subject) parameter summaries from a fitted model. SSRTcalc provides [ranef.ssrt_stan](#) for hierarchical Stan fits.

Usage

```
ranef(object, ...)
```

Arguments

<code>object</code>	A fitted model object.
<code>...</code>	Additional arguments passed to methods.

Value

A method-specific object (typically a data.frame of per-group summaries).

See Also

[ranef.ssrt_stan](#)

ranef.ssrt_stan	<i>Extract per-subject random effects from a hierarchical fit</i>
-----------------	---

Description

Extract per-subject random effects from a hierarchical fit

Usage

```
## S3 method for class 'ssrt_stan'
ranef(object, ...)
```

Arguments

object	An ssrt_stan object from a hierarchical model.
...	Unused.

Value

data.frame with per-subject posterior summaries.

run_all_mc	<i>Run all four Monte Carlo analyses in one call</i>
------------	--

Description

Run all four Monte Carlo analyses in one call

Usage

```
run_all_mc(
  data,
  n_iter = 1000,
  seed = 42,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa"
)
```

Arguments

data	data.frame in SSRTcalc long format.
n_iter	Iterations (shared). Default 1000.
seed	Random seed. Default 42.
stop_col, rt_col, acc_col, ssd_col	Column names.

Value

Named list: bootstrap, simulation, power, robustness.

Examples

```
## Not run:
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
res <- run_all_mc(d, n_iter=500)

## End(Not run)
```

ssrt_boot

Bootstrap confidence intervals for SSRT

Description

Resamples trials with replacement `n_iter` times and applies the chosen SSRT estimation function to each resample.

Usage

```
ssrt_boot(
  data,
  ssrt_fn = "integration_adaptiveSSD",
  n_iter = 2000,
  conf = 0.95,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  seed = 42,
  parallel = FALSE,
  n_cores = 2
)
```

Arguments

<code>data</code>	data.frame in SSRTcalc long format.
<code>ssrt_fn</code>	SSRT function name. Default "integration_adaptiveSSD".
<code>n_iter</code>	Bootstrap resamples. Default 2000.
<code>conf</code>	Confidence level. Default 0.95.
<code>stop_col, rt_col, acc_col, ssd_col</code>	Column names.
<code>seed</code>	Random seed. Default 42.
<code>parallel</code>	Use parallel::mclapply? (Unix/macOS only). Default FALSE.
<code>n_cores</code>	Cores when parallel=TRUE. Default 2.

Value

Object of class `ssrt_boot`.

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
b <- ssrt_boot(d, n_iter = 500)
print(b)
```

ssrt_power

Minimum trial count analysis via Monte Carlo

Description

Simulates datasets of increasing size and computes SSRT variance as a function of trial count (power curve).

Usage

```
ssrt_power(
  data,
  trial_counts = c(10, 20, 30, 50, 75, 100, 150, 200),
  n_iter = 500,
  target_se = 10,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  seed = 42
)
```

Arguments

<code>data</code>	data.frame for calibration.
<code>trial_counts</code>	Stop-trial counts to evaluate.
<code>n_iter</code>	MC iterations per count. Default 500.
<code>target_se</code>	Target SE in ms. Default 10.
<code>stop_col, rt_col, acc_col, ssd_col</code>	Column names.
<code>seed</code>	Random seed. Default 42.

Value

Object of class `ssrt_power`.

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
p <- ssrt_power(d, trial_counts=c(10,30,50), n_iter=200)
print(p)
```

ssrt_robustness *Sensitivity of SSRT estimates to horse-race assumption violations*

Description

Sweeps three violation types: go/stop process correlation, trigger failure rate, and go-RT shift on stop trials.

Usage

```
ssrt_robustness(
  data,
  violation = "all",
  n_iter = 500,
  corr_range = seq(0, 0.8, by = 0.1),
  trigger_range = seq(0, 0.4, by = 0.05),
  shift_range = seq(0, 100, by = 10),
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  seed = 42
)
```

Arguments

data	data.frame for calibration.
violation	"all", "correlation", "trigger_failure", or "go_shift".
n_iter	MC iterations per condition. Default 500.
corr_range	Correlations to test.
trigger_range	Trigger failure rates to test.
shift_range	Go-RT shifts in ms to test.
stop_col, rt_col, acc_col, ssd_col	Column names.
seed	Random seed. Default 42.

Value

Object of class ssrt_robustness.

Examples

```

data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
r <- ssrt_robustness(d, violation="trigger_failure", n_iter=20,
                    trigger_range=seq(0, 0.2, 0.1))
print(r)

```

ssrt_simulate

Parametric Monte Carlo SSRT estimation via ex-Gaussian simulation

Description

Fits an ex-Gaussian to observed go-RT data, simulates synthetic datasets under the horse-race model, and estimates SSRT on each.

Usage

```

ssrt_simulate(
  data,
  n_iter = 2000,
  n_trials = NULL,
  p_stop = NULL,
  ssrt_true = NULL,
  conf = 0.95,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  seed = 42
)

```

Arguments

data	data.frame in SSRTcalc long format.
n_iter	MC iterations. Default 2000.
n_trials	Trials per simulated dataset. NULL uses nrow(data).
p_stop	Stop-trial proportion. NULL uses observed proportion.
ssrt_true	Known true SSRT for parameter recovery. Default NULL.
conf	Confidence level. Default 0.95.
stop_col, rt_col, acc_col, ssd_col	Column names.
seed	Random seed. Default 42.

Value

Object of class `ssrt_simulate`.

Examples

```
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
s <- ssrt_simulate(d, n_iter=500)
print(s)
```

ssrt_stan

Bayesian SSRT estimation via Stan

Description

Fits the independent horse-race model with ex-Gaussian go- and stop-process distributions using Hamiltonian Monte Carlo via Stan. Supports single-subject and multi-subject hierarchical designs, with an optional trigger-failure parameter following Matzke et al. (2013).

Usage

```
ssrt_stan(
  data,
  model = c("single", "hierarchical"),
  subject_col = NULL,
  trigger_failure = FALSE,
  n_quad = 100,
  chains = 4,
  iter = 2000,
  warmup = 1000,
  cores = 4,
  backend = "auto",
  adapt_delta = 0.95,
  stop_col = "vol",
  rt_col = "RT_exp",
  acc_col = "correct",
  ssd_col = "soa",
  seed = 42,
  ...
)
```

Arguments

<code>data</code>	data.frame in SSRTcalc long format.
<code>model</code>	"single" (default) or "hierarchical".
<code>subject_col</code>	Column with subject IDs (required for hierarchical).

trigger_failure	Logical: include trigger-failure parameter. Default FALSE.
n_quad	Quadrature resolution for P(inhibit) integral. Default 100.
chains	Number of MCMC chains. Default 4.
iter	Total iterations per chain including warmup. Default 2000.
warmup	Warmup iterations. Default 1000.
cores	Parallel cores. Default 4.
backend	"cmdstan", "rstan", or "auto".
adapt_delta	HMC target acceptance rate. Increase to 0.99 if divergences appear. Default 0.95.
stop_col, rt_col, acc_col, ssd_col	Column names.
seed	Random seed. Default 42.
...	Additional arguments passed to the Stan sampler.

Value

An object of class `ssrt_stan`.

References

Matzke, D., et al. (2013). Bayesian parametric estimation of stop-signal reaction time distributions. *Journal of Experimental Psychology: General*, 142(4), 1047–1073.

Examples

```
## Not run:
data(adaptive)
d <- adaptive[adaptive$SubjID == 1, ]
fit <- ssrt_stan(d, chains = 4, iter = 2000)
print(fit)
plot(fit)

# Hierarchical
fit_h <- ssrt_stan(adaptive, model = "hierarchical",
                  subject_col = "SubjID", chains = 4)
ranef(fit_h)

## End(Not run)
```

ssrt_stan_compare	<i>Compare base model vs. trigger-failure model</i>
-------------------	---

Description

Fits both models and returns side-by-side SSRT posterior summaries.

Usage

```
ssrt_stan_compare(data, ...)
```

Arguments

data	data.frame in SSRTcalc long format.
...	Arguments forwarded to ssrt_stan().

Value

Invisibly, a list with base, tf, and comparison.

ssrt_stan_inhibition_fn	<i>Posterior inhibition function</i>
-------------------------	--------------------------------------

Description

Computes $P(\text{inhibit} \mid \text{SSD})$ across a range of SSD values using posterior parameter samples, with a 90

Usage

```
ssrt_stan_inhibition_fn(
  fit,
  ssd_range = seq(0, 600, by = 10),
  n_draws = 400,
  plot = TRUE
)
```

Arguments

fit	An ssrt_stan object (single-subject only).
ssd_range	SSD values to evaluate in ms. Default seq(0, 600, 10).
n_draws	Posterior draws to use. Default 400.
plot	Produce the plot? Default TRUE.

Value

Invisibly, a data.frame with columns: ssd, mean, lo90, hi90.

ssrt_stan_loo	<i>Leave-one-out cross-validation for an ssrt_stan fit</i>
---------------	--

Description

Requires the Stan model to include a log_lik vector in generated quantities. If absent, an informative error explains how to add it.

Usage

```
ssrt_stan_loo(fit)
```

Arguments

fit An ssrt_stan object.

Value

A loo object.

ssrt_stan_pp_check	<i>Posterior predictive checks for an ssrt_stan fit</i>
--------------------	---

Description

Overlays posterior-predictive RT distributions over observed data.

Usage

```
ssrt_stan_pp_check(fit, n_samples = 200)
```

Arguments

fit An ssrt_stan object (single-subject only).
n_samples Posterior draws to use. Default 200.

Value

Invisibly, a list with pp_go_rt and pp_ssrt.

Examples

```
## Not run:  
data(adaptive)  
fit <- ssrt_stan(adaptive[adaptive$SubjID == 1, ])  
ssrt_stan_pp_check(fit)  
  
## End(Not run)
```

Index

* datasets

adaptive, [2](#)

fixed, [3](#)

adaptive, [2, 3](#)

fixed, [3](#)

integration_adaptiveSSD, [4, 5](#)

integration_fixedSSD, [3, 5](#)

mean_adaptiveSSD, [6](#)

mean_fixedSSD, [3, 7](#)

ranef, [8](#)

ranef.ssrt_stan, [8, 9](#)

run_all_mc, [9](#)

ssrt_boot, [10](#)

ssrt_power, [11](#)

ssrt_robustness, [12](#)

ssrt_simulate, [13](#)

ssrt_stan, [14](#)

ssrt_stan_compare, [16](#)

ssrt_stan_inhibition_fn, [16](#)

ssrt_stan_loo, [17](#)

ssrt_stan_pp_check, [17](#)