

Package ‘RStanTVA’

December 17, 2025

Type Package

Title TVA Models in 'Stan' using 'R' and 'StanTVA'

Version 0.3.2

Description 'Stan' implementation of the Theory of Visual Attention (TVA; Bundesen, 1990; <[doi:10.1037/0033-295X.97.4.523](https://doi.org/10.1037/0033-295X.97.4.523)>) and numerous convenience functions for generating, compiling, fitting, and analyzing TVA models.

License GPL (>= 3)

Encoding UTF-8

BugReports <https://github.com/mmrabe/RStanTVA/issues>

URL <https://github.com/mmrabe/RStanTVA>

Date 2025-12-16

Depends rstan, R (>= 3.6)

Imports dplyr, readr, tidyr, methods, utils, tibble, stats, cli, lme4, brms, rlang

Suggests cmdstanr, knitr, rmarkdown, ggplot2

Additional_repositories <https://stan-dev.r-universe.dev>

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Maximilian M. Rabe [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-2556-5644>>),
Søren Kyllingsbæk [aut] (ORCID:
<<https://orcid.org/0000-0002-4100-437X>>)

Maintainer Maximilian M. Rabe <maximilian.rabe@uni-potsdam.de>

Repository CRAN

Date/Publication 2025-12-16 23:10:02 UTC

Contents

alias,stantvafit-method	2
coef,stantvafit-method	3
extract,stantvafit-method	4
fitted,stantvafit-method	4
fixef,stantvafit-method	5
logLik,stantvafit-method	6
model_code	6
names,stantvafit-method	7
optimizing	8
predict,stantvafit-method	9
ranef,stantvafit-method	9
read_stantva_fit	10
read_tva_data	11
sampling	11
show,stantvacode-method	12
show,stantvafit-method	13
show,stantvamodel-method	14
stancsv2stantvafit	14
stantvacode-class	15
stantvafit-class	15
stantvamodel-class	16
stantva_code	16
stantva_model	17
stantva_path	18
summary,stantvafit-method	18
tva_recovery	19
tva_recovery_true_params	19
tva_report	20
write_stantva_fit	20
write_stantva_model	21
write_tva_data	21
Index	23

alias,stantvafit-method
Retrieve parameters aliases

Description

Returns the StanTVA parameter aliases for the underlying RStan fit.

Usage

```
## S4 method for signature 'stantvafit'
alias(object)
```

Arguments

object The StanTVA fit object.

Value

A character vector of parameter aliases.

Examples

```
## Not run:  
al <- alias(fit)  
al  
  
## End(Not run)
```

coef,stantvafit-method

Model coefficients

Description

Returns the model coefficients (sum of fixed + random effects, grouped by random factor) for a StanTVA fit object.

Usage

```
## S4 method for signature 'stantvafit'  
coef(object)
```

Arguments

object The StanTVA fit object.

Value

The model coefficients, grouped by random factors.

Examples

```
## Not run:  
fixef <- coef(fit)  
fixef  
  
## End(Not run)
```

 extract,stantvafit-method

Extract samples from a fitted RStanTVA model

Description

Returns posterior samples from a fitted RStanTVA model.

Usage

```
## S4 method for signature 'stantvafit'
extract(object, pars, ...)
```

Arguments

object	The RStanTVA fit.
pars	(Optional) A character vector of variable names to extract.
...	Additional arguments passed to rstan::extract() , e.g. permuted and inc_warmup.

Value

See [rstan::extract\(\)](#) for details.

Examples

```
## Not run:
f <- read_stantva_fit("fit.rds")
extract(f, "C_Intercept")

## End(Not run)
```

 fitted,stantvafit-method

Retrieve fitted parameter values

Description

Returns the fitted values for latent model parameters. This is identical to calling `predict()` without new data.

Usage

```
## S4 method for signature 'stantvafit'
fitted(object, variables = names(object@stanmodel@code@df))
```

Arguments

object	The StanTVA fit object.
variables	The names of the parameters to retrieve.

Value

The fitted values.

Examples

```
## Not run:
p <- fitted(fit, variables = c("C","K"))
colMeans(p$C)

## End(Not run)
```

fixef,stantvafit-method

Fixed effects

Description

Returns the fixed effects for a StanTVA fit object.

Usage

```
## S4 method for signature 'stantvafit'
fixef(object)
```

Arguments

object	The StanTVA fit object.
--------	-------------------------

Value

The fixed effects.

Examples

```
## Not run:
fixed_effects <- fixef(fit)
fixed_effects

## End(Not run)
```

```
logLik, stantvafit-method
```

Log-likelihood

Description

Returns the pointwise log-likelihood of a StanTVA fit.

Usage

```
## S4 method for signature 'stantvafit'
logLik(object)
```

Arguments

`object` The StanTVA fit.

Value

The pointwise log likelihood.

Examples

```
## Not run:
loglik <- logLik(model, data, params)
loglik

## End(Not run)
```

```
model_code
```

Extract Stan code

Description

Extracts the Stan code from a StanTVA model object.

Usage

```
model_code(object, type)

## S4 method for signature 'stanmodel'
model_code(object, type = c("stan", "stan2", "cpp"))

## S4 method for signature 'stanfit'
model_code(object, type)
```

Arguments

object	A StanTVA model object or fit.
type	The type of code to return (stan: formatted StanTVA, stan2: ready-to-compile Stan code, cpp: generated C++ code).

Value

A RStanTVA model code object (stan), or a string containing the code (stan2 or cpp).

Methods (by class)

- `model_code(stanmodel)`: method
- `model_code(stanfit)`: Extract code from a model fit

Examples

```
## Not run:
model <- stantva_model(locations = 2)
model_code(model)

## End(Not run)
```

names,stantvafit-method

Retrieve model parameter names

Description

Returns the names of the fitted model parameters.

Usage

```
## S4 method for signature 'stantvafit'
names(x)
```

Arguments

x	The StanTVA fit.
---	------------------

Value

The list of parameter names and aliases.

Examples

```
## Not run:
f <- read_stantva_fit("fit.rds")
names(f)

## End(Not run)
```

optimizing

Maximum-likelihood estimation

Description

Obtain a point estimate by maximizing the joint posterior from the StanTVA model.

Usage

```
optimizing(object, ...)

## S4 method for signature 'stantvamodel'
optimizing(object, data, init, seed = sample.int(.Machine$integer.max, 1), ...)
```

Arguments

<code>object</code>	The StanTVA model object.
<code>...</code>	Further arguments passed to <code>rstan::optimizing()</code> .
<code>data</code>	The data to which the model should be fitted, usually a <code>data.frame</code> .
<code>init</code>	How to initialize the individual chains, see <code>rstan::optimizing()</code> . Note that for random, any lower-level hierarchical (e.g., subject-level) parameters are initialized to zero.
<code>seed</code>	Seed for random number generation and chain initialization

Value

A list, representing the maximum-likelihood estimate, see `rstan::optimizing()`.

Functions

- `optimizing(stantvamodel)`: method

predict,stantvafit-method
Predict parameter values

Description

Returns the predictions for latent model parameters.

Usage

```
## S4 method for signature 'stantvafit'  
predict(object, newdata, variables = names(object@stanmodel@code@df))
```

Arguments

object	The StanTVA fit object.
newdata	The new data (leave empty to use fitted data).
variables	The names of the parameters to predict.

Value

The predictions.

Examples

```
## Not run:  
p <- predict(fit, variables = c("C","K"))  
colMeans(p$C)  
  
## End(Not run)
```

ranef,stantvafit-method
Random effects

Description

Returns the random effects for a StanTVA fit object.

Usage

```
## S4 method for signature 'stantvafit'  
ranef(object)
```

Arguments

object The StanTVA fit object.

Value

The fixed effects.

Examples

```
## Not run:
random_effects <- ranef(fit)
random_effects

## End(Not run)
```

read_stantva_fit	<i>Read StanTVA fit</i>
------------------	-------------------------

Description

Reads a StanTVA fit object from one or more files. If multiple files are given, the fits will be combined into a single fit object (e.g., combining separately fitted chains).

Usage

```
read_stantva_fit(files)
```

Arguments

files The file names.

Value

The StanTVA fit object.

Examples

```
## Not run:
fit <- read_stantva_fit(c("chain1.rds", "chain2.rds"))
fit

## End(Not run)
```

read_tva_data	<i>Read TVA data</i>
---------------	----------------------

Description

Reads TVA data from a file.

Usage

```
read_tva_data(file, set = LETTERS, ...)
```

Arguments

file	The file name.
set	The set of items.
...	Additional arguments passed to read_table() .

Value

A TVA data object, which inherits from `data.frame`.

Examples

```
## Not run:
data <- read_tva_data("data.dat")
data

## End(Not run)
```

sampling	<i>Draw posterior samples from an RStanTVA model</i>
----------	--

Description

Draw samples from the model defined by object.

Usage

```
sampling(object, ...)

## S4 method for signature 'stantvamodel'
sampling(
  object,
  data,
  init = "random",
  seed = sample.int(.Machine$integer.max, 1),
```

```

...,
backend = c("rstan", "cmdstanr", "cmdstanr_mpi"),
cpp_options = if (match.arg(backend) == "cmdstanr") list(stan_threads =
  object@code@config$parallel) else if (match.arg(backend) == "cmdstanr_mpi") list(CXX
    = "mpicxx", TBB_CXX_TYPE = "gcc", STAN_MPI = TRUE)
)

```

Arguments

object	The StanTVA model object.
...	Further arguments passed to the sampling handler of the specified backend.
data	The data to which the model should be fitted, usually a <code>data.frame</code> .
init	How to initialize the individual chains, see rstan::sampling() . Note that for random, any lower-level hierarchical (e.g., subject-level) parameters are initialized to zero.
seed	Seed for random number generation and chain initialization
backend	Which backend to use for fitting (default: <code>rstan</code>)
cpp_options	Which options to pass to <code>stan_model()</code> for compiling the C++ code.

Value

Returns a `stantva_fit` object, which inherits from [stanfit](#), representing the fit of object to data.

Functions

- `sampling(stantvamodel)`: method

show,stantvacode-method

Show StanTVA code

Description

Display the content of the StanTVA code object in the console.

Usage

```
## S4 method for signature 'stantvacode'
show(object)
```

Arguments

object	The StanTVA code object.
--------	--------------------------

Value

Returns object invisibly but the function is usually only called for its side effects.

`show,stantvafit-method`*Print StanTVA fit*

Description

Prints a StanTVA fit object.

Usage

```
## S4 method for signature 'stantvafit'  
show(object)
```

```
## S4 method for signature 'stantvafit'  
print(x, digits_summary = 2, ...)
```

Arguments

<code>object</code>	The StanTVA fit object.
<code>x</code>	The StanTVA fit object.
<code>digits_summary</code>	The number of significant digits to display in posterior summaries.
<code>...</code>	Currently not used.

Value

Returns `x`. Usually called for its side effects (printing to the console).

Functions

- `show(stantvafit)`: Alias

Examples

```
## Not run:  
print(fit)  
  
## End(Not run)
```

```
show,stantvamodel-method
```

Show StanTVA model

Description

Prints a StanTVA model object.

Usage

```
## S4 method for signature 'stantvamodel'
show(object)
```

Arguments

object The StanTVA model object.

Value

The printed object.

Examples

```
## Not run:
model <- stantva_model(locations = 4)
show(model)

## End(Not run)
```

```
stancsv2stantvafit      Read StanTVA fit from CSV
```

Description

This function may be used to read an RStan or CmdStan fit from CSV files. Note that you also need to provide the fitted model.

Usage

```
stancsv2stantvafit(csv_file, data, model, contrasts = list())
```

Arguments

csv_file The CSV file to be read.
 data The data to which the model was fitted.
 model The fitted model as an StanTVA model or StanTVA code object.
 contrasts Any contrasts specified to factors in the data set.

Value

The StanTVA fit object.

Examples

```
## Not run:
data <- read_tva_data("data.dat")
model <- stantva_code(locations = 6)
fit <- stancsv2stantvafit("chain1.csv", data, model)
fit

## End(Not run)
```

stantvacode-class	<i>StanTVA code class</i>
-------------------	---------------------------

Description

StanTVA code class

Slots

code The generated Stan code.

config A list of model configuration parameters, as passed to `stantva_code()` or `stantva_model()`.

include_path The path to the StanTVA includes (usually identical to `stantva_path()`).

df The degrees of freedom of the model parameters.

dim The dimensions of the model parameters.

version The RStanTVA package version that was used to generate this model fit.

priors Priors for the model parameters.

initializers Stan code for the generation of initial samples

stantvafit-class	<i>StanTVA fit class</i>
------------------	--------------------------

Description

StanTVA fit class

Slots

stanmodel The StanTVA model object that was fitted to the data.

data The data to which the StanTVA model was fitted.

stantvamodel-class	StanTVA model class
Description	
StanTVA model class	
Slots	
code The StanTVA code object that was used to compile this model.	
initializers Compiled functions for generation of initial samples	
stantva_code	Generate StanTVA code

Description

Creates a StanTVA model code object.

Usage

```
stantva_code(  
  formula = NULL,  
  locations,  
  task = c("wr", "pr"),  
  regions = list(),  
  C_mode = c("equal", "locations", "regions"),  
  w_mode = c("locations", "regions", "equal"),  
  t0_mode = c("constant", "gaussian", "exponential", "shifted_exponential"),  
  K_mode = c("bernoulli", "free", "binomial", "betabinomial", "hypergeometric", "probit"),  
  max_K = locations,  
  fixed = NULL,  
  parallel = isTRUE(rstan_options("threads_per_chain") > 1L),  
  save_log_lik = FALSE,  
  priors = NULL,  
  sanity_checks = TRUE,  
  debug_neginf_loglik = FALSE  
)
```

Arguments

- formula Optional formulas for nested and hierarchical model parameters.
- locations The number of display locations (items).
- task The task. Currently implemented: wr (whole report) and pr (partial report)

regions	An optional list of groups of display locations (regions).
C_mode	The mode/family for the C parameter.
w_mode	The mode/family for the w parameter.
t0_mode	The mode/family for the t_0 parameter.
K_mode	The mode for the K parameter.
max_K	The upper bound of K .
fixed	Named vector or list of parameters that are not to be sampled/fitted but fixed to their respective values.
parallel	(logical) Whether to use parallel chains.
save_log_lik	(logical) Whether to save the log likelihood (needed for likelihood-based model comparison such as loo).
priors	The priors.
sanity_checks	(logical) Whether to perform sanity checks.
debug_neginf_loglik	(logical) Whether to debug negative infinity log likelihood.

Value

The StanTVA model code object.

Examples

```
model <- stantva_code(locations = 4, task = "pr")
model
```

stantva_model	<i>StanTVA model</i>
---------------	----------------------

Description

Creates a StanTVA model object.

Usage

```
stantva_model(..., stan_options = list())
```

Arguments

... Additional arguments passed to [stantva_code\(\)](#).
 stan_options The Stan options, passed to [stan_model\(\)](#)

Value

The StanTVA model object.

Examples

```
## Not run:
model <- stantva_model(locations = 2, task = "pr")
model

## End(Not run)
```

stantva_path	<i>StanTVA path</i>
--------------	---------------------

Description

Returns the path to the StanTVA directory.

Usage

```
stantva_path()
```

Details

This function is used internally by the `stantva_model()` method.

Value

The path to the StanTVA directory.

Examples

```
path <- stantva_path()
path
```

summary,stantvafit-method	<i>Summary method for RStanTVA fits</i>
---------------------------	---

Description

Summarize the distributions of estimated parameters and derived quantities using the posterior draws.

Usage

```
## S4 method for signature 'stantvafit'
summary(object, pars, ...)
```

Arguments

object The RStanTVA fit.
 pars (Optional) A character vector of variable names to extract.
 ... Additional arguments passed to `rstan::summary()`, e.g. probs and use_cache.

Value

See `rstan::summary()` for details.

Examples

```
## Not run:
f <- read_stantva_fit("fit.rds")
summary(f, "C_Intercept", probs = c(.025, .975))

## End(Not run)
```

tva_recovery	<i>True parameters for TVA recovery study</i>
--------------	---

Description

True parameters for TVA recovery study

Usage

```
tva_recovery
```

Format

An object of class `grouped_df` (inherits from `tbl_df`, `tbl`, `data.frame`) with 11700 rows and 9 columns.

tva_recovery_true_params	<i>True parameters for TVA recovery study</i>
--------------------------	---

Description

True parameters for TVA recovery study

Usage

```
tva_recovery_true_params
```

Format

An object of class `list` of length 5.

tva_report	<i>Generate typical descriptive statistics for TVA reports</i>
------------	--

Description

This function generates by-trial descriptive statistics, see ‘Value’ below.

Usage

```
tva_report(data)
```

Arguments

`data` The TVA report data as a `data.frame`.

Value

The function returns a transmuted `data.frame/tibble` with columns `condition` (copied from `data`), `exposure` (copied from `data$T`), `n_items`, `n_targets`, `n_distractors`, and `score` (number of correctly reported items).

Examples

```
## Not run: tva_report(tva_recovery)
```

write_stantva_fit	<i>Write StanTVA fit</i>
-------------------	--------------------------

Description

Writes a StanTVA fit object to a file.

Usage

```
write_stantva_fit(fit, file, ...)
```

Arguments

`fit` The StanTVA fit object.
`file` The file name.
`...` Additional arguments passed to [saveRDS\(\)](#).

Value

No return value, called for side effects.

Examples

```
## Not run: write_stantva_fit(fit, "fit.rds")
```

write_stantva_model	<i>Write StanTVA model</i>
---------------------	----------------------------

Description

Writes a StanTVA model to a file.

Usage

```
write_stantva_model(model, file = stdout())
```

Arguments

model	The StanTVA model object.
file	The file name.

Value

No return value, called for side effects.

Examples

```
## Not run: write_stantva_model(model, "model.stan")
```

write_tva_data	<i>Write TVA data</i>
----------------	-----------------------

Description

Writes TVA data to a file.

Usage

```
write_tva_data(data, file, ...)
```

Arguments

data	The TVA data object.
file	The file name.
...	Additional arguments passed to write_tsv() .

Value

No return value, called for side effects.

Examples

```
## Not run:  
data <- read_tva_data("data.dat")  
write_tva_data(data, "data.dat")  
  
## End(Not run)
```

Index

- * **datasets**
 - tva_recovery, [19](#)
 - tva_recovery_true_params, [19](#)
- alias, stantvafit-method, [2](#)
- coef, stantvafit-method, [3](#)
- extract, stantvafit-method, [4](#)
- fitted, stantvafit-method, [4](#)
- fixef, stantvafit-method, [5](#)
- logLik, stantvafit-method, [6](#)
- model_code, [6](#)
- model_code, stanfit-method (model_code), [6](#)
- model_code, stanmodel-method (model_code), [6](#)
- names, stantvafit-method, [7](#)
- optimizing, [8](#)
- optimizing, stantvamodel-method (optimizing), [8](#)
- predict, stantvafit-method, [9](#)
- print, stantvafit-method (show, stantvafit-method), [13](#)
- ranef, stantvafit-method, [9](#)
- read_stantva_fit, [10](#)
- read_table(), [11](#)
- read_tva_data, [11](#)
- rstan::extract(), [4](#)
- rstan::optimizing(), [8](#)
- rstan::sampling(), [12](#)
- rstan::summary(), [19](#)
- sampling, [11](#)
- sampling, stantvamodel-method (sampling), [11](#)
- saveRDS(), [20](#)
- show, stantvacode-method, [12](#)
- show, stantvafit-method, [13](#)
- show, stantvamodel-method, [14](#)
- stan_model(), [17](#)
- stancsv2stantvafit, [14](#)
- stanfit, [12](#)
- stantva_code, [16](#)
- stantva_code(), [17](#)
- stantva_model, [17](#)
- stantva_model(), [18](#)
- stantva_path, [18](#)
- stantvacode-class, [15](#)
- stantvafit-class, [15](#)
- stantvamodel-class, [16](#)
- summary, stantvafit-method, [18](#)
- tva_recovery, [19](#)
- tva_recovery_true_params, [19](#)
- tva_report, [20](#)
- write_stantva_fit, [20](#)
- write_stantva_model, [21](#)
- write_tsv(), [21](#)
- write_tva_data, [21](#)