Package: stanedit (via r-universe)

November 14, 2024

Title Facilitate editing of stan models

Version 0.1.0

Description Provides functionality for extracting particular bits of
stan models as well as removing and adding lines.

License MIT + file LICENSE

Imports checkmate, methods, utils

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://epiforecasts.r-universe.dev

RemoteUrl https://github.com/epiforecasts/stanedit

RemoteRef v0.1.0

RemoteSha 66a36c690499b4bc7d597192ffeaa6327e7c3058

Contents

add_block — *Add a block to a stan model*

Description

Add a block to a stan model. If that block exists, it will be removed first.

Usage

```
add_block(x, name, lines)
```

Arguments

x	a stanedit object
name	name of the block
lines	character vector, lines in the block

Value

a stanedit object containing the new block

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
reg <- add_block(
  reg,
  "transformed parameters",
  c("real gamma;", "gamma = alpha * beta;")
)
```

Equals.stanedit — *Check if two models are equal*

Description

Ignores differences in the model name.

Usage

```
## S3 method for class 'stanedit'
e1 == e2, ...
```

Arguments

e1	a stanedit
e2	a stanedit
...	ignored

Value

TRUE or FALSE, depending on whether the models are equal or not

Examples

```
model_file_name <- system.file(package = "rbi", "PZ.bi")
PZ <- stanedit(filename = model_file_name)
PZ == PZ # TRUE
```

Extract.stanedit *Subset model lines*

Description

Extracts a subset of lines from the model.

Usage

```
## S3 method for class 'stanedit'
x[i, ...]
```

Arguments

x	A stanedit
i	A vector of line numbers
...	ignored

Value

a character string of the extracted model lines(s)

Examples

```
model_file_name <- system.file(package = "rbi", "PZ.bi")
PZ <- stanedit(filename = model_file_name)
PZ[3:4]
```

`Extract_assign.stanedit` *Subset and replace model lines*

Description

Extracts a subset of lines from the model and assigns new character strings.

Usage

```
## S3 replacement method for class 'stanedit'
x[i, ...] <- value
```

Arguments

`x`	A stanedit
`i`	A vector of line numbers
`...`	ignored
`value`	A vector of the same length as `i`, containing the replacement strings

Value

the updated `stanedit` object

Examples

```
model_file_name <- system.file(package = "rbi", "PZ.bi")
PZ <- stanedit(filename = model_file_name)
PZ[3:4] <- c("const e = 0.4", "const m_l = 0.05")
```

`get_block` *Get the contents of a block in a stan model*

Description

Returns the contents of a block in a stan model as a character vector of lines.

Usage

```
get_block(x, name, shell = FALSE)
```

Arguments

x	a stanedit object
name	name of the block
shell	if TRUE (default:FALSE), will return the shell (i.e., the definition of the block) as well as content; this is useful, e.g., to see options passed to a `transition` or ode block

Value

a character vector of the lines in the block, or an empty character vector if the block is not found

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
get_block(reg, "parameters")
```

---

get_declaration   *Get the declaration of a variable in a stan model*

Description

Returns the contents of a block in a stan model as a character vector of lines.

Usage

```
get_declaration(x, name)
```

Arguments

x	a stanedit object
name	name of the variable

Value

a character vector of length 1 with the line where the variable is declared, or an empty character vector if no declaration is found

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
get_declaration(reg, "alpha")
```

get_vars — *Extract the variables declared in a stan model*

Description

Extract the variables declared in a stan model

Usage

```
get_vars(x, block)
```

Arguments

`x`	a `stanedit` object
`block`	character (optional), the block or blocks in which to search; by default the whole model will be searched

Value

a character vector of variable names

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
get_vars(reg, "parameters")
```

insert_lines — *Insert lines in a stan model*

Description

Inserts one or more lines into a libbi model. If one of `before` or `after` is given, the line(s) will be inserted before or after a given line number or block name, respectively. If one of `at_beginning_of` or `at_end_of` is given, the lines will be inserted at the beginning/end of the block, respectively.

Usage

```
insert_lines(x, lines, before, after, at_beginning_of, at_end_of)
```

Arguments

`x`	a `stanedit` object
`lines`	vector or line(s)
`before`	line number before which to insert line(s)
`after`	line number after which to insert line(s)
`at_beginning_of`	block at the beginning of which to insert lines(s)
`at_end_of`	block at the end of which to insert lines(s)

Value

the updated `stanedit` object

See Also

`stanedit`

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
insert_lines(reg, lines = "alpha ~ std_normal();", after = 12)
```

---

move *Move stan code elsewere*

Description

Move stan code elsewere

Usage

```
move(x, lines, ...)
```

Arguments

`x`	a `stanedit` object
`lines`	vector of line numbers to move
`...`	arguments passed to `insert_lines`

Value

a character vector of variable names

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
move(reg, find_declaration(reg, "alpha"), at_end_of = "data")
```

---

remove_lines *Remove line(s) and/or block(s) in a libbi model*

Description

Removes one or more lines in a libbi model.

Usage

```
remove_lines(x, what)
```

Arguments

x	a stanedit object
what	either a vector of line number(s) to remove, or a vector of blocks to remove (e.g., "parameters")

Value

the updated stanedit object

See Also

stanedit

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
reg <- remove_lines(reg, "model")
```

stanedit *stanedit*

Description

`stanedit` prepares stan model code for programmatically editing.

Usage

```
stanedit(object, filename)
```

Arguments

`object`	either a `stanmodel` object as generated by `rstan::stan_model()` or a `CmdStanModel` object as generated by `cmdstanr::cmdstan_model()`, or a character vector representing model code.
`filename`	name of a file containing model code, if `object` is not given.

Value

A `{stanedit}` object containing clean model code. This will have some formatting applied to it for further processing so line numbers may not correspond to the original model, and comments will be removed.

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
reg
```

Unequals.stanedit *Check if two models are unequal*

Description

Ignores differences in the model name.

Usage

```
## S3 method for class 'stanedit'
e1 != e2, ...
```

Arguments

`e1`	a `stanedit`
`e2`	a `stanedit`
`...`	ignored

Value

TRUE or FALSE, depending on whether the models are equal or not

Examples

```
model_file_name <- system.file(package = "rbi", "PZ.bi")
PZ <- stanedit(filename = model_file_name)
PZ != PZ # FALSE
```

`write_model` *Writes a bi model to a file.*

Description

Writes a bi model to a file given by `filename`. The extension '.bi' will be added if necessary.

Usage

```
write_model(x, filename)
```

Arguments

`x`	a `stanedit` object
`filename`	name of the file to be written

Value

the return value of the `writeLines` call.

See Also

`stanedit`

Examples

```
model_file_name <- system.file(package = "stanedit", "regression.stan")
reg <- stanedit(filename = model_file_name)
new_file_name <- tempfile("reg", fileext = ".stan")
write_model(reg, new_file_name)
```

Index