Package 'campsisnca'

October 19, 2026

Type Package

Title Non-Compartmental Analysis for Campsis Simulation Platform

Version 1.7.3

Maintainer Nicolas Luyckx <nicolas.luyckx@lynxsoft.be>

Description A flexible and user-friendly non-compartmental analysis (NCA) toolkit designed to work seamlessly with simulated pharmacokinetic data generated using the 'campsis' ecosystem. The package provides a comprehensive framework to compute standard and custom NCA metrics, including exposure (AUC), peak/trough concentrations, half-life and time-above/below thresholds, with support for configurable time windows and summary statistics. 'campsisnca' integrates tightly with 'campsis' and 'campsismod', enabling streamlined workflows from simulation to analysis. In addition, the package provides a JSON-based interface to define NCA analyses, metrics and options using formal schemas, allowing analyses to be created, validated and executed outside of R and facilitating reproducibility, automation and system integration. The package also includes utilities for generating formatted summary tables and exporting results in multiple formats suitable for reporting. Trapezoidal rule implementation for AUC calculation is based on the 'qpNCA' package by Huisman, Jolling, Mehta and Bergsma (2021) <doi:10.32614/CRAN.package.qpNCA>, following methodology from Rowland and Tozer (2011, ISBN:978-0-683-07404-8). The package itself is licensed under the GPL (>= 3); the JSON schema files shipped in inst/extdata are licensed separately under the Creative Commons Attribution 4.0 International (CC BY 4.0).

License GPL (>= 3)

URL https://github.com/campsisverse/campsisnca

BugReports https://github.com/campsisverse/campsisnca/issues

Depends campsismod, R (>= 4.0.0)

Imports assertthat, campsis, cards, dplyr, glue, gt, gtsummary, jsonlite, jsonvalidate, lifecycle, magrittr, methods, purrr, rlang, stringr, tibble, tidyr

Suggests devtools, ggplot2, mrgsolve, pkgdown, rmarkdown, roxygen2, rxode2, testthat

Config/roxygen2/version 8.1.0

Encoding UTF-8

Language en-US

LazyData true

Collate 'all_classes.R' 'check.R' 'utilities.R' 'special_operators.R' 'consistency_checks.R' 'stat_formula.R' 'export_type.R' 'generic.R' 'data.R' 'stat_utils.R' 'export_utils.R' 'nca_options.R' 'nca_time_window.R' 'nca_metric_utils.R' 'gtsummary_utils.R' 'filter_utils.R' 'rounding_utils.R' 'theoretical_metrics.R' 'nca_metric.R' 'nca_metrics.R' 'auc_trap_rule.R' 'auc.R' 'cmax.R' 'tmax.R' 'cmin.R' 'tmin.R' 'cat.R' 'ctrough.R' 'cavg.R' 'thalf.R' 'theoretical_thalf.R' 'time_above_below.R' 'change_from_baseline.R' 'custom_metric.R' 'nca_analysis.R' 'nca_analyses.R' 'nca_table.R' 'replicated_nca_table.R' 'json_interface.R' 'deprecated_methods.R'

NeedsCompilation no

Author Nicolas Luyckx [aut, cre], Calvagone [cph]

Repository CRAN

Date/Publication 2026-10-19 10:20:02 UTC

Contents

abstract_avg_metric-class . . . 5
abstract_last_metric-class . . . 5
abstract_max_metric-class . . . 5
abstract_min_metric-class . . . 5
abstract_time_above_or_below_limit_metric-class . . . 6
abstract_value_at_metric-class . . . 6
applyTimeWindow . . . 6
apply_time_window . . . 7
AUC . . . 7
auc_metric-class . . . 8
Avg . . . 8
avg_metric-class . . . 9
calculate . . . 9
campsisnca_output-class . . . 10
campsisnca_tbl-class . . . 10
campsis_output-class . . . 10
CAt . . . 11
Cavg . . . 12
cavg_metric-class . . . 12

CFB . . . 13
cfb_metric-class . . . 13
ChangeFromBaseline . . . 14
Cmax . . . 15
cmax_metric-class . . . 15
Cmin . . . 16
cmin_metric-class . . . 16
computeTimeAboveBelow . . . 17
compute_nca_metric_summary . . . 17
conc_at_metric-class . . . 18
Ctrough . . . 18
ctrough_metric-class . . . 19
CustomMetric . . . 19
CustomMetricTbl . . . 20
custom_metric-class . . . 20
cv . . . 21
dataframe_type-class . . . 21
deparse_digits . . . 21
deparse_one_line . . . 22
discardCategoryColumn . . . 22
discard_category_column . . . 23
export,replicated_nca_table,character-method . . . 23
extract_brace_values . . . 24
generate_table_code . . . 24
geocv . . . 25
geomean . . . 25
getCampsisncaOption . . . 26
get_character_cols_only . . . 26
get_default_name . . . 27
get_digits_code . . . 28
get_labels_code . . . 29
get_latex_name . . . 29
get_statistics_code . . . 30
get_strata . . . 31
get_table_summary_code . . . 31
get_unit . . . 32
get_variable_type_code . . . 33
glue_stat_display . . . 33
gtsummary_type-class . . . 34
gt_type-class . . . 34
individual_campsisnca_tbl-class . . . 34
individual_wide_campsisnca_tbl-class . . . 34
iValue . . . 35
i_value . . . 35
i_values . . . 36
i_value_tbl . . . 37
Last . . . 38
last_metric-class . . . 38

Max . . . 39
max_metric-class . . . 39
metrics.1cpt . . . 40
metrics.2cpt . . . 40
metrics.common . . . 41
Min . . . 41
min_metric-class . . . 42
NCAAnalysis . . . 42
NCAMetrics . . . 43
NCAMetricsTable . . . 43
NCAOptions . . . 44
NCATable . . . 45
nca_analyses-class . . . 46
nca_analysis-class . . . 46
nca_metric-class . . . 46
nca_metrics-class . . . 46
nca_options-class . . . 46
nca_table-class . . . 47
nca_time_window-class . . . 47
pk_bolus_md . . . 47
preserve_column_levels . . . 48
remove_column_levels . . . 49
ReplicatedNCATable . . . 49
replicated_nca_table-class . . . 50
restore_column_levels . . . 50
save_column_levels . . . 51
se . . . 52
stat_display_string . . . 52
summary_campsisnca_tbl-class . . . 53
summary_pretty_campsisnca_tbl-class . . . 53
summary_wide_campsisnca_tbl-class . . . 53
Thalf . . . 53
Thalf.1cpt . . . 54
thalf.1cpt.required . . . 55
Thalf.2cpt.dist . . . 55
Thalf.2cpt.eff . . . 56
thalf.2cpt.required . . . 56
Thalf.2cpt.z . . . 57
thalf_metric-class . . . 57
theoretical_thalf_metric-class . . . 58
TimeAboveLimit . . . 58
TimeBelowLimit . . . 59
timerange . . . 60
TimeWindow . . . 60
time_above_limit_metric-class . . . 61
time_below_limit_metric-class . . . 61
Tmax . . . 61
tmax_metric-class . . . 62

Tmin 62
tmin_metric-class 63
toGt 63
trap 64
UndefinedNCAOptions 65
UndefinedTimeWindow 65
undefined_nca_options-class 65
undefined_nca_time_window-class 66
ValueAt 66
value_at_metric-class 67

Index **68**

`abstract_avg_metric-class`
Abstract avg metric class.

Description

Abstract avg metric class.

`abstract_last_metric-class`
Abstract last metric class.

Description

Abstract last metric class.

`abstract_max_metric-class`
Abstract max metric class.

Description

Abstract max metric class.

`abstract_min_metric-class`
Abstract min metric class.

Description

Abstract min metric class.

`abstract_time_above_or_below_limit_metric-class` *Abstract time above/below limit class.*

Description

Abstract time above/below limit class.

`abstract_value_at_metric-class` *Abstract value at metric class.*

Description

Abstract value at metric class.

`applyTimeWindow` *Apply time window.*

Description

‘r lifecycle::badge("deprecated")‘

Usage

```
applyTimeWindow(x, window, data_time_unit, ...)
```

Arguments

`x`	input data for the calculation, data frame
`window`	time window
`data_time_unit`	time unit of TIME column in data (x argument)
`...`	extra arguments

Details

‘applyTimeWindow()‘ is deprecated in favor of ‘apply_time_window()‘.

Value

updated object

apply_time_window *Apply time window.*

Description

Apply time window.

Usage

```
apply_time_window(x, window, data_time_unit, ...)

## S4 method for signature 'ANY,nca_time_window,character'
apply_time_window(x, window, data_time_unit)
```

Arguments

x	input data for the calculation, data frame
window	time window
data_time_unit	time unit of TIME column in data (x argument)
...	extra arguments

Value

updated object

AUC *AUC.*

Description

AUC.

Usage

```
AUC(
  variable = NULL,
  window = NULL,
  method = 1,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
method	method: * 1: linear up - linear down * 2: linear up - logarithmic down * 3: linear before Tmax, logarithmic after Tmax
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Details

Trapezoidal rule implementation for AUC calculation is based on the 'qpNCA' package by Huisman, Jolling, Mehta and Bergsma (2021) <doi:10.32614/CRAN.package.qpNCA>, following methodology from Rowland and Tozer (2011, ISBN:978-0-683-07404-8).

Value

an object of class auc_metric

---

auc_metric-class *AUC metric class.*

Description

AUC metric class.

Avg *Avg.*

Description

Avg.

Usage

```
Avg(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class avg_metric

---

avg_metric-class *Avg metric class.*

Description

Avg metric class.

---

calculate *Calculate.*

Description

Calculate.

Usage

```
calculate(object, x, options = NULL, ...)

## S4 method for signature 'nca_metric,ANY'
calculate(object, x, options = NULL, ...)

## S4 method for signature 'theoretical_thalf_metric,ANY'
calculate(object, x, options = NULL, ...)

## S4 method for signature 'nca_analysis,ANY'
calculate(object, x, options = NULL, ...)

## S4 method for signature 'nca_analyses,ANY'
calculate(object, x, options = NULL, ...)
```

```
## S4 method for signature 'nca_table,ANY'
calculate(object, x, options = NULL, ...)

## S4 method for signature 'replicated_nca_table,campsis_output'
calculate(object, x, options = NULL, ...)
```

Arguments

object	object (NCA table, NCA analyses, NCA analysis, PK metric) where calculation is applied
x	input data for the calculation, data frame
options	NCA options
...	extra arguments

Value

updated object

---

campsisnca_output-class *The Campsisnca Output Class Union*

Description

A class union containing standard data frames, tibbles, and Campsisnca-specific table objects.

---

campsisnca_tbl-class *Campsisnca table class (see this class as an interface)*

Description

Campsisnca table class (see this class as an interface)

---

campsis_output-class *The Campsis Output Class Union*

Description

A class union containing standard data frames, tibbles, and Campsis-specific table objects.

CAt *CAt (concentration at specific time).*

Description

CAt (concentration at specific time).

Usage

```
CAt(
  variable = NULL,
  window = NULL,
  time = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
time	what time to read the concentrations. If not provided, last concentrations from x will be returned.
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class conc_at_metric

Cavg *Cavg.*

Description

Cavg.

Usage

```
Cavg(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class cavg_metric

cavg_metric-class *Cavg metric class.*

Description

Cavg metric class.

CFB *Alias for Change from Baseline (CFB).*

Description

Alias for Change from Baseline (CFB).

Usage

```
CFB(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL,
  method = "difference"
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README
method	Character string specifying the calculation method. Must be one of "difference" (default), "percent", "ratio", or "log".

Value

an object of class cfb_metric

cfb_metric-class *Change from Baseline metric class.*

Description

Change from Baseline metric class.

Slots

method Character string specifying the CFB method ("difference", "percent", "ratio", "log").

ChangeFromBaseline *Change from Baseline (CFB).*

Description

Change from Baseline (CFB).

Usage

```
ChangeFromBaseline(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL,
  method = "difference"
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README
method	Character string specifying the calculation method. Must be one of "difference" (default), "percent", "ratio", or "log".

Value

an object of class cfb_metric

Cmax *Cmax.*

Description

Cmax.

Usage

```
Cmax(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class cmax_metric

cmax_metric-class *Cmax metric class.*

Description

Cmax metric class.

Cmin *Cmin*

Description

Cmin

Usage

```
Cmin(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class cmin_metric

cmin_metric-class *Cmin metric class.*

Description

Cmin metric class.

computeTimeAboveBelow *Compute the duration of a line segment above or below a certain limit.*

Description

Compute the duration of a line segment above or below a certain limit.

Usage

```
computeTimeAboveBelow(x1, y1, x2, y2, above, strictly)
```

Arguments

x1	x-coordinate of the first point
y1	y-coordinate of the first point
x2	x-coordinate of the second point
y2	y-coordinate of the second point
above	whether the line segment is above the limit
strictly	whether the line segment is strictly above or below the limit

Value

the duration of the line segment above or below the limit

compute_nca_metric_summary *Compute NCA metric summary.*

Description

Compute NCA metric summary.

Usage

```
compute_nca_metric_summary(object, strat_vars, quantile_type)
```

Arguments

object	NCA metric
strat_vars	stratification variable names in data
quantile_type	type of quantile

Value

data frame

`conc_at_metric-class` *Concentration at metric class.*

Description

Concentration at metric class.

`Ctrough` *Ctrough. Last time in x shall be considered as the trough time. Similar to Last, but for concentrations.*

Description

Ctrough. Last time in x shall be considered as the trough time. Similar to Last, but for concentrations.

Usage

```
Ctrough(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

`variable`	dependent variable
`window`	time window on which this metric should be computed
`name`	custom metric name (will be exported into table headers)
`unit`	metric unit (will be exported into table headers if provided)
`stat_display`	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
`digits`	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class ctrough_metric

ctrough_metric-class *Ctrough metric class.*

Description

Ctrough metric class.

CustomMetric *Custom metric (input data as time and value vectors).*

Description

Custom metric (input data as time and value vectors).

Usage

```
CustomMetric(
  variable = NULL,
  window = NULL,
  fun,
  name = NULL,
  unit = NULL,
  categorical = FALSE,
  stat_display = get_stat_display_default(categorical),
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
fun	any custom function with exactly 2 arguments: time and value
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
categorical	categorical endpoint, logical
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class custom_metric

CustomMetricTbl *Custom metric (input data as tibble).*

Description

Custom metric (input data as tibble).

Usage

```
CustomMetricTbl(
  window = NULL,
  fun,
  name = NULL,
  unit = NULL,
  categorical = FALSE,
  stat_display = get_stat_display_default(categorical),
  digits = NULL
)
```

Arguments

window	time window on which this metric should be computed
fun	any custom function with exactly 1 argument: data
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
categorical	categorical endpoint, logical
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class custom_metric

custom_metric-class *Custom metric class.*

Description

Custom metric class.

cv *Compute the coefficient of variation.*

Description

Compute the coefficient of variation.

Usage

```
cv(x)
```

Arguments

x	numeric vector

Value

numeric value

dataframe_type-class *Dataframe export type class.*

Description

Dataframe export type class.

deparse_digits *Custom deparse function. Works similarly to deparse1. However, lines are trimmed before being concatenated.*

Description

Custom deparse function. Works similarly to deparse1. However, lines are trimmed before being concatenated.

Usage

```
deparse_digits(digits)
```

Arguments

digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these)

Value

a character vector, which will be pasted and given to gtsummary

deparse_one_line *Deparse 1 line.*

Description

Deparse 1 line.

Usage

```
deparse_one_line(x)
```

Arguments

x	expression to deparse

Value

a single string

discardCategoryColumn *Discard category column.*

Description

‘r lifecycle::badge("deprecated")‘

Usage

```
discardCategoryColumn(x, split = "_")
```

Arguments

x	summary export
split	string use to concatenate the ’stat’ and ’category’ column

Details

‘discardCategoryColumn()‘ is deprecated in favor of ‘discard_category_column()‘.

Value

updated object

discard_category_column *Discard category column.*

Description

Discard category column.

Usage

```
discard_category_column(x, split = "_")
```

Arguments

x	summary export
split	string use to concatenate the 'stat' and 'category' column

Value

updated data frame

export,replicated_nca_table,character-method *Export replicated NCA table.*

Description

Export replicated NCA table.

Usage

```
## S4 method for signature 'replicated_nca_table,character'
export(object, dest, ...)
```

Arguments

object	replicated_nca_table object
dest	destination for the summarised table, either "dataframe", "gtsummary" or "gt"
...	extra arguments

Value

a data frame, a gtsummary object or a gt object, depending on the destination

extract_brace_values *Extract values from within braces.*

Description

Extract values from within braces.

Usage

```
extract_brace_values(x)
```

Arguments

x	input string

Value

a character vector

generate_table_code *Generate table code.*

Description

Generate table code.

Usage

```
generate_table_code(object, init = NULL, ...)

## S4 method for signature 'nca_table,logical'
generate_table_code(object, init = NULL, ...)
```

Arguments

object	table object
init	generate initialization code to generate the individuals, default is TRUE
...	extra arguments

Value

a character vector containing the code

geocv *Compute the geometric CV.*

Description

Compute the geometric CV.

Usage

```
geocv(x)
```

Arguments

x	numeric vector

Value

numeric value

geomean *Compute the geometric mean.*

Description

Compute the geometric mean.

Usage

```
geomean(x)
```

Arguments

x	numeric vector

Value

numeric value

getCampsisncaOption *Get Campsisnca option logic.*

Description

Get Campsisnca option logic.

Usage

```
getCampsisncaOption(name, default)
```

Arguments

name	option to search
default	default value if option not found

Value

option value

get_character_cols_only *Filter Column Names to Character (and Optionally Factor) Columns*

Description

Filters a vector of column names to include only those present in a data frame that are of class character (and optionally factor).

Usage

```
get_character_cols_only(x, cols, include_factor = FALSE)
```

Arguments

x	A data frame or tibble.
cols	A character vector of candidate column names.
include_factor	Logical. If TRUE, factor columns are also included along with character columns. Defaults to FALSE.

Value

A character vector of column names matching the specified class criteria.

get_default_name *Get default name.*

Description

Get default name.

Usage

```
get_default_name(object, ...)

## S4 method for signature 'nca_metric'
get_default_name(object, ...)

## S4 method for signature 'auc_metric'
get_default_name(object, ...)

## S4 method for signature 'max_metric'
get_default_name(object, ...)

## S4 method for signature 'cmax_metric'
get_default_name(object, ...)

## S4 method for signature 'tmax_metric'
get_default_name(object, ...)

## S4 method for signature 'min_metric'
get_default_name(object, ...)

## S4 method for signature 'cmin_metric'
get_default_name(object, ...)

## S4 method for signature 'tmin_metric'
get_default_name(object, ...)

## S4 method for signature 'value_at_metric'
get_default_name(object, ...)

## S4 method for signature 'conc_at_metric'
get_default_name(object, ...)

## S4 method for signature 'last_metric'
get_default_name(object, ...)

## S4 method for signature 'ctrough_metric'
get_default_name(object, ...)
```

```
## S4 method for signature 'avg_metric'
get_default_name(object, ...)

## S4 method for signature 'cavg_metric'
get_default_name(object, ...)

## S4 method for signature 'thalf_metric'
get_default_name(object, ...)

## S4 method for signature 'theoretical_thalf_metric'
get_default_name(object, ...)

## S4 method for signature 'time_above_limit_metric'
get_default_name(object, ...)

## S4 method for signature 'time_below_limit_metric'
get_default_name(object, ...)

## S4 method for signature 'cfb_metric'
get_default_name(object, ...)
```

Arguments

object	get default name of this object
...	optional extra arguments

Value

a character string (the default name)

`get_digits_code` *Get digits code for gtsummary.*

Description

Get digits code for gtsummary.

Usage

```
get_digits_code(table)
```

Arguments

table	NCA table

Value

code

get_labels_code *Get labels code for gtsummary.*

Description

Get labels code for gtsummary.

Usage

```
get_labels_code(table, subscripts)
```

Arguments

table	NCA table
subscripts	use subscripts, logical value

Value

code

get_latex_name *Get the name of the metric in LaTeX notation (with subscript coded with an underscore and brackets).*

Description

Get the name of the metric in LaTeX notation (with subscript coded with an underscore and brackets).

Usage

```
get_latex_name(x, ...)

## S4 method for signature 'nca_metric'
get_latex_name(x)

## S4 method for signature 'abstract_max_metric'
get_latex_name(x)

## S4 method for signature 'tmax_metric'
get_latex_name(x)

## S4 method for signature 'abstract_min_metric'
get_latex_name(x)
```

```
## S4 method for signature 'tmin_metric'
get_latex_name(x)

## S4 method for signature 'abstract_value_at_metric'
get_latex_name(x)

## S4 method for signature 'last_metric'
get_latex_name(x)

## S4 method for signature 'ctrough_metric'
get_latex_name(x)

## S4 method for signature 'abstract_avg_metric'
get_latex_name(x)

## S4 method for signature 'thalf_metric'
get_latex_name(x)

## S4 method for signature 'theoretical_thalf_metric'
get_latex_name(x)

## S4 method for signature 'cfb_metric'
get_latex_name(x)
```

Arguments

x	metric
...	extra arguments, not used

Value

a character string

---

get_statistics_code *Get statistics code for gtsummary.*

Description

Get statistics code for gtsummary.

Usage

```
get_statistics_code(table)
```

Arguments

table	NCA table

Value

code

`get_strata` *Get all stratas.*

Description

Get all stratas.

Usage

```
get_strata(object, keep_single = NULL, ...)

## S4 method for signature 'nca_table,logical'
get_strata(object, keep_single = NULL, ...)
```

Arguments

`object`	table object
`keep_single`	keep single stratification values, logical (default TRUE)
`...`	extra arguments

Value

list of stratification variable names

`get_table_summary_code` *Get table summary code.*

Description

Get table summary code.

Usage

```
get_table_summary_code(
  variable,
  data,
  by,
  stats,
  type,
  labels,
  digits,
  combine_with,
  header_label
)
```

Arguments

variable	assigned variable name
data	data frame code
by	variable
stats	stats to compute
type	type of the variables
labels	the labels to display
digits	the digits to be used for rounding
combine_with	either 'tbl_stack' or 'tbl_merge'
header_label	header label name

Value

data frame

---

get_unit *Get the unit corresponding to the given metric.*

Description

Get the unit corresponding to the given metric.

Usage

```
get_unit(object, metric, ...)

## S4 method for signature 'nca_metrics,character'
get_unit(object, metric, ...)

## S4 method for signature 'nca_analysis,character'
get_unit(object, metric, ...)

## S4 method for signature 'nca_table,character'
get_unit(object, metric, ...)
```

Arguments

object	any object that contains units
metric	given metric name
...	extra arguments, not used

Value

a character string (the unit)

get_variable_type_code *Get the variable type code for gtsummary.*

Description

Get the variable type code for gtsummary.

Usage

```
get_variable_type_code(table, all_dichotomous_levels)
```

Arguments

table	NCA table
all_dichotomous_levels	show all dichotomous levels (0 and 1) when data is dichotomous

Value

code

glue_stat_display *Glue stat display string.*

Description

Glue stat display string.

Usage

```
glue_stat_display(stat_display, stats, summary, digits)
```

Arguments

stat_display	stat display string
stats	statistics, character vector
summary	summary data frame
digits	digits to be used for rounding

Value

glued string

`gtsummary_type-class` *Gt summary export type class.*

Description

Gt summary export type class.

`gt_type-class` *Gt export type class.*

Description

Gt export type class.

`individual_campsisnca_tbl-class` *Individual Campsisnca table class*

Description

Individual Campsisnca table class

`individual_wide_campsisnca_tbl-class` *Individual (wide format) Campsisnca table class*

Description

Individual (wide format) Campsisnca table class

iValue *Compute the individual value of an individual.*

Description

‘r lifecycle::badge("deprecated")‘

Usage

```
iValue(object, time, value = NULL)
```

Arguments

object	PK metric
time	time vector, numeric
value	value vector, numeric

Details

‘iValue()‘ is deprecated in favor of ‘i_value()‘.

Value

individual value

i_value *Compute the individual value of an individual.*

Description

Compute the individual value of an individual.

Usage

```
i_value(object, time, value = NULL)

## S4 method for signature 'auc_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'abstract_max_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'tmax_metric,numeric,numeric'
i_value(object, time, value = NULL)
```

```
## S4 method for signature 'abstract_min_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'tmin_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'abstract_value_at_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'abstract_last_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'abstract_avg_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'thalf_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature
## 'abstract_time_above_or_below_limit_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'cfb_metric,numeric,numeric'
i_value(object, time, value = NULL)

## S4 method for signature 'custom_metric,numeric,numeric'
i_value(object, time, value = NULL)
```

Arguments

object	PK metric
time	time vector, numeric
value	value vector, numeric

Value

individual value

---

i_values *Compute the individual values on a simulation output.*

Description

Compute the individual values on a simulation output.

Usage

```
i_values(object, x, options, strat_vars, ...)

## S4 method for signature 'nca_metric'
i_values(object, x, options, strat_vars, ...)
```

Arguments

object	PK metric
x	input data for the calculation, data frame
options	NCA options
strat_vars	stratification variable names
...	extra arguments

Value

individual values

---

i_value_tbl *Compute the individual value of an individual.*

Description

Compute the individual value of an individual.

Usage

```
i_value_tbl(object, data, ...)

## S4 method for signature 'custom_metric,tbl_df'
i_value_tbl(object, data)
```

Arguments

object	PK metric
data	individual data, tibble
...	extra arguments

Value

individual value

Last *Last value.*

Description

Last value.

Usage

```
Last(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class last_metric

last_metric-class *Last metric class.*

Description

Last metric class.

Max *Max.*

Description

Max.

Usage

```
Max(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class max_metric

max_metric-class *Max metric class.*

Description

Max metric class.

metrics.1cpt *Theoretical metrics for 1-cpt model results.*

Description

Theoretical metrics for 1-cpt model results.

Usage

```
metrics.1cpt(x, map = character(0))
```

Arguments

x	CAMPSIS/NONMEM dataframe
map	character vector used for column mapping, only one key is possible: K

Value

theoretical metrics

metrics.2cpt *Theoretical metrics for 2-cpt model results.*

Description

Theoretical metrics for 2-cpt model results.

Usage

```
metrics.2cpt(x, map = character(0))
```

Arguments

x	CAMPSIS/NONMEM dataframe
map	character vector used for column mapping, keys to be chosen among: DOSE, TAU, CL, V2, Q, V3, KA

Value

theoretical metrics

metrics.common *Pre-processing for metrics.1cpt and metrics.2cpt.*

Description

Pre-processing for metrics.1cpt and metrics.2cpt.

Usage

```
metrics.common(x, map, thalf.1cpt)
```

Arguments

x	CAMPSIS/NONMEM dataframe
map	character vector used for column mapping, only one key is possible: K
thalf.1cpt	logical value

Value

theoretical metrics

Min *Min.*

Description

Min.

Usage

```
Min(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class min_metric

---

min_metric-class *Min metric class.*

Description

Min metric class.

NCAAnalysis *Create an NCA analysis.*

Description

Create an NCA analysis.

Usage

```
NCAAnalysis(
  name = "Default",
  window = TimeWindow(),
  variable = NULL,
  strata = get_default_strata()
)
```

Arguments

name	name of this analysis, e.g. 'Day 1'
window	time window, see TimeWindow
variable	default variable which is analysed
strata	strata levels this analysis refers to, named vector, e.g. c(ARM='1g QD'). Note, the default strata are c(SCENARIO='all', ARM='all'). Use 'all' if this analysis refers to all levels for the specified stratification variable. By default, a stratification variable that has only 1 level is ignored.

Value

an object of class nca_analysis

`NCAMetrics` *NCA metrics*

Description

NCA metrics

Usage

```
NCAMetrics()
```

Value

an object of class nca_metrics

`NCAMetricsTable` *NCA table (deprecated).*

Description

NCA table (deprecated).

Usage

```
NCAMetricsTable(
  title = NULL,
  subtitle = NULL,
  swap_strat = FALSE,
  combine_with = "tbl_stack",
  show_all_levels = FALSE,
  header_label = "Metric",
  subscripts = TRUE,
  nca_options = NCAOptions(),
  tab_options = list(),
  json = NULL
)
```

Arguments

title	table title, optional character value
subtitle	table subtitle, optional character value
swap_strat	swap stratification variables in table (only useful when 2 stratification variables are given)
combine_with	either 'tbl_stack' or 'tbl_merge'
show_all_levels	show all dichotomous levels in table
header_label	'Metric' by default
subscripts	use LaTeX subcripts/superscripts notation when writing labels
nca_options	NCA options, see ?NCAOptions
tab_options	list of options to pass to gt::tab_options
json	path to JSON table file or JSON content in string form

Value

an object of class nca_table

NCAOptions *NCA options used for calculation of metrics.*

Description

NCA options used for calculation of metrics.

Usage

```
NCAOptions(
  quantile_type = 2L,
  data_time_unit = "hour",
  table_time_unit = "hour"
)
```

Arguments

quantile_type	type of quantile to use (see ?quantile), default value in campsisnca is 2 (aligned with gtsummary)
data_time_unit	time unit of the data given to 'calculate'
table_time_unit	time unit in table (for time-dependent metrics like AUC, Time above and below, etc.)

Value

an object of class nca_options

NCATable *NCA table.*

Description

NCA table.

Usage

```
NCATable(
  title = NULL,
  subtitle = NULL,
  swap_strat = FALSE,
  combine_with = "tbl_stack",
  show_all_levels = FALSE,
  header_label = "Metric",
  subscripts = TRUE,
  nca_options = NCAOptions(),
  tab_options = list(),
  json = NULL
)
```

Arguments

title	table title, optional character value
subtitle	table subtitle, optional character value
swap_strat	swap stratification variables in table (only useful when 2 stratification variables are given)
combine_with	either 'tbl_stack' or 'tbl_merge'
show_all_levels	show all dichotomous levels in table
header_label	'Metric' by default
subscripts	use LaTeX subcripts/superscripts notation when writing labels
nca_options	NCA options, see ?NCAOptions
tab_options	list of options to pass to gt::tab_options
json	path to JSON table file or JSON content in string form

Value

an object of class nca_table

`nca_analyses-class` *NCA analyses class.*

Description

NCA analyses class.

`nca_analysis-class` *NCA analysis class.*

Description

NCA analysis class.

`nca_metric-class` *NCA metric class. See this class as abstract class.*

Description

NCA metric class. See this class as abstract class.

`nca_metrics-class` *NCA metrics class. See this class as a list of NCA metrics.*

Description

NCA metrics class. See this class as a list of NCA metrics.

`nca_options-class` *NCA options class.*

Description

NCA options class.

nca_table-class *NCA table class.*

Description

NCA table class.

nca_time_window-class *NCA time window class.*

Description

NCA time window class.

pk_bolus_md *Simulated Pharmacokinetic Dataset (Multiple-Dose Bolus)*

Description

A simulated 2-compartment oral dataset containing 200 subjects administered 1000 mg bolus doses every 24 hours for 7 days. Includes rich sampling on Day 1 and Day 7, sparse sampling on intermediate days, and weight-based allometric scaling on clearance.

Usage

```
pk_bolus_md
```

Format

A tibble with 5,000 rows and 16 variables:

ID Subject identifier (1–200)

TIME Time after initial dose (hours)

ARM Study arm identifier

A_DEPOT Amount in the depot (absorption) compartment (mg)

A_CENTRAL Amount in the central compartment (mg)

A_PERIPHERAL Amount in the peripheral compartment (mg)

A_OUTPUT Eliminated amount (mg)

BW Body weight covariate (kg), sampled from Uniform(50, 100)

CL Individual clearance (L/h), allometrically scaled with body weight

V2 Central volume of distribution (L)

Q Inter-compartmental clearance (L/h)

V3 Peripheral volume of distribution (L)

KA Absorption rate constant (1/h)

CP True plasma concentration in central compartment (mg/L or mcg/mL)

OBS_CP Observed plasma concentration with ~15.8% proportional residual variability

Y Observation variable (identical to `OBS_CP`)

Source

Simulated using the `campsis` package (ADVAN4/TRANS4 model).

Examples

```
data(pk_bolus_md)
head(pk_bolus_md)
```

---

preserve_column_levels

Preserve Existing Column Value Order as Factor Levels

Description

Converts target columns into factors using their current unique row appearance order as the factor levels.

Usage

```
preserve_column_levels(x, cols)
```

Arguments

`x`	a data frame or tibble.
`cols`	a character vector of column names to convert.

Value

a data frame with updated factor columns.

remove_column_levels *Strip Factor Class from Columns*

Description

Converts target columns from factors into standard character vectors.

Usage

```
remove_column_levels(x, cols)
```

Arguments

x	a data frame or tibble.
cols	a character vector of column names to convert.

Value

a data frame with character columns.

ReplicatedNCATable *Replicated NCA table.*

Description

Replicated NCA table.

Usage

```
ReplicatedNCATable(
  title = NULL,
  subtitle = NULL,
  selected_statistics = character(),
  summary_stat_display = get_stat_display_default(),
  summary_stat_signif_digits = 3L,
  strata = get_default_strata(),
  tab_options = list(),
  json = NULL
)
```

Arguments

`title`	table title, optional character value
`subtitle`	table subtitle, optional character value
`selected_statistics`	NCA metrics statistics to keep (e.g. mean, etc) when summary statistics are computed on replicated output. Default is the empty character vector (all statistics are computed).
`summary_stat_display`	display format for replicate statistics, character vector. Default is `'{median} ({p5}-{p95})'`.
`summary_stat_signif_digits`	number of significant digits to display for replicate statistics, default is 3.
`strata`	strata levels this analysis refers to, named vector, e.g. c(ARM='1g QD'). Note, the default strata are c(SCENARIO='all', ARM='all'). Use 'all' if this analysis refers to all levels for the specified stratification variable. By default, a stratification variable that has only 1 level is ignored.
`tab_options`	list of options to pass to gt::tab_options
`json`	path to JSON table file or JSON content in string form

Value

an object of class replicated_nca_table

`replicated_nca_table-class` *Replicated NCA table class.*

Description

Replicated NCA table class.

`restore_column_levels` *Restore Factor Levels and Optionally Reorder Rows*

Description

Re-applies saved factor levels back onto target columns, optionally physically sorts the rows by the restored factor order, and optionally converts the columns back to character vectors.

Usage

```
restore_column_levels(x, saved_levels, arrange = TRUE, to_character = FALSE)
```

Arguments

x	a data frame or tibble.
saved_levels	a named list of character vectors representing factor levels (typically created by save_column_levels).
arrange	logical. If TRUE (default), physically reorders rows in x to match the restored factor level sequence.
to_character	logical. If TRUE, converts the target columns back to character vectors after restoring levels and sorting. Defaults to FALSE.

Value

a data frame with restored factor (or character) columns and optionally reordered rows.

---

save_column_levels *Extract and Save Column Factor Levels*

Description

Extracts factor levels (or unique value order for non-factors) for target columns and stores them in a named list for later restoration.

Usage

```
save_column_levels(x, cols)
```

Arguments

x	a data frame or tibble.
cols	a character vector of column names whose levels should be saved.

Value

a named list where names correspond to column names and values contain character vectors of factor levels.

se *Compute the standard error.*

Description

Compute the standard error.

Usage

```
se(x)
```

Arguments

x	numeric vector

Value

numeric value

stat_display_string *Return the evaluated statistics display string. This method was kept for backward compatibility in the tests. It is recommended to call 'export(dest="dataframe", type="summary_pretty")' on the NCA table instead.*

Description

Return the evaluated statistics display string. This method was kept for backward compatibility in the tests. It is recommended to call 'export(dest="dataframe", type="summary_pretty")' on the NCA table instead.

Usage

```
stat_display_string(object, ...)

## S4 method for signature 'nca_metric'
stat_display_string(object, ...)
```

Arguments

object	PK metric
...	extra arguments

Value

a string, e.g. 100 [45-143]

`summary_campsisnca_tbl-class` *Summary Campsisnca table class*

Description

Summary Campsisnca table class

`summary_pretty_campsisnca_tbl-class` *Summary (pretty format) Campsisnca table class*

Description

Summary (pretty format) Campsisnca table class

`summary_wide_campsisnca_tbl-class` *Summary (wide format) Campsisnca table class*

Description

Summary (wide format) Campsisnca table class

`Thalf` *Terminal half life computed by making a linear regression in the log domain on the given data x.*

Description

Terminal half life computed by making a linear regression in the log domain on the given data x.

Usage

```
Thalf(
  variable = NULL,
  window = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class thalf_metric

Thalf.1cpt *Theoretical half life for a 1-compartment model.*

Description

Theoretical half life for a 1-compartment model.

Usage

```
Thalf.1cpt(
  map = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

map	character vector used for column mapping, only one key is possible: K
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class theoretical_thalf_metric

thalf.1cpt.required *Thalf metric required columns for a 1-compartment model.*

Description

Thalf metric required columns for a 1-compartment model.

Usage

```
thalf.1cpt.required()
```

Value

character vector of required column names

Thalf.2cpt.dist *Theoretical distribution half life for a 2-compartment model.*

Description

Theoretical distribution half life for a 2-compartment model.

Usage

```
Thalf.2cpt.dist(
  map = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

map	character vector used for column mapping, keys to be chosen among: DOSE, TAU, CL, V2, Q, V3, KA
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class theoretical_thalf_metric

Thalf.2cpt.eff *Theoretical effective half life for a 2-compartment model.*

Description

Theoretical effective half life for a 2-compartment model.

Usage

```
Thalf.2cpt.eff(
  map = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

map	character vector used for column mapping, keys to be chosen among: DOSE, TAU, CL, V2, Q, V3, KA
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class theoretical_thalf_metric

thalf.2cpt.required *Thalf metric required columns for a 2-compartment model.*

Description

Thalf metric required columns for a 2-compartment model.

Usage

```
thalf.2cpt.required()
```

Value

character vector of required column names

Thalf.2cpt.z *Theoretical elimination half life for a 2-compartment model.*

Description

Theoretical elimination half life for a 2-compartment model.

Usage

```
Thalf.2cpt.z(
  map = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

map	character vector used for column mapping, keys to be chosen among: DOSE, TAU, CL, V2, Q, V3, KA
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class theoretical_thalf_metric

thalf_metric-class *Thalf metric class.*

Description

Thalf metric class.

`theoretical_thalf_metric-class` *Theoretical thalf metric class.*

Description

Theoretical thalf metric class.

`TimeAboveLimit` *Time above a certain limit.*

Description

Time above a certain limit.

Usage

```
TimeAboveLimit(
  variable = NULL,
  window = NULL,
  limit = NULL,
  strictly = FALSE,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

`variable`	dependent variable
`window`	time window on which this metric should be computed
`limit`	the limit to compare the variable against
`strictly`	whether the variable must be strictly above the limit
`name`	custom metric name (will be exported into table headers)
`unit`	metric unit (will be exported into table headers if provided)
`stat_display`	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
`digits`	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class time_above_limit_metric

TimeBelowLimit *Time below a certain limit.*

Description

Time below a certain limit.

Usage

```
TimeBelowLimit(
  variable = NULL,
  window = NULL,
  limit = NULL,
  strictly = FALSE,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
limit	the limit to compare the variable against
strictly	whether the variable must be strictly below the limit
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class time_below_limit_metric

timerange *Filter CAMPSIS dataset based on min and max time.*

Description

Filter CAMPSIS dataset based on min and max time.

Usage

```
timerange(
  x,
  min = 0,
  max = Inf,
  exclmin = FALSE,
  exclmax = FALSE,
  rebase = FALSE
)
```

Arguments

x	CAMPSIS/NONMEM dataframe
min	min time
max	max time
exclmin	exclude min time when filtering
exclmax	exclude max time when filtering
rebase	rebase first time to origin, logical value, FALSE by default

Value

dataset subset

TimeWindow *Create a time window object.*

Description

Create a time window object.

Usage

```
TimeWindow(
  start = 0,
  end = "last",
  time_unit = "hour",
  exclude_start = FALSE,
  exclude_end = FALSE
)
```

Arguments

start	start time of window
end	end time of window, use 'last' to specify the end of the simulation output
time_unit	time unit of 'start' and 'end'
exclude_start	exclude start time when filtering
exclude_end	exclude end time when filtering

Value

a time range object

---

time_above_limit_metric-class *Time above limit metric class.*

Description

Time above limit metric class.

---

time_below_limit_metric-class *Time below limit metric class.*

Description

Time below limit metric class.

Tmax *Tmax.*

Description

Tmax.

Usage

```
Tmax(
  variable = NULL,
  window = NULL,
  rebase = TRUE,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
rebase	rebase time according to start time of window
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class tmax_metric

---

tmax_metric-class *Tmax metric class.*

Description

Tmax metric class.

Tmin *Tmin.*

Description

Tmin.

Usage

```
Tmin(
  variable = NULL,
  window = NULL,
  rebase = TRUE,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

variable	dependent variable
window	time window on which this metric should be computed
rebase	rebase time according to start time of window
name	custom metric name (will be exported into table headers)
unit	metric unit (will be exported into table headers if provided)
stat_display	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
digits	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class tmin_metric

---

tmin_metric-class *Tmin metric class.*

Description

Tmin metric class.

---

toGt *Gtsummary to Gt.*

Description

Gtsummary to Gt.

Usage

```
toGt(
  x,
  title = NULL,
  subtitle = NULL,
  opts = list(),
  subscripts = FALSE,
  fmt_markdown = FALSE
)
```

Arguments

x	gtsummary table
title	table title
subtitle	table subtitle
opts	gt tab options
subscripts	use subscripts
fmt_markdown	transform any markdown-formatted text, logical value. Default is FALSE.

Value

a gt table object

`trap` *NOTE: This method has been adapted from the 'qpNCA' package by Huisman, Jolling, Mehta and Bergsma (2021) <doi:10.32614/CRAN.package.qpNCA>, following methodology from Rowland and Tozer (2011, ISBN:978-0-683-07404-8).*

Description

Calculate Area Under the Curve Using Trapezoids.

Usage

```
trap(x, y, method = 1)
```

Arguments

x	x variable, i.e. time
y	y variable, i.e. concentration
method	method: * 1: linear up - linear down * 2: linear up - logarithmic down * 3: linear before Tmax, logarithmic after Tmax

Details

Calculates AUC using the trapezoidal method. Assumes data represent a single profile. Despite choice of method, only linear interpolation is used for areas of intervals beginning or ending with y: 0.

Value

area (length-one numeric)

UndefinedNCAOptions *Undefined NCA options.*

Description

Undefined NCA options.

Usage

```
UndefinedNCAOptions()
```

Value

an object of class undefined_nca_options

UndefinedTimeWindow *Create an undefined time window.*

Description

Create an undefined time window.

Usage

```
UndefinedTimeWindow()
```

Value

undefined time window

undefined_nca_options-class *Undefined NCA options class.*

Description

Undefined NCA options class.

`undefined_nca_time_window-class` *Undefined NCA time window class.*

Description

Undefined NCA time window class.

`ValueAt` *Value at (value at specific time).*

Description

Value at (value at specific time).

Usage

```
ValueAt(
  variable = NULL,
  window = NULL,
  time = NULL,
  name = NULL,
  unit = NULL,
  stat_display = NULL,
  digits = NULL
)
```

Arguments

`variable`	dependent variable
`window`	time window on which this metric should be computed
`time`	what time to read the values. If not provided, last values from x will be returned.
`name`	custom metric name (will be exported into table headers)
`unit`	metric unit (will be exported into table headers if provided)
`stat_display`	statistics display, default is '{median} [{p5}-{p95}]' for continuous data or '{n} / {N} ({p}%)' for categorical data
`digits`	rounding digits definitions (integer, function, purrr-style lambda function or list of these, 1 item per statistic), see README

Value

an object of class value_at_metric

value_at_metric-class *Value at metric class.*

Description

Value at metric class.

Index

∗ **datasets**
pk_bolus_md, 47

abstract_avg_metric-class, 5
abstract_last_metric-class, 5
abstract_max_metric-class, 5
abstract_min_metric-class, 5
abstract_time_above_or_below_limit_metric-class, 6
abstract_value_at_metric-class, 6
apply_time_window, 7
apply_time_window,ANY,nca_time_window,character-method *(apply_time_window)*, 7
applyTimeWindow, 6
AUC, 7
auc_metric-class, 8
Avg, 8
avg_metric-class, 9

calculate, 9
calculate,nca_analyses,ANY-method *(calculate)*, 9
calculate,nca_analysis,ANY-method *(calculate)*, 9
calculate,nca_metric,ANY-method *(calculate)*, 9
calculate,nca_table,ANY-method *(calculate)*, 9
calculate,replicated_nca_table,campsis_output-method *(calculate)*, 9
calculate,theoretical_thalf_metric,ANY-method *(calculate)*, 9
campsis_output *(campsis_output-class)*, 10
campsis_output-class, 10
campsisnca_output *(campsisnca_output-class)*, 10
campsisnca_output-class, 10
campsisnca_tbl *(campsisnca_tbl-class)*, 10
campsisnca_tbl-class, 10
CAt, 11
Cavg, 12
cavg_metric-class, 12
CFB, 13
cfb_metric-class, 13
ChangeFromBaseline, 14
Cmax, 15
cmax_metric-class, 15
Cmin, 16
cmin_metric-class, 16
compute_nca_metric_summary, 17
computeTimeAboveBelow, 17
conc_at_metric-class, 18
Ctrough, 18
ctrough_metric-class, 19
custom_metric-class, 20
CustomMetric, 19
CustomMetricTbl, 20
cv, 21

dataframe_type-class, 21
deparse_digits, 21
deparse_one_line, 22
discard_category_column, 23
discardCategoryColumn, 22

export,replicated_nca_table,character-method, 23
extract_brace_values, 24

generate_table_code, 24
generate_table_code,nca_table,logical-method *(generate_table_code)*, 24
geocv, 25
geomean, 25
get_character_cols_only, 26
get_default_name, 27
get_default_name,auc_metric-method *(get_default_name)*, 27

get_default_name,avg_metric-method
(get_default_name), 27
get_default_name,cavg_metric-method
(get_default_name), 27
get_default_name,cfb_metric-method
(get_default_name), 27
get_default_name,cmax_metric-method
(get_default_name), 27
get_default_name,cmin_metric-method
(get_default_name), 27
get_default_name,conc_at_metric-method
(get_default_name), 27
get_default_name,ctrough_metric-method
(get_default_name), 27
get_default_name,last_metric-method
(get_default_name), 27
get_default_name,max_metric-method
(get_default_name), 27
get_default_name,min_metric-method
(get_default_name), 27
get_default_name,nca_metric-method
(get_default_name), 27
get_default_name,thalf_metric-method
(get_default_name), 27
get_default_name,theoretical_thalf_metric-method
(get_default_name), 27
get_default_name,time_above_limit_metric-method
(get_default_name), 27
get_default_name,time_below_limit_metric-method
(get_default_name), 27
get_default_name,tmax_metric-method
(get_default_name), 27
get_default_name,tmin_metric-method
(get_default_name), 27
get_default_name,value_at_metric-method
(get_default_name), 27
get_digits_code, 28
get_labels_code, 29
get_latex_name, 29
get_latex_name,abstract_avg_metric-method
(get_latex_name), 29
get_latex_name,abstract_max_metric-method
(get_latex_name), 29
get_latex_name,abstract_min_metric-method
(get_latex_name), 29
get_latex_name,abstract_value_at_metric-method
(get_latex_name), 29
get_latex_name,cfb_metric-method
(get_latex_name), 29
get_latex_name,ctrough_metric-method
(get_latex_name), 29
get_latex_name,last_metric-method
(get_latex_name), 29
get_latex_name,nca_metric-method
(get_latex_name), 29
get_latex_name,thalf_metric-method
(get_latex_name), 29
get_latex_name,theoretical_thalf_metric-method
(get_latex_name), 29
get_latex_name,tmax_metric-method
(get_latex_name), 29
get_latex_name,tmin_metric-method
(get_latex_name), 29
get_statistics_code, 30
get_strata, 31
get_strata,nca_table,logical-method
(get_strata), 31
get_table_summary_code, 31
get_unit, 32
get_unit,nca_analysis,character-method
(get_unit), 32
get_unit,nca_metrics,character-method
(get_unit), 32
get_unit,nca_table,character-method
(get_unit), 32
get_variable_type_code, 33
getCampsisncaOption, 26
glue_stat_display, 33
gt_type-class, 34
gtsummary_type-class, 34

i_value, 35
i_value,abstract_avg_metric,numeric,numeric-method
(i_value), 35
i_value,abstract_last_metric,numeric,numeric-method
(i_value), 35
i_value,abstract_max_metric,numeric,numeric-method
(i_value), 35
i_value,abstract_min_metric,numeric,numeric-method
(i_value), 35
i_value,abstract_time_above_or_below_limit_metric,numeric,
(i_value), 35
i_value,abstract_value_at_metric,numeric,numeric-method
(i_value), 35
i_value,auc_metric,numeric,numeric-method
(i_value), 35

i_value,cfb_metric,numeric,numeric-method (i_value), 35
i_value,custom_metric,numeric,numeric-method (i_value), 35
i_value,thalf_metric,numeric,numeric-method (i_value), 35
i_value,tmax_metric,numeric,numeric-method (i_value), 35
i_value,tmin_metric,numeric,numeric-method (i_value), 35
i_value_tbl, 37
i_value_tbl,custom_metric,tbl_df-method (i_value_tbl), 37
i_values, 36
i_values,nca_metric-method (i_values), 36
individual_campsisnca_tbl (individual_campsisnca_tbl-class), 34
individual_campsisnca_tbl-class, 34
individual_wide_campsisnca_tbl (individual_wide_campsisnca_tbl-class), 34
individual_wide_campsisnca_tbl-class, 34
iValue, 35

Last, 38
last_metric-class, 38

Max, 39
max_metric-class, 39
metrics.1cpt, 40
metrics.2cpt, 40
metrics.common, 41
Min, 41
min_metric-class, 42

nca_analyses-class, 46
nca_analysis-class, 46
nca_metric-class, 46
nca_metrics-class, 46
nca_options-class, 46
nca_table-class, 47
nca_time_window-class, 47
NCAAnalysis, 42
NCAMetrics, 43
NCAMetricsTable, 43
NCAOptions, 44
NCATable, 45

pk_bolus_md, 47
preserve_column_levels, 48

remove_column_levels, 49
replicated_nca_table-class, 50
ReplicatedNCATable, 49
restore_column_levels, 50

save_column_levels, *51*, 51
se, 52
stat_display_string, 52
stat_display_string,nca_metric-method (stat_display_string), 52
summary_campsisnca_tbl (summary_campsisnca_tbl-class), 53
summary_campsisnca_tbl-class, 53
summary_pretty_campsisnca_tbl (summary_pretty_campsisnca_tbl-class), 53
summary_pretty_campsisnca_tbl-class, 53
summary_wide_campsisnca_tbl (summary_wide_campsisnca_tbl-class), 53
summary_wide_campsisnca_tbl-class, 53

Thalf, 53
Thalf.1cpt, 54
thalf.1cpt.required, 55
Thalf.2cpt.dist, 55
Thalf.2cpt.eff, 56
thalf.2cpt.required, 56
Thalf.2cpt.z, 57
thalf_metric-class, 57
theoretical_thalf_metric-class, 58
time_above_limit_metric-class, 61
time_below_limit_metric-class, 61
TimeAboveLimit, 58
TimeBelowLimit, 59
timerange, 60
TimeWindow, *42*, 60
Tmax, 61
tmax_metric-class, 62
Tmin, 62
tmin_metric-class, 63
toGt, 63

trap, 64

undefined_nca_options-class, 65
undefined_nca_time_window-class, 66
UndefinedNCAOptions, 65
UndefinedTimeWindow, 65

value_at_metric-class, 67
ValueAt, 66