Package ‘campsis’

April 12, 2024

Type Package

Title Generic PK/PD Simulation Platform CAMPSIS

Version 1.5.2

Description A generic, easy-to-use and intuitive pharmacokinetic/pharmacodynamic (PK/PD) simulation platform based on R packages ‘rxode2’, ‘RxODE’ and ‘mrgsolve’. CAMPSIS provides an abstraction layer over the underlying processes of writing a PK/PD model, assembling a custom dataset and running a simulation. CAMPSIS has a strong dependency to the R package ‘campsismod’, which allows to read/write a model from/to files and adapt it further on the fly in the R environment. Package ‘campsis’ allows the user to assemble a dataset in an intuitive manner. Once the user’s dataset is ready, the package is in charge of preparing the simulation, calling ‘rxode2’, ‘RxODE’ or ‘mrgsolve’ (at the user’s choice) and returning the results, for the given model, dataset and desired simulation settings.

License GPL (>= 3)


BugReports https://github.com/Calvagone/campsis/issues

Depends campsismod (>= 1.1.0), R (>= 4.0.0)

Imports assertthat, digest, dplyr, ggplot2, purrr, future, MASS, methods, plyr, progressr, rlang, stats, tibble, tidyr

Suggests bookdown, devtools, gridExtra, knitr, mrgsolve, pkgdown, rmarkdown, roxygen2, rxode2, stringr, testthat, tictoc, vdiffr

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.3

Collate 'global.R' 'utilities.R' 'time_utilities.R' 'check.R'
  'generic.R' 'data.R' 'seed.R' 'distribution.R'
  'dataset_config.R' 'time_entry.R' 'occasion.R' 'occasions.R'
R topics documented:

'treatment_iov.R' 'treatment_iovs.R' 'dose_adaptation.R'
'dose_adaptations.R' 'treatment_entry.R' 'treatment.R'
'observations.R' 'observations_set.R' 'covariate.R'
'covariates.R' 'bootstrap.R' 'protocol.R' 'arm.R' 'arms.R'
'event.R' 'events.R' 'scenario.R' 'scenarios.R'
'simulation_engine.R' 'dataset.R' 'parameter_uncertainty.R'
'event_logic.R' 'dataset_summary.R' 'hardware_settings.R'
'simulation_progress.R' 'solver_settings.R' 'nocb_settings.R'
'declare_settings.R' 'progress_settings.R'
'internal_settings.R' 'simulation_settings.R' 'plan_setup.R'
'simulate_preprocess.R' 'simulate.R' 'results_processing.R'
'default_plot.R'

NeedsCompilation no

Author Nicolas Luyckx [aut, cre]
Maintainer Nicolas Luyckx <nicolas.luyckx@calvagone.com>
Repository CRAN
Date/Publication 2024-04-12 09:00:09 UTC
R topics documented:

distribution-class .................................................. 16
DoseAdaptation ......................................................... 16
dose_adaptation-class ............................................. 17
dose_adaptations-class ............................................. 17
dosingOnly ............................................................ 17
EtaDistribution ........................................................ 18
Event ................................................................. 18
EventCovariate ....................................................... 19
Events ................................................................. 20
events-class .......................................................... 20
event_covariate-class ................................................ 20
FixedDistribution ..................................................... 21
fixedCovariate-class ................................................ 21
fixed_distribution-class ........................................... 21
FunctionDistribution .................................................. 22
function_distribution-class ....................................... 22
generateIIV .......................................................... 23
generateIIV_ ........................................................ 23
getAvailableTimeUnits ............................................. 24
getCovariates ........................................................ 24
getEventCovariates .................................................. 25
getFixedCovariates ................................................... 25
getIOVs ............................................................... 26
getOccasions .......................................................... 27
getSeedForDatasetExport .......................................... 27
getSeedForIteration .................................................. 28
getSeedForParametersSampling ................................... 28
getSplittingConfiguration ........................................ 29
getTimes ............................................................... 29
g getTimeVaryingCovariates ....................................... 30
Hardware .............................................................. 31
hardware_settings-class .......................................... 32
hours ................................................................. 32
Infusion .............................................................. 33
infusion-class ........................................................ 34
internal_settings-class ............................................. 34
IOV ................................................................. 34
length,arm-method ................................................... 35
length,dataset-method ............................................. 35
LogNormalDistribution .............................................. 36
minutes ............................................................... 36
months ............................................................... 37
mrgsolve_engine-class ............................................. 37
nhanes ............................................................... 37
NOCB ............................................................... 38
nocb_settings-class ................................................ 39
NormalDistribution .................................................. 39
R topics documented:

Observations ......................................................... 40
observations-class .............................................. 40
observations_set-class ......................................... 40
obsOnly .............................................................. 41
Occasion ............................................................. 41
occasion-class .................................................... 42
occasions-class ................................................... 42
ParameterDistribution ............................................ 42
PI ................................................................. 43
Progress ............................................................ 43
progress_settings-class ....................................... 44
protocol-class .................................................. 44
retrieveParameterValue ........................................ 44
rxode_engine-class ............................................. 45
sample ............................................................. 45
scatterPlot ........................................................ 46
Scenario ........................................................... 47
scenario-class .................................................. 47
Scenarios .......................................................... 48
scenarios-class .................................................. 48
seconds ............................................................ 48
setLabel .......................................................... 49
setSubjects ....................................................... 49
Settings ............................................................ 50
setupPlanDefault ................................................ 50
setupPlanSequential ............................................ 50
shadedPlot ........................................................ 51
simulate ........................................................... 52
SimulationProgress ............................................... 56
simulation_engine-class ....................................... 56
simulation_progress-class ................................... 57
simulation_settings-class ..................................... 57
Solver .............................................................. 58
solver_settings-class ......................................... 58
spaghettiPlot ..................................................... 59
standardiseTime .................................................. 59
TimeVaryingCovariate ........................................... 60
time_varying_covariate-class ............................... 60
treatment-class ................................................ 60
treatment iov-class ............................................ 60
treatment_ivs-class ........................................... 60
undefined_distribution-class ............................... 61
UniformDistribution ............................................. 62
VPC .............................................................. 62
vpcPlot ........................................................... 63
weeks ............................................................. 63
years .............................................................. 64

Index 65
applyCompartmentCharacteristics

Apply compartment characteristics from model. In practice, only compartment infusion duration needs to be applied.

Description

Apply compartment characteristics from model. In practice, only compartment infusion duration needs to be applied.

Usage

applyCompartmentCharacteristics(table, properties)

Arguments

table current dataset

properties compartment properties from model

Value

updated dataset

Arm

Create a treatment arm.

Description

Create a treatment arm.

Usage

Arm(id = as.integer(NA), subjects = 1, label = as.character(NA))

Arguments

id unique identifier for this arm (available through dataset), integer. If NA (default), this identifier is auto-incremented.

subjects number of subjects in arm, integer

label arm label, single character string. If set, this label will be output in the ARM column of CAMPISIS instead of the identifier.

Value

an arm
arm-class  

Arm class.

Description
Arm class.

Slots
- id  arm unique ID, integer
- subjects number of subjects in arm, integer
- label  arm label, single character string
- protocol protocol
- covariates covariates
- bootstrap covariates to be bootstrapped

arms-class  

Arms class.

Description
Arms class.

BinomialDistribution  

Binomial distribution.

Description
Binomial distribution.

Usage
BinomialDistribution(trials, prob)

Arguments
- trials  number of Bernoulli trials per observation (=subject), integer
- prob  probability of success for each trial

Value
a binomial distribution
Bolus

Create one or several bolus(es).

Description
Create one or several bolus(es).

Usage
Bolus(
  time,
  amount,
  compartment = NA,
  f = NULL,
  lag = NULL,
  ii = NULL,
  addl = NULL
)

Arguments
time treatment time(s), numeric value or vector. First treatment time if used together with ii and addl.
amount amount to give as bolus, single numeric value
compartment compartment index, single integer value
f fraction of dose amount, distribution
lag dose lag time, distribution
ii interdose interval, requires argument 'time' to be a single numeric value
addl number of additional doses, requires argument 'time' to be a single integer value

Value
a single bolus or a list of boluses

bolus-class Bolus class.

Description
Bolus class.
Bootstrap

Create a bootstrap object.

Description

Create a bootstrap object.

Usage

Bootstrap(
  data,
  id = "BS_ID",
  replacement = FALSE,
  random = FALSE,
  export_id = FALSE
)

Arguments

data data frame to be bootstrapped. It must have a unique identifier column named according to the specified argument ‘id’ (default value is ‘BS_ID’). Other columns are covariates to bootstrap. They must all be numeric. Whatever the configuration of the bootstrap, these covariates are always read row by row and belong to a same individual.

id unique identifier column name in data
replacement values can be reused or not when drawn, logical
random values are drawn randomly, logical
export_id tell CAMPSIS if the identifier ‘BS_ID’ must be output or not, logical

Value

a bootstrap object

bootstrap-class

Bootstrap class.

Description

Bootstrap class.
**Slots**

- `data`  values to draw, numeric vector
- `replacement`  values can be reused or not, logical
- `random`  values are drawn randomly, logical

**Description**

Create a bootstrap distribution. During function sampling, CAMPSIS will generate values depending on the given data and arguments.

**Usage**

```r
BootstrapDistribution(data, replacement = FALSE, random = FALSE)
```

**Arguments**

- `data`  values to draw, numeric vector
- `replacement`  values can be reused or not, logical
- `random`  values are drawn randomly, logical

**Value**

a bootstrap distribution

**Description**

Bootstrap distribution class.
**campsis_handler**  
*Suggested Campsis handler for showing the progress bar.*

**Description**
Suggested Campsis handler for showing the progress bar.

**Usage**
campsis_handler()

**Value**
a progressr handler list

**ConstantDistribution**  
*Create a constant distribution. Its value will be constant across all generated samples.*

**Description**
Create a constant distribution. Its value will be constant across all generated samples.

**Usage**
ConstantDistribution(value)

**Arguments**
value covariate value, single numeric value

**Value**
a constant distribution (same value for all samples)

**constant_distribution-class**  
*Constant distribution class.*

**Description**
Constant distribution class.

**Slots**
value covariate value, single numeric value
convertTime

Convert numeric time vector based on the provided units.

Description

Convert numeric time vector based on the provided units.

Usage

convertTime(x, from, to)

Arguments

x     numeric time vector
from  unit of x, single character value
to    destination unit, single character value

Value

numeric vector with the converted times

Covariate

Create a non time-varying (fixed) covariate.

Description

Create a non time-varying (fixed) covariate.

Usage

Covariate(name, distribution)

Arguments

name     covariate name, single character value
distribution covariate distribution

Value

a fixed covariate
covariate-class  

* Covariate class. *

**Description**

Covariate class.

**Slots**

- `name`  
  covariate name, single character value
- `distribution`  
  covariate distribution

covariates-class  

* Covariates class. *

**Description**

Covariates class.

Dataset  

* Create a dataset. *

**Description**

Create a dataset.

**Usage**

```r
Dataset(subjects = NULL)
```

**Arguments**

- `subjects`  
  number of subjects in the default arm

**Value**

- a dataset
Description

Dataset class.

Slots

- arms: a list of treatment arms
- config: dataset configuration for export
- iiv: data frame containing the inter-individual variability (all ETAS) for the export

DatasetConfig

Create a dataset configuration. This configuration allows CAMPSIS to know which are the default depot and observed compartments.

Description

Create a dataset configuration. This configuration allows CAMPSIS to know which are the default depot and observed compartments.

Usage

```r
DatasetConfig(
  defDepotCmt = 1,
  defObsCmt = 1,
  exportTSLD = FALSE,
  exportTDOS = FALSE,
  timeUnitDataset = "hour",
  timeUnitExport = "hour"
)
```

Arguments

- defDepotCmt: default depot compartment, integer
- defObsCmt: default observation compartment, integer
- exportTSLD: export column TSLD (time since last dose), logical
- exportTDOS: export column TDOS (time of last dose), logical
- timeUnitDataset: unit of time in dataset, character (‘hour’ by default)
- timeUnitExport: unit of time in export, character (‘hour’ by default)

Value

a dataset configuration
dataset_config-class  

Dataset configuration class.

Description

Dataset configuration class.

Slots

def_depot_cmt  default depot compartment, integer

def_obs_cmt  default observation compartment, integer

export_tslbd  export column TSLD, logical

export_tdos  export column TDOS, logical

time_unit_dataset  unit of time in dataset, character ('hour' by default)

time_unit_export  unit of time in export, character ('hour' by default)


days  

Convert days to hours.

Description

Convert days to hours.

Usage

days(x)

Arguments

x  numeric vector in days

Value

numeric vector in hours
Declare

Create declare settings.

Description

Create declare settings.

Usage

Declare(\texttt{variables = character(0)})

Arguments

\texttt{variables} uninitialized variables to be declared, only needed with \texttt{mrgsolve}

Value

Declare settings

\begin{verbatim}

declare_settings-class

Declare settings class.
\end{verbatim}

Description

Declare settings class.

Slots

\texttt{variables} uninitialized variables to be declared, only needed with \texttt{mrgsolve}

\begin{verbatim}

DiscreteDistribution  Discrete distribution.
\end{verbatim}

Description

Discrete distribution.

Usage

\texttt{DiscreteDistribution(x, prob, replace = TRUE)}
**DoseAdaptation**

**Arguments**

- **x**: vector of one or more integers from which to choose
- **prob**: a vector of probability weights for obtaining the elements of the vector being sampled
- **replace**: should sampling be with replacement, default is TRUE

**Value**

a discrete distribution

---

**distribution-class**  
*Distribution class. See this class as an interface.*

---

**Description**

Distribution class. See this class as an interface.

---

**DoseAdaptation**  
*Create a dose adaptation.*

---

**Description**

Create a dose adaptation.

**Usage**

DoseAdaptation(formula, compartments = integer(0))

**Arguments**

- **formula**: formula to apply, single character string, e.g. "AMT*WT"
- **compartments**: compartment numbers where the formula needs to be applied, integer vector. Default is integer(0) (formula applied on all compartments)

**Value**

a fixed covariate
dose_adaptation-class  

Dose adaptation class.

Description
Dose adaptation class.

Slots
formula  formula to apply, single character string, e.g. "AMT*WT"
compartments  compartment numbers where the formula needs to be applied

dose_adaptations-class  

Dose adaptations class.

Description
Dose adaptations class.

dosingOnly  

Filter CAMPSIS output on dosing rows.

Description
Filter CAMPSIS output on dosing rows.

Usage
dosingOnly(x)

Arguments
x  data frame, CAMPSIS output

Value
a data frame with the dosing rows
EtaDistribution

Description
Create an ETA distribution. The resulting distribution is a normal distribution, with mean=0 and sd=sqrt(OMEGA).

Usage
EtaDistribution(model, omega)

Arguments
- model: model
- omega: corresponding THETA name, character

Value
an ETA distribution

Event

Description
Create an interruption event.

Usage
Event(name = NULL, times, fun, debug = FALSE)

Arguments
- name: event name, character value
- times: interruption times, numeric vector
- fun: event function to apply at each interruption
- debug: output the variables that were changed through this event

Value
an event definition
**event-class**

*Event class.*

**Description**

Event class.

**Slots**

- **name**  event name, character value
- **times**  interruption times, numeric vector
- **fun**  event function to apply at each interruption
- **debug**  output the variables that were changed through this event

---

**EventCovariate**  

Create an event covariate. These covariates can be modified further in interruption events.

**Description**

Create an event covariate. These covariates can be modified further in interruption events.

**Usage**

EventCovariate(name, distribution)

**Arguments**

- **name**  covariate name, character
- **distribution**  covariate distribution at time 0

**Value**

a time-varying covariate
Events

Create a list of interruption events.

Description
Create a list of interruption events.

Usage

Events()

Value

a events object

events-class

Events class.

Description
Events class.

event_covariate-class

Event covariate class.

Description
Event covariate class.
Create a fixed distribution. Each sample will be assigned a fixed value coming from vector 'values'.

Usage

FixedDistribution(values)

Arguments

values covariate values, numeric vector (1 value per sample)

Value

a fixed distribution (1 value per sample)

Description

Fixed covariate class.

Description

Fixed distribution class.

Slots

values covariate values, numeric vector (1 value per sample)
FunctionDistribution

Create a function distribution. During distribution sampling, the provided function will be responsible for generating values for each sample. If first argument of this function is not the size (n), please tell which argument corresponds to the size 'n' (e.g. list(size="n")).

Description

Create a function distribution. During distribution sampling, the provided function will be responsible for generating values for each sample. If first argument of this function is not the size (n), please tell which argument corresponds to the size 'n' (e.g. list(size="n")).

Usage

FunctionDistribution(fun, args)

Arguments

fun function name, character (e.g. 'rnorm')
args list of arguments (e.g list(mean=70, sd=10))

Value

a function distribution

---

function_distribution-class

Function distribution class.

---

Description

Function distribution class.

Slots

fun function name, character (e.g. 'rnorm')
args list of arguments (e.g list(mean=70, sd=10))
generateIIV  

Generate IIV matrix for the given Campsis model.

**Description**

Generate IIV matrix for the given Campsis model.

**Usage**

```r
generateIIV(model, n, offset = 0)
```

**Arguments**

- `model`: Campsis model
- `n`: number of subjects
- `offset`: if specified, resulting ID will be ID + offset

**Value**

IIV data frame with ID column

generateIIV_  

Generate IIV matrix for the given OMEGA matrix.

**Description**

Generate IIV matrix for the given OMEGA matrix.

**Usage**

```r
generateIIV_(omega, n)
```

**Arguments**

- `omega`: omega matrix
- `n`: number of subjects

**Value**

IIV data frame
getAvailableTimeUnits  

Return the list of available time units.

Usage

getAvailableTimeUnits()

Value

character vector

getCovariates  

Get all covariates (fixed / time-varying / event covariates).

Description

Get all covariates (fixed / time-varying / event covariates).

Usage

getcovariates(object)

## S4 method for signature 'covariates'
getcovariates(object)

## S4 method for signature 'arm'
getcovariates(object)

## S4 method for signature 'arms'
getcovariates(object)

## S4 method for signature 'dataset'
getcovariates(object)

Arguments

object  any object

Value

all covariates from object
**getEventCovariates**

### Description

Get all event-related covariates.

### Usage

```r
getEventCovariates(object)
```

#### S4 method for signature 'covariates'

```r
gEventCovariates(object)
```

#### S4 method for signature 'arm'

```r
gEventCovariates(object)
```

#### S4 method for signature 'arms'

```r
gEventCovariates(object)
```

#### S4 method for signature 'dataset'

```r
gEventCovariates(object)
```

### Arguments

- **object**: any object

### Value

- all event-related covariates from object

---

**getFixedCovariates**

### Description

Get all fixed covariates.

### Usage

```r
getFixedCovariates(object)
```

#### S4 method for signature 'covariates'

```r
gFixedCovariates(object)
```

#### S4 method for signature 'arm'

```r
gFixedCovariates(object)
```

---
getFixedCovariates(object)

## S4 method for signature 'arms'
getFixedCovariates(object)

## S4 method for signature 'dataset'
getFixedCovariates(object)

Arguments

object any object

Value

all fixed covariates from object

getIOVs

Get all IOV objects.

Description

Get all IOV objects.

Usage

getIOVs(object)

## S4 method for signature 'arm'
getIOVs(object)

## S4 method for signature 'arms'
getIOVs(object)

## S4 method for signature 'dataset'
getIOVs(object)

Arguments

object any object

Value

all IOV's from object
getOccasions

Get all occasions.

Description

Get all occasions.

Usage

getOccasions(object)

## S4 method for signature 'arm'
getOccasions(object)

## S4 method for signature 'arms'
getOccasions(object)

## S4 method for signature 'dataset'
getOccasions(object)

Arguments

object any object

Value

all occasions from object

getSeedForDatasetExport

Get seed for dataset export.

Description

Get seed for dataset export.

Usage

getSeedForDatasetExport(seed, progress)

Arguments

seed original seed
progress simulation progress
getSeedForIteration  Get seed for iteration.

Description
Get seed for iteration.

Usage
getSeedForIteration(seed, progress)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seed</td>
<td>original seed</td>
</tr>
<tr>
<td>progress</td>
<td>simulation progress</td>
</tr>
</tbody>
</table>

Value
the seed value to be used for the given replicate number and iteration

getSeedForParametersSampling
Get seed for parameter uncertainty sampling.

Description
Get seed for parameter uncertainty sampling.

Usage
getSeedForParametersSampling(seed)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seed</td>
<td>original seed</td>
</tr>
</tbody>
</table>

Value
the seed value used to sample parameter uncertainty
getSplittingConfiguration

Get splitting configuration for parallel export.

Description
Get splitting configuration for parallel export.

Usage
getSplittingConfiguration(dataset, hardware)

Arguments
- dataset: Campsis dataset to export
- hardware: hardware configuration

Value
splitting configuration list (if 'parallel_dataset' is enabled) or NA (if 'parallel_dataset' disabled or if the length of the dataset is less than the dataset export slice size)

getTimes
Get all distinct times for the specified object.

Description
Get all distinct times for the specified object.

Usage
getimes(object)

## S4 method for signature 'observations_set'
getimes(object)

## S4 method for signature 'arm'
getimes(object)

## S4 method for signature 'arms'
getimes(object)

## S4 method for signature 'events'
getimes(object)

## S4 method for signature 'dataset'
getimes(object)
getTimeVaryingCovariates

Get all time-varying covariates.

**Description**

Get all time-varying covariates.

**Usage**

getTimeVaryingCovariates(object)

## S4 method for signature 'covariates'
getTimeVaryingCovariates(object)

## S4 method for signature 'arm'
getTimeVaryingCovariates(object)

## S4 method for signature 'arms'
getTimeVaryingCovariates(object)

## S4 method for signature 'dataset'
getTimeVaryingCovariates(object)

**Arguments**

object any object

**Value**

all time-varying covariates from object
Create hardware settings.

Usage

Hardware(
    cpu = 1,
    replicate_parallel = FALSE,
    scenario_parallel = FALSE,
    slice_parallel = FALSE,
    slice_size = NULL,
    dataset_parallel = FALSE,
    dataset_slice_size = 500,
    auto_setup_plan = NULL
)

Arguments

cpu number of CPU cores to use, default is 1
replicate_parallel enable parallel computing for replicates, default is FALSE
scenario_parallel enable parallel computing for scenarios, default is FALSE
slice_parallel enable parallel computing for slices, default is FALSE
slice_size number of subjects per simulated slice, default is NULL (auto-configured by Campsis depending on the specified engine)
dataset_parallel enable parallelisation when exporting dataset into a table, default is FALSE
dataset_slice_size dataset slice size when exporting subjects to a table, default is 500. Only applicable if ‘dataset_parallel’ is enabled.
auto_setup_plan auto-setup plan with the library future, if not set (i.e. =NULL), plan will be setup automatically if the number of CPU’s > 1.

Value

hardware settings
hardware_settings-class

Hardware settings class.

Description

Hardware settings class.

Slots

cpu number of CPU cores to use, default is 1
replicate_parallel enable parallel computing for replicates, default is FALSE
scenario_parallel enable parallel computing for scenarios, default is FALSE
slice_parallel enable parallel computing for slices, default is FALSE
slice_size number of subjects per simulated slice, default is NULL (auto-configured by Campsis depending on the specified engine)
dataset_parallel enable parallelisation when exporting dataset into a table, default is FALSE
dataset_slice_size dataset slice size when exporting subjects to a table, default is 500. Only applicable if 'dataset_parallel' is enabled.
auto_setup_plan auto-setup plan with the library future, default is FALSE

hours

Convert hours to hours (do nothing).

Description

Convert hours to hours (do nothing).

Usage

hours(x)

Arguments

x numeric vector in hours

Value

numeric vector in hours
Infusion

Create one or several infusion(s).

Description

Create one or several infusion(s).

Usage

Infusion(
  time,
  amount,
  compartment = NA,
  f = NULL,
  lag = NULL,
  duration = NULL,
  rate = NULL,
  ii = NULL,
  addl = NULL
)

Arguments

time treatment time(s), numeric value or vector. First treatment time if used together with ii and addl.
amount total amount to infuse, numeric
compartment compartment index, integer
f fraction of infusion amount, distribution
lag infusion lag time, distribution
duration infusion duration, distribution
rate infusion rate, distribution
ii interdose interval, requires argument 'time' to be a single numeric value
addl number of additional doses, requires argument 'time' to be a single integer value

Value

a single infusion or a list of infusions.
infusion-class

Infusion class.

Description

Infusion class.

Slots

duration  infusion duration, distribution
rate  infusion rate, distribution

internal_settings-class

Internal settings class (transient object from the simulation settings).

Description

Internal settings class (transient object from the simulation settings).

Slots

dataset_summary  dataset summary
progress  simulation progress
iterations  list of event iterations

IOV

Define inter-occasion variability (IOV) into the dataset. A new variable of name 'colname' will be output into the dataset and will vary at each dose number according to the given distribution.

Description

Define inter-occasion variability (IOV) into the dataset. A new variable of name 'colname' will be output into the dataset and will vary at each dose number according to the given distribution.

Usage

IOV(colname, distribution, doseNumbers = NULL)
Arguments

colname  name of the column that will be output in dataset
distribution  distribution
doseNumbers  dose numbers, if provided, IOV is generated at these doses only. By default, IOV is generated for all doses.

Value

an IOV object

---

length,arm-method  Return the number of subjects contained in this arm.

Description

Return the number of subjects contained in this arm.

Usage

```r
## S4 method for signature 'arm'
length(x)
```

Arguments

x  arm

Value

a number

---

length,dataset-method  Return the number of subjects contained in this dataset.

Description

Return the number of subjects contained in this dataset.

Usage

```r
## S4 method for signature 'dataset'
length(x)
```

Arguments

x  dataset
LogNormalDistribution  Create a log normal distribution.

Description
Create a log normal distribution.

Usage
LogNormalDistribution(meanlog, sdlog)

Arguments
meanlog  mean value of distribution in log domain
sdlog   standard deviation of distribution in log domain

Value
a log normal distribution

minutes  Convert minutes to hours.

Description
Convert minutes to hours.

Usage
minutes(x)

Arguments
x  numeric vector in minutes

Value
numeric vector in hours
months

Convert pharma months (1 month = 4 weeks) to hours.

Description

Convert pharma months (1 month = 4 weeks) to hours.

Usage

months(x)

Arguments

x numeric vector in months

Value

numeric vector in hours

mrgsolve_engine-class

mrgsolve engine class.

Description

mrgsolve engine class.

nhanes

NHANES database (demographics and body measure data combined, from 2017-2018).

Description

NHANES database (demographics and body measure data combined, from 2017-2018).

Usage

nhanes
**Format**

data frame

<table>
<thead>
<tr>
<th>BS_ID</th>
<th>Original identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEX</td>
<td>Sex: 1 for males, 2 for females</td>
</tr>
<tr>
<td>AGE</td>
<td>Age in years</td>
</tr>
<tr>
<td>BW</td>
<td>Body weight in kg</td>
</tr>
<tr>
<td>BMI</td>
<td>Body mass index</td>
</tr>
<tr>
<td>HT</td>
<td>Height in cm</td>
</tr>
</tbody>
</table>

**Source**


---

**NOCB**

Create NOCB settings.

**Description**

Create NOCB settings.

**Usage**

```r
NOCB(enable = NULL, variables = character(0))
```

**Arguments**

- **enable**
  - enable/disable next-observation carried backward mode (NOCB), default value is TRUE for mrgsolve, FALSE for RxODE
- **variables**
  - variable names subject to NOCB behavior (see vignette for more info)

**Value**

NOCB settings
NOCB settings class

Description

NOCB settings class.

Slots

enable  enable/disable next-observation carried backward mode (NOCB), default value is TRUE for mrgsolve, FALSE for RxODE

variables  variable names subject to NOCB behavior (see vignette for more info)

NormalDistribution  Create a normal distribution.

Description

Create a normal distribution.

Usage

NormalDistribution(mean, sd)

Arguments

mean  mean value of distribution
sd  standard deviation of distribution

Value

a normal distribution
Observations

Create an observations list. Please note that the provided ‘times’ will automatically be sorted. Duplicated times will be removed.

Description

Create an observations list. Please note that the provided ‘times’ will automatically be sorted. Duplicated times will be removed.

Usage

Observations(times, compartment = NA)

Arguments

times observation times, numeric vector
compartment compartment index, integer

Value

an observations list

observations-class

Observations class.

Description

Observations class.

Slots

times observation times, numeric vector
compartment compartment index, integer
dv observed values, numeric vector (FOR EXTERNAL USE)

observations_set-class

Observations set class.

Description

Observations set class.
obsOnly

Filter CAMPSIS output on observation rows.

Description
Filter CAMPSIS output on observation rows.

Usage
obsOnly(x)

Arguments
x data frame, CAMPSIS output

Value
a data frame with the observation rows

Occasion
Define a new occasion. Occasions are defined by mapping occasion values to dose numbers. A new column will automatically be created in the exported dataset.

Description
Define a new occasion. Occasions are defined by mapping occasion values to dose numbers. A new column will automatically be created in the exported dataset.

Usage
Occasion(colname, values, doseNumbers)

Arguments
colname name of the column that will be output in dataset
values the occasion numbers, any integer vector
doseNumbers the related dose numbers, any integer vector of same length as 'values'

Value
occasion object
occasion-class  

Occasion class.

Description

Occasion class.

Slots

colname single character value representing the column name related to this occasion
values occasion values, integer vector, same length as dose_numbers
dose_numbers associated dose numbers, integer vector, same length as values

occasions-class  

Occasions class.

Description

Occasions class.

ParameterDistribution  

Create a parameter distribution. The resulting distribution is a log-normal distribution, with meanlog=log(THETA) and sdlog=sqrt(OMEGA).

Description

Create a parameter distribution. The resulting distribution is a log-normal distribution, with meanlog=log(THETA) and sdlog=sqrt(OMEGA).

Usage

ParameterDistribution(model, theta, omega = NULL)

Arguments

model model
theta corresponding THETA name, character
omega corresponding OMEGA name, character, NULL if not defined

Value

a parameter distribution
Description

Compute the prediction interval summary over time.

Usage

PI(x, output, scenarios = NULL, level = 0.9, gather = TRUE)

Arguments

x data frame
output variable to show, character value
scenarios scenarios, character vector, NULL is default
level PI level, default is 0.9 (90% PI)
gather FALSE: med, low & up columns, TRUE: metric column

Value

a summary table

Progress

Create progress settings.

Description

Create progress settings.

Usage

Progress(tick_slice = TRUE)

Arguments

tick_slice tick() is called after each simulated slice, default is TRUE. In some cases, when
the number of subjects per slice is low, it may be useful disable this flag, to
improve performance issues.

Value

progress settings
**progress_settings-class**

*Progress settings class.*

**Description**

Progress settings class.

**Slots**

- **tick_slice**
  
tick() is called after each simulated slice, default is TRUE. In some cases, when the number of subjects per slice is low, it may be useful disable this flag, to improve performance issues.

**protocol-class**

*Protocol class.*

**Description**

Protocol class.

**retrieveParameterValue**

*Retrieve the parameter value (standardized) for the specified parameter name.*

**Description**

Retrieve the parameter value (standardized) for the specified parameter name.

**Usage**

retrieveParameterValue(model, paramName, default = NULL, mandatory = FALSE)

**Arguments**

- **model**
  - model
- **paramName**
  - parameter name
- **default**
  - default value if not found
- **mandatory**
  - must be in model or not

**Value**

the standardized parameter value or the given default value if not found
**rxode_engine-class**  
RxODE/rxode2 engine class.

**Description**  
RxODE/rxode2 engine class.

**Slots**  
rxode2 logical field to indicate if CAMPSIS should use rxode2 (field set to TRUE) or RxODE (field set to FALSE). Default is TRUE.

**sample**  
Sample generic object.

**Description**  
Sample generic object.

**Usage**  
sample(object, n, ...)  
## S4 method for signature 'constant_distribution, integer'  
sample(object, n)  
## S4 method for signature 'fixed_distribution, integer'  
sample(object, n)  
## S4 method for signature 'function_distribution, integer'  
sample(object, n)  
## S4 method for signature 'bootstrap_distribution, integer'  
sample(object, n)  
## S4 method for signature 'bolus, integer'  
sample(object, n, ...)  
## S4 method for signature 'infusion, integer'  
sample(object, n, ...)  
## S4 method for signature 'observations, integer'  
sample(object, n, ...)  
## S4 method for signature 'covariate, integer'
sample(object, n)

## S4 method for signature 'bootstrap, integer'
sample(object, n)

## S4 method for signature 'campsis_model, integer'
sample(object, n)

Arguments

- object: generic object
- n: number of samples required
- ...: extra arguments

Value
	sampling result

---

scatterPlot

Scatter plot (or X vs Y plot).

Description

Scatter plot (or X vs Y plot).

Usage

scatterPlot(x, output, colour = NULL, time = NULL)

Arguments

- x: data frame
- output: the 2 variables to show, character vector
- colour: variable(s) to colour
- time: the time to look at those 2 variables, if NULL, min time is used (usually 0)

Value

a ggplot object
Scenario

Create an scenario.

Description

Create an scenario.

Usage

Scenario(name = NULL, model = NULL, dataset = NULL)

Arguments

name scenario name, single character string
model either a CAMPSIS model, a function or lambda-style formula
dataset either a CAMPSIS dataset, a function or lambda-style formula

Value

a new scenario

scenario-class  Scenario class.

Description

Scenario class.

Slots

name  scenario name, single character string
model  either a CAMPSIS model, a function or lambda-style formula
dataset  either a CAMPSIS dataset, a function or lambda-style formula
Scenarios

Description
Create a list of scenarios.

Usage
Scenarios()

Value
a scenarios object

---

scenarios-class

Description
Scenarios class.

---

seconds

Description
Convert seconds to hours.

Usage
seconds(x)

Arguments
x numeric vector in seconds

Value
numeric vector in hours
setLabel  

*Set the label.*

**Description**

Set the label.

**Usage**

setLabel(object, x)

```r
## S4 method for signature 'arm,character'
setLabel(object, x)
```

**Arguments**

- `object` any object that has a label
- `x` the new label

**Value**

the updated object

---

setSubjects  

*Set the number of subjects.*

**Description**

Set the number of subjects.

**Usage**

setSubjects(object, x)

```r
## S4 method for signature 'arm,integer'
setSubjects(object, x)
```

```r
## S4 method for signature 'dataset,integer'
setSubjects(object, x)
```

**Arguments**

- `object` any object
- `x` the new number of subjects
Value
the updated object

Description
Create advanced simulation settings.

Usage
Settings(...)

Arguments

Value
advanced simulation settings

setupPlanDefault
Setup default plan for the given simulation or hardware settings. This plan will prioritise the distribution of workers in the following order: 1) Replicates (if 'replicate_parallel' is enabled) 2) Scenarios (if 'scenario_parallel' is enabled) 3) Dataset export / slices (if 'dataset_export' or 'slice_parallel' is enabled)

Description
Setup default plan for the given simulation or hardware settings. This plan will prioritise the distribution of workers in the following order: 1) Replicates (if 'replicate_parallel' is enabled) 2) Scenarios (if 'scenario_parallel' is enabled) 3) Dataset export / slices (if 'dataset_export' or 'slice_parallel' is enabled)

Usage
setupPlanDefault(object)

Arguments
object simulation or hardware settings

Value
nothing
**setupPlanSequential**

Setup plan as sequential (i.e. no parallelisation).

**Description**

Setup plan as sequential (i.e. no parallelisation).

**Usage**

```r
setupPlanSequential()
```

**Value**

nothing

---

**shadedPlot**

Shaded plot (or prediction interval plot).

**Description**

Shaded plot (or prediction interval plot).

**Usage**

```r
shadedPlot(
  x, 
  output, 
  colour = NULL, 
  strat_extra = NULL, 
  level = 0.9, 
  alpha = 0.25
)
```

**Arguments**

- `x` data frame
- `output` variable to show
- `colour` variable(s) to colour
- `strat_extra` variable(s) to stratify, but not to colour (useful for use with facet_wrap)
- `level` PI level, default is 0.9 (90% PI)
- `alpha` alpha parameter (transparency) given to geom_ribbon

**Value**

a ggplot object
**Description**

Simulate function.

**Usage**

```r
simulate(
  model,
  dataset,
  dest = NULL,
  events = NULL,
  scenarios = NULL,
  tablefun = NULL,
  outvars = NULL,
  outfun = NULL,
  seed = NULL,
  replicates = 1,
  dosing = FALSE,
  settings = NULL
)
```

```r
## S4 method for signature
## 'campsis_model',
## dataset,
## character,
## events,
## scenarios,
## 'function',
## character,
## 'function',
## integer,
## integer,
## logical,
## simulation_settings'
simulate(
  model,
  dataset,
  dest = NULL,
  events = NULL,
  scenarios = NULL,
  tablefun = NULL,
  outvars = NULL,
  outfun = NULL,
  seed = NULL,
```
simulate(replicates = 1, dosing = FALSE, settings = NULL)

## S4 method for signature
## 'campsis_model,
## tbl_df,
## character,
## events,
## scenarios,
## 'function',
## character,
## 'function',
## integer,
## integer,
## logical,
## simulation_settings'
simulate(
  model,
  dataset,
  dest = NULL,
  events = NULL,
  scenarios = NULL,
  tablefun = NULL,
  outvars = NULL,
  outfun = NULL,
  seed = NULL,
  replicates = 1,
  dosing = FALSE,
  settings = NULL
)

## S4 method for signature
## 'campsis_model,
## data.frame,
## character,
## events,
## scenarios,
## 'function',
## character,
## 'function',
## integer,
## integer,
## logical,
## simulation_settings'
simulate(
  model,
dataset,
dest = NULL,
events = NULL,
scenarios = NULL,
tablefun = NULL,
outvars = NULL,
outfun = NULL,
seed = NULL,
replicates = 1,
dosing = FALSE,
settings = NULL
)

## S4 method for signature
## 'campsis_model,
## tbl_df,
## rxode_engine,
## events,
## scenarios,
## `function`,
## character,
## `function`,
## integer,
## integer,
## logical,
## simulation_settings'
simulate(
  model,
  dataset,
  dest = NULL,
  events = NULL,
  scenarios = NULL,
tablefun = NULL,
outvars = NULL,
outfun = NULL,
seed = NULL,
replicates = 1,
dosing = FALSE,
settings = NULL
)

## S4 method for signature
## 'campsis_model,
## tbl_df,
## mrgsolve_engine,
## events,
## scenarios,
## `function`,
simulate

## character,
## `function`,
## integer,
## integer,
## logical,
## `simulation_settings`

simulate(
  model,
  dataset,
  dest = NULL,
  events = NULL,
  scenarios = NULL,
  tablefun = NULL,
  outvars = NULL,
  outfun = NULL,
  seed = NULL,
  replicates = 1,
  dosing = FALSE,
  settings = NULL
)

**Arguments**

- **model**
  - generic CAMPSIS model
- **dataset**
  - CAMPSIS dataset or 2-dimensional table
- **dest**
  - destination simulation engine, default is 'RxODE'
- **events**
  - interruption events
- **scenarios**
  - list of scenarios to be simulated
- **tablefun**
  - function or lambda formula to apply on exported 2-dimensional dataset
- **outvars**
  - variables to output in resulting dataframe
- **outfun**
  - function or lambda formula to apply on resulting dataframe after each replicate
- **seed**
  - seed value
- **replicates**
  - number of replicates, default is 1
- **dosing**
  - output dosing information, default is FALSE
- **settings**
  - advanced simulation settings

**Value**

dataframe with all results
Create a simulation progress object.

Usage

```r
SimulationProgress(
    replicates = 1,
    scenarios = 1,
    progressor = NULL,
    hardware = NULL
)
```

Arguments

- `replicates`: total number of replicates to simulate
- `scenarios`: total number of scenarios to simulate
- `progressor`: progressr progressor
- `hardware`: hardware settings

Value

A progress bar

Simulation engine class

Description

Simulation engine class.
**simulation_progress-class**

*Simulation progress class.*

**Description**

Simulation progress class.

**Arguments**

- `replicates`: total number of replicates to simulate
- `scenarios`: total number of scenarios to simulate
- `iterations`: total number of iterations to simulate
- `slices`: total number of slices to simulate
- `replicate`: current replicate number being simulated
- `scenario`: current scenario number being simulated
- `iteration`: current iteration number being simulated
- `slice`: current slice number being simulated
- `progressor`: progressr progressor
- `hardware`: hardware settings

---

**simulation_settings-class**

*Simulation settings class.*

**Description**

Simulation settings class.

**Slots**

- `hardware`: hardware settings object
- `solver`: solver settings object
- `nocb`: NOCB settings object
- `declare`: declare settings (mrgsolve only)
- `progress`: progress settings
- `internal`: internal settings
Solver settings-class

Create solver settings.

Description

Create solver settings.

Usage

Solver(
  atol = 1e-08,
  rtol = 1e-08,
  hmax = NA,
  maxsteps = 70000L,
  method = "liblsoda"
)

Arguments

atol absolute solver tolerance, default is 1e-08
rtol relative solver tolerance, default is 1e-08
hmax limit how big a solver step can be, default is NA
maxsteps max steps between 2 integration times (e.g. when observations records are far
  apart), default is 70000
method solver method, for RxODE/rxode2 only: 'liblsoda' (default), 'lsoda', 'dop853',
  'indLin'. Mrgsolve's method is always 'lsoda'.

Value

solver settings

solver_settings-class

Solver settings class. See ?mrgsolve::update. See ?rxode2::rxSolve.

Description

Solver settings class. See ?mrgsolve::update. See ?rxode2::rxSolve.
Spaghetti plot.

Usage

spaghettiPlot(x, output, colour = NULL)

Arguments

x
output
colour

data frame
variable to show
variable(s) to colour

Value

plot

Standardise time to hours.

Usage

standardiseTime(x, unit)

Arguments

x
unit

numeric time vector
unit of x, single character value
Value

numeric vector with the times converted to hours

---

**TimeVaryingCovariate**  
*Create a time-varying covariate. This covariate will be implemented using EVID=2 rows in the exported dataset and will not use interruption events.*

---

**Description**

Create a time-varying covariate. This covariate will be implemented using EVID=2 rows in the exported dataset and will not use interruption events.

**Usage**

```r
TimeVaryingCovariate(name, table)
```

**Arguments**

- `name`: covariate name, character
- `table`: data.frame, must contain the mandatory columns 'TIME' and 'VALUE'. An 'ID' column may also be specified. In that case, ID’s between 1 and the max number of subjects in the dataset/arm can be used. All ID’s must have a VALUE defined for TIME 0.

**Value**

a time-varying covariate

---

**time_varying_covariate-class**  
*Time-varying covariate class.*

---

**Description**

Time-varying covariate class.

---

**treatment-class**  
*Treatment class.*

---

**Description**

Treatment class.
Description

Treatment IOV class.

Slots

colname  name of the column that will be output in dataset
distribution  distribution
dose_numbers  associated dose numbers, integer vector, same length as values

Description

Treatment IOV’s class.

undefined_distribution-class

Undefined distribution class. This type of object is automatically created in method toExplicitDistribution() when the user does not provide a concrete distribution. This is because S4 objects do not accept NULL values.

Description

Undefined distribution class. This type of object is automatically created in method toExplicitDistribution() when the user does not provide a concrete distribution. This is because S4 objects do not accept NULL values.
UniformDistribution

Create an uniform distribution.

Description

Create an uniform distribution.

Usage

UniformDistribution(min, max)

Arguments

<table>
<thead>
<tr>
<th>min</th>
<th>min value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>max value</td>
</tr>
</tbody>
</table>

Value

an uniform distribution

VPC

Compute the VPC summary. Input data frame must contain the following columns: replicate, low, med, up, any scenario column

Description

Compute the VPC summary. Input data frame must contain the following columns: replicate, low, med, up, any scenario column

Usage

VPC(x, scenarios = NULL, level = 0.9)

Arguments

<table>
<thead>
<tr>
<th>x</th>
<th>data frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>scenarios</td>
<td>scenarios, character vector, NULL is default</td>
</tr>
<tr>
<td>level</td>
<td>PI level, default is 0.9 (90% PI)</td>
</tr>
</tbody>
</table>
vpcPlot

Value

VPC summary with columns TIME, <scenarios> and all combinations of low, med, up (i.e. low_low, low_med, low_up, etc.)

Description

VPC plot.

Usage

vpcPlot(x, scenarios = NULL, level = 0.9, alpha = 0.15)

Arguments

x data frame, output of CAMPSIS with replicates
scenarios scenarios, character vector, NULL is default
level PI level, default is 0.9 (90% PI)
alpha alpha parameter (transparency) given to geom_ribbon

Value

a ggplot object

weeks

Convert weeks to hours.

Description

Convert weeks to hours.

Usage

weeks(x)

Arguments

x numeric vector in weeks

Value

numeric vector in hours
years

Convert pharma years (1 year = 12*4 weeks) to hours.

Description

Convert pharma years (1 year = 12*4 weeks) to hours.

Usage

years(x)

Arguments

x numeric vector in years

Value

numeric vector in hours
Index

* datasets
  - nhanes, 37

applyCompartmentCharacteristics, 5
Arm, 5
arm-class, 6
arms-class, 6

BinomialDistribution, 6
Bolus, 7
bolus-class, 7
Bootstrap, 8
bootstrap-class, 8
bootstrap_distribution-class, 9
BootstrapDistribution, 9

campsis_handler, 10
constant_distribution-class, 10
ConstantDistribution, 10
convertTime, 11
Covariate, 11
covariate-class, 12
covariates-class, 12

Dataset, 12
dataset-class, 13
dataset_config-class, 14
DatasetConfig, 13
days, 14
Declare, 15
declare_settings-class, 15
DiscreteDistribution, 15
distribution-class, 16
doze_adaptation-class, 17
doze_adaptations-class, 17
DoseAdaptation, 16
dosingOnly, 17

EtaDistribution, 18
Event, 18
event-class, 19
event_covariate-class, 20
EventCovariate, 19
Events, 20
events-class, 20

fixed_covariate-class, 21
fixed_distribution-class, 21
FixedDistribution, 21
function_distribution-class, 22
FunctionDistribution, 22

generateIIV, 23
generateIIV_, 23
getAvailableTimeUnits, 24
getCovariates, 24
getCovariates,arm-method
  (getCovariates), 24
getCovariates,arms-method
  (getCovariates), 24
getCovariates,covariates-method
  (getCovariates), 24
getCovariates,dataset-method
  (getCovariates), 24
getEventCovariates, 25
getEventCovariates,arm-method
  (getEventCovariates), 25
getEventCovariates,arms-method
  (getEventCovariates), 25
getEventCovariates,covariates-method
  (getEventCovariates), 25
getEventCovariates,dataset-method
  (getEventCovariates), 25
getFixedCovariates, 25
getFixedCovariates,arm-method
  (getFixedCovariates), 25
getFixedCovariates,arms-method
  (getFixedCovariates), 25
getFixedCovariates,covariates-method
  (getFixedCovariates), 25
getFixedCovariates, dataset-method (getFixedCovariates), 25
getIOVs, 26
getIOVs, arm-method (getIOVs), 26
getIOVs, arms-method (getIOVs), 26
getIOVs, dataset-method (getIOVs), 26
getOccasions, 27
getOccasions, arm-method (getOccasions), 27
getOccasions, arms-method (getOccasions), 27
getSeedForDatasetExport, 27
getSeedForIteration, 28
getSplittingConfiguration, 29
getTimes, 29
getTimes, arm-method (getTimes), 29
getTimes, arms-method (getTimes), 29
getTimes, dataset-method (getTimes), 29
getTimes, events-method (getTimes), 29
getTimes, observations_set-method (getTimes), 29
ggetTimeVaryingCovariates, 30
ggetTimeVaryingCovariates, arm-method (getTimeVaryingCovariates), 30
ggetTimeVaryingCovariates, arms-method (getTimeVaryingCovariates), 30
ggetTimeVaryingCovariates, covariates-method (getTimeVaryingCovariates), 30
ggetTimeVaryingCovariates, dataset-method (getTimeVaryingCovariates), 30

Hardware, 31
hardware_settings-class, 32
hours, 32

Infusion, 33
infusion-class, 34
internal_settings-class, 34
IOV, 34

length, arm-method, 35
length, dataset-method, 35
LogNormalDistribution, 36

minutes, 36
months, 37

mrgsolve_engine-class, 37
nhanes, 37
NOCB, 38
nocb_settings-class, 39
NormalDistribution, 39

Observations, 40
observations-class, 40
observations_set-class, 40
obsOnly, 41
Occasion, 41
occasion-class, 42
occasions-class, 42

ParameterDistribution, 42
PI, 43
Progress, 43
progress_settings-class, 44
protocol-class, 44

retrieveParameterValue, 44
rxode_engine-class, 45

sample, 45
sample, bolus, integer-method (sample), 45
sample, bootstrap, integer-method (sample), 45
sample, bootstrap_distribution, integer-method (sample), 45
sample, campsis_model, integer-method (sample), 45
sample, constant_distribution, integer-method (sample), 45
sample, covariate, integer-method (sample), 45
sample, fixed_distribution, integer-method (sample), 45
sample, function_distribution, integer-method (sample), 45
sample, infusion, integer-method (sample), 45
sample, observations, integer-method (sample), 45
scatterPlot, 46
Scenario, 47
scenario-class, 47
Scenarios, 48
scenarios-class, 48
seconds, 48
setLabel, 49
setLabel,arm,character-method
  (setLabel), 49
setSubjects, 49
setSubjects,arm,integer-method
  (setSubjects), 49
setSubjects,dataset,integer-method
  (setSubjects), 49
Settings, 50
setupPlanDefault, 50
setupPlanSequential, 51
shadedPlot, 51
simulate, 52
simulate,campsis_model,data.frame,character,events,scenarios,function,character,function,integer,integer
  (simulate), 52
simulate,campsis_model,dataset,character,events,scenarios,function,character,function,integer,integer,logical
  (simulate), 52
simulate,campsis_model,tbl_df,character,events,scenarios,function,character,function,integer,integer,logical
  (simulate), 52
simulate,campsis_model,tbl_df,mrgsolve_engine,events,scenarios,function,character,function,integer,integer,logical
  (simulate), 52
simulate,campsis_model,tbl_df,rxode_engine,events,scenarios,function,character,function,integer,integer,logical
  (simulate), 52
simulation_engine-class, 56
simulation_progress-class, 57
simulation_settings-class, 57
SimulationProgress, 56
Solver, 58
solver_settings-class, 58
spaghettiPlot, 59
standardiseTime, 59

time_varying_covariate-class, 60
TimeVaryingCovariate, 60
treatment-class, 60
treatment_iov-class, 61
treatment_iovs-class, 61

undefined_distribution-class, 61
UniformDistribution, 62

VPC, 62
vpcPlot, 63

weeks, 63

years, 64