Package ‘no.name’

June 29, 2020

Title Build Box/Compartment Models Via Excel

Version 12.03

Description This R package enables users to build
box/compartment models via an Excel workbook. ODE (ordinary
differential equation) or CTMC (continuous-time Markov chain) models
can be built with either single (ODE, CTMC) or multiple age groups
(ODE). This package also provides tools for users to perform
sensitivity analsyes/parameter sweeps on their models and to visualize
model results.

License CC BY 4.0

Depends R (>= 2.10)

VignetteBuilder knitr

Encoding UTF-8

Language en

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.0

Imports adaptivetau,
deSolve,
dplyr,
DT,
forcats,
ggplot2,
htmlwidgets,
lhs,
magrittr,
openxlsx,
plotly,
readxl,
scales,
tidyr,
triangle

Suggests klippy,
knitr,
rmarkdown

R topics documented:

`assess.parameter.importance` *Assess parameter importance*

Description

Assess parameter importance

Usage

```
assess.parameter.importance(don, X, Y, method)
```

Arguments

`don`	a data frame.
`X`	a numeric vector.
`Y`	a numeric vector.
`method`	a character element ("kendall-partial-correlation-slow", "pearson-partial-correlation-fast", #' "pearson-partial-correlation-slow", "spearman-partial-correlation-slow", "negative-log-p-value", "t-test").

Value

a vector.

compare.models *Compare results from box/compartment models*

Description

Compare results from box/compartment models

Usage

```
compare.models(
  solution1,
  solution2,
  age.suffix2 = "",
  ignore.vars = NULL,
  time.scope = c(0, Inf),
  tolerance = list(absolute = 2, relative = 1e-04)
)
```

Arguments

`solution1`	a box/compartment model object.
`solution2`	a box/compartment model object.
`age.suffix2`	a character element.
`ignore.vars`	a logical element.
`time.scope`	a numeric vector.
`tolerance`	a list with absolute and relative variables.

Value

a vector.

get.path *Return the complete path to a demo model Excel workbook*

Description

Return the complete path to a demo model Excel workbook

Usage

```
get.path(x)
```

Arguments

`x`	a character element representing a file name ("demo.model.1.xls", "demo.model.3.xls", "demo.model.4a.xls", "demo.model.4b.xls").

Value

a character element representing the complete path to the file. If the file cannot be found, an empty element will be returned.

Examples

```
path.to.demo.model <- get.path("demo.model.1.xls")
```

get.scatter.plot *Render a scatter plot*

Description

Render a scatter plot

Usage

```
get.scatter.plot(
  x,
  y,
  x_label_text = deparse(substitute(x)),
  y_label_text = deparse(substitute(y)),
  geom_point_size = 2,
  element_text_size = 12,
  height = NULL,
  width = NULL
)
```

Arguments

x	a numeric vector.
y	a numeric vector.
x_label_text	a character element (by default, the vector name).
y_label_text	a character element (by default, the vector name).
geom_point_size	a numeric element (by default, 2).
element_text_size	a numeric value (by default, 12).
height	an integer element representing the height of the plot in pixels.
width	an integer element representing the width of the plot in pixels.

Value

none.

Examples

```
# Load demo data
data("no.name.demo")

# Define results
outcomes.summary.df <- no.name.demo$results

get.scatter.plot(
  x = outcomes.summary.df$delta.overwrite,
  y = outcomes.summary.df$maxInc,
  height = 500,
  width = 756
)
```

`get.tornado.plot` *Render a tornado plot*

Description

Render a tornado plot

Usage

```
get.tornado.plot(
  outcome_variable,
  parameters = parms.tried.df,
  outcomes = outcomes.summary.df,
  method = "kendall-partial-correlation-slow",
  bin_width = 0.5,
  element_text_size = 12,
  order_by_absolute_value = FALSE,
  add_label = FALSE,
  height = NULL,
  width = NULL
)
```

Arguments

`outcome_variable`
a character element.

`parameters` the parms.tried.df data frame.

`outcomes` the outcomes.summary.df data frame.

`method` a character element ("kendall-partial-correlation-slow", "pearson-partial-correlation-fast", #' "pearson-partial-correlation-slow", "spearman-partial-correlation-slow", "negative-log-p-value", "t-test").

`bin_width` a numeric element representing the width of the bars (by default, 0.5).

`element_text_size`
a numeric element (by default, 12).

`order_by_absolute_value`
a logical element representing whether to order the bars by absolute value (by default, FALSE).

`add_label` a logical element representing whether to add labels to the bars (by default, FALSE).

`height` an integer element representing the height of the plot in pixels.

`width` an integer element representing the width of the plot in pixels.

Value

none.

Examples

```
# Load demo data
data("no.name.demo")

# Define results
parameters.swept <- no.name.demo$parameters
outcomes.summary.df <- no.name.demo$results

get.tornado.plot(
  outcome_variable = "maxInc",
  parameters = parameters.swept,
  outcomes = outcomes.summary.df,
  height = 500,
  width = 756
)
```

`get.tornado.table` *Render a tornado table*

Description

Render a tornado table

Usage

```
get.tornado.table(
  outcome.variable,
  parameters = parms.tried.df,
  outcomes = outcomes.summary.df,
  method = "kendall-partial-correlation-slow"
)
```

Arguments

`outcome.variable`
a character element.

`parameters` the parms.tried.df data frame.

`outcomes` the outcomes.summary.df data frame.

`method`	a character element ("kendall-partial-correlation-slow", "pearson-partial-correlation-fast", #' "pearson-partial-correlation-slow", "spearman-partial-correlation-slow", "negative-log-p-value", "t-test").

Value

none.

Examples

```
# Load demo data
data("no.name.demo")

# Define results
parameters.swept <- no.name.demo$parameters
outcomes.summary.df <- no.name.demo$results

get.tornado.table(
  outcome.variable = "maxInc",
  parameters = parameters.swept,
  outcomes = outcomes.summary.df
)
```

`no.name.demo` *Box/compartment models results from 108 parameter sweeps.*

Description

A dataset containing box/compartment model results from 108 parameter sweeps.

Usage

```
no.name.demo
```

Format

A list containing two data frames, outcomes.summary.df and parms.tried.df:

lambda.overwrite multiplier

delta.overweight multiplier ...

Source

https://github.com/barnzilla/no.name

save.model.in.workbook *Save box/compartment model as an Excel workbook*

Description

Save box/compartment model as an Excel workbook

Usage

```
save.model.in.workbook(input.info.list, file_name, map.names)
```

Arguments

`input.info.list`	a list.
`file_name`	a character element.
`map.names`	a list of workbook sheet names.

Value

none.

seir.n.age.classes *Main box/compartment model functions*

Description

Main box/compartment model functions

Usage

```
seir.n.age.classes(
  file.name,
  sheets.names,
  just.get.functions = FALSE,
  functions.kit = NULL,
  also.get.flows = NULL,
  agegrp.glue = "",
  CTMC.random.seeds = NULL
)
```

Arguments

`file.name` a character element.

`sheets.names` a list.

`just.get.functions` a logical element (by default, FALSE).

`functions.kit` a list containing one or more of the following: $differential.eqns.func, $post.processing.func, $post.processing.companion.kit, $CTMC.eqns.func, $CTMC.eqns.func.companion (by default, NULL).

`also.get.flows` a character element (by default, NULL).

`agegrp.glue` a character element (by default, an empty character element).

`CTMC.random.seeds` an integer vector (by default, NULL).

Value

a list containing the results of the box/compartment model.

Examples

```
# Get full path to demo model (in Excel workbook)
# that comes with the no.name package
model.1.workbook <- get.path("demo.model.1.xls")

# Define workbook sheet names
sheet.names <- list(
  parms.notime.0d = "Parameters any time any age",
  parms.0d = "Parameters any age",
  parms.1d = "Parameters by Age",
  parms.2d = "Parameters by Age x Age",
  initial.conditions = "Initial conditions",
  model.flow = "Model Specs (not lazy v1)",
  auxiliary.vars = "Intermediate calculations",
  post.processing = "Post Processing Empty"
)

# Fit model with model.flow = "Model Specs (not lazy v1)"
results.1 <- seir.n.age.classes(model.1.workbook, sheet.names)
```

`try.various.parms.values` *Perform a parameter sweep*

Description

Perform a parameter sweep

Usage

```
try.various.parms.values(
  seir.object,
  parm.cloud.grid.specs,
  only.show.parms.to.try = FALSE
)
```

Arguments

`seir.object` a box/compartment model object.

`parm.cloud.grid.specs` a data frame of parameter sweep specifications.

`only.show.parms.to.try` a logical value. By default, FALSE.

Value

a list of parameter sweep inputs and results.

Index