

Package ‘SimVitD’

April 2, 2022

Type Package

Title Simulation Tools for Planning Vitamin D Studies

Version 1.0.2

Description Simulation tools for planning Vitamin D studies. Individual vitamin D status profiles are simulated, modelling population heterogeneity in trial arms. Exposures to infectious agents are generated, with infection depending on vitamin D status.

License GPL (>= 2)

Maintainer Jason Wyse <wyseja@tcd.ie>

Imports simpleboot

Suggests parallel

Encoding UTF-8

NeedsCompilation no

Author Rebecca Mangan [aut],
Jason Wyse [aut, cre],
Lina Zgaga [aut]

Repository CRAN

Date/Publication 2022-04-02 18:20:10 UTC

R topics documented:

SimVitD-package	2
exposure.levels	3
infection.count	4
infection.count.plot	5
intensity.function	6
plot.exposure.levels	6
plot.power.calc	7
plot.vitd.curve	8
power.calc	8
print.exposure.levels	10
print.infection.count	11
print.power.calc	11

print.vitd.curve	12
rr.curve.plot	12
rr.profile.plot	13
summary.exposure.levels	14
summary.infection.count	15
summary.power.calc	15
summary.vitd.curve	16
vitd.curve	16

Index	18
--------------	-----------

SimVitD-package	<i>Simulation tools for vitamin D studies.</i>
-----------------	--

Description

A suite of tools for statistical planning of vitamin D trials.

Details

Package:	SimVitD
Type:	Package
Version:	1.0
Date:	2020-12-22
License:	GPL (>= 2)
LazyLoad:	yes

The SimVitD package uses simulation to aid in statistical planning of vitamin D trials. The core elements of the simulation based tools involve

- (i) simulation of an individual vitamin D status profiles taking into account natural cyclic variation in summer and winter
- (ii) simulation of exposures to potential infections in an individual's environment
- (iii) simulation of the possibility of developing infection conditional on vitamin D status at exposure
- (iv) Monte Carlo estimation of power of detecting supplementation effects for a selection of different supplementation scheme and two classes of hypothesis test.

The functions [vitd.curve](#) to generate individual status profiles and [exposure.levels](#) to simulate exposures. The accompanying vignette provides a gateway into the simulation tools.

Author(s)

Rebecca Mangan, Jason Wyse, Lina Zgaga

Examples

```
# simulate individual Vitamin D profiles
```

```

indprofiles <- vitd.curve( n=10, type="placebo" )

# make a plot of all these profiles
plot( indprofiles )

# intensity function for exposures to infection
intensfun <- intensity.function( summer.rate=0.1, winter.rate=0.9 )

# exposure times
expos <- exposure.levels( indprofiles, rate=2, intensfun, end=2 )

# plot of exposures on top of Vitamin D profiles
plot( expos )

# disease calculation
infect <- infection.count( expos )

# plot disease points on top of exposure points
infection.count.plot( expos, infect )

```

exposure.levels	<i>Generate exposures to infectious agent and vitamin D status</i>
-----------------	--

Description

`exposure.levels` creates a list of the exposures to an infectious agent and the status of vitamin D at the time of each exposure for each participant. The exposure times are simulated from a non homogeneous poisson process.

Usage

```
exposure.levels( x, rate, intensity.func = intensity.function(), start = 0, end = 1 )
```

Arguments

<code>x</code>	An object of class <code>vitd.curve</code> .
<code>rate</code>	Expected number of exposures per week, the rate at which exposures occur in an equivalent homogeneous Poisson process.
<code>intensity.func</code>	A function that takes time as sole argument and returns value between 0 and 1, scaling the rate argument. For example, exposures may be higher at certain times for seasonal infections.
<code>start</code>	Start time of the study in units of years.
<code>end</code>	End time of study in units of years.

Value

`exposure.levels` returns an object of class `exposure.levels` that is a list. The list has the following slots.

<code>exposures</code>	The exposure times to infection for each participant.
<code>levels</code>	The vitamin D status levels for each participant at the time of exposure to infection.

Author(s)

Rebecca Mangan and Jason Wyse

<code>infection.count</code>	<i>Generate whether a participant contracted an infection at exposure times.</i>
------------------------------	--

Description

Generates probabilities of developing infection, the relative risk and whether a participant becomes infected or not at exposure times.

Usage

```
infection.count( expos, baseline = 0.03, RR = 3, holding.time = 2, lohi.vit = c(10,70) )
```

Arguments

<code>expos</code>	An object of class <code>exposure.levels</code> .
<code>baseline</code>	Numeric, baseline prevalence of developing infection at any exposure time. Probability of developing infection when a participant has maximum (fully replete) vitamin D levels.
<code>RR</code>	Numeric, the relative risk of the study. The difference between becoming infected at highest and lowest vitamin D levels.
<code>holding.time</code>	Numeric, the mean number of weeks for after becoming infected when the participant can not become re-infected. Times are simulated independently from an exponential distribution with this mean.
<code>lohi.vit</code>	A vector of length 2 giving the level at which a participant is considered to have insufficient and sufficient vitamin D status levels.

Value

`infection.count` returns an object of class `infection.count` that is a list. The list has the following slots.

<code>baseline</code>	The baseline prevalence of disease.
<code>RR</code>	The relative risk of the study.

<code>inflection</code>	The inflection points of the relative risk curve.
<code>probs</code>	Matrix, the probability of developing infection at each exposure time for each participant.
<code>relativerisk</code>	Matrix, the relative risk of infection at each exposure time for each participant.
<code>infection</code>	Matrix, 1 if participant was infected at the corresponding exposure time, 0 if participant was not infected at exposure time.
<code>count</code>	Number of infections for each participant over the course of the study.
<code>mean</code>	The mean number of infections for the group over the course of the study.

Author(s)

Rebecca Mangan and Jason Wyse

`infection.count.plot` *Plotting the times of disease on the vitamin D curves.*

Description

Plot whether a participant was disease or not as points on top of the vitamin D curves plotted from `plot.vitd.curve`.

Usage

```
infection.count.plot( expos, infect, pch = 20, cex = 1.5, col = "red" )
```

Arguments

<code>expos</code>	An object of class <code>exposure.levels</code> .
<code>infect</code>	An object of class <code>infection.count</code> .
<code>pch</code>	Plotting symbol for the points.
<code>cex</code>	Standard graphical parameter.
<code>col</code>	Colour of the plotted points.

Author(s)

Rebecca Mangan and Jason Wyse

intensity.function *Generate an intensity function*

Description

Generates intensity function, function with time as sole argument.

Usage

```
intensity.function( summer.rate = 0, winter.rate = 1, flu = TRUE )
```

Arguments

summer.rate	Value between 0 and 1, the rate at which exposures occur in summer months.
winter.rate	Value between 0 and 1, the rate at which exposures occur in winter months.
flu	If TRUE define winter months as flu season (Sep-May in Northern Hemisphere, Mar-Nov in Southern Hemisphere). If FALSE have constant exposure rate all year.

Value

[intensity.function](#) returns a function that takes time as sole argument and returns value between 0 and 1, the returned function acts as `intensity.func` for input to [exposure.levels](#).

Author(s)

Rebecca Mangan and Jason Wyse

plot.exposure.levels *Plotting the exposures to infection on vitamin D status curves.*

Description

Plot the exposures to infection as points on top of the vitamin D curves plotted from `plot.vitd.curve`.

Usage

```
## S3 method for class 'exposure.levels'
plot( x, col = "blue", ... )
```

Arguments

x	An object of class <code>exposure.levels</code> .
col	Character, value for the colour of the points.
...	Additional arguments to be passed to the plot function.

Author(s)

Rebecca Mangan and Jason Wyse

plot.power.calc *Plotting a power.calc object*

Description

Plot the power between two groups as the number of participants and the relative risk vary.

Usage

```
## S3 method for class 'power.calc'
plot( x, col = "hotpink", lwd = 1.5, lty = 1, ylab = NULL,
      x.legend = NULL, y.legend = NULL, main.legend = "Risk scaling", legend.size = 1,
      target.power = NA, which = 1L, ... )
```

Arguments

x	An object of class power.calc.
col	Colour of the line.
lwd	Line width.
lty	Line type.
ylab	Title for the y-axis.
x.legend	The x position of the legend. If not given, tries to default to a sensible value.
y.legend	The y position of the legend. If not given, tries to default to a sensible value.
main.legend	Title for the legend.
legend.size	Size of the legend.
target.power	Add a horizontal dotted line at a target power. A value of NA suppresses plotting.
which	Type of plot. Value 1 gives a plot of power, and value 2 plot of estimated effect size.
...	Additional arguments to be passed to the plot function.

Author(s)

Rebecca Mangan and Jason Wyse

plot.vitd.curve *Plotting a vitd.curve object*

Description

Plot vitamin D status curves.

Usage

```
## S3 method for class 'vitd.curve'
plot( x, main = " ", xlab = " ", ylab = "25-hydroxyvitamin D",
      col=1:6, add = FALSE, ylim = NULL, ... )
```

Arguments

x	An object of class vitd.curve.
main	Main title for the plot.
xlab	A title for the x-axis.
ylab	A title for the y-axis.
col	A vector of colours for the plotted status curves.
add	If TRUE, add to current plotting window
ylim	Gives the plotting limits on the y-axis.
...	Additional arguments to the plot function.

Author(s)

Rebecca Mangan and Jason Wyse

power.calc *Generate the power to detect the difference between two groups*

Description

generates a value for the power between two groups

Usage

```
power.calc( n, ratio=1, N = 500, test.type, sig.level = 0.05,
            vitdcurves.placebo = NULL, vitdcurves.treatment = NULL,
            baseline = 0.03, RR = 3, rate = 1, intensity.func = intensity.function(),
            holding.time = 2, lohi.vit = c(10,70), clt = NULL, mc.error = 1, boot.rep = 9999,
            parallel = FALSE, num.cores = NULL, verbose=FALSE )
```


Arguments

n	Numeric, the number of participants in the control group.
ratio	Ratio greater than or equal to 1 giving size of treatment group as a multiple of n. Defaults to groups of equal size.
N	Number of simulations of the study to run.
test.type	Type of test to calculate the power, one of "count" or "proportions".
sig.level	Significance level used to test for a statistically significant difference between the groups.
vitdcurves.placebo	An object of class vitd.curve.
vitdcurves.treatment	An object of class vitd.curve.
baseline	Baseline prevalence of getting diseased at any exposure time. Probability of getting diseased when a participant has sufficient vitamin D levels.
RR	Fold risk difference between getting infection between the most deficient and most sufficient vitamin D levels.
rate	Expected number of exposures per week, the rate at which exposures occur in the equivalent homogeneous Poisson process.
intensity.func	Function taking time as sole argument and returns value between 0 and 1, input to nhpp function see poisson package.
holding.time	Expected number of weeks for the holding time.
lohi.vit	Inflection points of the relative risk curve used in infection.count.
clt	Logical or vector of logical values of same length as n indicating whether to use Central Limit Theorem approximation for the test when TRUE. Default is NULL and determination is automatic based on whether group size is less than 35 or not.
mc.error	Number of times to repeat the experiment at each n value to explore Monte Carlo error.
boot.rep	Number of bootstrap samples to carry out non-parametric tests of hypotheses.
parallel	Use parallel processing to carry out the simulations. This will parallelise over mc.error.
num.cores	Number of cores to exploit in parallel mode. Defaults to (cores available) - 1.
verbose	If TRUE print out regular updates on progression. Not available when parallel is set to TRUE.

Value

`power.calc` returns an object of class `power.calc` that is a list. The list has the following slots.

test.type	The type of study the power has been calculated on.
baseline	The baseline prevalence for disease.
RR	The relative risk of the study.

<code>npergroup</code>	The number of participants per group in the study.
<code>mc.error</code>	Number of repetitions of experiment to approximate Monte Carlo error.
<code>power</code>	A <code>length(RR) x mc.error x length(npergroup)</code> array of estimated power values.
<code>eff.size</code>	A <code>length(RR) x mc.error x length(npergroup)</code> array of estimated treatment effects (differences between groups).

Author(s)

Rebecca Mangan and Jason Wyse

Examples

```
# simulate placebo group
placebo <- vitd.curve( n = 10, type = "placebo" )

# simulate treatment group
treatment <- vitd.curve( n = 10, type = "dynamic-dose" )

# intensity function for exposures to infection
intensfun <- intensity.function( summer.rate = 0, winter.rate = 1 )

# calculate power: example only- run for much larger num.sims
pow <- power.calc( n = c(10,20,30),
  N = 10, test.type = 'count',
  vitdcurves.placebo = placebo, vitdcurves.treatment = treatment,
  baseline = 0.03, RR = c(2,4), rate = 1, intensity.func = intensfun, boot.rep=2000 )

### NOT RUN ###
# approximate the Monte Carlo error in estimation of the power-- takes longer to run
#pow <- power.calc( n = c(10,20,30),
#  N = 100, test.type = 'count',
#  vitdcurves.placebo = placebo, vitdcurves.treatment = treatment,
#  baseline = 0.03, RR = c(2,3,4), rate = 1,
#  intensity.func = intensfun, mc.error = 10 )

# plot power curves
plot( pow, xlab = "n", x.legend = 10, y.legend = 1, main.legend = "Relative Risk" )
```

`print.exposure.levels` *Print summary of an exposure.levels object.*

Description

Print a summary of a `exposure.levels` object.

Usage

```
## S3 method for class 'exposure.levels'
print( x, ... )
```

Arguments

x An object of class `exposure.levels`.
... Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`print.infection.count` *Print a summary of a `infection.count` object*

Description

Print a summary of a `infection.count` object.

Usage

```
## S3 method for class 'infection.count'  
print( x, ... )
```

Arguments

x An object of class `infection.count`.
... Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`print.power.calc` *Summary of a `power.calc` object*

Description

Print a summary of a `power.calc` object.

Usage

```
## S3 method for class 'power.calc'  
print( x, ... )
```

Arguments

x An object of class `power.calc`.
... Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`print.vitd.curve` *Print summary of a vitd.curve object*

Description

Print a summary of a vitd.curve object.

Usage

```
## S3 method for class 'vitd.curve'  
print( x, ... )
```

Arguments

`x` An object of class vitd.curve.
`...` Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`rr.curve.plot` *Plotting the relative risk curve for vitamin D status.*

Description

Plot the relative risk curve for vitamin D showing times of exposure and whether a participant developed infection from exposure.

Usage

```
rr.curve.plot( expos, infect, idx = 1, main = NULL, xlab = "25-hydroxyvitamin D",  
              ylab = "Risk scaling", col = "blue", ... )
```

Arguments

expos	An object of class <code>exposure.levels</code> .
infect	An object of class <code>infection.count</code> .
idx	A vector of indexes of specific exposures to plot.
main	Main title for the plot.
xlab	A title for the x-axis.
ylab	A title for the y-axis.
col	Character, value for the colour of the points.
...	Additional arguments to plot.

Author(s)

Rebecca Mangan and Jason Wyse

rr.profile.plot *Plotting a vitamin D profile and relative risk curve.*

Description

Plot a vitamin D status profile for a single participant and the relative risk curve for vitamin D (with exposure times and whether a participant was infected at that exposure time) side by side.

Usage

```
rr.profile.plot( x, expos, infect, idx = 1, ... )
```

Arguments

x	An object of class <code>vi td.curve</code> .
expos	An object of class <code>exposure.levels</code> .
infect	An object of class <code>infection.count</code> .
idx	Index of curve to plot.
...	Additional arguments to plot.

Author(s)

Rebecca Mangan and Jason Wyse

Examples

```
# individual profiles
indprofiles <- vitd.curve( n=10, type="placebo" )

# intensity function for exposures to infection
intensfun <- intensity.function( summer.rate=0.1, winter.rate=0.9 )

# exposure times
expos <- exposure.levels( indprofiles, rate=2, intensfun, end=2 )

# disease calculation
infect <- infection.count( expos )

# plot RR profile
rr.profile.plot( indprofiles, expos, infect )
```

summary.exposure.levels

Summary of a vitd.curve object

Description

Print a summary of a exposure.levels object.

Usage

```
## S3 method for class 'exposure.levels'
summary( object, ... )
```

Arguments

object	An object of class exposure.levels.
...	Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`summary.infection.count`*Summary of a infection.count object*

Description

Print a summary of a infection.count object.

Usage

```
## S3 method for class 'infection.count'  
summary( object, ... )
```

Arguments

<code>object</code>	An object of class <code>infection.count</code> .
<code>...</code>	Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

`summary.power.calc`*Summary of a power.calc object*

Description

Print a summary of a power.calc object.

Usage

```
## S3 method for class 'power.calc'  
summary( object, ... )
```

Arguments

<code>object</code>	An object of class <code>power.calc</code> .
<code>...</code>	Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

summary.vitd.curve *Summary of a vitd.curve object*

Description

Print a summary of a vitd.curve object.

Usage

```
## S3 method for class 'vitd.curve'
summary( object, ... )
```

Arguments

object An object of class vitd.curve.
 ... Optional arguments to lower level functions.

Author(s)

Rebecca Mangan and Jason Wyse

vitd.curve *Generate Vitamin D curves*

Description

Generates a vitamin D status profile curve for each individual in a group

Usage

```
vitd.curve( n = 1, type = c("placebo", "fixed-dose", "dynamic-dose"), start = 0, end = 1,
  mu = 45, amplitude = 35, dyn.dose.thresh = 50, sd.mu = 5, sd.amplitude = 5,
  sd.dyn.dose.thresh = 5, supp.dose = 20, supp.dose.rate = Inf, weight = 1,
  sd.weight = 0, min.thresh = 10, north.hemi = TRUE, res = 40 )
```

Arguments

n Number of curves to simulate.
 type One of "placebo", "fixed-dose", "dynamic-dose".
 start Time in units of years when trial started.
 end Time in units of years when trial ended.
 mu The mean level of 25OHD in the trial arm around which there is cosine variation.
 amplitude Amplitude of cosine function describing variation around mu

dyn.dose.thresh	Threshold for the concentration-controlled scheme.
sd.mu	Standard deviation levels around mean; this is the standard deviation of H in vignette.
sd.amplitude	Standard deviation of the amplitude.
sd.dyn.dose.thresh	Standard deviation of the participant retained concentration in the concentration-controlled trial.
supp.dose	The 25OHD nmol/l equivalent for dosage in fixed-dose supplementation.
supp.dose.rate	Concentration parameter for fixed-dose scheme uptake. Large values imply that all participants derive the same equivalent.
weight	For fixed-dose supplementation, this is the expected value of the proportion of the dose which is always utilized.
sd.weight	Standard deviation of weight
min.thresh	The minimum detectable threshold of 25OHD. Defaults to 10 nmol/l.
north.hemi	Summer/winter months as in the Northern Hemisphere if TRUE and Southern Hemisphere if FALSE.
res	Resolution parameter for plotting of vitamin D curves.

Value

[vitd.curve](#) returns an object of class `vitd.curve`. Curve parameter settings are returned. The slot `curves` give the curve specific parameters for each of the `n` generated curves. Additionally, `time` used for plotting gives time values passed to `plot.vitd.curve`.

Author(s)

Rebecca Mangan and Jason Wyse

Index

`exposure.levels`, [2](#), [3](#), [3](#), [4](#), [6](#)

`infection.count`, [4](#), [4](#)

`infection.count.plot`, [5](#)

`intensity.function`, [6](#), [6](#)

`plot.exposure.levels`, [6](#)

`plot.power.calc`, [7](#)

`plot.vitd.curve`, [8](#)

`power.calc`, [8](#), [9](#)

`print.exposure.levels`, [10](#)

`print.infection.count`, [11](#)

`print.power.calc`, [11](#)

`print.vitd.curve`, [12](#)

`rr.curve.plot`, [12](#)

`rr.profile.plot`, [13](#)

SimVitD-package, [2](#)

`summary.exposure.levels`, [14](#)

`summary.infection.count`, [15](#)

`summary.power.calc`, [15](#)

`summary.vitd.curve`, [16](#)

`vitd.curve`, [2](#), [16](#), [17](#)