

Package: epidemics (via r-universe)

June 13, 2024

Title Composable Epidemic Scenario Modelling

Version 0.4.0.9000

Description A library of compartmental epidemic models taken from the published literature, and classes to represent affected populations, public health response measures including non-pharmaceutical interventions on social contacts, non-pharmaceutical and pharmaceutical interventions that affect disease transmissibility, vaccination regimes, and disease seasonality, which can be combined to compose epidemic scenario models.

License MIT + file LICENSE

URL <https://github.com/epiverse-trace/epidemics>,
<https://epiverse-trace.github.io/epidemics/>

BugReports <https://github.com/epiverse-trace/epidemics/issues>

Imports checkmate, cli, data.table, glue, Rcpp, RcppEigen, stats, utils, withr

Suggests bench, bookdown, colorspace, dplyr, EpiEstim, finalsize, ggdist, ggplot2, knitr, purrr, rmarkdown, scales, socialmixr, spelling, testthat (>= 3.0.0), tibble, tidy

LinkingTo BH, Rcpp, RcppEigen

VignetteBuilder knitr

Config/Needs/website epiverse-trace/epiversetheme

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Repository <https://epiverse-trace.r-universe.dev>

RemoteUrl <https://github.com/epiverse-trace/epidemics>

RemoteRef HEAD

RemoteSha f706261bac19282e813ca43675ff16dcb79b3d40

Contents

.intervention_on_rates	2
as.intervention	3
as.vaccination	3
epidemic_peak	4
epidemic_size	5
intervention	7
model_default	10
model_diphtheria	12
model_ebola	16
model_vacamole	20
new_infections	24
outcomes_averted	25
population	28
print.population	29
print.vaccination	30
print_intervention	30
vaccination	31

Index	34
--------------	-----------

.intervention_on_rates

Apply interventions to rate parameters

Description

Apply interventions to rate parameters

Usage

.intervention_on_rates(t, interventions, parameters)

Arguments

t	A single number for the simulation time.
interventions	A named list of list-like objects that each have at least the three members "time_begin", "time_end", and "reduction". These are used to calculate the effect on each of the named parameters in the simulation.
parameters	A named list of numeric parameters affected by interventions. This represents the model parameters, such as the transmission rate, β , or the recovery rate, γ .

Value

A named list of the same length as parameters, with the same names. These parameters can then be used in a timestep of an ODE model.

as.intervention	<i>Convert a list to a intervention object</i>
-----------------	--

Description

Convert a list to a intervention object

Usage

```
as.intervention(x, type = c("contacts", "rate"))
```

Arguments

x	A list, or an object that inherits from a list.
type	A string for the type of intervention: "contacts" for a <contact_intervention> or "rate" for a <rate_intervention>.

Value

A [intervention](#) class object.

Examples

```
# prepare a list
npi <- list(
  name = "npi",
  type = "contacts",
  time_begin = 30,
  time_end = 60,
  reduction = rep(0.1, 3)
)

as.intervention(npi)
```

as.vaccination	<i>Convert a list to a vaccination object</i>
----------------	---

Description

Convert a list to a vaccination object

Usage

```
as.vaccination(x)
```

Arguments

`x` A list, or an object that inherits from a list.

Value

A `vaccination` class object.

Examples

```
# prepare a list
vax <- list(
  name = "vax_regime",
  time_begin = matrix(1),
  time_end = matrix(100),
  nu = matrix(0.001)
)

as.vaccination(vax)
```

epidemic_peak

Get the time and size of a compartment's highest peak

Description

Get the time and size of a compartment's highest peak for all demography groups.

Usage

```
epidemic_peak(data, compartments = "infectious")
```

Arguments

`data` A `<data.frame>` or `<data.table>` of model output, typically the output of a compartmental model.

`compartments` A character vector for the compartments of interest.

Details

This is used for epidemics with a single peak. It is useful from a public health policy point of view to determine how bad an epidemic will be and when that happens.

Value

A `<data.table>` with columns "demography_group", "compartment", "time" and "value"; these specify the name of the demography group, the epidemiological compartment, and the peak time and value for each compartment in compartments.

Examples

```
# create a population
uk_population <- population(
  name = "UK population",
  contact_matrix = matrix(1),
  demography_vector = 67e6,
  initial_conditions = matrix(
    c(0.9999, 0.0001, 0, 0, 0),
    nrow = 1, ncol = 5L
  )
)

# run epidemic simulation with no vaccination or intervention
data <- model_default(
  population = uk_population,
  time_end = 600
)

# get the timing and peak of the exposed and infectious compartment
epidemic_peak(data, c("exposed", "infectious"))
```

epidemic_size

Get the epidemic size

Description

Get the size of the epidemic at any stage between the start and the end. This is calculated as the number of individuals *recovered* from infection at that stage of the epidemic.

Usage

```
epidemic_size(
  data,
  stage = 1,
  time = NULL,
  by_group = TRUE,
  include_deaths = FALSE,
  simplify = TRUE
)
```

Arguments

data	A table of model output, typically the output of <code>model_default()</code> or similar functions.
stage	A numeric vector for the stage of the epidemic at which to return the epidemic size; here 0.0 represents the start time of the epidemic i.e., the initial conditions of the epidemic simulation, while 1.0 represents the end of the epidemic simulation model (100% of model time). Defaults to 1.0, at which stage returned

	values represent the <i>final size</i> of the epidemic. This value is overridden by any values passed to the <code>time</code> argument.
<code>time</code>	Alternative to <code>stage</code> , an integer-like vector for the timepoint of the epidemic at which to return the epidemic size. Overrides any values passed to <code>stage</code> .
<code>by_group</code>	A logical representing whether the epidemic size should be returned by demographic group, or whether a single population-wide value is returned. Defaults to <code>TRUE</code> .
<code>include_deaths</code>	A logical value that indicates whether to count dead individuals in the epidemic size calculation. Defaults to <code>FALSE</code> . Setting <code>include_deaths = TRUE</code> makes the function look for a "dead" compartment in the data. If there is no such column, the function returns only the final number of recovered or removed individuals in each demographic group.
<code>simplify</code>	A logical determining whether the epidemic size data should be simplified to a vector with one element for each demographic group. If the length of <code>stage</code> or <code>time</code> is ≤ 1 , this argument is overridden and the data are returned as a <code><data.table></code> .

Details

This function can be used to calculate the *final size* of the epidemic, by setting `stage = 1.0` (100% of model time; the default).

The function allows for the calculation of epidemic sizes by demographic group as well as the total epidemic size.

Value

If `simplify == TRUE` and a single timepoint is requested, returns a vector of epidemic sizes of the same length as the number of demographic groups. If `by_group == FALSE`, sums the epidemic size to return an overall value for the full population.

If multiple timepoints are requested, or if multiple replicates are present under a specially named column "replicate" (only from the Ebola model), no simplification to a vector is possible; returns a `<data.table>` of timepoints and epidemic sizes at each timepoint.

All options return the absolute sizes and not proportions.

Examples

```
# create a population
uk_population <- population(
  name = "UK population",
  contact_matrix = matrix(1),
  demography_vector = 67e6,
  initial_conditions = matrix(
    c(0.9999, 0.0001, 0, 0, 0),
    nrow = 1, ncol = 5L
  )
)

# run epidemic simulation with no vaccination or intervention
```

```

data <- model_default(
  population = uk_population
)

# get the final epidemic size if no other arguments are specified
epidemic_size(data)

# get the epidemic size at the halfway point
epidemic_size(data, stage = 0.5)

# alternatively, get the epidemic size at `time = 50`
epidemic_size(data, time = 50)

```

intervention

Create an intervention for an epidemic model

Description

Prepare an object of the `<intervention>` super-class that specifies a modification of the model parameters.

A `<contacts_intervention>` is used to simulate a non-pharmaceutical intervention (NPI) regime that reduces the population's social contacts.

A `<rate_intervention>` is used to simulate a reduction in the model's rate parameters (such as the transmission rate β), and can be used to represent pharmaceutical interventions such as improved treatment, but also NPIs such as wearing masks.

Interventions have a single start and end time that applies to all demographic groups in the population, but can have groups-specific effects on the reduction of contacts.

Combine `<intervention>`-inheriting objects to create sequential or overlapping intervention regimes using `c()` on two or more `<intervention>`-inheriting objects.

Usage

```
intervention(name = NA_character_, type, time_begin, time_end, reduction)
```

```
is_intervention(x)
```

```
is_contacts_intervention(x)
```

```
is_rate_intervention(x)
```

```
## S3 method for class 'contacts_intervention'
c(x, ...)
```

```
## S3 method for class 'rate_intervention'
c(x, ...)
```

Arguments

name	String for the name of the intervention.
type	String for the type of intervention. May be one of "contacts" or "rate", for a <contacts_intervention> or <rate_intervention> respectively.
time_begin	Single number for the start time of the intervention.
time_end	Single number for the end time of the intervention.
reduction	For <contacts_intervention>s, a matrix with as many rows as the number of demographic groups in the type population, and a single column. Each element gives the group-specific proportion reduction in contacts. For <rate_intervention>s, a single number giving the proportion reduction in a model parameter contacts. See details for how c() can be used to combine interventions of the same subclass.
x	An <intervention> object, or an object to be checked as an <intervention> object.
...	intervention objects to combine with x to create a multi-dose <intervention> object.

Details

Epidemic models that can accommodate interventions on contacts are able to accommodate any number of interventions with different start and end times and different group-specific effects.

Epidemic models that can accommodate interventions on rates are also able to accommodate any number of interventions with different start and end times, but with only a uniform effect on the relevant rate.

When multiple contact interventions are combined using c(), the reduction in contacts is stacked column wise to form a matrix $[i, j]$.

When multiple rate interventions are combined using c(), the reduction in the rate is concatenated into a vector of the same length as the number of interventions.

Models such as `model_default_cpp()` are set up to treat interventions with overlapping periods (i.e., overlap between the time when they are active) as having an *additive effect* on contact or rate reductions.

For contact reductions, the group-specific effect of J overlapping interventions is thus a vector $\sum_{j=1}^J x_{ij}$, for each demographic group i . This is handled internally by the epidemic model code. For example, a contact reduction matrix for two perfectly overlapping interventions ($J = 2$) with

different effects across three demographic groups ($I = 3$) would be represented as: $\begin{bmatrix} 0.1 & 0.05 \\ 0.1 & 0.1 \\ 0.1 & 0.0 \end{bmatrix}$

In epidemic models, the cumulative group-specific effect when both interventions are active would be (0.15, 0.2, 0.1).

For rate reductions, the effect of overlapping interventions that reduce a particular rate is also considered to be additive.

Value

An object of the `<intervention>` S3 super-class, with possible sub-classes `<contact_intervention>` and `<rate_intervention>`.

Concatenating two or more `<intervention>`-inheriting objects using `c()` also returns a `<intervention>`-inheriting object of the same sub-class. This object holds the intervention-specific start and end times, and reductions specified by all the constituent intervention actions (by demographic group if an intervention on contacts).

The combined effect of these actions on the population is handled internally by epidemic model functions.

A "null" intervention generated using `.no_contacts_intervention(population)` or `.no_rate_intervention()` returns a `<intervention>` of the appropriate sub-class that has its start and end times, and its effect all set to 0.0.

`is_intervention()`, `is_contacts_intervention()`, and `is_rate_intervention()` each return a logical value for whether the object is of the `<intervention>`, `<contacts_intervention>`, or `<rate_intervention>` class, respectively.

Examples

```
# assuming a population with two age groups, 0 -- 18, and 18+
# an example in which schools are closed for 30 days (or other time units)
close_schools <- intervention(
  name = "close schools",
  type = "contacts",
  time_begin = 50,
  time_end = 80,
  reduction = matrix(c(0.5, 0.01)) # reduces contacts differentially
)
close_schools

# Check for intervention class
is_contacts_intervention(close_schools)

# Concatenating interventions
# create first intervention
npi_1 <- intervention(
  type = "contacts",
  time_begin = 30,
  time_end = 60,
  reduction = matrix(0.1)
)

# second intervention
npi_2 <- intervention(
  type = "contacts",
  time_begin = 45,
  time_end = 75,
  reduction = matrix(0.1)
)

c(npi_1, npi_2)
```

 model_default

Model an SEIR-V epidemic with interventions

Description

Simulate an epidemic using a deterministic, compartmental epidemic model with the compartments "susceptible", "exposed", "infectious", "recovered", and "vaccinated". This model can accommodate heterogeneity in social contacts among demographic groups, as well as differences in the sizes of demographic groups.

The `population`, `transmission_rate`, `infectiousness_rate`, and `recovery_rate` arguments are mandatory, while passing an `intervention` and `vaccination` are optional and can be used to simulate scenarios with different epidemic responses or different levels of the same type of response. See **Details** for more information.

Usage

```
model_default(
  population,
  transmission_rate = 1.3/7,
  infectiousness_rate = 1/2,
  recovery_rate = 1/7,
  intervention = NULL,
  vaccination = NULL,
  time_dependence = NULL,
  time_end = 100,
  increment = 1
)
```

Arguments

- | | |
|----------------------------------|--|
| <code>population</code> | An object of the <code>population</code> class, which holds a population contact matrix, a demography vector, and the initial conditions of each demographic group. See population() . |
| <code>transmission_rate</code> | A numeric for the rate at which individuals move from the susceptible to the exposed compartment upon contact with an infectious individual. Often denoted as β , with $\beta = R_0/\text{infectious period}$. See Details for default values. |
| <code>infectiousness_rate</code> | A numeric for the rate at which individuals move from the exposed to the infectious compartment. Often denoted as σ , with $\sigma = 1.0/\text{pre-infectious period}$. This value does not depend upon the number of infectious individuals in the population. See Details for default values. |
| <code>recovery_rate</code> | A numeric for the rate at which individuals move from the infectious to the recovered compartment. Often denoted as γ , with $\gamma = 1.0/\text{infectious period}$. See Details for default values. |

intervention	A named list of <code><intervention></code> s representing optional non-pharmaceutical or pharmaceutical interventions applied during the epidemic. Only a single intervention on social contacts of the class <code><contacts_intervention></code> is allowed as the named element "contacts". Multiple <code><rate_interventions></code> on the model parameters are allowed; see Details for the model parameters for which interventions are supported.
vaccination	A <code><vaccination></code> object representing an optional vaccination regime with a single dose, followed during the course of the epidemic, with a start and end time, and age-specific vaccination rates.
time_dependence	A named list where each name is a model parameter, and each element is a function with the first two arguments being the current simulation time, and x , a value that is dependent on time (x represents a model parameter). See Details for more information, as well as the vignette on time-dependence <code>vignette("time_dependence", package = "epidemics")</code> .
time_end	The maximum number of timesteps over which to run the model. Taken as days, with a default value of 100 days. May be a numeric vector.
increment	The size of the time increment. Taken as days, with a default value of 1 day.

Value

A `<data.table>`. If the model parameters and composable elements are all scalars, a single `<data.table>` with the columns "time", "compartment", "age_group", and "value", giving the number of individuals per demographic group in each compartment at each timestep in long (or "tidy") format is returned.

If the model parameters or composable elements are lists or list-like, a nested `<data.table>` is returned with a list column "data", which holds the compartmental values described above. Other columns hold parameters and composable elements relating to the model run. Columns "scenario" and "param_set" identify combinations of composable elements (population, interventions, vaccination regimes), and infection parameters, respectively.

Details: SEIRV model suitable for directly transmitted infections

Model parameters:

This model only allows for single, population-wide rates of transitions between compartments per model run.

However, model parameters may be passed as numeric vectors. These vectors must follow Tidyverse recycling rules: all vectors must have the same length, or, vectors of length 1 will be recycled to the length of any other vector.

The default values are:

- Transmission rate (β , `transmission_rate`): 0.186, assuming an $R_0 = 1.3$ and an infectious period of 7 days.
- Infectiousness rate (σ , `infectiousness_rate`): 0.5, assuming a pre-infectious period of 2 days.
- Recovery rate (γ , `recovery_rate`): 0.143, assuming an infectious period of 7 days.

Examples

```

# create a population
uk_population <- population(
  name = "UK population",
  contact_matrix = matrix(1),
  demography_vector = 67e6,
  initial_conditions = matrix(
    c(0.9999, 0.0001, 0, 0, 0),
    nrow = 1, ncol = 5L
  )
)

# run epidemic simulation with no vaccination or intervention
# and three discrete values of transmission rate
data <- model_default(
  population = uk_population,
  transmission_rate = c(1.3, 1.4, 1.5) / 7.0, # uncertainty in R0
)

# view some data
data

# run epidemic simulations with differences in the end time
# may be useful when considering different start dates with a fixed end point
data <- model_default(
  population = uk_population,
  time_end = c(50, 100, 150)
)

data

```

model_diphtheria

Model a diphtheria outbreak using a compartmental ODE model

Description

Simulate a diphtheria outbreak using a deterministic, compartmental ordinary differential equation model with the compartments "susceptible", "exposed", "infectious", "hospitalised", and "recovered". The model is based on Finger et al. (2019) and is intended to be used in the context of internally displaced people (IDP) or refugee camps. This model allows for a proportion of each demographic group to be vaccinated at the start of the outbreak, and thus to not contribute to the outbreak. The model also allows for changes to the number of susceptibles in each age group to model influxes or evacuations from camps.

Usage

```

model_diphtheria(
  population,
  transmission_rate = 4/4.5,

```

```

infectiousness_rate = 1/3,
recovery_rate = 1/3,
reporting_rate = 0.03,
prop_hosp = 0.01,
hosp_entry_rate = 0.2,
hosp_exit_rate = 0.2,
prop_vaccinated = 0 * population[["demography_vector"]],
intervention = NULL,
time_dependence = NULL,
population_change = NULL,
time_end = 100,
increment = 1
)

```

Arguments

- population** An object of the `population` class, which holds a population contact matrix, a demography vector, and the initial conditions of each demographic group. See [population\(\)](#).
- transmission_rate** A numeric for the rate at which individuals move from the susceptible to the exposed compartment upon contact with an infectious individual. Often denoted as β , with $\beta = R_0/\text{infectious period}$. See **Details** for default values.
- infectiousness_rate** A numeric for the rate at which individuals move from the exposed to the infectious compartment. Often denoted as σ , with $\sigma = 1.0/\text{pre-infectious period}$. This value does not depend upon the number of infectious individuals in the population. See **Details** for default values.
- recovery_rate** A numeric for the rate at which individuals move from the infectious to the recovered compartment. Often denoted as γ , with $\gamma = 1.0/\text{infectious period}$. See **Details** for default values.
- reporting_rate** A numeric for the proportion of infectious cases that is reported; this is a precursor to hospitalisation as only reported cases are hospitalised.
- prop_hosp** A numeric for the proportion of reported cases that is hospitalised.
- hosp_entry_rate** A numeric for the rate at which reported cases of infectious individuals are hospitalised. This is calculated as $1 / \text{time to hospitalisation}$, denoted τ_1 .
- hosp_exit_rate** A numeric for the rate at which individuals are discharged from hospital to enter the 'recovered' compartment. This is calculated as $1 / \text{time to discharge}$, denoted τ_2 .
- prop_vaccinated** A numeric vector of the same length as the number of demographic groups indicated the proportion of each group that is vaccinated. These individuals are not included in the model dynamics.
- intervention** A named list of `<rate_intervention>` objects representing optional pharmaceutical or non-pharmaceutical interventions applied to the model parameters listed above.

time_dependence	A named list where each name is a model parameter, and each element is a function with the first two arguments being the current simulation time, and x , a value that is dependent on time (x represents a model parameter). See Details for more information, as well as the vignette on time-dependence <code>vignette("time_dependence", package = "epidemics")</code> .
population_change	A two-element list, with elements named "time" and "values", giving the times of population changes, and the corresponding changes in the population of each demographic group at those times. "time" must be a numeric vector, while "values" must be a list of the length of "time", with each element a numeric vector of the same length as the number of demographic groups in population.
time_end	The maximum number of timesteps over which to run the model. Taken as days, with a default value of 100 days. May be a numeric vector.
increment	The size of the time increment. Taken as days, with a default value of 1 day.

Value

A data table with the columns "time", "compartment", "age_group", "value", and "run", giving the number of individuals per demographic group in each compartment at each timestep in long (or "tidy") format, with "run" indicating the unique parameter combination.

Details: Model an infection outbreak in a humanitarian camp setting

This model has been developed for diphtheria outbreaks in settings where interventions on social contacts are difficult to implement. It is suitable for application to the outbreak of similar, directly transmitted infectious diseases as well.

Model parameters:

This model only allows for single, population-wide rates transitions between compartments per model run.

However, model parameters may be passed as numeric vectors. These vectors must follow Tidyverse recycling rules: all vectors must have the same length, or, vectors of length 1 will be recycled to the length of any other vector.

The default values are taken from Finger et al. (2019) where possible:

- Transmission rate (β , `transmission_rate`): 0.8888889, assuming an R_0 of 4.0 and a total infectious period of 4.5 days.
- Infectiousness rate (σ , `infectiousness_rate`): 0.333, assuming a pre-infectious period of 3 days.
- Reporting rate (r , `reporting_rate`): 0.03, assuming that 3% of infectious cases are detected or reported.
- Proportion hospitalised (η , `prop_hosp`): 0.01, assuming that 1% of reported cases need hospital treatment.
- Hospital entry rate (τ_1 , `hosp_entry_rate`): 0.2, assuming that it takes 5 days for infectious individuals to seek hospital treatment.
- Hospital exit rate (τ_2 , `hosp_exit_rate`): 0.2, assuming that individuals are discharged from hospital after 5 days.

- Recovery rate (γ , recovery_rate): 0.333, assuming an infectious period following symptoms, of 3 days.

Modelling population changes:

This model allows changes to the number of susceptibles in each demographic group, to represent influxes or evacuations from the camp as would be expected in humanitarian relief situations. Users can specify the times and changes (to each demographic group) of changes using the population_changes argument, to examine the effect on outbreak dynamics.

References

Finger, F., Funk, S., White, K., Siddiqui, M. R., Edmunds, W. J., & Kucharski, A. J. (2019). Real-time analysis of the diphtheria outbreak in forcibly displaced Myanmar nationals in Bangladesh. *BMC Medicine*, 17, 58. doi:10.1186/s1291601912887.

Examples

```
# create a dummy camp population with three age groups
# diphtheria model is SEIHR
# assume that most are susceptible, some infectious
# values taken from supplementary material in Finger et al. for the
# Kutupalong camp, rounded to the nearest 100
n_age_groups <- 3
demography_vector <- c(83000, 108200, 224600)
initial_conditions <- matrix(0, nrow = n_age_groups, ncol = 5)

# set susceptibles and infectious
initial_conditions[, 1] <- demography_vector - 1
initial_conditions[, 3] <- rep(1, n_age_groups)

camp_pop <- population(
  contact_matrix = matrix(1, nrow = n_age_groups, ncol = n_age_groups),
  demography_vector = demography_vector,
  initial_conditions = initial_conditions / demography_vector
)

# assume younger age groups are vaccinated
prop_vaccinated <- c(0.2, 0.10, 0.1)

# run model for single, default parameter set
data <- model_diphtheria(
  camp_pop,
  prop_vaccinated = prop_vaccinated
)
head(data)
tail(data)

# run model with increase in population
# create population change data
p <- list(
  time = 70,
  values = list(
```

```

      c(1e4, 2e5, 1e5)
    )
  )

data <- model_diphtheria(
  camp_pop,
  prop_vaccinated = prop_vaccinated,
  population_change = p
)
head(data)
tail(data)

```

 model Ebola

Model an Ebola virus disease epidemic

Description

Simulate an epidemic using a discrete-time, stochastic SEIR compartmental model with compartments based on Li et al. (2019), and with Erlang passage times based on a model developed by Getz and Dougherty (2017), developed to model the West African Ebola virus disease (EVD) outbreak of 2013 – 2016. See **Details** for more information.

model Ebola_cpp() is an Rcpp implementation of this model that currently lags behind the R implementation, and is likely to be removed.

Usage

```

model Ebola(
  population,
  transmission_rate = 1.5/12,
  erlang_subcompartments = 2,
  infectiousness_rate = erlang_subcompartments/5,
  removal_rate = erlang_subcompartments/12,
  prop_community = 0.9,
  etu_risk = 0.7,
  funeral_risk = 0.5,
  intervention = NULL,
  time_dependence = NULL,
  time_end = 100,
  replicates = 100
)

```

Arguments

population	An object of the <population> class, see population() . This model only accepts a <population> without demographic structure, that is, the demography_vector must be a single number representing the total size of the affected population.
------------	---

The model also does not account for demographic differences in social contacts, which means that the `contact_matrix` is ignored. For consistency, the matrix must be square and have as many rows as demography groups, which is one.

<code>transmission_rate</code>	A numeric vector for the rate at which individuals move from the susceptible to the exposed compartment upon contact with an infectious individual. Often denoted as β , with $\beta = R_0/\text{infectious period}$. See Details for default values.
<code>erlang_subcompartments</code>	A numeric, integer-like vector for the number of Erlang sub-compartments assumed for the exposed, infectious, and hospitalised compartments. Defaults to 2.
<code>infectiousness_rate</code>	A numeric vector for the rate at which individuals move from the exposed to the infectious compartment. Often denoted as σ , with $\sigma = 1.0/\text{pre-infectious period}$. This value does not depend upon the number of infectious individuals in the population. See Details for default values.
<code>removal_rate</code>	A numeric vector for the rate at which infectious individuals transition from the infectious or hospitalised compartments to the funeral or removed compartments. This model does not distinguish between recoveries and deaths. Denoted in Getz and Dougherty as γ^I (see Details).
<code>prop_community</code>	A numeric vector for the proportion of infectious individuals who remain in the community and are not hospitalised for treatment. Defaults to 0.9.
<code>etu_risk</code>	A numeric vector for the relative risk of onward transmission of EVD from hospitalised individuals, with values between 0.0 and 1.0, where 0.0 indicates that hospitalisation completely prevents onward transmission, and 1.0 indicates that hospitalisation does not prevent onward transmission at all; values are relative to the baseline transmission rate β . Defaults to 0.7.
<code>funeral_risk</code>	A numeric vector for the relative risk of onward transmission of EVD from funerals of individuals who died with EVD. Must be between 0.0 and 1.0, where 0.0 indicates that there is no onward transmission, and 1.0 indicates that funeral transmission is equivalent to the baseline transmission rate in the community β . Defaults to 0.5.
<code>intervention</code>	An optional named list of <code><rate_intervention></code> objects representing optional pharmaceutical or non-pharmaceutical interventions applied to the model parameters listed above. May also be a list of such lists, in which case each set of interventions is treated as a separate scenario. See Details below.
<code>time_dependence</code>	An optional named list where each element is a function with the first two arguments being the current simulation time, and <code>x</code> , a value that is dependent on time (<code>x</code> represents a model parameter). List names must correspond to model parameters modified by the function. Alternatively, may be a list of such lists, in which case each set of functions is treated as a distinct scenario. See Details for more information, as well as the vignette <code>vignette("time_dependence", package = "epidemics")</code> .
<code>time_end</code>	A numeric, integer-like vector for the maximum number of
<code>replicates</code>	A single number for replicates to run. Defaults to 100. timesteps over which to run the model, in days. Defaults to 100 days.

Value

A `<data.table>`. If the model parameters and composable elements are all scalars, a single `<data.table>` with the columns "time", "compartment", "age_group", and "value", giving the number of individuals per demographic group in each compartment at each timestep in long (or "tidy") format is returned.

If the model parameters or composable elements are lists or list-like, a nested `<data.table>` is returned with a list column "data", which holds the compartmental values described above. Other columns hold parameters and composable elements relating to the model run. Columns "scenario" and "param_set" identify combinations of composable elements (population, interventions, vaccination regimes), and infection parameters, respectively.

Details: Discrete-time Ebola virus disease model

This model has compartments adopted from the consensus model for Ebola virus disease presented in Li et al. (2019), and with transitions between epidemiological compartments modelled using Erlang sub-compartments adapted from Getz and Dougherty (2018); see **References**.

The R code for this model is adapted from code by Ha Minh Lam and initially made available on *Epreicipes* (<https://github.com/epirecipes/epicookbook>) under the MIT license.

The shape of the Erlang distributions of passage times through the exposed and infectious compartments (k^E and k^I) are recommended to be set to 2 as a sensible choice, which is the default value for the `erlang_ssubcompartments` argument, but can be allowed to vary (but not independently).

The transition rates between the exposed and infectious, and infectious and funeral compartments (and also hospitalised to removed), γ^E and γ^I in Getz and Dougherty's notation, are passed by the user as the `infectiousness_rate` and `removal_rate` respectively.

Getz and Dougherty's equation (6) gives the relationship between these parameters and the mean pre-infectious ρ^E and infectious ρ^I periods.

$$\gamma^E = \frac{k^E}{\rho^E} = \frac{2}{\rho^E} \text{ and } \gamma^I = \frac{k^I}{\rho^I} = \frac{2}{\rho^I}$$

In this discrete time model, γ^E and γ^I are used to determine the passage times of newly exposed or infectious individuals through their respective compartments (thus allowing for variation in passage times).

Hospitalisation, funerals, and removal:

A proportion, `1 - prop_community`, of infectious individuals are transferred to the hospitalised compartment in each timestep. This compartment represents Ebola Treatment Units (ETUs), and individuals in the hospitalised compartment are considered to be infectious but no longer in the community.

The passage time of individuals in the hospitalised compartment is similar to that of individuals in the infectious compartment (i.e., infectious in the community), which means that an infectious individual with N timesteps before exiting the infectious compartment will exit the hospitalised compartment in the same time.

Hospitalised individuals can contribute to transmission of Ebola to susceptibles depending on the value of `etu_risk` which scales the baseline transmission rate β for hospitalised individuals.

We assume that deaths in hospital lead to Ebola-safe funerals, and individuals exiting the hospitalised compartment move to the 'removed' compartment, which holds both recoveries and deaths.

We assume that deaths outside of hospital lead to funerals that are potentially unsafe burials, and the `funeral_risk` argument scales the baseline transmission rate β for funeral transmission of Ebola to susceptibles.

Individuals are assumed to spend only a single timestep in the funeral transmission compartment, before they move into the 'removed' compartment.

Individuals in the 'removed' compartment do not affect model dynamics.

Model parameters:

The default values are:

- Transmission rate (β , `transmission_rate`): 0.125, resulting from an $R_0 = 1.5$ and an infectious period of 12 days.
- Infectiousness rate (γ^E , `infectiousness_rate`): 0.4, assuming a pre-infectious period of 5 days and two Erlang subcompartments.
- Removal rate (γ^I , `recovery_rate`): 0.1667, assuming an infectious period of 12 days and two Erlang subcompartments.

Implementing vaccination:

Vaccination cannot currently be implemented in this model as it does not have a "vaccinated" epidemiological compartment. This prevents the use of a `<vaccination>` object.

Instead, users can use the `time_dependence` argument to pass a function that modifies model parameters — specifically, the transmission rate — in a way that is consistent with the effect of vaccination. An example is shown in the vignette about this model; run this code to open the vignette: `vignette("ebola_model", package = "epidemics")`

Vector inputs:

Vector parameter inputs:

The model infection parameters and the model duration may be passed as numeric or integer-like vectors (as appropriate to the parameter), to simulate the effect of parameter uncertainty. All parameter vectors must be of the same length, or any one parameter vector may have a length > 1 while all other have a length of 1. In the first case, each i -th combination of parameters is treated as a parameter set. In the second case, all single value parameters (scalars) are recycled to the same length as the non-scalar parameter.

The model is run for N stochastic realisations of each parameter set. Random number seeds are preserved across parameter sets, so that differences in outcomes in each j -th run are due to differences in parameters alone.

Vector inputs for composable elements:

The `intervention` and `time_dependence` arguments also accept vectorised inputs in the form of lists of intervention and time dependence sets. Each combination of intervention and time-dependence sets is treated as a distinct 'scenario', and realisations of each parameter set are run for each scenario.

References

- Li, S.-L., Ferrari, M. J., Bjørnstad, O. N., Runge, M. C., Fonesbeck, C. J., Tildesley, M. J., Pannell, D., & Shea, K. (2019). Concurrent assessment of epidemiological and operational uncertainties for optimal outbreak control: Ebola as a case study. *Proceedings of the Royal Society B: Biological Sciences*, 286(1905), 20190774. doi:10.1098/rspb.2019.0774
- Getz, W. M., & Dougherty, E. R. (2018). Discrete stochastic analogs of Erlang epidemic models. *Journal of Biological Dynamics*, 12(1), 16–38. doi:10.1080/17513758.2017.1401677

Examples

```

# create a population with 6 compartments
population <- population(
  contact_matrix = matrix(1),
  demography_vector = 14e6,
  initial_conditions = matrix(
    c(0.999998, 0.000001, 0.000001, 0, 0, 0),
    nrow = 1, ncol = 6L
  )
)

# run epidemic simulation with no vaccination or intervention
data <- model Ebola(
  population = population
)

# view some data
head(data)

```

model_vacamole

Model leaky, two-dose vaccination in an epidemic using Vacamole

Description

Simulate an epidemic using the *Vacamole* model for Covid-19 developed at RIVM, the National Institute for Public Health and the Environment in the Netherlands. This model is aimed at estimating the impact of 'leaky' vaccination on an epidemic. See **Details** and **References** for more information.

Usage

```

model_vacamole(
  population,
  transmission_rate = 1.3/7,
  transmission_rate_vax = 0.8 * transmission_rate,
  infectiousness_rate = 1/2,
  hospitalisation_rate = 1/1000,
  hospitalisation_rate_vax = 0.8 * hospitalisation_rate,
  mortality_rate = 1/1000,
  mortality_rate_vax = 0.8 * mortality_rate,
  recovery_rate = 1/7,
  intervention = NULL,
  vaccination = NULL,
  time_dependence = NULL,
  time_end = 100,
  increment = 1
)

```

Arguments

population	An object of the population class, which holds a population contact matrix, a demography vector, and the initial conditions of each demographic group. See population() .
transmission_rate	A numeric for the rate at which individuals move from the susceptible to the exposed compartment upon contact with an infectious individual. Often denoted as β , with $\beta = R_0/\text{infectious period}$. See Details for default values.
transmission_rate_vax	A numeric of values between 0.0 and 1.0 giving the transmission_rate of the infection to individuals who have received two doses of the vaccine. The default value is 80% of the transmission_rate of the infection to individuals who are not doubly vaccinated.
infectiousness_rate	A numeric for the rate at which individuals move from the exposed to the infectious compartment. Often denoted as σ , with $\sigma = 1.0/\text{pre-infectious period}$. This value does not depend upon the number of infectious individuals in the population. See Details for default values.
hospitalisation_rate	A numeric for the hospitalisation rate of infectious individuals.
hospitalisation_rate_vax	A numeric of values between 0.0 and 1.0 giving the hospitalisation rate of infectious individuals who have received two doses of the vaccine. The default value is 80% of the hospitalisation rate of individuals who are not doubly vaccinated.
mortality_rate	A numeric for the mortality rate of infectious or hospitalised individuals.
mortality_rate_vax	A numeric of values between 0.0 and 1.0 giving the mortality of infectious and hospitalised individuals who have received two doses of the vaccine. The default value is 80% of the mortality rate of individuals who are not doubly vaccinated.
recovery_rate	A numeric for the rate at which individuals move from the infectious to the recovered compartment. Often denoted as γ , with $\gamma = 1.0/\text{infectious period}$. See Details for default values.
intervention	A named list of <intervention>s representing optional non-pharmaceutical or pharmaceutical interventions applied during the epidemic. Only a single intervention on social contacts of the class <contacts_intervention> is allowed as the named element "contacts". Multiple <rate_interventions> on the model parameters are allowed; see Details for the model parameters for which interventions are supported.
vaccination	An optional <vaccination> object representing a vaccination regime with two doses followed during the course of the epidemic, with a start and end time, and age-specific vaccination rates for each dose. See vaccination() .
time_dependence	A named list where each name is a model parameter, and each element is a function with the first two arguments being the current simulation time, and x, a value that is dependent on time (x represents a model parameter). See Details

	for more information, as well as the vignette on time-dependence <code>vignette("time_dependence", package = "epidemics")</code> .
time_end	The maximum number of timesteps over which to run the model. Taken as days, with a default value of 100 days. May be a numeric vector.
increment	The size of the time increment. Taken as days, with a default value of 1 day.

Value

A `<data.table>`. If the model parameters and composable elements are all scalars, a single `<data.table>` with the columns "time", "compartment", "age_group", and "value", giving the number of individuals per demographic group in each compartment at each timestep in long (or "tidy") format is returned.

If the model parameters or composable elements are lists or list-like, a nested `<data.table>` is returned with a list column "data", which holds the compartmental values described above. Other columns hold parameters and composable elements relating to the model run. Columns "scenario" and "param_set" identify combinations of composable elements (population, interventions, vaccination regimes), and infection parameters, respectively.

Details: Vacamole Covid-19 model with leaky, two-dose vaccination

The Vacamole model has the compartments "susceptible", "vaccinated_one_dose", "vaccinated_two_dose", "exposed", "infectious", "infectious_vaccinated", "hospitalised", "hospitalised_vaccinated", "recovered", and "dead".

This model allows for:

1. A 'hospitalised' compartment along with a hospitalisation rate;
2. Two doses of vaccination, with 'leaky' protection, i.e., vaccination does not prevent infection completely but allows for a reduction in the infection rate, as well as reduced rates of moving into states considered more serious, such as 'hospitalised' or 'dead'.

Model parameters:

This model only allows for single, population-wide rates of transitions between compartments per model run.

However, model parameters may be passed as numeric vectors. These vectors must follow Tidyverse recycling rules: all vectors must have the same length, or, vectors of length 1 will be recycled to the length of any other vector.

- Transmission rate (β , `transmission_rate`): 0.186, resulting from an $R_0 = 1.3$ and an infectious period of 7 days. The transmission rate for doubly vaccinated individuals (β_v) is 80% of β , 0.1488.
- Infectiousness rate (σ , `infectiousness_rate`): 0.5, assuming a pre-infectious period of 2 days.
- Hospitalisation rate (η , `hospitalisation_rate`): 1.0 / 1000, assuming that one in every thousand infectious individuals is hospitalised. The hospitalisation rate of doubly vaccinated individuals (η_v) is 80% of η , 0.8 / 1000.
- Mortality rate (ω , `mortality_rate`): 1.0 / 1000, assuming that one in every thousand infectious and hospitalised individuals dies. The mortality rate of the doubly vaccinated (ω_v) is 80% of ω , 0.8 / 1000.
- Recovery rate (γ , `recovery_rate`): 0.143, assuming an infectious period of 7 days.

References

Ainslie, K. E. C., Backer, J. A., Boer, P. T. de, Hoek, A. J. van, Klinkenberg, D., Altes, H. K., Leung, K. Y., Melker, H. de, Miura, F., & Wallinga, J. (2022). A scenario modelling analysis to anticipate the impact of COVID-19 vaccination in adolescents and children on disease outcomes in the Netherlands, summer 2021. *Eurosurveillance*, 27(44), 2101090. doi:10.2807/1560-7917.ES.2022.27.44.2101090

Examples

```
# create a population, note eleven columns for compartments
population <- population(
  contact_matrix = matrix(1),
  demography_vector = 67e6,
  initial_conditions = matrix(
    c(0.9999, 0, 0, 0, 0, 0.0001, 0, 0, 0, 0, 0),
    nrow = 1, ncol = 11L
  )
)

# create a vaccination regime
double_vax <- vaccination(
  nu = matrix(1e-3, ncol = 2, nrow = 1),
  time_begin = matrix(c(10, 30), nrow = 1),
  time_end = matrix(c(50, 80), nrow = 1)
)

# run epidemic simulation with vaccination but no intervention
# with a single set of parameters
data <- model_vacamole(
  population = population,
  vaccination = double_vax
)

# view some data
head(data)

# run epidemic simulation with no vaccination or intervention
# and three discrete values of transmission_rate
data <- model_vacamole(
  population = population,
  transmission_rate = c(1.3, 1.4, 1.5) / 7.0, # uncertainty in R0
)

# view some data
head(data)
tail(data)
```

new_infections	<i>Get new infections over model time</i>
----------------	---

Description

Get new infections over model time

Usage

```
new_infections(data, compartments_from_susceptible = NULL, by_group = TRUE)
```

Arguments

data	A table of model output, typically the output of <code>model_default()</code> or similar functions.
compartments_from_susceptible	An optional argument, for a character vector of the names of model compartments into which individuals transition from the "susceptible" compartment, and which are not related to infection. A common example is a compartment for "vaccinated" individuals who are no longer susceptible, but who should also not be counted as infected.
by_group	A logical representing whether the epidemic size should be returned by demographic group, or whether a single population-wide value is returned.

Value

A table with the same columns as data, but with the additional variable under compartment, "new_infections", resulting in additional rows.

Examples

```
# create a population
uk_population <- population(
  contact_matrix = matrix(1),
  demography_vector = 67e6,
  initial_conditions = matrix(
    c(0.9999, 0.0001, 0, 0, 0),
    nrow = 1, ncol = 5L
  )
)

# run epidemic simulation with no vaccination or intervention
data <- model_default(
  population = uk_population,
  time_end = 200,
  increment = 1
)
```



```
new_infections(data)
```

outcomes_averted	<i>Calculate outcomes averted by interventions</i>
------------------	--

Description

Calculate outcomes averted by interventions

Usage

```
outcomes_averted(baseline, scenarios, by_group = TRUE, summarise = TRUE)
```

Arguments

baseline	A nested <code><data.table></code> of a single model outcome with parameter uncertainty. This is expected to be the output of a single call to model functions such as <code>model_default()</code> or <code>model_ebola()</code> .
scenarios	A nested <code><data.table></code> of any number of model outcomes with infection parameter sets identical to those in baseline for comparability, i.e., the scenarios must differ from the baseline only in any interventions applied.
by_group	A single logical value that controls whether outcomes averted are calculated separately for each demographic group in the model outputs. This is passed to the <code>by_group</code> argument in <code>epidemic_size()</code> . Defaults to TRUE.
summarise	A single logical value that controls whether outcomes averted are summarised (returning the median and 95% uncertainty intervals), aggregating over parameter sets for each scenario, or whether the raw differences between the baseline and comparator scenarios are returned while matching by parameter set.

Details

Both deterministic and stochastic models (currently only the Ebola model) are supported.

When comparing deterministic model scenarios, users are expected to ensure that outputs comparable in terms of demographic groups and parameters. The output is expected to have parameter uncertainty, and differences between each scenario and the baseline are calculated after matching on parameter sets.

When comparing stochastic model scenarios, each scenario is matched against the baseline on the replicate number as well as the parameter set to reduce the effect of initial conditions on differences in outcomes.

Value

A `<data.table>`. When `summarise = TRUE`, a `<data.table>` of the same number of rows as scenarios, with the columns "scenario", "averted_median", "averted_lower", and "averted_upper".

When `summarise = FALSE`, a `<data.table>` with one row per "scenario", parameter set ("param_set"), and demography group ("demography_group), with the additional column "outcomes_averted" giving the difference between the baseline and the comparator scenario for each parameter set for each demography group.

Examples

```

polymod <- socialmixr::polymod
contact_data <- socialmixr::contact_matrix(
  polymod,
  countries = "United Kingdom",
  age.limits = c(0, 20, 40),
  symmetric = TRUE
)

# prepare contact matrix
contact_matrix <- t(contact_data$matrix)

# prepare the demography vector
demography_vector <- contact_data$demography$population
names(demography_vector) <- rownames(contact_matrix)

# initial conditions
initial_i <- 1e-6
initial_conditions <- c(
  S = 1 - initial_i, E = 0, I = initial_i, R = 0, V = 0
)

# build for all age groups
initial_conditions <- rbind(
  initial_conditions,
  initial_conditions,
  initial_conditions
)

# create population object
uk_population <- population(
  name = "UK",
  contact_matrix = contact_matrix,
  demography_vector = demography_vector,
  initial_conditions = initial_conditions
)

# create vector of parameters
beta <- withr::with_seed(
  1,
  rnorm(100, mean = 1.3 / 7, sd = 0.005)
)

```

```
baseline <- model_default(  
  population = uk_population,  
  transmission_rate = beta  
)  
  
max_time <- 100  
# prepare durations as starting at 25% of the way through an epidemic  
# and ending halfway through  
time_begin <- max_time / 4  
time_end <- max_time / 2  
  
# create three distinct contact interventions  
# prepare an intervention that models school closures for 180 days  
close_schools <- intervention(  
  name = "School closure",  
  type = "contacts",  
  time_begin = time_begin,  
  time_end = time_end,  
  reduction = matrix(c(0.3, 0.01, 0.01))  
)  
  
# prepare an intervention which mostly affects adults 20 -- 65  
close_workplaces <- intervention(  
  name = "Workplace closure",  
  type = "contacts",  
  time_begin = time_begin,  
  time_end = time_end,  
  reduction = matrix(c(0.01, 0.3, 0.01))  
)  
  
intervention_sets <- list(  
  list(  
    contacts = close_schools  
  ),  
  list(  
    contacts = close_workplaces  
  )  
)  
  
scenarios <- model_default(  
  population = uk_population,  
  transmission_rate = beta,  
  intervention = intervention_sets  
)  
  
# Defaults to summarise = TRUE  
outcomes_averted(  
  baseline = baseline,  
  scenarios = scenarios  
)  
  
# Set summarise = FALSE to get raw difference data
```

```

outcomes_averted(
  baseline = baseline,
  scenarios = scenarios,
  summarise = FALSE
)

```

population

Construct a new population for an epidemic model

Description

Construct a new population for an epidemic model

Check whether an object is a <population>

Usage

```

population(
  name = NA_character_,
  contact_matrix,
  demography_vector,
  initial_conditions
)

```

```
is_population(x)
```

Arguments

name	Optional string for the population name.
contact_matrix	A matrix giving the contacts between the demographic groups in the population. Must be a square matrix.
demography_vector	A vector of the sizes of each demographic group. Must have the same length as the dimensions of the contact matrix.
initial_conditions	Matrix representing the initial proportions of each demographic group in the four model compartments: 'susceptible', 'infected/infectious', 'recovered', and 'vaccinated'. Must have as many rows as the number of demographic groups. Each compartment is represented in the columns of the matrix, so that the element M_{ij} represents the proportion of individuals of demographic group i in compartment j .
x	An object to be checked as a valid population.

Value

An object of the <population> S3 class.

is_population() returns a logical for whether the object is a <population>.

Examples

```
uk_pop <- population(  
  name = "UK population",  
  contact_matrix = matrix(1),  
  demography_vector = 67e6,  
  initial_conditions = matrix(  
    c(0.9999, 0.0001, 0, 0),  
    nrow = 1, ncol = 4  
  )  
)  
  
# print to check  
uk_pop  
  
# check for class <population>  
is_population(uk_pop)
```

print.population *Print a <population> object*

Description

Print a <population> object

Usage

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

Arguments

x A <population> object.
... Other parameters passed to `print()`.

Value

Invisibly returns the <population> object x. Called for printing side-effects.

print.vaccination *Print a <vaccination> object*

Description

Print a <vaccination> object

Usage

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

Arguments

x A <vaccination> object.
... Other parameters passed to `print()`.

Value

Invisibly returns the <vaccination> object x. Called for printing side-effects.

print_intervention *Print an object of the <intervention> super-class*

Description

Print an object of the <intervention> super-class
Print objects of the <contact_intervention> class
Print objects of the <rate_intervention> class

Usage

```
## S3 method for class 'intervention'  
print(x, ...)  
  
## S3 method for class 'contact_intervention'  
print(x, ...)  
  
## S3 method for class 'rate_intervention'  
print(x, ...)
```

Arguments

<code>x</code>	An object that inherits from the <code><intervention></code> class. For <code>print.contacts_intervention()</code> , an object of the <code><contacts_intervention></code> class. For <code>print.rate_intervention()</code> , an object of the <code><rate_intervention></code> class.
<code>...</code>	Other parameters passed to <code>print()</code> .

Value

Invisibly returns the object `x`. Called for printing side-effects.

<code>vaccination</code>	<i>Construct a new vaccination regime for an epidemic model</i>
--------------------------	---

Description

Prepare a `<vaccination>` object that specifies a vaccination regime for use in an epidemic model. These objects can handle different vaccination start and end times, as well as different vaccination rates, for each demographic group in the epidemic modelling scenario.

Combine `<vaccination>` objects to create multi-dose vaccination regimes using `c()` on two or more `<vaccination>` objects.

Usage

```
vaccination(name = NA_character_, nu, time_begin, time_end)
```

```
is_vaccination(x)
```

```
## S3 method for class 'vaccination'
c(x, ...)
```

Arguments

<code>name</code>	String for the name of the vaccination regime.
<code>nu</code>	Matrix for the group-specific rate of vaccination, expressed as the rate parameter nu . Each element of the matrix nu_{ij} represents the rate of delivering vaccine dose j to demographic group i .
<code>time_begin</code>	Matrix for the start time of delivering vaccination dose j to demographic group i . demographic group i .
<code>time_end</code>	Matrix for the end time of delivering vaccination dose j to demographic group i .
<code>x</code>	A <code><vaccination></code> object, or an object to be checked as being a <code><vaccination></code> .
<code>...</code>	Vaccination objects to combine with <code>x</code> to create a multi-dose <code><vaccination></code> object.

Details

Multi-dose vaccinations can be passed to all epidemic models, but not all models accommodate multi-dose vaccinations. For example, the default SEIR-V model provided by `model_default()` has only a single vaccinated compartment, and will only use the first parameter set of a multi-dose regime to determine how individuals transition into the vaccinated compartment.

In contrast, the Vacamole model considers two doses, and will make use of the first two parameter sets of a multi-dose regime. More doses can be specified, but will be disregarded by this model.

Value

An object of the `<vaccination>` S3 class.

`vaccination()` returns a `<vaccination>` object with the specified parameters.

Concatenating two or more `<vaccination>` objects using `c()` also returns a `<vaccination>` object. This object holds the group-specific start and end times, and group-specific vaccination rates specified by all the constituent vaccination regimes.

`.no_vaccination()` returns a `<vaccination>` that has no effect on the population, with start and end times set to 0.0, and the rate of vaccination *nu* also set to 0.0.

`is_vaccination()` return a logical for whether the object is of the `<vaccination>` class.

Examples

```
# Assuming a population with two age groups, children 0 -- 5, and others 5+
# an example for childhood vaccination only
childhood_vaccination <- vaccination(
  name = "childhood_vaccination",
  time_begin = matrix(c(0, 100)), # assuming a simulation over 100 days
  time_end = matrix(c(100, 100)),
  nu = matrix(c(0.0001, 0.0)) # over 5s never vaccinated
)
childhood_vaccination

# check whether the object is a <vaccination>
is_vaccination(childhood_vaccination)

# Concatenating vaccinations
# create first dose regime
vax_1 <- vaccination(
  name = "vax_regime",
  time_begin = matrix(1),
  time_end = matrix(100),
  nu = matrix(0.001)
)

# second dose regime
vax_2 <- vaccination(
  name = "vax_regime",
  time_begin = matrix(101),
  time_end = matrix(200),
  nu = matrix(0.001)
```


)

c(vax_1, vax_2)

Index

.intervention_on_rates, 2

as.intervention, 3
as.vaccination, 3

c.contacts_intervention(intervention),
7
c.rate_intervention(intervention), 7
c.vaccination(vaccination), 31

epidemic_peak, 4
epidemic_size, 5
epidemic_size(), 25

intervention, 3, 7
is_contacts_intervention
(intervention), 7
is_intervention(intervention), 7
is_population(population), 28
is_rate_intervention(intervention), 7
is_vaccination(vaccination), 31

model_default, 10
model_default(), 24, 25, 32
model_default_cpp(), 8
model_diphtheria, 12
model_ebola, 16
model_ebola(), 25
model_vacamole, 20

new_infections, 24

outcomes_averted, 25

population, 28
population(), 10, 13, 16, 21
print(), 29–31
print.contact_intervention
(print_intervention), 30
print.intervention
(print_intervention), 30
print.population, 29
print.rate_intervention
(print_intervention), 30
print.vaccination, 30
print_intervention, 30
vaccination, 4, 31
vaccination(), 21