Package 'SIR'

February 1, 2026

Type Package

Title Simulation and Inference for Deterministic Epidemic Models

Version 0.2.0

Author Vicente Castellar

Maintainer Vicente Castellar <vic.castellar@gmail.com>

Description Simulation and exploration of deterministic compartmental epidemiological models formulated as systems of ordinary differential equations (ODEs). The package provides tools to define custom epidemic models, simulate latent state trajectories over time, and explore model dynamics under different parameter settings. Several classical models are included, such as SI, SIR, SIRS, SEIR, and SEIRS. An optional Shiny application is provided for interactive visualization and experimentation with epidemic scenarios.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports deSolve, stats, shiny, registry

Suggests ggplot2, devtools, roxygen2, knitr, rmarkdown

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL https://github.com/vcastellar/SIR

BugReports https://github.com/vcastellar/SIR/issues

Collate 'constructor.R'
'MODEL_SEIR.R'
'MODEL_SEIRS.R'
'MODEL_SI.R'
'MODEL_SIR.R'
'MODEL_SIRS.R'
'metodos.R'
'metrics.R'
'models.R'
'run_epi_app.R'
'simulate_epi.R'

Contents

`attack_rate` *Attack rate*

Description

Computes the cumulative number of infections over the course of the epidemic, defined as the integral (sum) of the incidence curve.

This metric requires the epidemiological role `"incidence"`.

Usage

```
attack_rate(sim)
```

Arguments

sim	An object of class "sim_epi".

Details

For models with permanent immunity (e.g. SIR, SEIR), this corresponds to the final epidemic size. For models with waning immunity (e.g. SIRS, SEIRS), the attack rate measures total infections and may exceed the population size.

Value

A numeric scalar giving the attack rate.

Examples

```
sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:160,
  parms = c(beta = 0.3, gamma = 0.1),
  init  = c(S = 999, I = 1, R = 0)
)

attack_rate(sim)
```

---

doubling_time_ts *Time-varying doubling time*

Description

Computes the time-varying epidemic doubling time based on the instantaneous growth rate.

As the epidemic approaches its peak, the doubling time increases and tends to infinity when growth vanishes.

Requires the epidemiological role "incidence".

Usage

```
doubling_time_ts(sim)
```

Arguments

sim	An object of class "sim_epi".

Value

A data.frame with columns:

- `time`: Time points.
- `doubling_time`: Time-varying doubling time.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:100,
  parms = c(beta = 0.3),
  init  = c(S = 999, I = 1)
)

head(doubling_time_ts(sim))
```

`epi_model` *Define an epidemic model*

Description

Defines a deterministic compartmental epidemic model governed by a system of ordinary differential equations (ODEs). The model explicitly declares its state variables, parameters, model outputs, and optional epidemiological roles.

Epidemiological roles assign semantic meaning (e.g. infectious, incidence) to specific state variables or auxiliary outputs, enabling generic epidemiological metrics and summaries to operate consistently across models.

Usage

```
epi_model(
  name,
  rhs,
  state_names,
  par_names,
  outputs = state_names,
  roles = NULL,
  lower = NULL,
  upper = NULL,
  defaults = NULL,
  init = NULL
)
```

Arguments

`name`	Character scalar. Human-readable name of the model (e.g. "SIR").
`rhs`	Function defining the ODE system. Must have signature `function(time, state, parms)` and return a list whose first element is the vector of state derivatives.
`state_names`	Character vector of state variable names.
`par_names`	Character vector of parameter names.
`outputs`	Character vector of named outputs returned by the RHS. Must include all state variables.
`roles`	Optional named list mapping epidemiological roles to variable names (states or outputs). Role names must belong to the standard vocabulary defined in `.epi_role_vocab()`.
`lower`	Optional named numeric vector of lower parameter bounds.
`upper`	Optional named numeric vector of upper parameter bounds.
`defaults`	Optional named numeric vector of default parameter values.
`init`	Optional named numeric vector of default initial conditions.

Details

Epidemiological roles:

Roles provide a semantic layer on top of model variables. For example, different models may use different variable names for infectious individuals, but assigning the `"infectious"` role allows epidemiological metrics to remain model-agnostic.

Supported roles are:

- **susceptible**
- **exposed**
- **infectious**
- **recovered**
- **deceased**
- **incidence**
- **deaths**

Not all roles must be defined for every model. Metrics depending on missing roles should fail explicitly.

Value

An object of class `"epi_model"`.

Examples

```
sir_model <- epi_model(
  name = "SIR",
  rhs  = sir_rhs,
  state_names = c("S", "I", "R"),
  par_names   = c("beta", "gamma"),
```

```
  outputs     = c("S", "I", "R", "incidence"),
  roles = list(
    susceptible = "S",
    infectious  = "I",
    recovered   = "R",
    incidence   = "incidence"
  )
)
```

`initial_doubling_time` *Initial doubling time*

Description

Computes the epidemic doubling time during the initial exponential growth phase.

This is defined as $\log(2)/r$, where r is the initial exponential growth rate.

Usage

```
initial_doubling_time(sim)
```

Arguments

`sim`	An object of class `"sim_epi"`.

Value

A numeric scalar giving the initial doubling time.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:50,
  parms = c(beta = 0.4),
  init  = c(S = 999, I = 1)
)

initial_doubling_time(sim)
```

initial_growth_rate *Initial exponential growth rate*

Description

Estimates the early exponential growth rate of the epidemic by fitting a log-linear model to the initial segment of the incidence curve.

This metric characterises the initial exponential phase of the epidemic.

Requires the epidemiological role `"incidence"`.

Usage

```
initial_growth_rate(sim, n = 7)
```

Arguments

`sim`	An object of class `"sim_epi"`.
`n`	Integer. Number of initial time points to use for estimation.

Value

A numeric scalar giving the initial exponential growth rate.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:50,
  parms = c(beta = 0.4),
  init  = c(S = 999, I = 1)
)

initial_growth_rate(sim, n = 10)
```

instantaneous_growth_rate *Instantaneous growth rate*

Description

Computes the time-varying exponential growth rate of the epidemic based on the incidence curve.

This metric captures departures from exponential growth due to susceptible depletion or other non-linear effects.

Requires the epidemiological role `"incidence"`.

Usage

```
instantaneous_growth_rate(sim)
```

Arguments

sim	An object of class "sim_epi".

Value

A data.frame with columns:

- time: Time points.
- r: Instantaneous growth rate.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:100,
  parms = c(beta = 0.3),
  init  = c(S = 999, I = 1)
)

head(instantaneous_growth_rate(sim))
```

---

peak_incidence *Peak incidence*

Description

Computes the maximum value of the incidence curve and the time at which it occurs.

This metric requires the epidemiological role "incidence".

Usage

```
peak_incidence(sim)
```

Arguments

sim	An object of class "sim_epi".

Value

A named list with elements:

- time: Time at which peak incidence occurs.
- value: Maximum incidence value.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:100,
  parms = c(beta = 0.3),
  init  = c(S = 999, I = 1)
)

peak_incidence(sim)
```

`peak_prevalence` *Peak prevalence*

Description

Computes the maximum number of infectious individuals during the epidemic.

This metric requires the epidemiological role `"infectious"`.

Usage

```
peak_prevalence(sim)
```

Arguments

`sim`	An object of class `"sim_epi"`.

Value

A numeric scalar giving the peak prevalence.

Examples

```
sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:160,
  parms = c(beta = 0.3, gamma = 0.1),
  init  = c(S = 999, I = 1, R = 0)
)

peak_prevalence(sim)
```

`plot.sim_epi` *Plot a simulated epidemic*

Description

Plot method for objects of class `"sim_epi"` as returned by `simulate_epi`. The function provides a flexible visualization interface for simulated epidemic trajectories by allowing the user to plot either all model states, the observed incidence time series, or a single model-defined output.

Unlike earlier versions, this method does not assume a specific compartmental structure (e.g. SIR or SEIR). All available quantities that can be plotted are taken directly from the outputs declared in the underlying `epi_model` used to generate the simulation.

Usage

```
## S3 method for class 'sim_epi'
plot(x, what = "states", scale = c("auto", "full", "small", "log"), ...)
```

Arguments

`x`	Object of class `"sim_epi"` as returned by `simulate_epi`.
`what`	Character string specifying what to plot. One of: `"states"` (all state variables), `"incidence"` (observed incidence counts), or the name of a single model output declared in `x$model$outputs` (e.g. `"I"`, `"S"`).
`scale`	Character string specifying the scale for state plots. One of `"auto"`, `"full"`, `"small"`, or `"log"`. This argument is only used when `what = "states"`.
`...`	Additional graphical parameters passed to base plotting functions such as `plot` and `matplot`.

Details

The behavior of the function depends on the value of the `what` argument:

- `what = "states"`Plots the time evolution of all state variables defined in `x$model$state_names`, using the simulated trajectories stored in `x$states`. The time variable is taken from the `time` column of `x$states` and is not considered a state.
- `what = "incidence"`Plots the observed incidence counts generated by the observation model and stored in `x$incidence`. If no observation model was specified during simulation (i.e. `obs = "none"`), this option results in an error.
- `what = <output>`Plots a single model-defined output, where `<output>` is the name of one of the quantities declared in `x$model$outputs` (e.g. `"I"`, `"S"`, `"R"`, or `"incidence"`). The corresponding trajectory is extracted from `x$states`.

If available, the time unit stored in the `sim_epi` object (i.e. `x$time_unit`) is used to label the time axis. This affects only the plot labels and does not change the numerical values of time.

Value

Invisibly returns the input object `x`.

See Also

`simulate_epi`, `summary.sim_epi`, `print.sim_epi`

Examples

```
sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:200,
  time_unit = "days",
  parms = c(beta = 0.3, gamma = 0.1),
  init  = c(S = 999990, I = 10, R = 0, C = 10),
  obs   = "poisson",
  seed  = 1
)

# Plot all model states
plot(sim)

# Plot observed incidence (requires an observation model)
plot(sim, what = "incidence")

# Plot a single state variable
plot(sim, what = "I")
```

`print.epi_model` *Print an epidemic model object*

Description

Provides a concise, human-readable summary of an `epi_model` object. The printed output includes the model name, state variables, parameters, the declared model outputs, and the underlying system of differential equations.

This method is automatically called when an object of class `"epi_model"` is printed at the console.

Usage

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

Arguments

`x`	An object of class `"epi_model"`.
`...`	Further arguments (ignored).

Details

The epi_model class explicitly declares the set of model outputs via the outputs field. These outputs may include state variables, derived quantities such as incidence, or any other observable returned by the model's right-hand side (rhs) function.

Parameter bounds are shown when available. The model equations are printed by deparsing the rhs function for inspection.

Value

Invisibly returns the input object x.

Examples

```
SIR_MODEL
```

---

print.sim_epi *Print a simulated epidemic*

Description

Print method for objects of class "sim_epi" as returned by simulate_epi. The function provides a concise, human-readable summary of a simulated epidemic, including the underlying epidemic model, the simulation time horizon, the parameter values used, and selected outcome metrics derived from the simulation results.

Unlike earlier versions, this method does not assume a specific compartmental structure (such as SIR or SEIR). Model-specific information is obtained directly from the epi_model object stored in x$model.

Usage

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

Arguments

x	Object of class "sim_epi" as returned by simulate_epi.
...	Additional arguments ignored by this method; included for compatibility with the generic print function.

Details

This method is automatically dispatched when an object of class `"sim_epi"` is printed. It is intended to give a quick overview of the simulation without displaying the full internal structure of the object.

If available, the time unit stored in the `sim_epi` object (i.e. `x$time_unit`) is used to label the simulation horizon and reported event times. This affects only the printed labels and does not change the numerical values of time.

The printed output typically includes:

- the name of the epidemic model,
- the simulated time horizon,
- the parameter values used in the simulation,
- basic outcome summaries (e.g. peak prevalence or total infections), when such quantities are available.

Value

Invisibly returns the input object `x`.

See Also

`simulate_epi`, `summary.sim_epi`, `plot.sim_epi`

Examples

```
sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:200,
  time_unit = "days",
  parms = c(beta = 0.30, gamma = 0.10),
  init  = c(S = 999990, I = 10, R = 0, C = 10),
  seed  = 1
)

sim
```

---

run_epi_app *Run the epidemic model simulator*

Description

Launches the interactive Shiny application for exploring and simulating registered epidemiological models.

Usage

```
run_epi_app()
```

Details

The app automatically detects all models available through `list_epi_models()`, including user-defined models registered with `register_epi_model()`.

`SEIRS_MODEL` *SEIRS epidemic model with latent period and waning immunity*

Description

An `epi_model` object representing a deterministic **SEIRS** (Susceptible–Exposed–Infectious–Recovered–Susceptible) compartmental epidemic model with a latent (exposed) period and waning immunity.

The model describes the spread of an infection in a closed population where susceptible individuals become infected at rate $\lambda(t)$ and enter the exposed compartment `E`. Exposed individuals progress to the infectious compartment at rate `sigma`. Infectious individuals recover at rate `gamma`, and recovered individuals lose immunity at rate `omega`, returning to the susceptible compartment.

Usage

```
SEIRS_MODEL
```

Format

An object of class `"epi_model"`.

Details

State variables:

The model is defined in terms of the following state variables:

S(t) Number of susceptible individuals at time t.

E(t) Number of exposed (infected but not yet infectious) individuals at time t.

I(t) Number of infectious (actively infected) individuals at time t.

R(t) Number of recovered individuals with temporary immunity at time t.

The total population size is conserved:

$$N = S(t) + E(t) + I(t) + R(t).$$

Model outputs:

The SEIRS model declares the following outputs:

`"S"` Susceptible population size.

`"E"` Exposed (latent) population size.

`"I"` Infectious population size.

`"R"` Recovered (temporarily immune) population size.

`"incidence"` Rate of progression from `E` to `I`, $\sigma E(t)$, representing the instantaneous incidence of new infectious cases returned by the model's right-hand side.

All declared outputs may be used as observables in generic utilities such as `fit_epi_model` via the `target` argument.

Parameters:

The SEIRS model depends on the following parameters:

beta Transmission rate (per day).

sigma Rate of progression from exposed to infectious (per day); $1/\sigma$ is the mean latent period.

gamma Recovery/removal rate from infectious to recovered (per day); $1/\gamma$ is the mean infectious period.

omega Rate of waning immunity from `R` back to `S` (per day); $1/\omega$ is the mean immunity duration.

Model equations:

New infections occur at rate

$$\lambda(t) = \beta \frac{S(t)\, I(t)}{N}.$$

Case incidence (entries into `I`) occurs at rate

$$\text{incidence}(t) = \sigma E(t).$$

The system of ordinary differential equations is:

$$\begin{aligned}
\frac{dS}{dt} &= -\lambda(t) + \omega R(t), \\
\frac{dE}{dt} &= \lambda(t) - \sigma E(t), \\
\frac{dI}{dt} &= \sigma E(t) - \gamma I(t), \\
\frac{dR}{dt} &= \gamma I(t) - \omega R(t).
\end{aligned}$$

Usage:

This predefined model object is intended to be used with generic utilities such as `simulate_epi`, `fit_epi_model`, and `predict.fit_epi_model` that operate on `epi_model` objects.

See Also

`simulate_epi`, `fit_epi_model`, `new_epi_model`

Examples

```
## Simulate a SEIRS epidemic
sim <- simulate_epi(
  model = SEIRS_MODEL,
  times = 0:300,
  parms = c(beta = 0.3, sigma = 0.2, gamma = 0.14, omega = 0.01),
  init  = c(S = 1e6, E = 0, I = 20, R = 0),
  obs   = "poisson"
)

plot(sim)
```

```
## Plot observed incidence
plot(sim, what = "incidence")

## Fit the model to observed incidence
fit_inc <- fit_epi_model(
  x = sim$incidence$inc,
  model = SEIRS_MODEL,
  init = SEIRS_MODEL$init,
  target = "incidence"
)

fit_inc
```

SEIR_MODEL *SEIR epidemic model with latent (exposed) period*

Description

An `epi_model` object representing a deterministic **SEIR** (Susceptible–Exposed–Infectious–Recovered) compartmental epidemic model with a latent (exposed) period.

The model describes the spread of an infection in a closed population where susceptible individuals become infected at rate $\lambda(t)$ and enter the exposed compartment `E`. Exposed individuals progress to the infectious compartment at rate `sigma` and subsequently recover with permanent immunity.

Usage

```
SEIR_MODEL
```

Format

An object of class `"epi_model"`.

Details

State variables:

The model is defined in terms of the following state variables:

S(t) Number of susceptible individuals at time t.

E(t) Number of exposed (infected but not yet infectious) individuals at time t.

I(t) Number of infectious (actively infected) individuals at time t.

R(t) Number of recovered (immune) individuals at time t.

The total population size is conserved:

$$N = S(t) + E(t) + I(t) + R(t).$$

Model outputs:

The SEIR model declares the following outputs:

`"S"` Susceptible population size.

`"E"` Exposed (latent) population size.

`"I"` Infectious population size.

`"R"` Recovered (immune) population size.

`"incidence"` Rate of progression from `E` to `I`, $\sigma E(t)$, representing the instantaneous incidence of new infectious cases returned by the model's right-hand side.

All declared outputs may be used as observables in generic utilities such as `fit_epi_model` via the `target` argument.

Parameters:

The SEIR model depends on the following parameters:

beta Transmission rate (per day).

sigma Rate of progression from exposed to infectious (per day); $1/\sigma$ is the mean latent period.

gamma Recovery/removal rate from infectious to recovered (per day); $1/\gamma$ is the mean infectious period.

Model equations:

New infections occur at rate

$$\lambda(t) = \beta \frac{S(t)\, I(t)}{N}.$$

Progression from exposed to infectious occurs at rate

$$\text{incidence}(t) = \sigma E(t).$$

The system of ordinary differential equations is:

$$\begin{aligned}
\frac{dS}{dt} &= -\lambda(t), \\
\frac{dE}{dt} &= \lambda(t) - \sigma E(t), \\
\frac{dI}{dt} &= \sigma E(t) - \gamma I(t), \\
\frac{dR}{dt} &= \gamma I(t).
\end{aligned}$$

Usage:

This predefined model object is intended to be used with generic utilities such as `simulate_epi`, `fit_epi_model`, and `predict.fit_epi_model` that operate on `epi_model` objects.

See Also

`simulate_epi`, `fit_epi_model`, `new_epi_model`

Examples

```
## Simulate a SEIR epidemic
sim <- simulate_epi(
  model = SEIR_MODEL,
  times = 0:200,
  parms = c(beta = 0.3, sigma = 0.2, gamma = 0.14),
  init  = c(S = 1e6, E = 5, I = 10, R = 0),
  obs   = "poisson"
)

plot(sim)

## Plot observed incidence
plot(sim, what = "incidence")

## Fit the model to observed incidence
fit_inc <- fit_epi_model(
  x = sim$incidence$inc,
  model = SEIR_MODEL,
  init = SEIR_MODEL$init,
  target = "incidence"
)

fit_inc
```

---

simulate_epi *Simulate an epidemic model defined by an* epi_model *object*

Description

Simulates a deterministic compartmental epidemic model specified via an epi_model object. The model is solved as a system of ordinary differential equations (ODEs) using ode. Optionally, a stochastic observation process can be applied to a model-defined incidence process to generate reported incidence counts.

The function is fully model-agnostic: all model-specific information (state variables, parameters, default values, ODE right-hand side, and optional incidence definition) is read directly from the supplied epi_model.

Usage

```
simulate_epi(
  model,
  times,
  time_unit = "days",
  parms = NULL,
  init = NULL,
  obs = c("none", "poisson", "negbin"),
```

```
  size = 20,
  seed = NULL,
  ...
)
```

Arguments

model	An object of class "epi_model" defining the epidemic model to simulate.
times	Numeric vector of time points at which to solve the ODE system. Must be strictly increasing and of length >= 2. The first value typically corresponds to the initial time (e.g. 0).
time_unit	Character string specifying the unit of time associated with times. One of "days", "weeks", "month", or "year". This is used only for printing and plotting labels and does not affect the numerical simulation. The optional time_unit argument can be used to label the time axis in printed summaries and plots, but does not affect the numerical solution.
parms	Named numeric vector of model parameters. Names must match model$par_names. Any missing parameters are taken from model$defaults, if available.
init	Named numeric vector giving the initial values of the state variables. Names must exactly match model$state_names.
obs	Character string specifying the observation model. One of "none", "poisson", or "negbin".
size	Numeric. Dispersion (size) parameter for the negative binomial observation model.
seed	Optional integer. If provided, sets the random seed for reproducible observation draws.
...	Additional arguments passed directly to ode, such as method, rtol, or atol.

Details

Model structure:

The model argument must be an object of class "epi_model", typically created with new_epi_model. At minimum, the model must define:

- an ODE right-hand side function (model$rhs);
- the state variables (model$state_names);
- the model parameters (model$par_names).

Optionally, the model may define a latent incidence process by returning a variable named "incidence" as an additional output of the ODE right-hand side function. If no such variable is defined, no incidence time series is produced.

Time grid:

The model is simulated at the time points specified by the numeric vector times. This vector must be strictly increasing and of length >= 2. The first value typically corresponds to the initial time (e.g. 0).

This design follows the interface of ode and provides full control over the temporal resolution of the simulation, including irregular or non-integer time grids.

Initial conditions:

The initial state `init` must be supplied explicitly as a named numeric vector whose names match `model$state_names`. No automatic construction of initial conditions is performed.

Observation model:

If the model defines a latent incidence process, it can optionally be converted into reported incidence counts using a simple observation model:

`obs = "poisson"` Reported counts are drawn from a Poisson distribution with mean equal to the latent incidence.

`obs = "negbin"` Reported counts are drawn from a negative binomial distribution with mean equal to the latent incidence and dispersion parameter `size`.

`obs = "none"` No stochastic observation model is applied; the observed incidence is taken to be equal to the latent incidence.

If the model does not define a latent incidence process, the arguments `obs` and `size` are ignored and no incidence outputs are produced.

Numerical integration:

The ODE system is solved using `ode`. Additional arguments passed via `...` are forwarded directly to `deSolve::ode()`, allowing full control over the numerical integration. In particular, the integration method can be specified using the `method` argument.

If `method` is not supplied, `deSolve::ode()` uses its default integration method (typically `"lsoda"`).

Value

An object of class `"sim_epi"`, containing:

model The `epi_model` object used to generate the simulation.

params A list of model parameter values used in the simulation.

states A data frame with columns `time` and the model state variables, containing the simulated state trajectories.

incidence A data frame with columns `time` and `inc` containing the observed (reported) incidence counts, or `NULL` if no incidence process is defined by the model.

incidence_cum A data frame with columns `time` and `cases_cum` containing cumulative observed cases, or `NULL`.

See Also

`new_epi_model`, `ode`, `plot.sim_epi`, `print.sim_epi`

Examples

```
## ------------------------------------------------------------------
## Example 1: SIR model with model-defined incidence
## ------------------------------------------------------------------

sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:200,
```

```
  time_unit = "week",
  parms = c(beta = 0.30, gamma = 0.10),
  init  = SIR_MODEL$init,
  seed  = 1,
  method = 'lsoda'
)

# Plot state trajectories defined by the model
plot(sim)

# Plot observed incidence (requires the model to define incidence)
plot(sim, what = "incidence")
plot(sim, what = "CR")


## ------------------------------------------------------------------
## Example 2: Using a different numerical integration method
## ------------------------------------------------------------------

sim_rk4 <- simulate_epi(
  model = SIR_MODEL,
  times = 0:200,
  parms = c(beta = 0.30, gamma = 0.10),
  init  = c(S = 1e6 - 10, I = 10, R = 0, C = 10),
  method = "rk4"
)

plot(sim_rk4)


## ------------------------------------------------------------------
## Example 3: Model without an incidence definition
## ------------------------------------------------------------------

sim <- simulate_epi(
  model = SI_MODEL,
  times = 30:100,
  parms = c(beta = 0.25),
  init  = c(S = 999, I = 1)
)

plot(sim)
plot(sim, what = "incidence")


## ------------------------------------------------------------------
## Example 4: Stochastic observation model
## ------------------------------------------------------------------

sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:150,
  parms = c(beta = 0.35, gamma = 0.12),
```

```
  init  = c(S = 1e5 - 5, I = 5, R = 0, C = 5),
  obs   = "negbin",
  size  = 20,
  seed  = 123
)

plot(sim, what = "incidence")
```

`SIRS_MODEL` *SIRS epidemic model with waning immunity*

Description

An `epi_model` object representing a deterministic **SIRS** (Susceptible–Infectious–Recovered–Susceptible) compartmental epidemic model with waning immunity.

The model describes the spread of an infection in a closed population where susceptible individuals become infectious, subsequently recover, and may lose immunity over time, returning to the susceptible compartment at rate `omega`.

Usage

```
SIRS_MODEL
```

Format

An object of class `"epi_model"`.

Details

State variables:

The model is defined in terms of the following state variables:

S(t) Number of susceptible individuals at time t.

I(t) Number of infectious (actively infected) individuals at time t.

R(t) Number of recovered individuals with temporary immunity at time t.

The total population size is conserved:

$$N = S(t) + I(t) + R(t).$$

Model outputs:

The SIRS model declares the following outputs:

`"S"` Susceptible population size.

`"I"` Infectious population size.

`"R"` Recovered (temporarily immune) population size.

`"incidence"` Instantaneous rate of new infections $\lambda(t)$ returned by the model's right-hand side.

All declared outputs may be used as observables in generic utilities such as `fit_epi_model` via the `target` argument.

Parameters:

The SIRS model depends on the following parameters:

beta Transmission rate (per day).

gamma Recovery/removal rate (per day).

omega Rate of waning immunity from `R` back to `S` (per day).

Model equations:

New infections occur at rate

$$\lambda(t) = \beta \frac{S(t)\, I(t)}{N}.$$

The system of ordinary differential equations is:

$$\begin{aligned}
\frac{dS}{dt} &= -\lambda(t) + \omega R(t), \\
\frac{dI}{dt} &= \lambda(t) - \gamma I(t), \\
\frac{dR}{dt} &= \gamma I(t) - \omega R(t).
\end{aligned}$$

Usage:

This predefined model object is intended to be used with generic utilities such as `simulate_epi`, `fit_epi_model`, and `predict.fit_epi_model` that operate on `epi_model` objects.

See Also

`simulate_epi`, `fit_epi_model`, `new_epi_model`

Examples

```
## Simulate a SIRS epidemic
sim <- simulate_epi(
  model = SIRS_MODEL,
  times = 0:200,
  parms = c(beta = 0.3, gamma = 0.1, omega = 0.02),
  init  = c(S = 1e6, I = 20, R = 0)
)

plot(sim)

## Plot observed incidence (if an observation model is used)
plot(sim, what = "incidence")

## Fit the model to observed incidence
fit_inc <- fit_epi_model(
  x = sim$incidence$inc,
  model = SIRS_MODEL,
  init = SIRS_MODEL$init,
```

```
  target = "incidence"
)

fit_inc
```

SIR_MODEL *SIR epidemic model*

Description

An `epi_model` object representing a deterministic **SIR** (Susceptible–Infectious–Recovered) compartmental epidemic model.

The model describes the spread of an infection in a closed population where susceptible individuals become infectious and subsequently recover with permanent immunity.

Usage

```
SIR_MODEL
```

Format

An object of class `"epi_model"`.

Details

State variables:

The model is defined in terms of the following state variables:

S(t) Number of susceptible individuals at time t.

I(t) Number of infectious (actively infected) individuals at time t.

R(t) Number of removed/recovered individuals at time t.

The total population size is conserved:

$$N = S(t) + I(t) + R(t).$$

Model outputs:

The SIR model declares the following outputs:

`"S"` Susceptible population size.

`"I"` Infectious population size.

`"R"` Recovered (removed) population size.

`"incidence"` Instantaneous rate of new infections $\lambda(t)$ returned by the model's right-hand side.

All declared outputs may be used as observables in generic utilities such as `fit_epi_model` via the `target` argument.

Parameters:

The SIR model depends on the following parameters:

beta Transmission rate (per day).

gamma Recovery/removal rate (per day).

Model equations:

New infections occur at rate

$$\lambda(t) = \beta \frac{S(t)\, I(t)}{N}.$$

The system of ordinary differential equations is:

$$\begin{aligned}
\frac{dS}{dt} &= -\lambda(t), \\
\frac{dI}{dt} &= \lambda(t) - \gamma I(t), \\
\frac{dR}{dt} &= \gamma I(t).
\end{aligned}$$

Usage:

This predefined model object is intended to be used with generic utilities such as `simulate_epi`, `fit_epi_model`, and `predict.fit_epi_model` that operate on `epi_model` objects.

See Also

`simulate_epi`, `fit_epi_model`, `new_epi_model`

Examples

```
## Simulate a SIR epidemic
sim <- simulate_epi(
  model = SIR_MODEL,
  times = 0:200,
  parms = c(beta = 0.3, gamma = 0.1),
  init  = c(S = 1e6, I = 10, R = 0)
)

plot(sim)

## Plot observed incidence (if an observation model is used)
plot(sim, what = "incidence")

## Fit the model to observed incidence
fit_inc <- fit_epi_model(
  x = sim$incidence$inc,
  model = SIR_MODEL,
  init = SIR_MODEL$init,
  target = "incidence"
)

fit_inc
```

SI_MODEL *SI epidemic model*

Description

An `epi_model` object representing a deterministic **SI** (Susceptible–Infectious) compartmental epidemic model.

The model describes the spread of an infection in a closed population where individuals move irreversibly from the susceptible compartment `S` to the infectious compartment `I`. No recovery or removal process is included.

Usage

```
SI_MODEL
```

Format

An object of class `"epi_model"`.

Details

State variables:

The model is defined in terms of the following state variables:

S(t) Number of susceptible individuals at time t.

I(t) Number of infectious individuals at time t.

The total population size is conserved:

$$N = S(t) + I(t).$$

Model outputs:

The SI model declares the following outputs:

`"S"` Susceptible population size.

`"I"` Infectious population size.

`"incidence"` Instantaneous rate of new infections $\lambda(t)$ returned by the model's right-hand side.

All declared outputs may be used as observables in generic utilities such as `fit_epi_model` via the `target` argument.

Parameters:

The SI model depends on a single parameter:

beta Transmission rate (per day).

Model equations:

New infections occur at rate

$$\lambda(t) = \beta \frac{S(t)\, I(t)}{N}.$$

The system of ordinary differential equations is:

$$\frac{dS}{dt} = -\lambda(t),$$
$$\frac{dI}{dt} = \lambda(t).$$

Usage:

This predefined model object is intended to be used with generic utilities such as `simulate_epi`, `fit_epi_model`, and `predict.fit_epi_model` that operate on `epi_model` objects.

See Also

`simulate_epi`, `fit_epi_model`, `new_epi_model`

Examples

```
## Simulate an SI epidemic
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:100,
  parms = c(beta = 0.4),
  obs = "negbin",
  init = SI_MODEL$init
)

plot(sim)

## Plot observed incidence
plot(sim, what = "incidence")

## Fit the model to observed incidence
fit_inc <- fit_epi_model(
  x = sim$incidence$inc,
  model = SI_MODEL,
  init = SI_MODEL$init,
  target = "incidence"
)

fit_inc
```

`summary.sim_epi` *Summarize a simulated epidemic*

Description

Summary method for objects of class `"sim_epi"` as returned by `simulate_epi`. The function computes and returns a small set of epidemiologically meaningful summary quantities derived from the simulation.

Usage

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

Arguments

object	Object of class "sim_epi" as returned by simulate_epi.
...	Currently unused; included for compatibility with generic summary.

Details

The summary includes:

model The epidemic model simulated (e.g. "sir" or "sirs").

R0 The basic reproduction number, computed as $R_0 = \beta/\gamma$.

peak_I The maximum number of infectious individuals observed during the simulation.

total_infections The total number of infection events, computed as the maximum value of the cumulative infection variable C(t).

This method is automatically dispatched when calling summary() on an object of class "sim_epi".

Value

A named list with summary statistics describing the simulated epidemic.

See Also

simulate_epi, plot.sim_epi

Examples

```
sim <- simulate_epi(n_days = 300, model = SIRS_MODEL, omega = 1/180, seed = 1)

summary(sim)
```

---

time_to_peak *Time to peak incidence*

Description

Returns the time at which incidence reaches its maximum.

This is a convenience wrapper around peak_incidence().

Usage

```
time_to_peak(sim)
```

Arguments

sim	An object of class "sim_epi".

Value

A numeric scalar giving the time to peak incidence.

Examples

```
sim <- simulate_epi(
  model = SI_MODEL,
  times = 0:100,
  parms = c(beta = 0.3),
  init  = c(S = 999, I = 1)
)

time_to_peak(sim)
```

Index