

Package: daedalus (via r-universe)

October 8, 2026

Title Model Health, Social, and Economic Costs of a Pandemic

Version 0.3.9

Description Model the health, education, and economic costs of directly transmitted respiratory virus pandemics, under different scenarios of prior vaccine investment, policy interventions, and public behavioural change, using the 'DAEDALUS' integrated health-economics model adapted from Haw et al. (2022) <[doi:10.1038/s43588-022-00233-0](https://doi.org/10.1038/s43588-022-00233-0)>.

License MIT + file LICENSE

URL <https://github.com/jameel-institute/daedalus>,
<https://jameel-institute.github.io/daedalus/>

BugReports <https://github.com/jameel-institute/daedalus/issues>

Depends R (>= 3.5)

Imports abind, checkmate, cli, daedalus.data (>= 0.0.3), data.table, dust2, Rcpp, rlang

Suggests bookdown, countrycode, dplyr, ggdist, ggplot2, knitr, latex2exp, rmarkdown, scales, socialmixr, spelling, testthat (>= 3.0.0), tidyr

LinkingTo BH, cpp11, dust2, monty, Rcpp, RcppEigen

VignetteBuilder knitr

Remotes jameel-institute/daedalus.data, mrc-ide/dust2, mrc-ide/monty

Config/Needs/website jameel-institute/jameelinst.rpkg.theme

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Repository <https://jameel-institute.r-universe.dev>

Date/Publication 2026-09-08 13:30:59 UTC
RemoteUrl <https://github.com/jameel-institute/daedalus>
RemoteRef HEAD
RemoteSha 152b06bf6405aa35de86895b1d2109b7e2cd003b

Contents

class_behaviour	2
class_country	4
class_daedalus_output	6
class_infection	6
class_npi	8
class_vaccination	11
daedalus	13
daedalus_costs	16
daedalus_multi_infection	20
econ_constants	21
epi_constants	22
epi_output_helpers	23
get_data	24
get_fiscal_costs	26
get_life_years_lost	28
model_constants	29
set_data	31
summary_constants	32
Index	33

class_behaviour	<i>Represent behaviour-change mechanisms in DAEDALUS</i>
-----------------	--

Description

Helper functions to prepare <daedalus_behaviour> class responses corresponding to either old or new behavioural change mechanisms. Inherits from <daedalus_response> to allow for more control in future over when these mechanisms are active.

Usage

```
daedalus_old_behaviour(rate = 0.001, lower_limit = 0.2)

daedalus_new_behaviour(
  hospital_capacity,
  behav_effectiveness = 0.5,
  baseline_optimism = 0.5,
  responsiveness = 1.5,
```

```

    k0 = 4.59,
    k1 = -9.19
)

is_daedalus_behaviour(x)

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

Arguments

rate	The marginal rate of decrease in the scaling factor for each additional daily death in the 'old' behavioural model.
lower_limit	The lower limit of the scaling factor in the 'old' behavioural model; prevents scaling factor from reaching zero.
hospital_capacity	The emergency hospital capacity of a country, but may also be a <daedalus_country> object from which the hospital capacity is extracted. See $\bar{H}$ in Details.
behav_effectiveness	A single double value for the effectiveness of adopting behavioural measures against the risk of infection. Expected to be in the range $[0, 1]$, where 0 represents no effectiveness, and 1 represents full effectiveness. See δ in Details. Defaults to 0.5.
baseline_optimism	A single double value for the baseline optimism about pandemic outcomes. Expected to be in the range $[0, 1]$. See $\bar{B}$ in Details. Defaults to 0.5.
responsiveness	A single double value for the population responsiveness to an epidemic signal. Must have a lower value of 0, but the upper bound is open. See k_2 in Details. Defaults to 1.5.
k0	A single, optional double value which is a scaling parameter for the sigmoidal relationship between p_t and behav_effectiveness. See k_0 in Details. Defaults to 4.59.
k1	A single, optional double value which is another scaling parameter for the sigmoidal relationship between p_t and behav_effectiveness. See k_1 in Details. Defaults to -9.19. Expected to be a negative number.
x	A <daedalus_behaviour> object.
...	Other arguments passed to <code>format()</code> .

Details

Daedalus currently supports two behavioural models (and the option to have neither model active). These models simulate population-level changes towards adopting behaviours that help reduce the risk of infection, which might be expected during a major disease outbreak.

In both models, the transmission rate of the infection β is reduced for all transmissions by a scaling factor; the models differ in how the scaling factor is calculated.

Note that a major issue with including this in a model run (any value other than "off") is that it leads to substantially lower response costs, and generally better health outcomes (lives lost), **without accounting for any attendant economic or social costs**. As such, please treat the behavioural models as experimental.

Deaths-based public concern model:

This model existed prior to daedalus v0.2.25 and is referred to as the "old" behavioural model. The scaling factor for β in this model is

$$(1 - \text{rate})^{\Delta D} \times (1 - L) + L$$

where rate is the marginal rate of reduction for each additional death per day (ΔD), and L is the lower limit of the scaling factor to prevent the FOI going to zero.

Optimism-responsiveness-effectiveness model:

This model is referred to as the "new" behavioural model, and the scaling factor for β is given by

$$\mathcal{B}(p_t, \delta) = p_t(1 - \delta)[p_t(1 - \delta) + (1 - p_t)] + (1 - p_t)[p_t(1 - \delta) + (1 - p_t)]$$

where p_t is the proportion of the population taking protective behaviour and δ is the effectiveness of the behaviour, both in the range $[0, 1]$. When either is zero, β is not scaled down.

p_t is computed during the model run, and is given by

$$p_t = \frac{1}{1 + \exp \left\{ - \left(k_0 + k_1 \bar{B} + k_2 \frac{H_t}{\bar{H}} \right) \right\}}$$

where $k_0 = 4.59$, $k_1 = -9.19$, k_2 are scaling parameters. k_0, k_1 have constant values that cannot be changed by the user, and these have been chosen to produce a sigmoidal relationship between $\bar{B}$ and p_t .

k_2 a constant user-supplied parameter that should be interpreted as population responsiveness to a signal of epidemic severity $H_t/\bar{H}$, which is the relative burden on hospital capacity (H_t : hospital demand at time t , and $\bar{H}$: emergency hospital capacity).

$\bar{B}$ is a constant user-supplied parameter that captures the baseline population optimism about the outbreak.

Value

An object of class <daedalus_behaviour> which inherits from <daedalus_response>. This is primarily a list holding behavioural parameters.

class_country

Represent countries and territories for DAEDALUS

Description

Helper functions to create and work with S3 class <daedalus_country> objects for use with [daedalus\(\)](#). These objects store country parameters for reuse and have methods for easy parameter access and editing, as well as processing raw country characteristics for the DAEDALUS model.

Usage

```
daedalus_country(country, contact_matrix = NULL, group_working_age = NULL)

is_daedalus_country(x)

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

Arguments

country	A string giving the country or territory name, or ISO2 or ISO3 code; must be from among <code>daedalus.data::country_codes_iso2c</code> or <code>daedalus.data::country_codes_iso3c</code> or <code>daedalus.data::country_names</code> .
contact_matrix	An optional named list of numeric matrices, representing age-specific contact matrices in different contexts or settings. Defaults to NULL, which selects the country-specific contact matrix provided in <code>daedalus.data::country_data</code> .
group_working_age	An optional value for the age-group that is considered to be the working-age group. Defaults to 3, which is taken from an internal constant.
x	An object of the <code><daedalus_country></code> class.
...	Other parameters passed to <code>print()</code> .

Value

- `daedalus_country()` returns an object of the S3 class `<daedalus_country>`
- `is_daedalus_country()` returns a logical for whether an object is a `<daedalus_country>`.
- `print.daedalus_country()` invisibly returns the `<daedalus_country>` object `x`. Called for printing side-effects.

Examples

```
x <- daedalus_country("Canada")

x

# check whether `x` is a <country> object
is_daedalus_country(x)

# using assignment operators; must be assigned as a list
x$contact_matrix <- list(home = matrix(99, 4, 4), school = matrix(1, 4, 4))
x
```

class_daedalus_output *Create and work with <daedalus_output> objects*

Description

Create and work with <daedalus_output> objects

Print <daedalus_output> class objects

Usage

```
as_daedalus_output(x)
```

```
validate_daedalus_output(x)
```

```
is_daedalus_output(x)
```

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

Arguments

x An object of the <daedalus_output> class.
 ... Not used; added for compatibility with the generic.

Value

An object of the <daedalus_output> class. Holds the raw ODE solution and ODE events logs in **dust2** forma), as well as country, infection, vaccination, and behaviour classes representing input arguments to [daedalus\(\)](#).

Invisibly returns x; called primarily for its side effects of erroring when the object does not satisfy the <daedalus_output> class requirements.

A logical for whether x is of the <daedalus_output> class.

None; called for its printing side effects.

class_infection *Represent infection parameters for DAEDALUS*

Description

Helper functions to create and work with S3 class <daedalus_infection> objects for use with [daedalus\(\)](#). These objects store infection parameters for reuse and have methods for easy parameter access and editing, as well as processing raw infection characteristics for the DAEDALUS model.

Usage

```
daedalus_infection(name, ...)

is_daedalus_infection(x)

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

Arguments

name	An epidemic name from among daedalus.data::epidemic_names . Selecting an epidemic automatically pulls in infection parameters associated with the epidemic; these are stored as packaged data in <code>daedalus.data::infection_data</code> . Default infection parameters for epidemics can be over-ridden by passing them as a named list to ...
...	Other parameters passed to print() .
x	An object of the <code><daedalus_infection></code> class.

Details**Included epidemics:**

Epidemics for which data are available are given below (pathogen in parentheses). The string indicates the name that must be passed to the name argument.

- "sars_cov_1": SARS 2004 (SARS-CoV-1),
- "influenza_2009": influenza 2009 (influenza A H1N1),
- "influenza_1957": influenza 1957 (influenza A H2N2),
- "influenza_1918": influenza 1918 (influenza A H1N1),
- "sars_cov_2_pre_alpha": Covid-19 wild type (SARS-Cov-2 wild type),
- "sars_cov_2_omicron": Covid-19 Omicron (SARS-CoV-2 omicron),
- "sars_cov_2_pre_delta": (SARS-CoV-2 delta).

Infection parameters:

All infections have the following parameters, which take default values stored in the package under [daedalus.data::infection_data](#). Users can pass custom values for these parameters as arguments via ...

- r_0 : A single numeric value for the basic reproduction value of the infection R_0 .
- sigma: A single numeric value > 0.0 for the rate of transition from the exposed compartment to one of two infectious compartments.
- p_sigma: A single numeric value in the range $[0.0, 1.0]$ for the proportion of infectious individuals who are also symptomatic. Asymptomatic individuals can have a different contribution to the force of infection from symptomatic individuals.
- epsilon: A single numeric value for the relative contribution of asymptomatic infectious individuals to the force of infection (compared to symptomatic individuals).
- gamma_Is: A single numeric value for the recovery rate of infectious individuals who are not hospitalised.

- `gamma_Ia`: A single numeric value for the recovery rate from asymptomatic infection.
- `ifr`: A numeric vector of length `N_AGE_GROUPS` (4) for the age-specific infection fatality risk.
- `eta`: A numeric vector of length `N_AGE_GROUPS` (4) for the age-specific hospitalisation rate for individuals who are infectious and symptomatic.
- `hfr`: A numeric vector of length `N_AGE_GROUPS` (4) for the age-specific probability of death conditional on hospitalisation.
- `gamma_H_recovery`: A single numeric value for the recovery rate of hospitalised individuals.
- `gamma_H_death`: A single numeric value for the death rate of hospitalised individuals.
- `rho`: A single numeric value for the rate at which infection-derived immunity wanes, returning individuals in the 'recovered' compartment to the 'susceptible' compartment.

Value

- `daedalus_infection()` returns an object of the S3 class `<daedalus_infection>`.
- `is_daedalus_infection()` returns a logical for whether an object is a `<daedalus_infection>`.
- `print.daedalus_infection()` invisibly returns the `<daedalus_infection>` object `x`. Called for printing side-effects.

Examples

```
# make a <daedalus_infection> object with default parameter values
daedalus_infection("sars_cov_1")

# modify infection parameters R0 and immunity waning rate
daedalus_infection("influenza_1918", r0 = 2.5, rho = 0.01)
```

class_npi

Represent non-pharmaceutical intervention strategies for DAEDALUS

Description

Helper functions to create and work with S3 class `<daedalus_npi>` objects for use with `daedalus()`. These objects store NPI parameters for reuse and have methods for easy parameter access and editing, as well as processing raw NPI characteristics for the DAEDALUS model.

Most NPIs must be initialised with an associated country and infection; these are used to determine flag positions as well as the state-values (hospital capacity) and incidence-prevalence ratio (IPR) at which the NPI starts and ends reactively.

`daedalus_timed_npi()` is a helper function to create a `<daedalus_npi>` that is only triggered by time, and is not responsive to state variables. Primarily intended for use in real time modelling.

Usage

```

daedalus_npi(
  name,
  country,
  infection,
  openness = NULL,
  start_time = 30,
  end_time = NULL,
  max_duration = 365
)

is_daedalus_npi(x)

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

daedalus_timed_npi(start_time, end_time, openness, country)

```

Arguments

name	An NPI strategy name from among daedalus.data::closure_strategy_names, or NA. Passing a pre-defined strategy name automatically pulls in openness parameters for the associated response strategy; these are stored as packaged data in daedalus.data::closure_strategy_data. Pass NA to define a custom response strategy by passing a vector to openness.
country	A country or territory object of class <code><daedalus_country></code>, or a country or territory name from those included in the package; see daedalus.data::country_names, or a country ISO2 or ISO3 code; see daedalus.data::country_codes_iso2c and daedalus.data::country_codes_iso3c. Country-specific data such as the community and workplace contacts, the demography, and the distribution of the workforce into economic sectors is automatically accessed from package data for the relevant country name if it is passed as a string. To override package defaults for country characteristics, pass a <code><daedalus_country></code> object instead. See daedalus_country() for more.
infection	An infection parameter object of the class <code><daedalus_infection></code>, or an epidemic name for which data are provided in the package; see daedalus.data::epidemic_names for historical epidemics or epidemic waves for which parameters are available. Passing the name as a string automatically accesses the default parameters of an infection. Create and pass a <code><daedalus_infection></code> to tweak infection parameters using daedalus_infection().
openness	For <code>daedalus_npi()</code>, an optional numeric matrix giving the openness of each setting in the model when the NPI is in effect. For <code>daedalus_timed_npi()</code>, a list of numeric matrices giving the openness coefficients for each interval specified by corresponding elements of <code>start_time</code> and <code>end_time</code>. Expected to have a minimum size of <code>N_ECON_SECTORS + N_AGE_GROUPS</code> (currently 98), corresponding to two settings with 49 strata per setting, with all val-

	ues are in the range $[0, 1]$,
start_time	For <code>daedalus_npi()</code> , a single number giving the number of days after the start of the model that the NPI response begins. Defaults to 30. For <code>daedalus_timed_npi()</code> : may be a vector of NPI start times, with no default values.
end_time	For <code>daedalus_npi()</code> , a single number giving the number of days after the start of the model that the NPI response ends. Defaults to <code>NULL</code> , indicating that the response ends (if no other condition is met), at the maximum duration. For <code>daedalus_timed_npi()</code> : may be a vector of NPI end times, with no default values. Timed NPIs have no default duration.
max_duration	The maximum number of days that an NPI response is active whether started by passing the <code>start_time</code> or when a state threshold is crossed. Defaults to 365 days. NPIs created using <code>daedalus_timed_npi()</code> have no default maximum duration and are encoded by pairs of start and end times.
x	An object to be tested or printed as a <code><daedalus_npi></code> .
...	For <code>daedalus_npi()</code> , other parameters passed to <code>new_daedalus_response()</code> . For the print method, other parameters passed to <code>print()</code> .

Details

Note that NPIs created using `daedalus_npi()` are reactive to the model state (i.e., the epidemic state), and can trigger multiple times when state conditions are met.

Value

`daedalus_npi()`: A `<daedalus_npi>` class object that specifies an NPI that may be responsive to both model state and time.

`daedalus_timed_npi()`: A `<daedalus_npi>` class object which specifies time-limited interventions only.

Examples

```
# for a school closure strategy
daedalus_npi("school_closures", "GBR", "sars_cov_1")

# set custom openness
daedalus_npi(
  NA,
  "GBR", "sars_cov_1",
  openness = cbind(rep(1, 49), rep(0.1, 49))
)

# time-limited NPI with multiple phases
phase_1 <- cbind(
  rep(1, 49),
  rep(0.5, 49)
)
phase_2 <- cbind(
```

```

      rep(1, 49),
      rep(0.3, 49)
    )
    phase_3 <- cbind(
      rep(1, 49),
      rep(0.8, 49)
    )

    daedalus_timed_npi(
      start_time = c(10, 20, 30),
      end_time = c(15, 25, 40),
      openness = list(
        phase_1, phase_2, phase_3
      ),
      country = "GBR"
    )

```

class_vaccination	<i>Represent vaccine investment scenarios for DAEDALUS</i>
-------------------	--

Description

Helper functions to create and work with S3 class <daedalus_vaccination> objects for use with [daedalus\(\)](#). These objects store vaccination parameters for reuse and have methods for easy parameter access and editing, as well as processing raw vaccination characteristics for the DAEDALUS model.

Usage

```

daedalus_vaccination(
  name,
  country,
  start_time = NULL,
  rate = NULL,
  uptake_limit = NULL,
  efficacy = 50,
  waning_period = 180
)

is_daedalus_vaccination(x)

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

```

Arguments

name	A vaccination investment scenario name as a string. Passing a name from among <code>daedalus.data::vaccination_scenario_names</code> leads to any optional parameters (see below) being drawn from pre-canned scenarios. These are stored as packaged data in <code>daedalus.data::vaccination_scenario_data</code> . Passing a name that is not among pre-canned scenarios causes all following parameters to become compulsory and non-optional.
country	A <code><daedalus_country></code> object or a 2- or 3-character string that can be coerced to a <code><daedalus_country></code> (e.g. "GBR" for the United Kingdom). Used to determine when vaccination should end.
start_time	The number of days after the start of the epidemic that vaccination begins. Must be a single number. Defaults to NULL and is optional if name is a pre-canned strategy. If so, the start time is taken from the vaccination scenarios specified by name. Passed to the <code>time_on</code> argument in <code>new_daedalus_response()</code> via the class constructor <code>new_daedalus_vaccination()</code> .
rate	A single number for the <i>percentage</i> of the total population that can be vaccinated each day. This is converted into a proportion automatically within <code>daedalus()</code> . Defaults to NULL and is optional if name is a pre-canned strategy; if so rate is taken from pre-canned scenario data.
uptake_limit	A single number giving the upper limit for the <i>percentage</i> of the population that can be vaccinated. When this limit is reached, vaccination ends. Passed to the <code>value_state_off</code> argument in <code>new_daedalus_response()</code> via the class constructor <code>new_daedalus_vaccination()</code> . Defaults to NULL and is optional if name is a pre-canned strategy in which case <code>uptake_limit</code> is taken from pre-canned scenario data.
efficacy	A single number in the range $[0, 100]$ giving the efficacy of vaccination in preventing infection. A value of 0 indicates that vaccinated individuals are as susceptible to infection as unvaccinated ones, while 100 would indicate completely non-leaky vaccination that completely protects against infection. Defaults to 50%.
waning_period	A single number representing the number of days over which the average individual wanes out of the vaccinated stratum to the unvaccinated stratum. Only individuals in the susceptible and recovered compartments can wane out of being vaccinated. Defaults to 180 days.
x	An object to be tested or printed as a <code><daedalus_vaccination></code> .
...	For <code>daedalus_vaccination()</code> , other parameters passed to <code>new_daedalus_response()</code> . For the print method, other parameters passed to <code>print()</code> .

Details

Note that vaccination once ended by reaching the `uptake_limit` does not restart once individuals wane out of the vaccinated compartment.

Examples

```
# for no advance vaccine investment in the UK
```

```

daedalus_vaccination("none", "GBR")

# modifying parameters for pre-defined strategies during initialisation
# set daily vaccination rate to 1.5% of population
daedalus_vaccination("low", "GBR", rate = 1.5)

# a fully customised strategy
daedalus_vaccination(
  "custom", "GBR",
  start_time = 100, rate = 1.5, uptake_limit = 70
)

```

daedalus	<i>DAEDALUS model for health, social and economic costs of a pandemic</i>
----------	---

Description

Run the DAEDALUS model from R. This is a work in progress.

Usage

```

daedalus(
  country,
  infection,
  response_strategy = NULL,
  vaccine_investment = NULL,
  behaviour = NULL,
  response_time = 30,
  initial_state_manual = NULL,
  time_end = 600,
  ...
)

```

Arguments

country	A country or territory object of class <daedalus_country>, or a country or territory name from those included in the package; see daedalus.data::country_names , or a country ISO2 or ISO3 code; see daedalus.data::country_codes_iso2c and daedalus.data::country_codes_iso3c . Country-specific data such as the community and workplace contacts, the demography, and the distribution of the workforce into economic sectors is automatically accessed from package data for the relevant country name if it is passed as a string. To override package defaults for country characteristics, pass a <daedalus_country> object instead. See daedalus_country() for more.
infection	An infection parameter object of the class <daedalus_infection>, or an epidemic name for which data are provided in the package; see daedalus.data::epidemic_names for historical epidemics or epidemic waves for which parameters are available.

Passing the name as a string automatically accesses the default parameters of an infection. Create and pass a `<daedalus_infection>` to tweak infection parameters using `daedalus_infection()`.

<code>response_strategy</code>	A string for the name of response strategy followed, a numeric of length 45 (number of economic sectors), a matrix with dimensions 49 x N where N is the number of contacts settings, or a <code><daedalus_npi></code> object. Defaults to NULL, which simulates an unmitigated epidemic. While the response strategy is active, economic contacts are scaled using the package data object <code>daedalus.data::closure_strategy_data</code> .
<code>vaccine_investment</code>	Either a single string or a <code><daedalus_vaccination></code> object specifying the vaccination parameters associated with an advance vaccine-investment scenario. Defaults to NULL for absolutely no vaccination in the model. A vaccination investment of "none" indicates no <i>prior</i> investment, but the model will include vaccination beginning after 1 year, at a low rate across all age groups. Other accepted values are "low", "medium" and "high". See <code>daedalus_vaccination()</code> for more information.
<code>behaviour</code>	An optional object of class <code><daedalus_behaviour></code> which determines how population-level perception of epidemic signals affects infection transmission. May be NULL for no behavioural modification of infection transmission. See the <code><daedalus_behaviour></code> class documentation for more details on the available behavioural mechanisms
<code>response_time</code>	A single numeric value for the time in days at which the selected response is activated. This is ignored if the response has already been activated by the hospitalisation threshold being reached. Defaults to 30 days. Responses have a default maximum duration of 365 days. This can be changed by passing a <code><daedalus_npi></code> object to <code>response_strategy</code> .
<code>initial_state_manual</code>	An optional named list with the names <code>p_infectious</code> , <code>p_asymptomatic</code> , and <code>p_immune</code> . <code>p_infectious</code> and <code>p_asymptomatic</code> give the proportion of infectious and symptomatic individuals in each age group and economic sector. Defaults to <code>1e-6</code> and <code>0.0</code> respectively. <code>p_immune</code> may be a single number in the range <code>0.0 <= p_immune <= 1.0</code> or a 4-element vector in that range (the number of age groups in the model), for the proportion of individuals in the population or in each age group that have some pre-existing immunity to infection (reduced susceptibility). See Details for more.
<code>time_end</code>	An integer-like value for the number of timesteps at which to return data. This is treated as the number of days with data returned for each day. Defaults to 300 days.
<code>...</code>	Optional arguments that are passed to <code>dust2::dust_ode_control()</code> .

Details

Details: Initial state:

Users can pass the following initial state parameters to `initial_state_manual`:

- `p_infectious`: A single numeric value in the range `[0.0, 1.0]` giving the proportion of individuals in each age group and economic sector that are to be initialised as infectious. Defaults to `1e-6`, or one in every one million as infectious.

- `p_asymptomatic`: A single numeric value in the range $[0.0, 1.0]$ for the proportion of initially infectious individuals who are considered to be asymptomatic. Defaults to 0.0.
- `p_immune`: Either a single number or a vector of 4 elements (the number of age groups) in the range $[0.0, 1.0]$ for the proportion of the population (or each age group) that is considered to have pre-existing immunity. This is a stop-gap implementation that assumes one of two cases: (1) if no vaccination is intended in the model and `vaccine_investment` is `NULL`, the susceptibility of individuals pre-existing immunity is 50%; or (2) if a vaccination strategy is specified, the pre-existing immunity is assumed to be from a prior rollout, and the susceptibility is determined by the chosen vaccination strategy (as $1 - \text{efficacy}$).

Value

A `<daedalus_output>` object if infection is a string or a single `<daedalus_infection>` object.

Examples

```
# country and infection specified by strings using default characteristics
output <- daedalus(
  "Canada", "influenza_1918"
)

# print output
output

# country passed as <daedalus_country> with a new 'total' contact matrix
conmat_new <- get_data(daedalus_country("Canada"), "contact_matrix")
country_x <- daedalus_country(
  "Canada",
  contact_matrix = list(total = conmat_new * 2.0) # simulate new data
)
output <- daedalus(country_x, "influenza_1918")

# with some infection parameters over-ridden by the user
output <- daedalus(
  "United Kingdom",
  daedalus_infection("influenza_1918", r0 = 1.3)
)

# with default initial conditions over-ridden by the user
output <- daedalus(
  "United Kingdom", "influenza_1918",
  initial_state_manual = list(p_infectious = 1e-3)
)

# including behavioural modification
output <- daedalus(
  "Canada", "influenza_1918",
  behaviour = daedalus_old_behaviour(),
  time_end = 100
)
```

daedalus_costs

Get epidemic losses from a DAEDALUS model run

Description

Calculate the costs of an epidemic and any mitigation measures, either from a Daedalus model run, or from a timeseries of epidemic data and relevant demographic and economic parameters.

Usage

```
## S3 method for class 'daedalus_output'
get_costs(
  x,
  summarise_as = c("none", "total", "domain"),
  productivity_loss_infection = 1,
  ...
)

## S3 method for class 'data.frame'
get_costs(
  x,
  comp_non_working,
  comp_infected,
  comp_dead,
  workforce,
  daily_gva,
  productivity_loss_infection,
  vsl_by_age,
  life_expectancy,
  value_school_year,
  n_students,
  edu_effectiveness_remote,
  npi_data = NULL,
  summarise_as = c("none", "total", "domain"),
  ...
)

get_costs(x, ...)
```

Arguments

x A <daedalus_output> object from a call to `daedalus()`, or a <data.frame> of an epidemic timeseries.

If **x** is a <data.frame>, it must have the columns "time", "compartment", "age_group", "econ_sector" and "value", giving the model time, the epidemiological compartment, and the number of individuals of each age group

	and economic sector in each compartment at each time. See Details for more on the expectations around how this dataset is organised.
summarise_as	A string from among "none", "total", or "domain", for how the costs should be returned. Select "none", the default, for the raw costs along with overall and domain-specific totals; "total" for the overall cost, and "domain" for the total costs per domain; the domains are 'economic', 'education', and 'life years'.
productivity_loss_infection	A single number in the range [0, 1] giving the loss in productivity associated with symptomatic infection. Currently defaults to 1.0 for compatibility with earlier function versions.
...	<dynamic-dots> Other arguments to class methods.
comp_non_working	A character vector giving the names of compartments in which individuals are assumed to have a reduced ability to work. This may (and should) include compartments where individuals are infected and symptomatic, as well as hospitalised or dead.
comp_infected	A character vector giving the name of compartments in which individuals are assumed to be infected with reduced ability to work. This should not include compartments where individuals are assumed to have no ability to work (hospitalised or dead).
comp_dead	A string giving the name of the compartment holding the count of deaths. Note that only a single deaths compartment is supported, so models with multiple death states should pass the final state . Used to calculate the value of lives lost.
workforce	A numeric vector of the number of workers in each economic sector of the population. Must be absolute numbers (i.e., not scaled to thousands or millions). Must have the same length as daily_gva.
daily_gva	A numeric vector of the daily gross-value added (GVA) in million dollars, by each economic sector of the population (or territory or state) in which the epidemic occurs.
vsl_by_age	A numeric vector of the value of a statistical life (VSL) for each age group in x.
life_expectancy	A numeric vector of the life-expectancy per age group. Expected to be the same length as vsl_by_age.
value_school_year	A single number giving the value of a school year. Users can use the helper function <code>get_value_school_year()</code> if a population's gross income per-capita is known to apply the internal Daedalus model for valuing a school year, or a value obtained in some other way.
n_students	A single number for the number of students in the population, whose education is expected to be impacted by the epidemic.
edu_effectiveness_remote	A single number for the effectiveness of remote education, in the range [0, 1].
npi_data	A list with numeric elements that mirrors the elements of a <daedalus_npi>, with the following names:

- `npi_times_start`: A numeric vector of the start times of any NPIs modelled.
- `npi_times_end`: A numeric vector of the end times of modelled NPIs.
- `npi_durations`: A numeric vector of the durations of NPIs.
- `npi_periods`: A numeric vector giving integer-ish elements in the ranges specified by the start and end times.
- `openness`: Either a numeric vector (if modelling a single NPI), or a list of numeric vectors (if modelling multiple NPIs), giving the openness coefficients of NPI regimes. Vector lengths must be the same as `workforce`.

Details

Expectations for non-Daedalus model data:

When a `<data.frame>` `x` is passed, it must have at least one row, at least some non-missing values, and:

1. Columns `"time"`, `"age_group"`, `"compartment"`, `"econ_sector"`, and `"value"`, giving the model time (numeric), age group identifier (character), epidemiological compartment identifier (character), economic sector identifier (character), and the value of the number of individuals in each age, epi-compartment, and economic group at each time (numeric). Numeric values must be non-negative > 0 , non-missing, and finite.
2. Compartments in `comp_infected`, `comp_non_working`, and `comp_dead` must be found in `x$compartment`.
3. All compartments of `comp_infected` and `comp_dead` must be in `comp_non_working`.

Output:

The total cost in million dollars is returned as `total_cost`. This is comprised of the following costs.

Economic costs:

A three element list of `economic_cost_total`, the total costs from pandemic impacts on economic sectors, including both costs of lost gross value added (GVA) due to pandemic-control restrictions or closures (`economic_cost_closures`), and pandemic-related absences due to illness and death (`economic_cost_absences`).

Educational costs:

A three element list of `education_cost_total`, the total costs from pandemic impacts on education due to pandemic-control restrictions or closures (`education_cost_closures`), and pandemic-related absences due to illness and death (`education_cost_absences`).

Life-value lost:

A four-element vector (for the number of age groups) giving the value of life-years lost per age group. This is calculated as the life-expectancy of each age group times the value of a statistical life, with all years assumed to have the same value.

Life-years lost:

A four-element vector (for the number of age groups) giving the value of life-years lost per age group. This is calculated as the life-expectancy of each age group times the number of deaths in that age group. No quality adjustment is applied.

Value

A list of different cost values, including the total cost. See **Details** for more information.

Examples

```
# for <daedalus_output> objects from `daedalus()`
o <- daedalus("CAN", "sars_cov_1", time_end = 100)
get_costs(o, "domain")

# for a data.frame of epi compartment timeseries, as from any epi model
output <- daedalus("Canada", "influenza_1918", time_end = 100)

data <- get_data(output)
comp_non_working <- c(
  "infect_symp",
  "hospitalised_recov",
  "hospitalised_death",
  "dead"
)
comp_infected <- "infect_symp"
comp_dead <- "dead"

daily_gva <- output$country$gva
workforce <- output$country$workers
vsl_by_age <- output$country$vsl
life_expectancy <- output$country$life_expectancy

value_school_year <- 1e6 # 1 million dollars
n_students <- output$country$demography[3L]

edu_effectiveness_remote <- 0.33

productivity_loss_infection <- 1.0

get_costs(
  data,
  comp_non_working,
  comp_infected,
  comp_dead,
  daily_gva,
  workforce,
  vsl_by_age,
  life_expectancy,
  value_school_year,
  n_students,
  productivity_loss_infection = productivity_loss_infection,
  edu_effectiveness_remote = edu_effectiveness_remote,
  summarise_as = "domain"
)
```

daedalus_multi_infection

Run daedalus for multiple infection parameter sets

Description

Run [daedalus\(\)](#) for multiple parameter sets, with the intended use case of running the model under uncertainty in infection parameters.

Usage

```
daedalus_multi_infection(
  country,
  infection,
  response_strategy = NULL,
  vaccine_investment = NULL,
  behaviour = NULL,
  response_time = 30,
  initial_state_manual = NULL,
  time_end = 600,
  ...
)
```

Arguments

- | | |
|--------------------|---|
| country | A country or territory object of class <code><daedalus_country></code> , or a country or territory name from those included in the package; see daedalus.data::country_names , or a country ISO2 or ISO3 code; see daedalus.data::country_codes_iso2c and daedalus.data::country_codes_iso3c . Country-specific data such as the community and workplace contacts, the demography, and the distribution of the workforce into economic sectors is automatically accessed from package data for the relevant country name if it is passed as a string. To override package defaults for country characteristics, pass a <code><daedalus_country></code> object instead. See daedalus_country() for more. |
| infection | A list of <code><daedalus_infection></code> objects. Must have a minimum length of 2. |
| response_strategy | A string for the name of response strategy followed, a numeric of length 45 (number of economic sectors), a matrix with dimensions 49 x N where N is the number of contacts settings, or a <code><daedalus_npi></code> object. Defaults to NULL, which simulates an unmitigated epidemic. While the response strategy is active, economic contacts are scaled using the package data object <code>daedalus.data::closure_strategy_data</code> . |
| vaccine_investment | Either a single string or a <code><daedalus_vaccination></code> object specifying the vaccination parameters associated with an advance vaccine-investment scenario. Defaults to NULL for absolutely no vaccination in the model. A vaccination investment of "none" indicates no <i>prior</i> investment, but the model will include |

	vaccination beginning after 1 year, at a low rate across all age groups. Other accepted values are "low", "medium" and "high". See daedalus_vaccination() for more information.
behaviour	An optional object of class <daedalus_behaviour> which determines how population-level perception of epidemic signals affects infection transmission. May be NULL for no behavioural modification of infection transmission. See the <daedalus_behaviour> class documentation for more details on the available behavioural mechanisms
response_time	A single numeric value for the time in days at which the selected response is activated. This is ignored if the response has already been activated by the hospitalisation threshold being reached. Defaults to 30 days. Responses have a default maximum duration of 365 days. This can be changed by passing a <daedalus_npi> object to response_strategy.
initial_state_manual	An optional named list with the names p_infectious, p_asymptomatic, and p_immune. p_infectious and p_asymptomatic give the proportion of infectious and symptomatic individuals in each age group and economic sector. Defaults to 1e-6 and 0.0 respectively. p_immune may be a single number in the range 0.0 <= p_immune <= 1.0 or a 4-element vector in that range (the number of age groups in the model), for the proportion of individuals in the population or in each age group that have some pre-existing immunity to infection (reduced susceptibility). See Details for more.
time_end	An integer-like value for the number of timesteps at which to return data. This is treated as the number of days with data returned for each day. Defaults to 300 days.
...	Optional arguments that are passed to dust2::dust_ode_control() .

Value

A list of <daedalus_output> objects of the same length as infection.

econ_constants	<i>Economic constants used in DAEDALUS</i>
----------------	--

Description

Economic constants used in DAEDALUS

Usage

EDU_EFFECTIVENESS_REMOTE

EDU_ANNUAL_ROR

WORK_EXPECTED_YEARS

EARNINGS_LOSS_DISCOUNT

Details

The Daedalus model uses these constants for economic value calculations:

1. edu_effectiveness_remote: The effectiveness of remote education.
2. edu_annual_ror: The rate of return on a year of education.
3. work_expected_years: The expected number of years of work for school-age children.
4. earnings_loss_discount: The discounting rate used in the calculation of lost earnings.

epi_constants

Epidemiological compartments and indices

Description

Names and indices for the epidemiological compartments used in DAEDALUS, for reuse in model code. The duplication is for ease of extracting indices from a named list, and of extracting names without having to call names().

Usage

COMPARTMENTS

idx_COMPARTMENTS

N_EPI_COMPARTMENTS

i_EPI_COMPARTMENTS

N_MODEL_COMPARTMENTS

N_INFECTION_SUBSYSTEM

Details

DAEDALUS has 8 epidemiological compartments: susceptible, exposed, infectious and symptomatic ("infect_symp"), infectious and asymptomatic ("infect_asymp"), hospitalised leading to recovery ("hospitalised_recov"), hospitalised leading to death ("hospitalised_death"), recovered, and dead.

There are 2 additional compartments that track the number of new infections and new hospitalisations.

Value

COMPARTMENTS returns a character vector of the epidemiological compartment names. idx_COMPARTMENTS returns a list with the compartment index.

All other constants return integer values.

epi_output_helpers *Calculate daily incidences and summarise epidemic measures*

Description

Functions to quickly summarise timeseries data from `daedalus()` to provide daily values for infections, hospitalisations, deaths, and vaccinations, while allowing grouping by different strata.

Usage

```
get_incidence(
  data,
  measures = c("infections", "hospitalisations", "deaths"),
  groups = NULL
)

get_epidemic_summary(
  data,
  measures = c("infections", "hospitalisations", "deaths"),
  groups = NULL
)

get_new_vaccinations(data, groups = NULL)

get_attack_rate(
  data,
  measures = c("infections", "hospitalisations", "deaths"),
  groups = c("econ_sector", "age_group")
)
```

Arguments

data	Either a <code><data.frame></code> from a call to <code>get_data()</code> on a <code><daedalus_output></code> object, or such an object directly.
measures	A character vector of one or more of the following, passed to <code>get_incidence()</code> and <code>get_epidemic_summary()</code> : "infections", "hospitalisations" or "deaths" for the measure to return. Defaults to returning all three in long format. <code>get_daily_vaccinations()</code> does not accept a measures argument and only provides the number of daily vaccinations.
groups	An optional character vector of grouping variables that correspond to model strata. Defaults to NULL which gives incidence across the whole population. Allowed groups correspond to modelled strata: "age_group", "vaccine_group", and "econ_sector". <code>get_attack_rate()</code> does not support "vaccine_group" due to a shifting baseline effect. <code>get_daily_vaccinations()</code> only accepts "age_group" and "econ_sector".

Value

A `<data.frame>` in long format, with one entry per model timestep, measure, and group chosen.

- `get_incidence()` returns a data frame with the number of daily new infections, new hospitalisations, and/or new deaths in each of the groups specified by groups.
- `get_epidemic_summary()` returns a data frame with the total number of the value specified in measure for each of the groups specified by groups.
- `get_daily_vaccinations()` returns a data frame with columns for the number of new daily vaccination in each combination of groups if provided. Columns for the groups are added when groups are specified.
- `get_attack_rate()` returns a data frame similar to `get_epidemic_summary()`, but with a `p_affected` column giving the value as a proportion of the initial group size (capped at 1.0).

Examples

```
data <- daedalus("Canada", "sars_cov_1")

# new infections
new_infections <- get_incidence(data, "infections")

# epidemic summary
get_epidemic_summary(
  data,
  groups = "age_group"
)

# get daily vaccinations
daily_vaccinations <- get_new_vaccinations(data)

# get attack rate
get_attack_rate(data, "infections", groups = "age_group")
```

get_data

Get parameters from DAEDALUS classes

Description

Generic and methods for S3 classes for safely getting class parameters.

Usage

```
## S3 method for class 'daedalus_country'
get_data(x, to_get, setting = c("default", "all"), ...)

## S3 method for class 'daedalus_infection'
get_data(x, to_get, ...)
```

```
## S3 method for class 'daedalus_npi'
get_data(x, to_get, ...)

## S3 method for class 'daedalus_output'
get_data(x, to_get = NULL, ...)

## S3 method for class 'daedalus_vaccination'
get_data(x, to_get, ...)

get_data(x, ...)
```

Arguments

x	An S3 class object from the daedalus package of the <daedalus_country>, <daedalus_infection>, <daedalus_npi>, <daedalus_output>, and <daedalus_vaccination> class.
to_get	A string giving the name of the element of x to return. For <daedalus_output> the additional internal data for R_{eff} can be accessed by passing "rt_data".
setting	Optional string for a contact setting. Defaults to the first provided contact matrix if multiple are present.
...	<dynamic-dots> Other arguments to class methods. Class methods do not currently support any other arguments.

Value

Returns a member of x, with the class preserved (e.g. numeric vector for a country "demography").

For <daedalus_output> objects, returns the model timeseries data when no element is specified.

Examples

```
# simple example of getting data
country_A <- daedalus_country("United Kingdom")
get_data(country_A, "demography")

get_data(country_A, "contact_matrix")

disease_x <- daedalus_infection("sars_cov_1", r0 = 1.9)
get_data(disease_x, "r0")

# get model data
output <- daedalus("Canada", "influenza_1918")
head(
  get_data(output)
)
```

get_fiscal_costs	<i>Get pandemic fiscal costs from a model run</i>
------------------	---

Description

A helper function that post-processes a <daedalus_output> object to calculate the costs to a national government (fiscal costs) of responding to a pandemic. Includes costs of economic support, vaccinations given, and NPIs administered or implemented.

Usage

```
get_fiscal_costs(
  x,
  support_level = 0.2,
  price_vax = 1,
  price_npi = 1,
  uptake_npi = 1,
  interest_rate = 4,
  tax_rate = 35,
  spending_rate = 45,
  starting_debt = 0,
  productivity_loss_infection = 1
)
```

Arguments

x	A <daedalus_output> object from a call to daedalus() , or a <data.frame> of an epidemic timeseries. If x is a <data.frame>, it must have the columns "time", "compartment", "age_group", "econ_sector" and "value", giving the model time, the epidemiological compartment, and the number of individuals of each age group and economic sector in each compartment at each time. See Details for more on the expectations around how this dataset is organised.
support_level	The proportion of pandemic-related economic losses that a government compensates, as a proportion.
price_vax	The per-dose price of vaccination.
price_npi	The per-day and per-person price of implementing any pandemic response. May include costs such as testing or masks.
uptake_npi	The extent to which NPIs are taken up by the population; essentially a number that modifies (reduces) the total cost of implementing an NPI.
interest_rate	The annual interest rate on government borrowing for pandemic response.
tax_rate	The annual mean tax rate on economic output; used to calculate government revenue.
spending_rate	The annual mean rate of public spending as a percentage of GDP.

- `starting_debt` The value of national starting debt. Currently assumed to be zero while country-specific data are collected.
- `productivity_loss_infection` A single number in the range $[0, 1]$ giving the loss in productivity associated with symptomatic infection. Currently defaults to 1.0 for compatibility with earlier function versions.

Details

Default argument values are intended to be representative. This function is intended to be called after `daedalus()` and parameters required to compute fiscal costs may need to be bundled along with model outputs. Note that all rates (interest rate, spending rate, and tax rate) are given as annual percentages. Only the interest rate is converted to an daily value from an annual one for use in compounding.

Public spending:

Spending is calculated as:

$$G_t = \sigma(\text{GVA}^* - \text{GVA}_t) + Cv_t + Cp_t$$

where Cv_t is the time-specific cost of vaccination, and is calculated as the cost of new vaccinations in each timestep: $P_v \times \delta V_t$.

Cp_t is the time-specific cost of implementing pandemic response, and is calculated as $\psi(\bar{N} - D_t)P_p$, where $\bar{N} - D_t$ is the remaining number of individuals in the population, and ψ is the proportion taking up any protection offered by the response.

Interest on spending and fiscal cost:

We assume that the government borrows to spend on pandemic mitigation measures outlined above such that the total cost to the public is then

$$\text{TCG}_t = G_t + (1 + R_t^T)\text{TCG}_{t-1}$$

where R_t^T is the daily rate of interest to be paid on the borrowed amount.

The interest rate is modelled as being constant over time. Users pass the annual rate of interest as a percentage, and this is converted to a daily rate using the internal function `annual_rate_daily()` as $(1 + R_t^T)^{1/365} - 1$.

Total public debt:

The total public debt at the end of the pandemic b_t is then the sum of :

- total public spending on the pandemic TCG_t ,
- existing day-to-day public spending $\bar{G}$ which is assumed to be a fraction of daily GDP νGDP ,
- existing debt owed due to past daily spending (including on pandemic mitigation), and interest to be paid on the debt,
- less the revenues collected from taxation, μGVA_{t-1} , where μ is the mean rate of taxation.

The daily GVA is the pre-pandemic GVA scaled by the available labour supply during the pandemic, taking into account labour restrictions due to illness-related absences and deaths, and response-related restrictions.

GDP is calculated as the sum of sector-specific daily GVA, and existing debt is currently assumed to be zero and is not included in the equation.

$$b_t = \bar{G} + \text{TCG}_{t-1} + (1 + R_{t-1}^T)b_{t-1} - \mu\text{GVA}_{t-1}$$

Value

A two-element list giving:

- A list of `fiscal_costs` with elements giving the fiscal costs TCG_t and a breakdown of these costs, as well as interest;
- A list of `public_debt` of the public debt b_t , which is the net of baseline public spending, pandemic response costs, and pandemic-impacted revenue.

Examples

```
# get fiscal costs for UK with SARS-CoV-2
# and both closures and vaccinations
o <- daedalus(
  "GBR", "sars_cov_2_pre_alpha",
  "economic_closures", "high",
  time_end = 100
)
fc <- get_fiscal_costs(o)

# also works when no closures are applied
o <- daedalus(
  "CAN", "influenza_2009",
  time_end = 30
)

# Compare public debt added estimates with other estimates from Covid-19:
# https://www.ctf.ca/EN/EN/Newsletters/Perspectives/2020/3/200302.aspx
get_fiscal_costs(o)
```

get_life_years_lost	<i>Get life-years lost by demographic group.</i>
---------------------	--

Description

Get life-years lost by demographic group.

Usage

```
get_life_years_lost(output, groups = c("none", "age_group"))
```

Arguments

output	A <daedalus_output> object.
groups	Whether to get the life-years lost by age group. Selecting "none" gives the total life-years lost.

Value

A <data.frame> with the number of life-years lost, optionally per age-group.

model_constants	<i>DAEDALUS model constants</i>
-----------------	---------------------------------

Description

Frequently used values for the DAEDALUS model related to the model population and structure. See also [epi_constants](#) for constants specific to the epidemiological model.

Usage

```

N_AGE_GROUPS

i_AGE_GROUPS

N_VACCINE_STRATA

N_VACCINE_DATA_GROUPS

AGE_GROUPS

i_WORKING_AGE

i_SCHOOL_AGE

N_ECON_SECTORS

i_ECON_SECTORS

i_EDUCATION_SECTOR

N_ECON_STRATA

i_NOT_WORKING

DIM_AGE_GROUPS

DIM_EPI_COMPARTMENTS

```

DIM_ECON_SECTORS
 DIM_VACCINE_STRATA
 i_UNVACCINATED_STRATUM
 i_VACCINATED_STRATUM
 i_NEW_VAX_STRATUM
 VACCINE_GROUPS
 N_OUTPUT_COLS
 N_FLAGS
 FLAG_NAMES
 MIN_DATA_COL_NAMES

Details

The DAEDALUS model requires the following values to be fixed.

- Number of age groups: 4
- Number of vaccination strata: 2
- Age group bins: 0-4 years, 5-19 years, 20-65 years (working age), 65+ years
- Index of the working-age age groups: 3; see AGE_GROUPS
- Number of economic sectors: 45
- Index of individuals not working as a layer in the 3D state tensor: 1
- Number of economic strata: 46; note that this is always one more than the number of economic sectors, with the additional stratum for those not in work which includes all individuals not of working age, and a proportion of working age individuals.
- Array dimension of age groups: 1
- Array dimension of epidemiological compartments: 2
- Array dimension of economic sectors: 3
- Array dimension of vaccination strata: 4
- Indices and numbers of key groups.
- Number of state variables that are flags, switches, or indicators.

Value

Values for the model constants.

set_data	<i>Set parameters in DAEDALUS classes</i>
----------	---

Description

Generic and methods for S3 classes for safely setting class parameters. Only parameters considered safe to change – mostly contact data in the <country>, but all parameters in <infection> – can be changed in this way.

Usage

```
## S3 method for class 'daedalus_infection'
set_data(x, ...)

## S3 method for class 'daedalus_vaccination'
set_data(x, ...)

set_data(x, ...)
```

Arguments

x	An S3 class object from the daedalus package of the <daedalus_country> or <infection> class.
...	<dynamic-dots> Named optional arguments for parameters to be changed, with their new values. The only values allowed for <daedalus_country> objects are "contact_matrix", "contacts_workplace", and "contacts_consumer_worker".

Value

An S3 object of the same class as input x.

Examples

```
# simple example of setting all contacts to 1
country_A <- daedalus_country("United Kingdom")
country_A

disease_x <- daedalus_infection("sars_cov_1")
disease_x <- set_data(disease_x, r0 = 3.0)
disease_x
```

summary_constants	<i>Model output measures and groups for summaries and derived time-series</i>
-------------------	---

Description

Model output measures and groups for summaries and derived time-series

Usage

SUMMARY_MEASURES

SUMMARY_GROUPS

Index

- * **econ_constant**
 - econ_constants, 21
- * **epi_constants**
 - epi_constants, 22
- * **epi_constant**
 - epi_constants, 22
- * **generics**
 - get_data, 24
 - set_data, 31
- * **model_constant**
 - model_constants, 29
- * **summary_constants**
 - summary_constants, 32
- AGE_GROUPS (model_constants), 29
- as_daedalus_output
 - (class_daedalus_output), 6
- class documentation, 14, 21
- class_behaviour, 2
- class_country, 4
- class_daedalus_output, 6
- class_infection, 6
- class_npi, 8
- class_vaccination, 11
- COMPARTMENTS (epi_constants), 22
- daedalus, 13
- daedalus(), 4, 6, 8, 11, 12, 16, 20, 23, 26, 27
- daedalus.data::closure_strategy_data, 9
- daedalus.data::closure_strategy_names, 9
- daedalus.data::country_codes_iso2c, 5, 9, 13, 20
- daedalus.data::country_codes_iso3c, 5, 9, 13, 20
- daedalus.data::country_data, 5
- daedalus.data::country_names, 5, 9, 13, 20
- daedalus.data::epidemic_names, 7, 9, 13
- daedalus.data::infection_data, 7
- daedalus.data::vaccination_scenario_names, 12
- daedalus_costs, 16
- daedalus_country (class_country), 4
- daedalus_country(), 9, 13, 20
- daedalus_infection (class_infection), 6
- daedalus_infection(), 9, 14
- daedalus_multi_infection, 20
- daedalus_new_behaviour
 - (class_behaviour), 2
- daedalus_npi (class_npi), 8
- daedalus_old_behaviour
 - (class_behaviour), 2
- daedalus_timed_npi (class_npi), 8
- daedalus_vaccination
 - (class_vaccination), 11
- daedalus_vaccination(), 14, 21
- DIM_AGE_GROUPS (model_constants), 29
- DIM_ECON_SECTORS (model_constants), 29
- DIM_EPI_COMPARTMENTS (model_constants), 29
- DIM_VACCINE_STRATA (model_constants), 29
- dust2::dust_ode_control(), 14, 21
- EARNINGS_LOSS_DISCOUNT
 - (econ_constants), 21
- econ_constants, 21
- EDU_ANNUAL_ROR (econ_constants), 21
- EDU_EFFECTIVENESS_REMOTE
 - (econ_constants), 21
- epi_constants, 22, 29
- epi_output_helpers, 23
- FLAG_NAMES (model_constants), 29
- format(), 3
- get_attack_rate (epi_output_helpers), 23
- get_costs (daedalus_costs), 16

get_data, 24
 get_epidemic_summary
 (epi_output_helpers), 23
 get_fiscal_costs, 26
 get_incidence(epi_output_helpers), 23
 get_life_years_lost, 28
 get_new_vaccinations
 (epi_output_helpers), 23
 get_value_school_year(), 17

 i_AGE_GROUPS(model_constants), 29
 i_ECON_SECTORS(model_constants), 29
 i_EDUCATION_SECTOR(model_constants), 29
 i_EPI_COMPARTMENTS(epi_constants), 22
 i_NEW_VAX_STRATUM(model_constants), 29
 i_NOT_WORKING(model_constants), 29
 i_SCHOOL_AGE(model_constants), 29
 i_UNVACCINATED_STRATUM
 (model_constants), 29
 i_VACCINATED_STRATUM(model_constants),
 29
 i_WORKING_AGE(model_constants), 29
 idx_COMPARTMENTS(epi_constants), 22
 is_daedalus_behaviour
 (class_behaviour), 2
 is_daedalus_country(class_country), 4
 is_daedalus_infection
 (class_infection), 6
 is_daedalus_npi(class_npi), 8
 is_daedalus_output
 (class_daedalus_output), 6
 is_daedalus_vaccination
 (class_vaccination), 11

 MIN_DATA_COL_NAMES(model_constants), 29
 model_constants, 29

 N_AGE_GROUPS(model_constants), 29
 N_ECON_SECTORS(model_constants), 29
 N_ECON_STRATA(model_constants), 29
 N_EPI_COMPARTMENTS(epi_constants), 22
 N_FLAGS(model_constants), 29
 N_INFECTION_SUBSYSTEM(epi_constants),
 22
 N_MODEL_COMPARTMENTS(epi_constants), 22
 N_OUTPUT_COLS(model_constants), 29
 N_VACCINE_DATA_GROUPS
 (model_constants), 29
 N_VACCINE_STRATA(model_constants), 29

 new_daedalus_response(), 10, 12

 print(), 5, 7, 10, 12
 print.daedalus_behaviour
 (class_behaviour), 2
 print.daedalus_country(class_country),
 4
 print.daedalus_infection
 (class_infection), 6
 print.daedalus_npi(class_npi), 8
 print.daedalus_output
 (class_daedalus_output), 6
 print.daedalus_vaccination
 (class_vaccination), 11

 set_data, 31
 summary_constants, 32
 SUMMARY_GROUPS(summary_constants), 32
 SUMMARY_MEASURES(summary_constants), 32

 VACCINE_GROUPS(model_constants), 29
 validate_daedalus_output
 (class_daedalus_output), 6

 WORK_EXPECTED_YEARS(econ_constants), 21