Package ‘Bernadette’

January 12, 2024

Type Package

Title Bayesian Inference and Model Selection for Stochastic Epidemics

Version 1.1.5

Description Bayesian analysis for stochastic extensions of non-linear dynamic systems using advanced computational algorithms. Described in Bouranis, L., Demiris, N., Kalogeropoulos, K., and Ntzoufras, I. (2022) <arXiv:2211.15229>.

License GPL (>= 3)

URL `https://bernadette-eu.github.io/`

BugReports `https://github.com/bernadette-eu/Bernadette/issues/`

Depends R (>= 3.0.2)

Imports ggplot2, grid, gridExtra, magrittr, methods, Rcpp (>= 1.0.8.3), RcppParallel, rstantools, rstan (>= 2.26.0), scales, stats, utils

Suggests knitr, rmarkdown, roxygen2, StanHeaders (>= 2.26.0), testthat (>= 3.0.0)

LinkingTo BH (>= 1.78.0-0), Rcpp (>= 1.0.8.3), RcppEigen (>= 0.3.3.9.1), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

VignetteBuilder knitr

Biarch true

Encoding UTF-8

Language en-US

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

SystemRequirements GNU make

Config/testthat/edition 3

NeedsCompilation yes

Author Lampros Bouranis [aut, cre, cph]
(<https://orcid.org/0000-0002-1291-2192>),
European Union [fnd] (Funding by the European Union's Horizon 2020
research and innovation programme under the Marie Sklodowska-Curie
grant agreement No 101027218)

Maintainer Lampros Bouranis <bernadette.aueb@gmail.com>

Repository CRAN

Date/Publication 2024-01-12 22:30:05 UTC

R topics documented:

`Bernadette-package` *Bayesian inference and model selection for stochastic epidemics*

Description

`Bernadette` provides Bayesian analysis for stochastic extensions of dynamic non-linear systems using advanced computational algorithms.

References

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: `https://arxiv.org/abs/2211.15229`

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.3. `https://mc-stan.org`

`age_distribution` *Country-specific age distribution*

Description

Function to extract the age distribution of a country for a given year, broken down by 5-year age bands and gender, following the United Nations 2019 Revision of World Population Prospects.

Usage

```
age_distribution(country, year)
```

Arguments

`country`	character; country identifier, following the List of United Nations Member States. See countries_un.
`year`	numeric; calendar year.

Value

An object of class *data.frame* that contains the age distribution.

References

United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019, Online Edition. Rev. 1.

Prem, K., van Zandvoort, K., Klepac, P. et al (2017). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

Examples

```
# Age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)
```

`age_specific_cusum_infection_counts`

Age distribution of cumulative reported infections for Greece

---

Description

A dataset containing the age distribution of cumulative reported infections in Greece from 2020-08-31 to 2021-03-28 (30 weeks). The dataset has been extracted from the Hellenic National Public Health Organization database.

Usage

```
data(age_specific_cusum_infection_counts)
```

Format

A data frame with 210 rows and 5 variables:

Date Date, format; date in the format "2020-08-31"

Total_Cases numeric; count of total cumulative reported infections on a given date

0-39 numeric; count of cumulative reported infections on a given date for the age group "0-39"

40-64 numeric; count of cumulative reported infections on a given date for the age group "40-64"

65+ numeric; count of cumulative reported infections on a given date for the age group "65+"

Value

A data.frame object with 210 rows and 5 variables.

References

Sandbird (2022). Daily regional statistics for covid19 cases in Greece.

Source

https://github.com/Sandbird/covid19-Greece/

`age_specific_infection_counts`
Age distribution of new reported infections for Greece

Description

A dataset containing the age distribution of new reported infections in Greece from 2020-08-31 to 2021-03-28 (30 weeks). The dataset has been extracted from the Hellenic National Public Health Organization database.

Usage

```
data(age_specific_infection_counts)
```

Format

A data frame with 210 rows and 8 variables:

Index integer; a sequence of integer numbers from 1 to 210

Right numeric; Index + 1

Date Date, format; date in the format "2020-08-31"

Week_ID numeric; index of the week that each day falls into. A week is assumed to have 7 days

Total_Cases numeric; count of total reported infections on a given date

0-39 numeric; count of reported infections on a given date for the age group "0-39"

40-64 numeric; count of reported infections on a given date for the age group "40-64"

65+ numeric; count of reported infections on a given date for the age group "65+"

Value

A data.frame object with 210 rows and 8 variables.

References

Sandbird (2022). Daily regional statistics for covid19 cases in Greece.

Source

`https://github.com/Sandbird/covid19-Greece/`

`age_specific_mortality_counts`

Age distribution of new reported deaths for Greece

Description

A dataset containing the age distribution of reported deaths in Greece from 2020-08-31 to 2021-03-28 (30 weeks). The dataset has been extracted from the Hellenic National Public Health Organization database.

Usage

```
data(age_specific_mortality_counts)
```

Format

A data frame with 210 rows and 8 variables:

Index integer; a sequence of integer numbers from 1 to 210

Right numeric; Index + 1

Date Date, format; date in the format "2020-08-31"

Week_ID numeric; index of the week that each day falls into. A week is assumed to have 7 days

New_Deaths numeric; count of new total reported deaths on a given date

0-39 numeric; count of new reported deaths on a given date for the age group "0-39"

40-64 numeric; count of new reported deaths on a given date for the age group "40-64"

65+ numeric; count of new reported deaths on a given date for the age group "65+"

Value

A data.frame object with 210 rows and 8 variables.

References

Sandbird (2022). Daily regional statistics for covid19 cases in Greece.

Source

`https://github.com/Sandbird/covid19-Greece/`

`aggregate_age_distribution`

Aggregate the age distribution matrix

Description

Function to aggregate the age distribution according to user-defined age groups.

Usage

```
aggregate_age_distribution(x, lookup_table)
```

Arguments

`x`	data.frame; an age distribution matrix. See age_distribution.
`lookup_table`	data.frame; a user-defined dataframe which maps the sixteen 5-year age bands to a new set of age bands.

Value

An object of class *data.frame* that contains the aggregated age distribution.

References

United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019, Online Edition. Rev. 1.

Prem, K., van Zandvoort, K., Klepac, P. et al (2020). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

Examples

```
# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                           Mapping = c(rep("0-39",  8),
                                       rep("40-64", 5),
                                       rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Plot the aggregated age distribution matrix:
plot_age_distribution(aggr_age)
```

`aggregate_contact_matrix`

Aggregate a contact matrix

Description

Function to aggregate a contact matrix according to user-defined age groups.

Usage

```
aggregate_contact_matrix(object, lookup_table, age_distr)
```

Arguments

`object`	data.frame; a contact matrix. See contact_matrix.
`lookup_table`	data.frame; a user-defined data.frame which maps the sixteen 5-year age bands to a new set of age bands.
`age_distr`	data.frame; the aggregated age distribution. See aggregate_contact_matrix.

Value

An object of class *data.frame*.

Examples

```
# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                           Mapping = c(rep("0-39",  8),
                                       rep("40-64", 5),
                                       rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Plot the contact matrix:
plot_contact_matrix(aggr_cm)
```

`aggregate_ifr_react` *Aggregate the Infection Fatality Ratio*

Description

Function to aggregate the age-specific Infection Fatality Ratio (IFR) estimates reported by the REACT-2 large-scale community study of SARS-CoV-2 seroprevalence in England according to user-defined age groups.

Usage

```
aggregate_ifr_react(x, user_AgeGrp, data_cases)
```

Arguments

`x`	data.frame; an age distribution matrix. See age_distribution.
`user_AgeGrp`	vector; a user-defined vector which maps the four age groups considered in REACT-2 to a new set of age groups.
`data_cases`	data.frame; time series dataset containing the age-stratified infection counts. See age_specific_infection_counts.

Value

A list of two data frames that contains the aggregated IFR estimates.

References

Ward, H., Atchison, C., Whitaker, M. et al. (2021). SARS-CoV-2 antibody prevalence in England following the first peak of the pandemic. Nature Communications 12, 905

Examples

```
# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

age_mapping <- c(rep("0-39",  8),
                 rep("40-64", 5),
                 rep("65+",   3))

data(age_specific_infection_counts)

# Aggregate the IFR:
aggr_age_ifr <- aggregate_ifr_react(age_distr, age_mapping, age_specific_infection_counts)
```

`contact_matrices` *Contact matrices per country*

Description

A list of 16 by 16 contact matrices for 177 countries. Row i of a column j of a contact matrix corresponds to the number of contacts made by an individual in group i with an individual in group j.

Usage

```
data(contact_matrices)
```

Format

A list of 16 by 16 dataframes for 177 countries.

Value

A list object of 16 by 16 dataframes for 177 countries.

References

Prem, K., van Zandvoort, K., Klepac, P. et al (2020). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

`contact_matrix` *Country-specific contact matrix*

Description

A 16 by 16 contact matrix whose row i of a column j corresponds to the number of contacts made by an individual in group i with an individual in group j.

Usage

```
contact_matrix(country)
```

Arguments

`country`	A character indicating the country identifier. See country_contact_matrices.

Value

An object of class "data.frame".

References

Prem, K., van Zandvoort, K., Klepac, P. et al (2020). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

Examples

```
conmat <- contact_matrix(country = "GRC")
```

`countries_un` *Names of countries with an available age distribution*

Description

Function to extract the names of the countries whose discrete age distribution is available by the United Nations.

Usage

```
countries_un()
```

Value

A character vector that contains the full names of 201 countries/areas.

References

United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019, Online Edition. Rev. 1.

Prem, K., van Zandvoort, K., Klepac, P. et al (2017). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

Examples

```
countries_un()
```

`country_contact_matrices` *Names of countries for which a contact matrix is available*

Description

Function to extract the names of the countries whose projected contact matrix is available.

Usage

```
country_contact_matrices()
```

Value

A character vector of length 177 with the IDs of each of the 177 geographical regions.

References

Prem, K., van Zandvoort, K., Klepac, P. et al (2017). Projecting contact matrices in 177 geographical regions: an update and comparison with empirical data for the COVID-19 era. medRxiv 2020.07.22.20159772; doi: https://doi.org/10.1101/2020.07.22.20159772

Examples

```
country_contact_matrices()
```

`itd_distribution` *Distribution of the time between infection and death*

Description

Function to discretize the infection-to-death distribution

Usage

```
itd_distribution(
  ts_length,
  gamma_mean = 24.19231,
  gamma_cv = 0.3987261,
  gamma_shape = 6.29,
  gamma_rate = 0.26
)
```

Arguments

`ts_length`	integer; time from infection to death in days.
`gamma_mean`	numeric; mean of a gamma distribution, for a given shape and rate. See also `GammaDist`.
`gamma_cv`	numeric; coefficient of variation of a gamma distribution, for a given shape and rate. See also `GammaDist`.
`gamma_shape`	numeric; shape parameter of a gamma distribution. See also `GammaDist`.
`gamma_rate`	numeric; rate parameter of a gamma distribution. See also `GammaDist`.

Value

A vector of length *ts_length*.

References

Flaxman et al (2020). Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. Nature, 584, 257-261.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)
```

`plot_age_distribution` *Bar plot of the age distribution*

Description

Bar plot of the age distribution

Usage

```
plot_age_distribution(x)
```

Arguments

`x`	data.frame; the age distribution matrix. See age_distribution and aggregate_age_distribution.

Value

A ggplot object that can be further customized using the **ggplot2** package.

References

United Nations, Department of Economic and Social Affairs, Population Division (2019). World Population Prospects 2019, Online Edition. Rev. 1.

Examples

```
# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

plot_age_distribution(age_distr)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                           Mapping = c(rep("0-39",  8),
                                       rep("40-64", 5),
                                       rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Plot the aggregated age distribution matrix:
plot_age_distribution(aggr_age)
```

`plot_contact_matrix` *Contact matrix heatmap*

Description

Contact matrix heatmap

Usage

```
plot_contact_matrix(x)
```

Arguments

`x`	data.frame; a contact matrix. See contact_matrix.

Value

A ggplot object that can be further customized using the **ggplot2** package.

Examples

```
# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

plot_contact_matrix(conmat)
```

`plot_posterior_cm` *Plot the posterior contact matrix*

Description

Plot the posterior contact matrix

Usage

```
plot_posterior_cm(object, y_data, ...)
```

Arguments

`object`	An object of class `stanigbm`. See `stan_igbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.
`...`	Optional arguments passed to `theme`.

Value

A `grid.arrange` object which can be further customised using the **gridExtra** package.

References

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: https://arxiv.org/abs/2211.15229

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")
```

```
# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

# Visualise the posterior distribution of the random contact matrix:
plot_posterior_cm(object = igbm_fit,
                  y_data = age_specific_mortality_counts)
```

`plot_posterior_infections` *Plot the posterior distribution of the infection counts*

Description

Plot the posterior distribution of the infection counts

Usage

```
plot_posterior_infections(
  object,
  type = c("age-specific", "aggregated"),
  xlab = NULL,
  ylab = NULL,
  ...
)
```

Arguments

`object`	A dataframe from `posterior_infections`.
`type`	character; Plot the output for the 'age-specific' infection counts or the 'aggregated' infections.
`xlab`	character; title of x-axis.
`ylab`	character; title of y-axis.
`...`	Optional arguments passed to `scale_x_date`.

Value

A `ggplot` object which can be further customised using the **ggplot2** package.

See Also

`posterior_infections`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)
```

```
# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_inf_summary <- posterior_infections(object = igbm_fit,
                                         y_data = age_specific_mortality_counts)

# Visualise the posterior distribution of the infection counts:
plot_posterior_infections(post_inf_summary, type = "age-specific")
plot_posterior_infections(post_inf_summary, type = "aggregated")
```

`plot_posterior_mortality` *Plot the posterior distribution of the mortality counts*

Description

Plot the posterior distribution of the mortality counts

Usage

```
plot_posterior_mortality(
  object,
  type = c("age-specific", "aggregated"),
  xlab = NULL,
  ylab = NULL,
  ...
)
```

Arguments

`object`	A dataframe from `posterior_mortality`.
`type`	character; Plot the output for the 'age-specific' mortality counts or the 'aggregated' mortality counts.
`xlab`	character; title of x-axis.
`ylab`	character; title of y-axis.
`...`	Optional arguments passed to `scale_x_date`.

Value

A `ggplot` object which can be further customised using the **ggplot2** package.

See Also

`posterior_mortality`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
```

```
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)
```

```
post_mortality_summary <- posterior_mortality(object = igbm_fit,
                                              y_data = age_specific_mortality_counts)

# Visualise the posterior distribution of the mortality counts:
plot_posterior_mortality(post_mortality_summary, type = "age-specific")
plot_posterior_mortality(post_mortality_summary, type = "aggregated")
```

`plot_posterior_rt` *Plot the estimated effective reproduction number trajectory*

Description

Plot the estimated effective reproduction number trajectory

Usage

```
plot_posterior_rt(object, xlab = NULL, ylab = NULL, ...)
```

Arguments

`object`	A data frame from `posterior_rt`.
`xlab`	character; Title of x-axis.
`ylab`	character; Title of y-axis.
`...`	Optional arguments passed to `scale_x_date` and `theme`.

Value

A `ggplot` object which can be further customised using the **ggplot2** package.

See Also

`posterior_rt`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
```

```
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_rt_summary <- posterior_rt(object                      = igbm_fit,
                              y_data                    = age_specific_mortality_counts,
```

```
                                    age_distribution_population = aggr_age,
                                    infectious_period           = 4)

# Visualise the posterior distribution of the effective reproduction number:
plot_posterior_rt(post_rt_summary)
```

`plot_posterior_transmrate`

Plot the estimated age-specific transmission rate

Description

Plot the estimated age-specific transmission rate

Usage

```
plot_posterior_transmrate(object, xlab = NULL, ylab = NULL, ...)
```

Arguments

`object`	A dataframe from `posterior_transmrate`.
`xlab`	character; Title of x-axis.
`ylab`	character; Title of y-axis.
`...`	Optional arguments passed to `scale_x_date`.

Value

A `ggplot` object which can be further customised using the **ggplot2** package.

See Also

`posterior_transmrate`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
```

```
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_transmrate_summary <- posterior_transmrate(object = igbm_fit,
                                                y_data = age_specific_mortality_counts)
```

```
# Visualise the posterior distribution of the age-specific transmission rate:
plot_posterior_transmrate(post_transmrate_summary)
```

`posterior_infections` *Summarize the posterior distribution of the infection counts*

Description

Summarize the posterior distribution of the infection counts

Usage

```
posterior_infections(object, y_data)
```

Arguments

`object`	An object of class `stanigbm`. See `stan_igbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.

Value

A named list with elements `Age_specific` and `Aggregated` which can be visualized using `plot_posterior_infections`.

References

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: https://arxiv.org/abs/2211.15229

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)
```

```
# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_inf_summary <- posterior_infections(object = igbm_fit,
                                         y_data = age_specific_mortality_counts)

# Visualise the posterior distribution of the infection counts:
plot_posterior_infections(post_inf_summary, type = "age-specific")
plot_posterior_infections(post_inf_summary, type = "aggregated")
```

`posterior_mortality` *Summarize the posterior distribution of the mortality counts*

Description

Summarize the posterior distribution of the mortality counts

Usage

```
posterior_mortality(object, y_data)
```

Arguments

`object`	An object of class `stanigbm`. See `stan_igbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.

Value

#' A named list with elements `Age_specific` and `Aggregated` which can be visualised using `plot_posterior_mortality`.

References

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: `https://arxiv.org/abs/2211.15229`

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"   , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
```

```
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_mortality_summary <- posterior_mortality(object = igbm_fit,
                                              y_data = age_specific_mortality_counts)

# Visualise the posterior distribution of the mortality counts:
plot_posterior_mortality(post_mortality_summary, type = "age-specific")
plot_posterior_mortality(post_mortality_summary, type = "aggregated")
```

`posterior_rt` *Estimate the effective reproduction number with the next generation matrix approach*

Description

Estimate the effective reproduction number with the next generation matrix approach

Usage

```
posterior_rt(object, y_data, age_distribution_population, infectious_period)
```

Arguments

`object`	An object of class `stanigbm`. See `stan_igbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.
`age_distribution_population`	data.frame; the age distribution of a given population. See `aggregate_age_distribution`.
`infectious_period`	integer; length of infectious period in days. Must be >=1.

Value

A data.frame which can be visualised using `plot_posterior_rt`.

References

Diekmann, O., Heesterbeek, J., and Roberts, M. (2010). The construction of next-generation matrices for compartmental epidemic models. *J. R. Soc. Interface*, 7, 873—885.

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: https://arxiv.org/abs/2211.15229

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                           Mapping = c(rep("0-39",  8),
                                       rep("40-64", 5),
                                       rep("65+"  , 3)))
```

```
# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_rt_summary <- posterior_rt(object                      = igbm_fit,
                              y_data                    = age_specific_mortality_counts,
                                age_distribution_population = aggr_age,
```

```
                                  infectious_period           = 4)

# Visualise the posterior distribution of the effective reproduction number:
plot_posterior_rt(post_rt_summary)
```

`posterior_transmrate` *Estimate the age-specific transmission rate*

Description

Estimate the age-specific transmission rate

Usage

```
posterior_transmrate(object, y_data)
```

Arguments

`object`	An object of class `stanigbm`. See `stan_igbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.

Value

A data.frame which can be visualised using `plot_posterior_transmrate`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
```

```
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

post_transmrate_summary <- posterior_transmrate(object = igbm_fit,
                                                y_data = age_specific_mortality_counts)

# Visualise the posterior distribution of the age-specific transmission rate:
plot_posterior_transmrate(post_transmrate_summary)
```

`priors` *Prior distributions and options*

Description

The functions described on this page are used to specify the prior-related arguments of the modeling functions in the **Bernadette** package.

The default priors used in the **Bernadette** modeling functions are intended to be *weakly informative*. For many applications the defaults will perform well, but prudent use of more informative priors is encouraged. Uniform prior distributions are possible (e.g. by setting `stan_igbm`'s `prior` argument to `NULL`) but, unless the data is very strong, they are not recommended and are *not* non-informative, giving the same probability mass to implausible values as plausible ones.

Usage

```
normal(location = 0, scale = NULL)

student_t(df = 1, location = 0, scale = NULL)

cauchy(location = 0, scale = NULL)

gamma(shape = 2, rate = 1)

exponential(rate = 1)
```

Arguments

`location`	Prior location. In most cases, this is the prior mean, but for `cauchy` (which is equivalent to `student_t` with `df=1`), the mean does not exist and `location` is the prior median. The default value is 0.
`scale`	Prior scale. The default depends on the family (see **Details**).
`df`	Degrees of freedom.
`shape`	Prior shape for the gamma distribution. Defaults to `2`.
`rate`	Prior rate for the exponential distribution. Defaults to `1`. For the exponential distribution, the rate parameter is the *reciprocal* of the mean.

Details

The details depend on the family of the prior being used:

Student t family: Family members:

- `normal(location, scale)`
- `student_t(df, location, scale)`
- `cauchy(location, scale)`

As the degrees of freedom approaches infinity, the Student t distribution approaches the normal distribution and if the degrees of freedom are one, then the Student t distribution is the Cauchy distribution. If `scale` is not specified it will default to 2.5.

Value

A named list to be used internally by the **Bernadette** model fitting functions.

See Also

The vignette for the **Bernadette** package discusses the use of some of the supported prior distributions.

Examples

```
# Age-specific mortality/incidence count time series:
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Can assign priors to names:
N05      <- normal(0, 5)
Gamma22  <- gamma(2,2)
igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      likelihood_variance_type    = "quadratic",
                      prior_volatility            = N05,
                      prior_nb_dispersion         = Gamma22,
                      algorithm_inference         = "optimizing")
```

`stan_igbm` *Bayesian diffusion-driven multi-type epidemic models via Stan*

Description

A Bayesian evidence synthesis approach to model the age-specific transmission dynamics of COVID-19 based on daily age-stratified mortality counts. The temporal evolution of transmission rates in populations containing multiple types of individual is reconstructed via independent diffusion processes assigned to the key epidemiological parameters. A suitably tailored Susceptible-Exposed-Infected-Removed (SEIR) compartmental model is used to capture the latent counts of infections and to account for fluctuations in transmission influenced by phenomena like public health interventions and changes in human behaviour.

Usage

```
stan_igbm(
  y_data,
  contact_matrix,
  age_distribution_population,
  age_specific_ifr,
  itd_distr,
  incubation_period = 3,
  infectious_period = 4,
  likelihood_variance_type = c("quadratic", "linear"),
  ecr_changes = 1,
  prior_scale_x0 = 1,
  prior_scale_x1 = 1,
  prior_scale_contactmatrix = 0.05,
  pi_perc = 0.1,
  prior_volatility = normal(location = 0, scale = 2.5),
  prior_nb_dispersion = gamma(shape = 2, rate = 1),
  algorithm_inference = c("sampling", "optimizing", "meanfield", "fullrank"),
  nBurn = 500,
  nPost = 500,
  nThin = 1,
  adapt_delta = 0.8,
  max_treedepth = 14,
  seed = 1,
  ...
)

stan_igbm.fit(
  standata_preprocessed,
  prior_volatility,
  prior_nb_dispersion,
```

```
  algorithm,
  nBurn,
  nPost,
  nThin,
  adapt_delta = NULL,
  max_treedepth = NULL,
  seed,
  ...
)
```

Arguments

`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.
`contact_matrix`	matrix; a squared matrix representing the the number of contacts between age groups.
`age_distribution_population`	data.frame; the age distribution of a given population. See `aggregate_age_distribution`.
`age_specific_ifr`	data.frame; time-varying age-specific infection-fatality ratio. See `aggregate_ifr_react`.
`itd_distr`	vector; Infection-to-death distribution. A vector of length *ts_length*.
`incubation_period`	integer; length of incubation period in days. Must be >=1.
`infectious_period`	integer; length of infectious period in days. Must be >=1.
`likelihood_variance_type`	integer; If 0, the variance of the over-dispersed count model is a quadratic function of the mean; if 1, the variance of the over-dispersed count model is a linear function of the mean.
`ecr_changes`	integer; between 1 and 7, defaults to 1. Expresses the number of changes of the effective contact rate during the course of 7 days.
`prior_scale_x0`	double; scale parameter of a Normal prior distribution assigned to the age-specific log(transmissibility) at time $t = 0$.
`prior_scale_x1`	double; scale parameter of a Normal prior distribution assigned to the age-specific log(transmissibility) at time $t = 1$.
`prior_scale_contactmatrix`	double; defaults to 0.05. A positive number that scales the informative Normal prior distribution assigned to the random contact matrix.
`pi_perc`	numeric; between 0 and 1. It represents the proportion of Exposed individuals in each age group of a given population at time $t = 0$. while the rest $100 * (1 - pi_perc)$ remain Susceptible.
`prior_volatility`	Prior distribution for the volatility parameters of the age-specific diffusion processes. `prior_volatility` can be a call to `exponential` to use an exponential distribution, `gamma` to use a Gamma distribution or one of `normal`, `student_t` or `cauchy` to use a half-normal, half-t, or half-Cauchy prior. See `priors` for details on these functions.

`prior_nb_dispersion`	Prior distribution for the dispersion parameter `phi` of the over-dispersed count model. Same options as for `prior_volatility`.
`algorithm_inference`	One of the sampling algorithms that are implemented in Stan. See `stan`.
`nBurn`	integer; number of burn-in iterations at the beginning of an MCMC run. See `sampling`.
`nPost`	integer; number of MCMC iterations after burn-in. See `sampling`.
`nThin`	integer; a positive integer specifying the period for saving samples. The default is 1, which is usually the recommended value. See `sampling`.
`adapt_delta`	double; between 0 and 1, defaults to 0.8. See `stan`.
`max_treedepth`	integer; defaults to 14. See `stan`.
`seed`	integer; seed for the random number generator. See `set.seed`.
`...`	Additional arguments, to be passed to lower-level functions.
`standata_preprocessed`	A named list providing the data for the model. See `sampling`.
`algorithm`	See `algorithm` in stan_igbm.

Details

The `stan_igbm` function performs full Bayesian estimation (if `algorithm_inference` is `"sampling"`) via MCMC. The Bayesian model adds priors (i) on the diffusion processes used to express the time-varying transmissibility of the virus, the probability that a contact between an infectious person in age group alpha and a susceptible person in age group alpha leads to transmission at time t and (ii) on a random contact matrix which represents the average number of contacts between individuals of age group alpha and age group alpha' The `stan_igbm` function calls the workhorse `stan_igbm.fit` function.

Value

An object of class *stanigbm* representing the fitted results. Slot mode for this object indicates if the sampling is done or not.

An object of S4 class *stanfit* representing the fitted results. Slot mode for this object indicates if the sampling is done or not.

References

Bouranis, L., Demiris, N. Kalogeropoulos, K. and Ntzoufras, I. (2022). Bayesian analysis of diffusion-driven multi-type epidemic models with application to COVID-19. arXiv: `https://arxiv.org/abs/2211.15229`

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)
```

```
# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
```

```
                          nThin                         = 1,
                          chains                        = chains,
                          adapt_delta                   = 0.6,
                          max_treedepth                 = 14,
                          seed                          = 1)

# print_summary <- summary(object = igbm_fit, y_data = age_specific_mortality_counts)$summary
```

`summary.stanigbm` *Summary of stanigbm posterior output*

Description

This function summarizes the MCMC output for `stanigbm` objects.

Usage

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

Arguments

`object`	An R object of class `stanigbm`.
`y_data`	data.frame; age-specific mortality counts in time. See `data(age_specific_mortality_counts)`.
`...`	Additional arguments, to be passed to lower-level functions.

Value

A named list with elements `summary` and `c_summary`, which contain summaries for for all Markov chains merged and individual chains, respectively. See `stanfit-method-summary`.

Examples

```
# Age-specific mortality/incidence count time series:
data(age_specific_mortality_counts)
data(age_specific_cusum_infection_counts)

# Import the age distribution for Greece in 2020:
age_distr <- age_distribution(country = "Greece", year = 2020)

# Lookup table:
lookup_table <- data.frame(Initial = age_distr$AgeGrp,
                          Mapping = c(rep("0-39",  8),
                                      rep("40-64", 5),
                                      rep("65+"  , 3)))

# Aggregate the age distribution table:
```

```
aggr_age <- aggregate_age_distribution(age_distr, lookup_table)

# Import the projected contact matrix for Greece:
conmat <- contact_matrix(country = "GRC")

# Aggregate the contact matrix:
aggr_cm <- aggregate_contact_matrix(conmat, lookup_table, aggr_age)

# Aggregate the IFR:
ifr_mapping <- c(rep("0-39", 8), rep("40-64", 5), rep("65+", 3))

aggr_age_ifr <- aggregate_ifr_react(age_distr, ifr_mapping, age_specific_cusum_infection_counts)

# Infection-to-death distribution:
ditd <- itd_distribution(ts_length  = nrow(age_specific_mortality_counts),
                         gamma_mean = 24.19231,
                         gamma_cv   = 0.3987261)

# Posterior sampling:

rstan::rstan_options(auto_write = TRUE)
chains <- 1
options(mc.cores = chains)

igbm_fit <- stan_igbm(y_data                      = age_specific_mortality_counts,
                      contact_matrix              = aggr_cm,
                      age_distribution_population = aggr_age,
                      age_specific_ifr            = aggr_age_ifr[[3]],
                      itd_distr                   = ditd,
                      incubation_period           = 3,
                      infectious_period           = 4,
                      likelihood_variance_type    = "linear",
                      ecr_changes                 = 7,
                      prior_scale_x0              = 1,
                      prior_scale_x1              = 1,
                      prior_scale_contactmatrix   = 0.05,
                      pi_perc                     = 0.1,
                      prior_volatility            = normal(location = 0, scale = 1),
                      prior_nb_dispersion         = exponential(rate = 1/5),
                      algorithm_inference         = "sampling",
                      nBurn                       = 10,
                      nPost                       = 30,
                      nThin                       = 1,
                      chains                      = chains,
                      adapt_delta                 = 0.6,
                      max_treedepth               = 14,
                      seed                        = 1)

# print_summary <- summary(object = igbm_fit, y_data = age_specific_mortality_counts)$summary
```

Index