Package ‘hhdynamics’

March 26, 2026

Type Package

Title Fitting Household Transmission Model to Estimate Household Transmission Dynamics of Influenza

Version 1.3.2

Maintainer Tim Tsang <`timkltsang@gmail.com`>

Description A Bayesian household transmission model to estimate household transmission dynamics, with accounting for infection from community and tertiary cases.

License GPL (>= 2)

Encoding UTF-8

LazyData TRUE

URL `https://github.com/timktsang/hhdynamics`

BugReports `https://github.com/timktsang/hhdynamics/issues`

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.9), RcppParallel

LinkingTo Rcpp, RcppArmadillo, RcppParallel

SystemRequirements GNU make

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation yes

Author Tim Tsang [aut, cre] (ORCID: <`https://orcid.org/0000-0001-5037-6776`>)

Repository CRAN

Date/Publication 2026-03-26 10:00:15 UTC

Contents

`coef.hhdynamics_fit` *Extract model coefficients from hhdynamics_fit*

Description

Returns named vector of posterior means for all estimated parameters.

Usage

```
## S3 method for class 'hhdynamics_fit'
coef(object, ...)
```

Arguments

`object`	An object of class `hhdynamics_fit`.
`...`	Additional arguments (unused).

Value

A named numeric vector of posterior means.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
coef(fit)
```

`create_wide_data` *Create a wide format of household data*

Description

Reshapes long-format household data (one row per individual) into wide format (one row per household) for the C++ MCMC backend. Also constructs design matrices for infectivity and susceptibility covariates.

Usage

```
create_wide_data(input, inf_factor, sus_factor)
```

Arguments

`input`	The input data, in long format. Must contain columns: hhID, member, inf, onset, size, end.
`inf_factor`	Formula for factors affecting infectivity (e.g. `~sex` or `~sex + age`). Use `NULL` for no factors.
`sus_factor`	Formula for factors affecting susceptibility (e.g. `~age`). Use `NULL` for no factors.

Details

The wide-format output has the following column structure per household:

- Columns 1–5: hhID, size, size (duplicate), index onset, end date
- Columns 6+: per-member blocks of (inf, onset, random effect placeholder, infectivity covariates, susceptibility covariates)

Missing values (members not present in a household) are filled with -1, which the C++ code treats as a sentinel. Missing factor covariates are coded as -99 (a distinct sentinel) and will be imputed during MCMC.

Value

A list with 5 elements: (1) data frame in wide format, (2) number of infectivity parameters, (3) number of susceptibility parameters, (4) integer vector mapping each dummy column to its factor group, (5) integer vector of factor levels per dummy column.

See Also

`household_dynamics` for the high-level interface, `run_MCMC` for the low-level MCMC function.

Examples

```
wide_data <- create_wide_data(inputdata, ~sex, ~age)
```

`household_dynamics` *Fit a household transmission model via MCMC*

Description

The main function to fit the household transmission model to data. Estimates the daily probability of infection from the community, the probability of person-to-person transmission within households, and effects of covariates on infectivity and susceptibility.

Usage

```
household_dynamics(
  input,
  inf_factor = NULL,
  sus_factor = NULL,
  SI = NULL,
  n_iteration = 15000,
  burnin = 5000,
  thinning = 1,
  estimate_SI = FALSE
)
```

Arguments

`input`	The input data, in long format (each row is an individual). Required columns: **hhID** Household identifier. **member** Member index (0 = index case, 1+ = contacts). **size** Number of individuals in the household. **end** End date of follow-up for that individual. **inf** Infection status (1 = infected, 0 = not). Index cases must have `inf = 1`. **onset** Onset time of symptoms.
`inf_factor`	Formula for factors affecting infectivity (e.g. `~sex` or `~sex + age`). Use `NULL` for no factors. Default is `NULL`.
`sus_factor`	Formula for factors affecting susceptibility (e.g. `~age`). Use `NULL` for no factors. Default is `NULL`.
`SI`	The mass function of the serial interval distribution. Must be a numeric vector of length 14 summing to approximately 1. Defaults to the bundled influenza serial interval from Tsang et al. (2014). Not used when `estimate_SI = TRUE` (SI is estimated from data via Weibull parameterization).

`n_iteration`	Total number of MCMC iterations. Default is 15000.
`burnin`	Number of initial iterations to discard. Default is 5000.
`thinning`	Thinning interval for posterior samples. Default is 1.
`estimate_SI`	Logical. If `TRUE`, jointly estimate the serial interval distribution as a Weibull(shape, scale) alongside other model parameters. Two additional parameters (`si_shape`, `si_scale`) are added to the MCMC. Priors: shape ~ Uniform(0.1, 10), scale ~ Uniform(0.1, 20). Default is `FALSE`.

Details

The model assumes that each household contact can be infected either from the community (at a constant daily rate) or from an infected household member (with probability governed by the serial interval distribution). Tertiary transmission within households is accounted for. Infection times for non-index cases are treated as latent variables and imputed via data augmentation during MCMC.

Covariate effects on infectivity and susceptibility enter multiplicatively on the log scale. The `summary()` method reports exponentiated estimates for interpretation as relative risks.

Missing covariate imputation: Factor covariates with missing values (`NA`) are automatically imputed during MCMC via Bayesian data augmentation, using a uniform categorical prior over factor levels. Only factor covariates are supported; continuous covariates with `NA` will produce an error. Interaction terms with missing data are not supported.

The returned `hhdynamics_fit` object stores the full MCMC output, enabling custom convergence diagnostics and post-processing. Key fields:

`$samples` Posterior parameter samples (post-burnin, thinned). Columns named by parameter.

`$log_likelihood` Log-likelihood trace for convergence assessment (full chain).

`$acceptance` Per-parameter acceptance rates from the Metropolis-Hastings sampler.

`$imputed_data` Final imputed dataset (wide format) with augmented infection times.

Value

An object of class `print.hhdynamics_fit{hhdynamics_fit}`. Use `summary()` to get parameter estimates, `print()` for a brief overview, and `coef()` for posterior means. When `estimate_SI = TRUE`, the output includes `si_shape` and `si_scale` parameters.

See Also

`simulate_data` for simulating from the model, `create_wide_data` for data preparation, `run_MCMC` for the low-level MCMC interface.

Examples

```
data(inputdata)

# Fit with default flu SI
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
print(fit)
```

```
summary(fit)
coef(fit)
```

`inputdata` *Example of input data*

Description

This is an example of the input data used in the `hhdynamics` function. This data frame illustrates the format of the input data. This is a simulated data.

Usage

```
data(inputdata)
```

Format

A example data with 8 variables, where each row represents an individual. For user's data, more predictors could be added by adding columns. For categorical variables, it should be specificed as factor, by using as.factor() function. The date is the dataset is an integer, by selecting a reference date as day 1. In the example dataset, 2008-01-01 is day 1.

hhID	The household id
member	The member id in a households, use 0 to index, 1 and so on as household contact
size	The number of individuals in the household
end	The end date of follow-up of that individual
inf	The infection status of the member. By defintion, index must be infected
onset	The onset time of infected individual
age_group	The age group of individual. 0: 0-19, 1: 20-64, 2: 65 or above
sex	The sex of the individual. 0: Female, 1: Male

Examples

```
data(inputdata)
str(inputdata)
head(inputdata)
table(inputdata$inf)
```

`para` *Example of parameter vector for the main model*

Description

This is an example of the parameter vector for the main model used in the `hhdynamics` function. This vector specifies the format of the parameter vector for the main model.

Usage

```
data(para)
```

Format

A vector with 7 elements, where each of them is a model parameter:

- **element 1** the random effect of individual infectivity (not available in this version. Models assumed no indiviudal heterogeneity after accounting for factors affecting infectivity)
- **element 2** the probability of infection from the community
- **element 3** the probability of person-to-person transmission in households
- **element 4** the parameter of the relationship between the number of household contacts and transmission (default 0 in this version)
- **element 5** the parameter for infectivity of male (Reference group: male)
- **element 6** the relative susceptibility of age group 1 (Reference group: age group 0)
- **element 7** the relative susceptibility of age group 2 (Reference group: age group 0)

See Also

Other example_data: `SI`

Examples

```
data(para)
print(para)
names(para)
```

`plot.hhdynamics_fit` *Plot method for hhdynamics_fit objects*

Description

Plot method for hhdynamics_fit objects

Usage

```
## S3 method for class 'hhdynamics_fit'
plot(x, type = "diagnostics", ...)
```

Arguments

`x`	An object of class `hhdynamics_fit`.
`type`	Type of plot: `"diagnostics"` (default), `"transmission"`, `"attack_rate"`, or `"covariates"`.
`...`	Additional arguments passed to the underlying plot function.

Value

Invisible NULL.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
plot(fit)                         # diagnostics (default)
plot(fit, type = "transmission")   # transmission probability curve
```

`plot_attack_rate` *Plot secondary attack rates*

Description

Forest plot showing observed secondary attack rates (SAR) with Wilson score 95% confidence intervals. Supports overall SAR, stratification by one or more covariates, and combinations thereof. Variable names are used as bold section headers; strata are indented below. The layout mirrors `plot_covariates()`: labels on the left, point estimates and CI bars in the middle, and `n/N` counts plus SAR percentages on the right.

Usage

```
plot_attack_rate(
  fit,
  by = NULL,
  include_overall = FALSE,
  labels = NULL,
  file = NULL,
  width = 8,
  height = NULL,
  xlim = NULL,
  cex = 0.85,
  ...
)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`by`	Formula, character string, or a *list* of formulas naming the stratification variable(s). Examples: `~age`, `"age"`, `list(~sex, ~age)`. Default: `NULL` (overall SAR only).
`include_overall`	Logical. When `TRUE`, an "Overall" row is prepended even when `by` is specified. Default: `FALSE`.
`labels`	Optional named list of custom display labels for variables and their levels. Each element is a list with `name` (display name for the section header) and/or `levels` (character vector of level labels in the same order as `sort(unique(variable))`). Names must match variable names in the data. Example: `list(age = list(name = "Age Group", levels = c("0-5", "6-17", "18+")))`.
`file`	Optional file path for PDF output. Height is auto-calculated from the number of rows. Default: `NULL` (current device).
`width`	PDF width in inches. Default: 8.
`height`	PDF height in inches. Default: `0.45 * n_rows + 1.8`.
`xlim`	Numeric vector of length 2 for the x-axis range (probability scale). Default: auto-determined from the data.
`cex`	Character expansion factor. Default: 0.85.
`...`	Additional graphical parameters passed to `plot()`.

Value

Invisible data frame of the estimate rows (Stratum, N_contacts, N_infected, SAR, Lower, Upper).

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
```

```
# Overall only
plot_attack_rate(fit)

# Stratified by age with section header
plot_attack_rate(fit, by = ~age)

# Combined: overall + age + sex in one figure
plot_attack_rate(fit, by = list(~sex, ~age), include_overall = TRUE,
  labels = list(sex = list(name = "Sex", levels = c("Male", "Female")),
               age = list(name = "Age Group", levels = c("0-5", "6-17", "18+"))))
```

`plot_covariates` *Forest plot of covariate effects*

Description

Produces a forest plot showing estimated relative risks for covariate effects on susceptibility and infectiousness. Covariates are grouped by variable with bold headers, reference categories labeled, alternating row shading, and estimate text with credible intervals on the right.

Usage

```
plot_covariates(
  fit,
  probs = c(0.025, 0.975),
  labels = NULL,
  file = NULL,
  width = 11,
  height = NULL,
  xlim = NULL,
  xlab_left = "Lower Risk",
  xlab_right = "Higher Risk",
  cex = 0.85,
  ...
)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`probs`	Numeric vector of length 2 for credible interval bounds. Default: `c(0.025, 0.975)` (95% CrI).
`labels`	Optional named list of custom labels for covariates. Each element is a list with `name` (display name for the variable header) and `levels` (character vector of level labels, including the reference level first). Names must match variable names in the formula. Example: `list(sex = list(name = "Sex", levels = c("Male", "Female")))`.

`file`	Optional file path for PDF output. When provided, a PDF is created with auto-calculated width and height based on the number of rows. Default: `NULL` (plot to current device).
`width`	PDF width in inches. Default: 11. Only used when `file` is not `NULL`.
`height`	PDF height in inches. Default: auto-calculated as `0.45 * n_rows + 1.8`. Only used when `file` is not `NULL`.
`xlim`	Numeric vector of length 2 for the x-axis range on the natural scale. Default: auto-determined from the credible intervals.
`xlab_left`	Label for the left direction arrow. Default: `"Lower Risk"`.
`xlab_right`	Label for the right direction arrow. Default: `"Higher Risk"`.
`cex`	Character expansion factor. Default: 0.85.
`...`	Additional graphical parameters passed to `plot()`.

Details

When `file` is provided, the plot is saved to a PDF with dimensions automatically calculated from the number of covariate rows. When `file` is `NULL`, the plot is drawn to the current graphics device.

Value

Invisible NULL. Called for its side effect of producing a plot.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
plot_covariates(fit)

# Save to PDF with auto-sized dimensions
plot_covariates(fit, file = tempfile(fileext = ".pdf"),
  labels = list(sex = list(name = "Sex", levels = c("Male", "Female")),
                age = list(name = "Age Group", levels = c("0-5", "6-17", "18+"))))
```

`plot_diagnostics` *MCMC diagnostic plots*

Description

Produces trace plots and posterior density plots for each estimated parameter. Trace plots show the MCMC chain with the posterior mean (red dashed line). Density plots show the marginal posterior with 95% credible interval bounds (blue dashed lines). Set `show_ess = TRUE` to annotate with effective sample size.

Usage

```
plot_diagnostics(fit, params = NULL, show_ess = FALSE)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`params`	Optional character vector of parameter names to plot. If `NULL` (default), all estimated parameters are plotted (fixed parameters like `size_param` are skipped).
`show_ess`	Logical. If `TRUE`, show the effective sample size (ESS) in the density plot title. Default: `FALSE`.

Details

Community and household parameters are shown on the probability scale (via the `1 - exp(-x)` transform).

Value

Invisible NULL. Called for its side effect of producing plots.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
plot_diagnostics(fit)
plot_diagnostics(fit, params = c("community", "household"))
```

`plot_household` *Plot household infection timeline*

Description

Visualizes the infection timeline for a single household. The index case is shown as a filled triangle, infected contacts as filled circles at their (imputed) onset times, and uninfected contacts as open circles spanning their follow-up period.

Usage

```
plot_household(fit, hh_id, col = NULL, ...)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`hh_id`	Household identifier to visualize.
`col`	Colors for infected and uninfected members. Default: `c("firebrick", "grey60")`.
`...`	Additional graphical parameters.

Value

Invisible NULL.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
plot_household(fit, hh_id = 1)
```

`plot_transmission` *Plot transmission probability over time since onset*

Description

Shows the daily probability of person-to-person transmission as a function of days since the infector's symptom onset. The serial interval distribution shapes this curve. The median and 95% credible interval are computed from the posterior samples.

Usage

```
plot_transmission(fit, hh_size = NULL, col = "steelblue", ...)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`hh_size`	Reference household size. Default: median from data.
`col`	Color for the median line and credible interval polygon. Default: `"steelblue"`.
`...`	Additional graphical parameters passed to `plot()`.

Details

When the model was fitted with `estimate_SI = TRUE`, the uncertainty band incorporates serial interval uncertainty (via the Weibull shape/scale posterior).

Value

Invisible data frame with columns Day, Median, Lower, Upper.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
plot_transmission(fit)
```

`print.hhdynamics_fit` *Print method for hhdynamics_fit*

Description

Print method for hhdynamics_fit

Usage

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

Arguments

`x`	An object of class `hhdynamics_fit`.
`...`	Additional arguments (unused).

Value

Invisibly returns `x`.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
print(fit)
```

`print.summary.hhdynamics_fit` *Print method for summary.hhdynamics_fit*

Description

Print method for summary.hhdynamics_fit

Usage

```
## S3 method for class 'summary.hhdynamics_fit'
print(x, digits = 3, ...)
```

Arguments

`x`	An object of class `summary.hhdynamics_fit`.
`digits`	Number of significant digits for printing. Default is 3.
`...`	Additional arguments (unused).

Value

Invisibly returns `x`.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
s <- summary(fit)
print(s, digits = 4)
```

`run_MCMC` *Run the MCMC for the household transmission model*

Description

Low-level function that runs the C++ MCMC sampler. Most users should use `household_dynamics` instead, which handles data preparation, validation, and returns an S3 object with named parameters.

Usage

```
run_MCMC(
  data_w,
  SI = NULL,
  n_iteration = 15000,
  burnin = 5000,
  thinning = 1,
  n_inf,
  n_sus,
  with_rm,
  factor_group = integer(0),
  n_levels_vec = integer(0),
  estimate_SI = FALSE
)
```

Arguments

`data_w`	The input data, in wide format (each row is a household), as produced by `create_wide_data`.
`SI`	The mass function of the serial interval distribution. Defaults to the bundled influenza serial interval from Tsang et al. (2014).
`n_iteration`	The number of iterations of the MCMC.
`burnin`	The number of burn-in iterations to discard.
`thinning`	The thinning interval for posterior samples.
`n_inf`	The number of parameters affecting infectivity in the model.

`n_sus`	The number of parameters affecting susceptibility in the model.
`with_rm`	Indicator if the model has a random effect on individual infectivity (1) or not (0). **Experimental:** when `with_rm = 1`, the random-effects output records one value per household (index case only), not per individual. A warning is issued at runtime.
`factor_group`	Integer vector mapping each dummy covariate column to its original factor group (from `create_wide_data`).
`n_levels_vec`	Integer vector of the number of levels for each dummy column's factor (from `create_wide_data`).
`estimate_SI`	Logical. If `TRUE`, jointly estimate Weibull shape/scale for the serial interval. Default is `FALSE`.

Details

The MCMC uses a Metropolis-Hastings algorithm with adaptive proposal variances. After 500 iterations, proposal standard deviations are set to the empirical posterior standard deviation, with multiplicative tuning based on acceptance rate (target: 20–30%). Infection times for household contacts are jointly updated at each iteration via a data augmentation step.

The parameter vector has the following structure:

1. Standard deviation of random effect on infectivity (fixed at initial value if `with_rm = 0`)
2. Rate of infection from community (log scale)
3. Rate of person-to-person transmission in households (log scale)
4. Household size parameter (currently fixed at 0)
5. Infectivity covariate effects (`n_inf` parameters)
6. Susceptibility covariate effects (`n_sus` parameters)

Value

A list with 6 elements from the C++ MCMC:

1. Posterior samples matrix (post-burnin, thinned)
2. Log-likelihood matrix (full chain, 3 columns: total, component 1, component 2)
3. Random effect samples (post-burnin). When `with_rm = 0`, returns a zero-variance placeholder matrix. When `with_rm = 1` (**experimental**), returns one random-effect value per household (index case only, not per individual).
4. Acceptance rates (per-parameter, numeric vector)
5. Infection-time update acceptance rates (iterations x household members)
6. Final imputed data matrix

See Also

`household_dynamics` for the high-level interface, `create_wide_data` for data preparation.

Examples

```
result_list <- create_wide_data(inputdata, ~sex, ~age)
data_w <- result_list[[1]]
n_inf <- result_list[[2]]
n_sus <- result_list[[3]]
mcmc_result <- run_MCMC(data_w,
  n_iteration = 1000, burnin = 500,
  thinning = 1, n_inf = n_inf, n_sus = n_sus, with_rm = 0)
```

`SI` *Example of serial interval distribution of influenza*

Description

This is an example of the serial interval distribution used in the `hhdynamics` function. This vector specifies the format of the serial interval distribution. This is estimated from Tsang et al. Association between antibody titers and protection against influenza virus infection within households. J Infect Dis. 2014 Sep 1;210(5):684-92.

Usage

```
data(SI)
```

Format

A numeric vector of length 14. Element `i` gives the probability that symptom onset occurs `i` days after the infector's onset. The vector sums to 1.

See Also

Other example_data: `para`

Examples

```
data(SI)
print(SI)
barplot(SI, names.arg = seq_along(SI),
        xlab = "Days since infector onset",
        ylab = "Probability",
        main = "Serial interval distribution (influenza)")
```

`simulate_data` *Simulate household transmission data*

Description

Generates synthetic datasets from the household transmission model for validation, power analysis, or posterior predictive checks.

Usage

```
simulate_data(
  input,
  rep_num,
  inf_factor = NULL,
  sus_factor = NULL,
  SI = NULL,
  para,
  with_rm
)
```

Arguments

`input`	The dataset in long format (same structure as for `household_dynamics`).
`rep_num`	The number of replications of the input dataset, to increase the sample size.
`inf_factor`	Formula for factors affecting infectivity (e.g. `~sex`). Use `NULL` for no factors. Default is `NULL`.
`sus_factor`	Formula for factors affecting susceptibility (e.g. `~age`). Use `NULL` for no factors. Default is `NULL`.
`SI`	The mass function of the serial interval distribution. Defaults to the bundled influenza serial interval from Tsang et al. (2014).
`para`	The parameter vector, matching the structure from `coef.hhdynamics_fit`: (1) random effect SD, (2) community rate, (3) household rate, (4) size parameter, (5+) covariate effects.
`with_rm`	Indicator if the model has a random effect on individual infectivity (1) or not (0).

Details

The simulation uses the same household structure (sizes, follow-up periods, covariate values) as the input data. The `rep_num` parameter replicates the household structure to increase sample size. Infection outcomes and onset times are simulated from the model given the parameter vector.

The output is in wide format (one row per household), matching the internal representation used by the C++ backend. Use this with `run_MCMC()` or `household_dynamics()` to verify model recovery.

Value

A simulated dataset in wide format (one row per household) based on the input parameter vectors.

See Also

`household_dynamics` for fitting the model, `coef.hhdynamics_fit` for extracting parameter estimates to use as simulation inputs.

Examples

```
data(inputdata)
data(SI)
para <- c(1, 0.01, 0.1, 0, 0.1, 0.1, 0.1)
simulated <- simulate_data(inputdata, 2, ~sex, ~age,
  SI = SI, para = para, with_rm = 0)
```

`summary.hhdynamics_fit` *Summary method for hhdynamics_fit*

Description

Computes posterior summaries (mean, 2.5%, 97.5% credible intervals) for all model parameters. For community and household parameters, reports the daily probability (via 1-exp(-x) transform). For covariate effects, additionally reports exponentiated estimates.

Usage

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

Arguments

`object`	An object of class `hhdynamics_fit`.
`...`	Additional arguments (unused).

Value

An object of class `summary.hhdynamics_fit`, which is a data frame with columns: Variable, Point.estimate, Lower.bound, Upper.bound, and optionally exp columns for covariates.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
summary(fit)
```

`table_attack_rates` *Table of secondary attack rates*

Description

Computes observed secondary attack rates (SAR) from the data, optionally stratified by a covariate. Confidence intervals use the Wilson score method.

Usage

```
table_attack_rates(fit, by = NULL)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`by`	Formula or character string naming the stratification variable (e.g. `~age` or `"age"`). Default: no stratification.

Value

A data frame with columns: Stratum, N_contacts, N_infected, SAR, Lower, Upper.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
table_attack_rates(fit)
table_attack_rates(fit, by = ~age)
```

`table_covariates` *Table of covariate effects (odds ratios)*

Description

Returns a data frame of covariate effects on infectivity and susceptibility, with exponentiated estimates interpretable as relative risks.

Usage

```
table_covariates(fit, probs = c(0.025, 0.975))
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`probs`	Numeric vector of length 2 for credible interval bounds.

Value

A data frame with columns: Covariate, Type, Estimate, Lower, Upper, exp_Estimate, exp_Lower, exp_Upper. Returns an empty data frame if no covariates were used.

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
table_covariates(fit)
```

`table_parameters` *Table of transmission parameter estimates*

Description

Returns a clean data frame of posterior parameter estimates with credible intervals and effective sample sizes. Community and household parameters are reported on the probability scale.

Usage

```
table_parameters(fit, probs = c(0.025, 0.975), show_ess = FALSE)
```

Arguments

`fit`	An object of class `hhdynamics_fit`.
`probs`	Numeric vector of length 2 for credible interval bounds. Default: `c(0.025, 0.975)` (95% CrI).
`show_ess`	Logical. If `TRUE`, include an ESS (effective sample size) column. Default: `FALSE`.

Value

A data frame with columns: Parameter, Mean, Median, Lower, Upper, Acceptance (and ESS if `show_ess = TRUE`).

Examples

```
data(inputdata)
fit <- household_dynamics(inputdata, ~sex, ~age,
  n_iteration = 1000, burnin = 500, thinning = 1)
table_parameters(fit)
```

Index