Package ‘surveil’

August 22, 2022

Title Time Series Models for Disease Surveillance

Version 0.2.1

URL https://connordonegan.github.io/surveil/,
https://github.com/ConnorDonegan/surveil/,
https://github.com/ConnorDonegan/surveil/,
https://cran.r-project.org/web//packages/surveil/

Description Fits time series models for routine disease surveillance tasks and returns probability distributions for a variety of quantities of interest, including age-standardized rates, period and cumulative percent change, and measures of health inequality. Calculates Theil's index to measure inequality among multiple groups, and can be extended to measure inequality across multiple groups nested within geographies. Inference is completed using Markov chain Monte Carlo via the Stan modeling language. The models are appropriate for count data such as disease incidence and mortality data, employing a Poisson or binomial likelihood and the first-difference (random-walk) prior for unknown risk. Optionally add a covariance matrix for multiple, correlated time series models. References: Donegan, Hughes, and Lee (2022) <doi:10.2196/34589>; Stan Development Team (2021) <https://mc-stan.org>; Theil (1972, ISBN:0-444-10378-3).

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Biarch true

Depends R (>= 3.5.0)

Imports rstantools (>= 2.1.1), methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), tidybayes (>= 3.0.0), dplyr (>= 1.0.7), rlang (>= 0.4.0), tidyr (>= 1.1.0), ggplot2 (>= 3.0.0), gridExtra (>= 2.0), scales (>= 0.4.0), ggdist (>= 3.0.0)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Suggests rmarkdown, knitr, testthat

SystemRequirements GNU make

VignetteBuilder knitr

NeedsCompilation yes

Author Connor Donegan [aut, cre] (<https://orcid.org/0000-0002-9698-5443>)

Maintainer Connor Donegan <connor.donegan@gmail.com>

Repository CRAN

Date/Publication 2022-08-22 07:50:02 UTC

R topics documented:

surveil-package *The 'surveil' package*

Description

Fits time series models for routine disease surveillance tasks and returns probability distributions for a variety of quantities of interest, including measures of health inequality, period and cumulative percent change, and age-standardized rates. Calculates Theil's index to measure inequality among multiple groups, and can be extended to measure inequality across multiple groups nested within geographies. Inference is completed using Markov chain Monte Carlo via the Stan modeling language. The models are appropriate for disease incidence and mortality data, employing a Poisson or binomial likelihood and first-difference (random-walk) prior for unknown risk, and optional covariance matrix for multiple correlated time series models.

References

Brandt P, Williams JT. Multiple time series models. Thousand Oaks, CA: SAGE Publications, 2007. ISBN:9781412906562

Clayton DG. Generalized linear mixed models. In: Gilks WR, Richardson S, Spiegelhalter DJ, editors. Markov chain Monte Carlo in practice. Boca Raton, FL: CRC Press, 1996. p. 275-302. ISBN:9780412055515

Conceicao P, Galbraith JK, Bradford P. The Theil Index in sequences of nested and hierarchic grouping structures: implications for the measurement of inequality through time, with data aggregated at different levels of industrial classification. Eastern Economic Journal 2001;27(4):491-514.

Donegan C, Hughes AE, and Lee SC (2022). Colorectal Cancer Incidence, Inequalities, and Prevention Priorities in Urban Texas: Surveillance Study With the "surveil" Software Package. *JMIR Public Health & Surveillance* 8(8):e34589. doi:10.2196/34589

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

Theil H. Statistical decomposition analysis. Amsterdam, The Netherlands: North-Holland Publishing Company, 1972. ISBN:0444103783

---

apc *Annual and cumulative percent change*

Description

Summarize annual and cumulative percent change in risk

Usage

```
apc(x)

## S3 method for class 'surveil'
apc(x)

## S3 method for class 'stand_surveil'
apc(x)
```

Arguments

`x`	A fitted `surviel` model, or standardized rates (a `stand_surveil` object).

Value

An apc (list) object containing the following data frames:

apc A data frame containing a summary of the posterior distribution for period-specific percent change. This contains the posterior mean (`apc`) 95 percent credible intervals (`lwr` and `upr` bounds).

cpc A data frame containing a summary of the posterior distribution for the cumulative percent change in risk at each time period. This contains the posterior mean (cpc) and 95 percent credible interval (lwr and upr bounds).

apc_samples MCMC samples from the posterior distribution for period percent change

cpc_samples MCMC samples from the posterior distribution for cumulative percent change

See Also

plot.apc print.apc stan_rw standardize

Examples

```
data(cancer)

 fit <- stan_rw(cancer, time = Year, group = Age,
               iter = 900) # low iter for speed only
 x <- apc(fit)
 print(x)
 plot(x, cumulative = TRUE)
```

---

cancer *US cancer incidence by age, 1999-2017*

Description

Annual cancer cases (all sites) by age group for the United States.

Usage

```
cancer
```

Format

A data frame with the following columns:

Year Year of diagnosis

Age Age group

Count Number of cancer cases

Population Age-specific population estimates

Source

United States Cancer Statistics - Incidence: 1999 - 2017, WONDER Online Database. United States Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute; 2020. Accessed at http://wonder.cdc.gov/cancer-v2017.html on Oct 6, 2021 12:38:09 PM

Examples

```
data(cancer)
head(cancer)
```

---

group_diff *Measures of pairwise inequality*

Description

Calculate pairwise measures of health inequality from a fitted surveil time series model, with credible intervals and MCMC samples. Calculates absolute and fractional rate differences (RD and population attributable risk (PAR)), rate ratios, and excess cases.

Usage

```
group_diff(x, target, reference)

## S3 method for class 'surveil'
group_diff(x, target, reference)

## S3 method for class 'list'
group_diff(x, ...)
```

Arguments

x	Either a fitted surveil time series model, or a list of two stand_surveil objects (i.e., surveil models with age-standardized rates, as returned by standardize). If x is a list of stand_surveil objects, see details below and note that the models must contain the same number of MCMC samples—to ensure this is the case, when using stan_rw set iter and chains to the same values for each of the two models.
target	The name (character string) of the disadvantaged group that is the target of inference. If x is a list of stand_surveil objects, the target argument is ignored and the first listed model will serve as the target group.
reference	The name (character string) of the reference group to which target will be compared. If x is a list of stand_surveil objects, the reference argument is ignored and the second listed model will serve as the reference group.
...	Additional arguments (not used).

Details

Comparing incidence rates:

For the following calculations, the terms reference and target refer to incidence rates for the respective groups; p is the size of the target population. (Target is the group that is the 'target' of our inferences, so that it is the numerator in rate ratios, etc.) The following measures are calculated by group_diff:

```
# rate difference
RD = target - reference
# population attributable fraction
PAR = RD/target = (RR - 1)/RR
# rate ratio
RR = target/reference
# excess cases
EC = RD * p
```

As the math communicates, the PAR is the rate difference expressed as a fraction of total risk for the target population. This could also be read as the fraction of risk in the target population that would have been removed had the target rate equaled the reference rate (Menvielle et al. 2017).

Comparing age-standardized rates:

If the user provides a list of `stand_surveil` objects with age-standardized rates (instead of a single `surveil` model), then the exact calculations will be completed as follows. The RR is simply the ratio of age-standardized rates, and the rate difference is similarly the difference between age-standardized rates. However, excess cases is calculated for each age group separately, and the total excess cases across all age groups is returned. Similarly, the attributable risk is calculated by taking the total excess cases across all age groups per year and dividing by the total risk (i.e., by the sum of the whole number of cases across all age groups). Cumulative excess cases is the sum of the time-period specific total number of excess cases. (Notice that the PAR is not equal to (RR-1)/RR when the PAR is derived from a number of age-specific rates and the RR is based on age-standardized rates.)

Value

A list, also of class "surveil_diff", with the following elements:

summary A tibble with a summary of posterior distributions (mean and 95 percent cred. intervals) for the target group incidence rate, the RD, RR, PAR, and excess cases.

cumulative_cases Summary of the posterior distribution for the cumulative number of excess cases and the PAR (mean and 95 percent cred. intervals)

groups Character string with target and reference population names

samples A data frame of MCMC samples for each quantity of interest (target and reference rates, RD, RR, PAR, and EC, as well as `Trend_Cases = Rate * Population`). Indexed by time.

cum_samples MCMC samples of the cumulative number of excess cases.

Author(s)

Connor Donegan (Connor.Donegan@UTSouthwestern.edu)

Source

Menvielle, G, Kulhanaova, I, Machenbach, JP. Assessing the impact of a public health intervention to reduce social inequalities in cancer. In: Vaccarella, S, Lortet-Tieulent, J, Saracci, R, Conway, D, Straif, K, Wild, CP, editors. Reducing Social Inequalities in Cancer: Evidence and Priorities for Research. Geneva, Switzerland: WHO Press, 2017:185-192.

See Also

plot.surveil_diff print.surveil_diff theil

Examples

```
data(msa)
houston <- msa[grep("Houston", msa$MSA), ]
fit <- stan_rw(houston, time = Year, group = Race,
               chains = 2, iter = 900) # low iter for speed only
gd <- group_diff(fit, "Black or African American", "White")
print(gd, scale = 100e3)
plot(gd, scale = 100e3)
```

---

msa *Colorectal cancer incidence by Texas MSA, 1999-2017, ages 50-79*

Description

Annual counts of colorectal cancer (cancer of colon or rectum), ages 50-79, for Texas's top four metropolitan statistical areas (MSAs), with population at risk estimates, by race-ethnicity (non-Hispanic White, non-Hispanic Black, Hispanic/Latino).

Usage

```
msa
```

Format

A `tibble` with the following attributes:

Year Year of diagnosis

Race Race-ethnicity designation

MSA Metropolitan statistical area

Count Number of CRC cases

Population Age-specific population estimate

Source

United States Cancer Statistics–Incidence: 1999-2017, WONDER Online Database. United States Department of Health and Human Services, Centers for Disease Control and Prevention and National Cancer Institute; 2020. Accessed at http://wonder.cdc.gov/cancer-v2017.html on Nov 9, 2020 2:59:24 PM.

Examples

```
data(msa)
head(msa)
```

---

plot.surveil *Methods for fitted* surveil *models*

Description

Print and plot methods for surveil model results

Usage

```
## S3 method for class 'surveil'
print(x, scale = 1, ...)

## S3 method for class 'surveil'
plot(
  x,
  scale = 1,
  style = c("mean_qi", "lines"),
  facet = FALSE,
  facet_scales = c("fixed", "free"),
  ncol = NULL,
  base_size = 14,
  palette = "Dark2",
  M = 250,
  alpha,
  lwd,
  fill = "gray80",
  size = 1.5,
  ...
)

## S3 method for class 'list'
plot(
  x,
  scale = 1,
  style = c("mean_qi", "lines"),
  facet = FALSE,
  ncol,
  facet_scales = c("fixed", "free"),
  M = 250,
  base_size = 14,
  palette = "Dark2",
  fill = "gray80",
  size = 1.5,
  alpha,
  lwd,
  ...
)
```

Arguments

`x` A fitted `surveil` model, or a list of `stand_surveil` objects (as produced by `standardize`).

`scale` Scale the rates by this amount; e.g., `scale = 100e3` will print rates per 100,000 at risk.

`...` For the plot method, additional arguments will be passed to ‘`theme`; for the print method, additional arguments will be passed to `print.data.frame`.

`style` If `style = "mean_qi"`, then the posterior mean and 95 percent credible interval will be plotted; if `style = "lines"`, then `M` samples from the joint probability distribution of the annual rates will be plotted.

`facet` If `facet = TRUE`, `facet_wrap` will be used instead of differentiating by line color.

`facet_scales` When `facet = TRUE`, this argument controls behavior of the scales for each subplot. See the `scales` argument to `facet_wrap`.

`ncol` Number of columns for the plotting device; optional and only used if `facet = TRUE`. If `ncol = 1`, the three plots will be aligned vertically in one column; if `ncol = 3` they will b aligned horizontally in one row. Defaults to `ncol = NULL` to allow `facet_wrap` to automatically determine the number of columns.

`base_size` Passed to `theme_classic()` to control size of plot components (text).

`palette` For multiple groups, choose the color palette. For a list of options, see `scale_color_brewer`. The default is `palette = "Dark2"`. Not used if `facet = TRUE`.

`M` If `style = "lines"`, then `M` is the number of samples from the posterior distribution that will be plotted; the default is `M = 250`.

`alpha` Numeric value from zero to one. When `style = "lines"`, this controls transparency of lines; passed to `geom_line`. For ‘style = "mean_qi", this controls the transparency of the shaded credible interval; passed to `geom_ribbon`.

`lwd` Numeric value indicating linewidth. Passed to `geom_line`

`fill` Color for the shaded credible intervals; only used when `style = "mean_qi"`.

`size` Positive numeric value. For `style = "mean_qi"`, this controls the size of the points representing crude rates. To exclude these points from the plot altogether, use `size = 0`.

Value

The plot method returns a `ggplot` object; the print method returns nothing but prints a summary of results to the R console. If `x` is a list of `stand_surveil` objects, the plotted lines will be labeled using the names returned by `names(x)`; if elements of the list are not named, plotted lines will simply be numbered.

Author(s)

Connor Donegan (Connor.Donegan@UTSouthwestern.edu)

See Also

stan_rw

Examples

```
data(msa)
houston <- msa[grep("Houston", msa$MSA), ]
fit <- stan_rw(houston, time = Year, group = Race,
              chains = 2, iter = 900) # for speed only

print(fit)

## plot probability distribution for disease risk
plot(fit, style = "lines")
plot(fit, facet = TRUE, scale = 100e3)

 ## as a ggplot, you can customize the output
library(ggplot2)
plot(fit) + theme_bw()
```

plot.theil *Methods for Theil's index*

Description

Printing and plotting methods for Theil's inequality index

Usage

```
## S3 method for class 'theil'
plot(
  x,
  style = c("mean_qi", "lines"),
  M = 250,
  col = "black",
  fill = "black",
  alpha,
  lwd,
  base_size = 14,
  scale = 100,
  labels = x$summary$time,
  ...
)

## S3 method for class 'theil_list'
plot(
  x,
  style = c("mean_qi", "lines"),
  M = 250,
  col = "black",
  fill = "black",
  alpha,
  lwd,
  between_title = "Between",
  within_title = "Within",
  total_title = "Total",
  scale = 100,
  plot = TRUE,
  ncol = 3,
  base_size = 14,
  ...
)

## S3 method for class 'theil'
print(x, scale = 100, digits = 3, ...)

## S3 method for class 'theil_list'
print(x, scale = 100, digits = 3, ...)
```

Arguments

x — An object of class thiel' or theil_list, as returned by calling theilon a list of fittedsurveil‘ models

style — If style = "mean_qi", then the posterior mean and 95 percent credible interval will be plotted; if style = "lines", then M samples from the joint probability distribution will be plotted.

M — If style = "lines", then M is the number of samples from the posterior distribution that will be plotted; the default is M = 250.

col — Line color

fill — Fill color

alpha — For style = "mean_qi", this controls the transparency for the credible interval (passed to geom_ribbon) and defaults to alpha = 0.5; for style = "lines", this controls the transparency of the lines and defaults to alpha = 0.7.

lwd — Line width; for style = "mean_qi", the default is lwd = 1; for style = "lines", the default is lwd = 0.05.

base_size — Passed to theme_classic to control size of plot elements (e.g., text) `scale` Scale Theil's index by `scale`

`labels` x-axis labels (time periods)

`...` additional arguments

`between_title` Plot title for the between geography component of Theil's T; defaults to "Between".

`within_title` Plot title for the within geography component of Theil's T; defaults to "Within".

`total_title` Plot title for Theil's index; defaults to "Total".

`plot` If `FALSE`, return a list of `ggplot`s. Not used when `style = "lines"`.

`ncol` Number of columns for the plotting device. If `ncol = 1`, the three plots will be aligned vertically in one column; if `ncol = 3` they will b aligned horizontally in one row.

`digits` number of digits to print (passed to `print.data.frame`)

Value

plot.theil:

The plot method returns an object of class `ggplot`.

plot.theil_list:

If `style = "lines"`, the plot method for `theil_list` objects returns a `ggplot` with facets for each component of inequality (between-areas, within-areas, and total). For `style = "mean_qi"`, the plot method returns either a list of plots (all of class `ggplot`) or, when `plot = TRUE`, it will draw them to current plotting device using `grid.arrange`.

print.theil:

The print returns nothing and method prints a summary of results to the R console.

See Also

`theil`

`print.apc` *Methods for APC objects*

Description

Methods for APC objects

Usage

```
## S3 method for class 'apc'
print(x, digits = 1, max = 20, ...)

## S3 method for class 'apc'
plot(
  x,
  cumulative = FALSE,
  style = c("mean_qi", "lines"),
  M = 250,
  col = "black",
  fill = "black",
  alpha,
  lwd,
  base_size = 14,
  ...
)
```

Arguments

`x` An apc object returned by `apc`

`digits` Print this many digits (passed to `print.data.frame`)

`max` Maximum number of time periods (rows) to print

`...` additional arguments; for the print argument, these will be passed to `print.data.frame`. For the plot method, these will be passed to `theme`.

`cumulative` Plot cumulative percent change? Defaults to `cumulative = FALSE`

`style` If `style = "mean_qi"`, then the posterior mean and 95 percent credible interval will be plotted; if `style = "lines"`, then `M` samples from the joint probability distribution will be plotted.

`M` If `style = "lines"`, then `M` is the number of samples from the posterior distribution that will be plotted; the default is `M = 250`.

`col` Line color

`fill` Fill color for the 95 percent credible interval

`alpha` For `style = "mean_qi"`, this controls the transparency for the credible interval (passed to `geom_ribbon`) and defaults to `alpha = 0.5`; for `style = "lines"`, this controls the transparency of the lines and defaults to `alpha = 0.7`.

`lwd` Line width

`base_size` Size of plot attributes, passed to ‘`theme_classic`

Value

print:
The print method does not have a return value, but prints a summary of results to the R console.

Plot:
The plot method returns a `ggplot`.

See Also

apc

print.stand_surveil *Methods for age-standardized rates*

Description

Print and plot methods for stand_surveil (standardized rates obtained from a fitted surveil model)

Usage

```
## S3 method for class 'stand_surveil'
print(x, scale = 1, digits = 3, ...)

## S3 method for class 'stand_surveil'
plot(
  x,
  scale = 1,
  style = c("mean_qi", "lines"),
  M = 250,
  base_size = 14,
  col = "black",
  fill = "gray80",
  alpha,
  lwd,
  ...
)
```

Arguments

x	An object of stand_surveil obtained by calling standardize on a fitted surveil model
scale	Scale the rates by this amount; e.g., scale = 100e3 will print rates per 100,000 at risk.
digits	Number of digits to print
...	additional arguments
style	If style = "mean_qi", then the posterior means and 95 percent credible intervals will be plotted; if style = "lines", then M samples from the joint posterior distribution will be plotted.
M	Number of samples to plot when style = "lines"
base_size	Passed to theme_classic() to control size of plot components (text).
col	Line color

`fill`	Fill color for the 95 percent credible intervals
`alpha`	For `style = "mean_qi"`, this controls the transparency for the credible interval (passed to `geom_ribbon`) and defaults to `alpha = 0.5`; for `style = "lines"`, this controls the transparency of the lines and defaults to `alpha = 0.7`.
`lwd`	Line width; for `style = "mean_qi"`, the default is `lwd = 1`; for `style = "lines"`, the default is `lwd = 0.05`.

Details

Calling `standardize` on a fitted `surveil` model will create a new object that contains the `surveil` model results as well standardized rates. This new `stand_surveil` object has its own methods for printing and plotting.

print.stand_surveil:

Any additional arguments (`...`) will be passed to `print.data.frame`

plot.stand_surveil:

Any additional arguments (`...`) will be passed to ‘`theme`.

Value

print.stand_surveil:

The print method returns nothing but prints a summary of results to the console.

plot.stand_surveil:

The plot method returns an object of class `ggplot`.

See Also

`standardize` `stan_rw`

`priors` *Prior distributions*

Description

Prior distributions

Usage

```
normal(location = 0, scale, k = 1)

lkj(eta)
```

Arguments

`location` Location parameter (numeric)

`scale` Scale parameter (positive numeric)

`k` Optional; number of groups for which priors are needed. This is a shortcut to avoid using the `rep` function to repeat the same prior for each group, as in: `normal(location = rep(0, times = 3), scale = rep(1, times = 3)`. To provide distinct priors for each group, simply specify them individually, as in `normal(location = c(-5, -6, -8), scale = c(2, 2, 2))`.

`eta` The shape parameter for the LKJ prior

Details

The prior distribution functions are used to set the values of prior parameters.

Users can control the values of the parameters, but the distribution (model) itself is fixed. The first log-rate (`eta[t]`, `t=1`) and the scale parameters (sigma) are assigned Gaussian (`normal`) prior distribution. (The scale parameter, sigma, is constrained to be positive, making it a half-normal prior.) For correlated time series, the correlation matrix is assigned the LKJ prior.

Parameterizations:

For details on how any distribution is parameterized, see the Stan Language Functions Reference document: https://mc-stan.org/users/documentation/.

LKJ prior:

The LKJ prior for correlation matrix has a single parameter, eta (eta > 0). If `eta=1`, then you are placing a uniform prior on any K-by-K correlation matrix. For eta > 1, there is a higher probability on the identity matrix, such that as eta increases beyond 1, you are expressing greater skepticism towards large correlations. If 0 < eta < 1, then you will be expressing skepticism towards correlations of zero and favoring non-zero correlations. See Stan documentation: https://mc-stan.org/docs/2_27/functions-reference/lkj-correlation.html.

Value

An object of class `prior` which will be used internally by **surveil** to set parameters of prior distributions.

Source

Stan Development Team. Stan Functions Reference Version 2.27. https://mc-stan.org/docs/2_27/functions-reference/lkj-correlation.html

Examples

```
# note there are three groups in the data, each requires a prior
prior <- list()
prior$eta_1 <- normal(location = -6, scale = 4, k = 3)
## by default, location = 0
prior$sigma <- normal(scale = 1, k = 3)
```

```
prior$omega <- lkj(2)


dfw <- msa[grep("Dallas", msa$MSA), ]
fit <- stan_rw(dfw, time = Year, group = Race, prior = prior,
               chains = 2, iter = 900) # for speed only
plot(fit)
```

`standard` *2000 U.S. standard million population*

Description

2000 U.S. standard million population

Usage

```
standard
```

Format

An object of class `data.frame` with 19 rows and 3 columns.

Source

National Cancer Institute. Standard Populations - 19 Age Groups. Accessed at `https://seer.cancer.gov/stdpopulations/stdpop.19ages.html` on Oct. 8, 2021.

Examples

```
data(standard)
head(standard)
```

`standardize` *Age-standardized rates*

Description

Convert `surveil` model results to age standardized rates using a fixed age distribution

Usage

```
standardize(x, label, standard_pop)
```

Arguments

`x`	A fitted `surveil` model
`label`	Labels (character strings) for the age groups that correspond to the values of `stand_pop`. The labels must match the grouping variable used to fit the model (i.e., `all(label %in% names(x$data$cases))` must be true).
`standard_pop`	Standard population values corresponding to the age groups specified by `label`

Value

A list, also of class "stand_surveil", containing the entire contents of the user-provided `surveil` model plus the following:

standard_summary summary data frame of standardized rates (means and 95 percent credible intervals)

standard_samples a data frame of Markov chain Monte Carlo (MCMC) samples from the posterior probability distribution for the standardized rates

standard_label user-provided age-group labels

standard_pop user-provided standardized population sizes (ordered as `standard_label`)

See Also

`vignette("age-standardization", package = "surveil")` `stan_rw` `plot.stand_surveil` `print.stand_surveil`

Examples

```
data(cancer)
data(standard)

head(standard)
head(cancer)


fit <- stan_rw(cancer,
              time = Year,
              group = Age,
              chains = 2, iter = 900 # for speed only
              )

stands <- standardize(fit,
                      label = standard$age,
                      standard_pop = standard$standard_pop)
print(stands)
plot(stands, style = "lines")
```

stan_rw *Time series models for mortality and disease incidence*

Description

Model time-varying incidence rates given a time series of case (or death) counts and population at risk.

Usage

```
stan_rw(
  data,
  group,
  time,
  cor = FALSE,
  family = poisson(),
  prior = list(),
  chains = 4,
  cores = 1,
  iter = 3000,
  refresh = 1500,
  control = list(adapt_delta = 0.98),
  ...
)
```

Arguments

`data` A `data.frame` containing the following columns:

- **Count** Number of cases or deaths; this column must be named 'Count'.
- **Population** Size of population at risk; this column must be named 'Population'.
- **time** Time period indicator. (Provide the (unquoted) column name using the `time` argument.)
- **group** Optional grouping variable. (Provide the (unquoted) column name using the `group` argument.)

`group` If `data` is aggregated by demographic group, provide the (unquoted) name of the column in `data` containing the grouping structure, such as age brackets or race-ethnicity. E.g., if `data` has column names `Year`, `Race`, `Cases`, and `Population`, then you would provide `group = Race`.

`time` Specify the (unquoted) name of the time variable in `data`, as in `time = Year`. This variable must be numeric-alike (i.e., `as.numeric(data$time)` will not fail).

`cor` For correlated random walks use `cor = TRUE`; default value is `FALSE`. Note this only applies when the `group` argument is used.

`family` The default specification is a Poisson model with log link function (`family = poisson()`). For a Binomial model with logit link function, use `family = binomial()`.

`prior` Optionally provide a named `list` with prior parameters. If any of the following items are missing, default priors will be assigned and printed to the console.

eta_1 The first value of log-risk in each series must be assigned a Gaussian prior probability distribution. Provide the location and scale parameters for each demographic group in a list, where each parameter is a `k`-length vector. For example, with `k=2` demographic groups, the following code will assign priors of `normal(-6.5, 5)` to the starting values of both series: `prior = list(eta_1 = normal(loc` Note, `eta` is the log-rate, so centering the prior for `eta_1` on `-6.5` is similar to centering the prior rate on `exp(-6.5)*100,000 = 150` cases per 100,000 person-years at risk. Note, however, that the translation from log-rate to rate is non-linear.

sigma Each demographic group has a scale parameter assigned to its log-rate. This is the scale of the annual deviations from the previous year's log-rate. The scale parameters are assigned independent half-normal prior distributions (these `half` normal distributions are restricted to be positive-valued only).

omega If `cor = TRUE`, an LKJ prior is assigned to the correlation matrix, Omega.

`chains` Number of independent MCMC chains to initiate (passed to `sampling`).

`cores` The number of cores to use when executing the Markov chains in parallel (passed to `sampling`).

`iter` Total number of MCMC iterations. Warmup draws are automatically half of `iter`.

`refresh` How often to print the MCMC sampling progress to the console.

`control` A named list of parameters to control Stan's sampling behavior. The most common parameters to control are `adapt_delta`, which may be raised to address divergent transitions, and `max_treedepth`. For example, `control = list(adapt_delta = 0.99, max_treedepth = 13)`, may be a reasonable specification to address a divergent transitions or maximum treedepth warning from Stan.

`...` Other arguments passed to `sampling`.

Details

By default, the models have Poisson likelihoods for the case counts, with log link function. Alternatively, a Binomial model with logit link function can be specified using the `family` argument (`family = binomial()`).

For time t = 1,...n, the models assign Poisson probability distribution to the case counts, given log-risk `eta` and population at tirks P; the log-risk is modeled using the first-difference (or random-walk) prior:

```
y_t ~ Poisson(p_t * exp(eta_t))
eta_t ~ Normal(eta_{t-1}, sigma)
eta_1 ~ Normal(-6, 5) (-Inf, 0)
sigma ~ Normal(0, 1) (0, Inf)
```

This style of model has been discussed in Bayesian (bio)statistics for quite some time. See Clayton (1996).

The above model can be used for multiple distinct groups; in that case, each group will have its own independent time series model.

It is also possible to add a correlation structure to that set of models. Let `Y_t` be a k-length vector of observations for each of k groups at time t (the capital letter now indicates a vector), then:

```
Y_t ~ Poisson(P_t * exp(Eta_t))
Eta_t ~ MVNormal(Eta_{t-1}, Sigma)
Eta_1 ~ Normal(-6, 5)  (-Inf, 0)
Sigma = diag(sigma) * Omega * diag(sigma)
Omega ~ LKJ(2)
sigma ~ Normal(0, 1) (0, Inf)
```

where `Omega` is a correlation matrix and `diag(sigma)` is a diagonal matrix with scale parameters on the diagonal. This was adopted from Brandt and Williams (2007); for the LKJ prior, see the Stan Users Guide and Reference Manual.

If the binomial model is used instead of the Poisson, then the first line of the model specifications will be:

```
y_t ~ binomial(P_t, inverse_logit(eta_t))
```

All else is remains the same. The logit function is `log(r/(1-r))`, where `r` is a rate between zero and one; the inverse-logit function is `exp(x)/(1 + exp(x))`.

Value

The function returns a list, also of class `surveil`, containing the following elements:

summary A `data.frame` with posterior means and 95 percent credible intervals, as well as the raw data (Count, Population, time period, grouping variable if any, and crude rates).

samples A `stanfit` object returned by `sampling`. This contains the MCMC samples from the posterior distribution of the fitted model.

cor Logical value indicating if the model included a correlation structure.

time A list containing the name of the time-period column in the user-provided data and a `data.frame` of observed time periods and their index.

group If a grouping variable was used, this will be a list containing the name of the grouping variable and a `data.frame` with group labels and index values.

family The user-provided `family` argument.

Author(s)

Connor Donegan (Connor.Donegan@UTSouthwestern.edu)

Source

Brandt P and Williams JT. Multiple time series models. Thousand Oaks, CA: SAGE Publications, 2007.

Clayton, DG. Generalized linear mixed models. In: Gilks WR, Richardson S, Spiegelhalter DJ, editors. Markov Chain Monte Carlo in Practice: Interdisciplinary Statistics. Boca Raton, FL: CRC Press, 1996. p. 275-302.

Donegan C, Hughes AE, and Lee SC (2022). Colorectal Cancer Incidence, Inequalities, and Prevention Priorities in Urban Texas: Surveillance Study With the "surveil" Software Package. *JMIR Public Health & Surveillance* 8(8):e34589. doi:10.2196/34589

Stan Development Team. Stan Modeling Language Users Guide and Reference Manual, 2.28. 2021. https://mc-stan.org

See Also

`vignette("demonstration", package = "surveil")` `vignette("age-standardization", package = "surveil")` `apc` `standardize`

Examples

```
library(rstan)
data(msa)
austin <- msa[grep("Austin", msa$MSA), ]

fit <- stan_rw(austin,
               time = Year,
               group = Race,
               chains = 2, iter = 900) # for speed only

## MCMC diagnostics
rstan::stan_mcse(fit$samples)
rstan::stan_rhat(fit$samples)
print(fit$samples)

## print the surveil object
print(fit)
head(fit$summary)

## plot time trends
plot(fit, style = 'lines')

## age-specific rates and cumulative percent change
data(cancer)
fit <- stan_rw(cancer, time = Year, group = Age,
              chains = 2, iter = 900) # for speed only
fit_apc <- apc(fit)
plot(fit_apc, cumulative = TRUE)

# age-standardized rates
data(standard)
fit_stands <- standardize(fit,
                          label = standard$age,
                          standard_pop = standard$standard_pop)
print(fit_stands)
```

```
plot(fit_stands)
fit_stands_apc <- apc(fit_stands)
plot(fit_stands_apc)
```

`surveil_diff` *Methods for* `surveil_diff` *objects*

Description

Methods for `surveil_diff` objects

print surveil_diff objects for analyses of inequality

Usage

```
## S3 method for class 'surveil_diff'
plot(
  x,
  style = c("mean_qi", "lines"),
  M = 250,
  col = "black",
  fill = "gray80",
  lwd,
  alpha,
  plot = TRUE,
  scale = 1e+05,
  PAR = TRUE,
  ncol = 3,
  base_size = 14,
  ...
)

## S3 method for class 'surveil_diff'
print(x, scale = 1, ...)
```

Arguments

`x` Object of class `surveil_diff`, as returned by calling `group_diff` on a fitted `surveil` model

`style` If `style = "mean_qi"`, then the posterior mean and 95 percent credible interval will be plotted; if `style = "lines"`, then `M` samples from the joint probability distribution of the annual rates will be plotted.

`M` If `style = "lines"`, then `M` is the number of samples from the posterior distribution that will be plotted; the default is `M = 250`.

`col` Line color

`fill` Fill color for credible intervals, passed to `geom_ribbon`

`lwd` Linewidth

`alpha` transparency; for `style = "mean_qi", controls the credible interval shading; for style` = "lines"‘, this is applied to the lines

`plot` If `plot = FALSE`, a list of ggplots will be returned

`scale` Print rates and rate differences as per `scale` at risk, e.g., per 10,000 at risk.

`PAR` Return population attributable risk? IF `FALSE`, then the rate ratio will be used instead of PAR.

`ncol` Number of columns for the plotting device. If `ncol = 1`, the three plots will be aligned vertically in one column; if `ncol = 3` they will b aligned horizontally in one row.

`base_size` Passed to `theme_classic` to control size of plot elements (e.g., text)

`...` additional print arguments

Value

plot.surveil_diff:

By default or whenever `plot = TRUE`, the plot method draws a series of plots to the current plotting device using `grid.arrange`. If `plot = FALSE`, then a list of `ggplots` is returned.

print.surveil_diff:

The print method returns nothing and prints a summary of results to the console.

`theil` *Theil's inequality index*

Description

Theil's entropy-based index of inequality

Usage

```
theil(x)

theil2(Count, Population, rates, total = TRUE)

## S3 method for class 'surveil'
theil(x)

## S3 method for class 'list'
theil(x)
```

Arguments

x A fitted surveil model, from stan_rw; or, a list of fitted surveil models, where each model represents a different geographic area (e.g., states).

Count Case counts, integers

Population Population at risk, integers

rates If Count is not provided, then rates must be provided (Count = rates * Population).

total If total = TRUE, Theil's index will be returned. Each unit contributes to Theil's index; if total = FALSE, all of the elements that sum to Theil's index will be returned.

Details

Theil's index is a good index of inequality in disease and mortality burdens when multiple groups are being considered. It provides a summary measure of inequality across a set of demographic groups that may be tracked over time (and/or space). Also, it is interesting because it is additive, and thus admits of simple decompositions.

The index measures discrepancies between a population's share of the disease burden, omega, and their share of the population, eta. A situation of zero inequality would imply that each population's share of cases is equal to its population share, or, omega=eta. Each population's contribution to total inequality is calculated as:

```
T_i = omega_i * [log(omega_i/eta_i)],
```

the log-ratio of case-share to population-share, weighted by their share of cases. Theil's index for all areas is the sum of each area's T_i:

```
T = sum_(i=1)^n T_i.
```

Theil's T is thus a weighted mean of log-ratios of case shares to population shares, where each log-ratio (which we may describe as a raw inequality score) is weighted by its share of total cases. The index has a minimum of zero and a maximum of log(N), where N is the number of units (e.g., number of states).

Theil's index, which is based on Shannon's information theory, can be extended to measure inequality across multiple groups nested within non-overlapping geographies (e.g., states).

Value

theil2:

If total = TRUE (the default), theil2 returns Theil's index as a numeric value. Else, theil2 returns a vector of values that sum to Theil's index.

theil.surveil:

A named list with the following elements:

summary A data.frame summarizing the posterior probability distribution for Theil's T, including the mean and 95 percent credible interval for each time period

samples A data.frame with MCMC samples for Theil's T

theil.list:

A list (also of class theil_list) containing a summary data frame and a tbl_df containing MCMC samples for Theil's index at each time period.

The summary data frame includes the following columns:

time time period

Theil Posterior mean for Theil's index; equal to the sum of Theil_between and Theil_within.

Theil_between The between-areas component to Theil's inequality index

Theil_within The within-areas component to Theil's inequality index

Additional columns contain the upper and lower limits of the 95 percent credible intervals for each component of Theil's index.

The data frame of samples contains the following columns:

time Time period indicator

.draw An id for each MCMC sample; note that samples are from the joint distribution

Theil_between The between-geographies component of Theil's index

Theil_within The within-geographies component of Theil's index

Theil Theil's inequality index (T = Between + Within).

Source

Conceicao, P. and P. Ferreira (2000). The young person's guide to the Theil Index: Suggesting intuitive interpretations and exploring analytical applications. University of Texas Inequality Project. UTIP Working Paper Number 14. Accessed May 1, 2021 from https://utip.gov.utexas.edu/papers.html

Conceicao, P, Galbraith, JK, Bradford, P. (2001). The Theil Index in sequences of nested and hierarchic grouping structures: implications for the measurement of inequality through time, with data aggregated at different levels of industrial classification. *Eastern Economic Journal*. 27(4): 491-514.

Theil, Henri (1972). *Statistical Decomposition Analysis.* Amsterdam, The Netherlands and London, UK: North-Holland Publishing Company.

Shannon, Claude E. and Weaver, Warren (1963). *The Mathematical Theory of Communication*. Urbana and Chicago, USA: University if Illinois Press.

See Also

plot.theil print.theil plot.theil_list

Examples

```
houston <- msa[grep("Houston", msa$MSA), ]
fit <- stan_rw(houston, time = Year, group = Race,
               chains = 2, iter = 900) # for speed only
theil_dfw <- theil(fit)
plot(theil_dfw)
```

```
Count <- c(10, 12, 3, 111)
Pop <- c(1000, 1200, 4000, 9000)
theil2(Count, Pop)
theil2(Count, Pop, total = FALSE)
```

waic *Widely Applicable Information Criteria*

Description

Widely Application Information Criteria (WAIC) for model comparison

Usage

```
waic(fit, pointwise = FALSE, digits = 2)
```

Arguments

`fit`	An `surveil` object
`pointwise`	Logical (defaults to `FALSE`); if `pointwise = TRUE`, a vector of values for each observation will be returned.
`digits`	Round results to this many digits.

Value

A vector of length 3 with `WAIC`, a rough measure of the effective number of parameters estimated by the model `Eff_pars`, and log predictive density `Lpd`. If `pointwise = TRUE`, results are returned in a `data.frame`.

Source

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely application information criterion in singular learning theory. Journal of Machine Learning Research 11, 3571-3594.

Examples

```
data(msa)
austin <- msa[grep("Austin", msa$MSA), ]
austin.w <- austin[grep("White", austin$Race),]
fit <- stan_rw(austin.w, time = Year,
               chains = 2, iter = 1200) # for speed only
waic(fit)
```

Index