Package 'cfr'

February 21, 2025

Title Estimate Disease Severity and Case Ascertainment

Version 0.2.0

Description Estimate the severity of a disease and ascertainment of cases, as discussed in Nishiura et al. (2009) <doi:10.1371/journal.pone.0006852>.

License MIT + file LICENSE

URL https://github.com/epiverse-trace/cfr, https://epiverse-trace.github.io/cfr/

BugReports https://github.com/epiverse-trace/cfr/issues

Depends R (>= 3.5.0)

Imports checkmate, stats

Suggests bookdown, data.table, distcrete, distributional, dplyr, forcats, ggplot2, incidence2, knitr, magrittr, purrr, rmarkdown, scales, spelling, testthat (>= 3.0.0), tidyr

VignetteBuilder knitr

Config/Needs/website epiverse-trace/epiversetheme

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Pratik R. Gupte [aut, cph] (<https://orcid.org/0000-0001-5294-7819>), Adam Kucharski [aut, cph, cre] (<https://orcid.org/0000-0001-8814-9421>), Tim Russell [aut, cph] (<https://orcid.org/0000-0001-5610-6080>), Joshua W. Lambert [rev] (<https://orcid.org/0000-0001-5218-3046>), Hugo Gruson [rev] (<https://orcid.org/0000-0002-4094-1476>), Tim Taylor [rev] (<https://orcid.org/0000-0002-8587-7113>), James M. Azam [rev] (<https://orcid.org/0000-0001-5782-7330>),

Abdoelnaser M. Degoot [rev] (<https://orcid.org/0000-0001-8788-2496>),
Sebastian Funk [rev] (<https://orcid.org/0000-0002-2842-3406>)

Maintainer Adam Kucharski <adam.kucharski@lshtm.ac.uk>

Repository CRAN

Date/Publication 2025-02-21 10:20:02 UTC

Contents

`cfr_rolling` *Estimate static severity for an expanding time series*

Description

Calculates the CFR at each time point in the case and death time series supplied, using an expanding window of time. The static CFR is calculated for each time point, using the time series from the start to each time point, and increasing the number of time points included by one in each iteration.

Usage

```
cfr_rolling(data, delay_density = NULL, poisson_threshold = 100)
```

Arguments

`data`	A `<data.frame>` containing the outbreak data. A daily time series with dates or some other absolute indicator of time (e.g. epiday or epiweek) and the numbers of new cases and new deaths at each time point. Note that the required columns are "date" (for the date), "cases" (for the number of reported cases), and "deaths" (for the number of reported deaths) on each day of the outbreak. Note that the `<data.frame>` is required to have an unbroken sequence of dates with no missing dates in between. The "date" column must be of class `Date` (see `as.Date()`). Note also that the total number of cases must be greater than the total number of reported deaths.

`delay_density` An optional argument that controls whether delay correction is applied in the severity estimation. May be `NULL`, for no delay correction, or a function that returns the density function of a distribution to evaluate density at user-specified values, e.g. `function(x) stats::dgamma(x = x, shape = 5, scale = 1)`.

`poisson_threshold` The case count above which to use Poisson approximation. Set to 100 by default. Must be > 0.

Details

When delay correction is applied by passing a delay distribution density function to `delay_density`, the internal function `.estimate_severity()` is used to calculate the rolling severity.

Note that in the naive method the severity estimate and confidence intervals cannot be calculated for days on which the cumulative number of cases since the start of the time-series, and for days on which the cumulative number of deaths reported exceeds the cumulative reported cases, and is returned as `NA`.

`cfr_rolling()` applies the internal function `.estimate_severity()` to an expanding time-series of total cases, total estimated outcomes, and total deaths. The method used to generate a profile likelihood for each day depends on the outbreak size and initial severity estimate for that day. This is essentially the same as running `cfr_static()` on each new day. The method used for each day is not communicated to the user, in order to prevent cluttering the terminal with messages.

Value

A <data.frame> with the date, maximum likelihood estimate and 95% confidence interval of the daily severity estimates, named "severity_estimate", "severity_low", and "severity_high", with one row for each day in the original data.frame.

Examples

```
# load package data
data("ebola1976")

# estimate severity without correcting for delays
cfr_static(ebola1976)

# estimate severity for each day while correcting for delays
# obtain onset-to-death delay distribution parameters from Barry et al. 2018
# The Lancet. <https://doi.org/10.1016/S0140-6736(18)31387-4>
# view only the first values
estimate <- cfr_rolling(
  ebola1976,
  delay_density = function(x) dgamma(x, shape = 2.40, scale = 3.33)
)

head(estimate)
```

`cfr_static` *Estimate a static disease severity measure*

Description

Calculates the severity of a disease, while optionally correcting for reporting delays using an epidemiological delay distribution of the time between symptom onset and death (onset-to-death).

Other delay distributions may be passed to calculate different disease severity measures such as the hospitalisation fatality risk.

Usage

```
cfr_static(data, delay_density = NULL, poisson_threshold = 100)
```

Arguments

`data`	A `<data.frame>` containing the outbreak data. A daily time series with dates or some other absolute indicator of time (e.g. epiday or epiweek) and the numbers of new cases and new deaths at each time point. Note that the required columns are "date" (for the date), "cases" (for the number of reported cases), and "deaths" (for the number of reported deaths) on each day of the outbreak. Note that the `<data.frame>` is required to have an unbroken sequence of dates with no missing dates in between. The "date" column must be of class `Date` (see `as.Date()`). Note also that the total number of cases must be greater than the total number of reported deaths.
`delay_density`	An optional argument that controls whether delay correction is applied in the severity estimation. May be `NULL`, for no delay correction, or a function that returns the density function of a distribution to evaluate density at user-specified values, e.g. `function(x) stats::dgamma(x = x, shape = 5, scale = 1)`.
`poisson_threshold`	The case count above which to use Poisson approximation. Set to 100 by default. Must be > 0.

Value

A `<data.frame>` with the maximum likelihood estimate and 95% confidence interval of the severity estimates, named "severity_estimate", "severity_low", and "severity_high".

Details: Adjusting for delays between two time series

The method used in `cfr_static()` follows Nishiura et al. (2009). The function calculates a quantity u_t for each day within the input data, which represents the proportion of cases estimated to have a known outcome on day t. Following Nishiura et al., u_t is calculated as:

$$u_t = \frac{\sum_{i=0}^{t} \sum_{j=0}^{\infty} c_i f_{j-i}}{\sum_{i=0} c_i}$$

where f_t is the value of the probability mass function at time t and c_t, d_t are the number of new cases and new deaths at time t, (respectively). We then use u_t at the end of the outbreak in the following likelihood function to estimate the severity of the disease in question.

$$\mathsf{L}(\theta \mid y) = \log \binom{u_t C}{D} + D \log \theta + (u_t C - D) \log (1.0 - \theta)$$

C and D are the cumulative number of cases and deaths (respectively) up until time t. θ is the parameter we wish to estimate, the severity of the disease. We estimate θ using simple maximum-likelihood methods, allowing the functions within this package to be quick and easy tools to use.

The precise severity measure — CFR, IFR, HFR, etc — that θ represents depends upon the input data given by the user.

The epidemiological delay-distribution density function passed to `delay_density` is used to evaluate the probability mass function parameterised by time; i.e. $f(t)$ which gives the probability that a case has a known outcome (usually death) at time t, parameterised with disease-specific parameters before it is supplied here.

Profile likelihood methods:

The naive CFR estimate (without delay correction) is the outcome of a Binomial test on deaths and cases using `stats::binom.test()`. The confidence intervals around the estimate are also taken from the test.

The delay-corrected CFR estimates are however obtained by generating a profile likelihood over the sequence `seq(1e-4, 1.0, 1e-4)`. The method used depends on the outbreak size and the initial expectation of disease severity. This is implemented in the internal function `.estimate_severity()`.

- **Delay correction, small outbreaks**: For outbreaks where the total cases are below the user-specified 'Poisson threshold' (`poisson_threshold`, default = 100), the CFR and uncertainty around it is taken from a profile likelihood generated from a Binomial model of deaths (successes) and estimated known outcomes (trials).
  - **Delay correction, large outbreaks with low severity**: For outbreaks with total cases greater than the Poisson threshold (default = 100) and with initial severity estimates < 0.05, the CFR and uncertainty are taken from a Poisson approximation of the Binomial profile likelihood (taking $\lambda = np$ for n estimated outcomes and p as the severity estimate).
- **Delay correction, large outbreaks with higher severity**: For outbreaks with total cases greater than the Poisson threshold (default = 100) and with initial severity estimates ≥ 0.05, the CFR and uncertainty are taken from a Normal approximation of the Binomial profile likelihood.

References

Nishiura, H., Klinkenberg, D., Roberts, M., & Heesterbeek, J. A. P. (2009). Early Epidemiological Assessment of the Virulence of Emerging Infectious Diseases: A Case Study of an Influenza Pandemic. PLOS ONE, 4(8), e6852. doi:10.1371/journal.pone.0006852

Examples

```
# load package data
data("ebola1976")
```

```
# estimate severity without correcting for delays
cfr_static(ebola1976)

# estimate severity for each day while correcting for delays
# obtain onset-to-death delay distribution parameters from Barry et al. 2018
# The Lancet. <https://doi.org/10.1016/S0140-6736(18)31387-4>
cfr_static(
  ebola1976,
  delay_density = function(x) dgamma(x, shape = 2.40, scale = 3.33)
)
```

`cfr_time_varying` *Estimate a severity measure that varies over time*

Description

Calculates how the severity of a disease changes over time while optionally correcting for reporting delays using an epidemiological delay distribution of the time between symptom onset and outcome (e.g. onset-to-death for the fatality risk).

Usage

```
cfr_time_varying(
  data,
  delay_density = NULL,
  burn_in = 7,
  smoothing_window = NULL
)
```

Arguments

`data`	A <`data.frame`> containing the outbreak data. A daily time series with dates or some other absolute indicator of time (e.g. epiday or epiweek) and the numbers of new cases and new deaths at each time point. Note that the required columns are "date" (for the date), "cases" (for the number of reported cases), and "deaths" (for the number of reported deaths) on each day of the outbreak. Note that the <`data.frame`> is required to have an unbroken sequence of dates with no missing dates in between. The "date" column must be of class `Date` (see `as.Date()`). Note also that the total number of cases must be greater than the total number of reported deaths.
`delay_density`	An optional argument that controls whether delay correction is applied in the severity estimation. May be `NULL`, for no delay correction, or a function that returns the density function of a distribution to evaluate density at user-specified values, e.g. `function(x) stats::dgamma(x = x, shape = 5, scale = 1)`.

`burn_in` A single integer-like value for the number of time-points (typically days) to disregard at the start of the time-series, if a burn-in period is desired.

Defaults to 7, which is a sensible default value that disregards the first week of cases and deaths, assuming daily data.

To consider all case data including the start of the time-series, set this argument to 0.

`smoothing_window`

An *odd* number determining the smoothing window size to use when smoothing the case and death time-series, using a rolling median procedure (as the `k` argument to `stats::runmed()`) before calculating the time-varying severity.

The default behaviour is to apply no smoothing. The minimum value of this argument is 1.

Value

A `<data.frame>` with the date, maximum likelihood estimate and 95% confidence interval of the daily severity estimates, named "severity_estimate", "severity_low", and "severity_high", with one row for each day in the original data.frame.

Details: Adjusting for delays between two time series

This function estimates the number of cases which have a known outcome over time, following Nishiura et al. (2009). The function calculates a quantity k_t for each day within the input data, which represents the number of cases estimated to have a known outcome, on day t. k_t is calculated in the following way:

$$k_t = \sum_{j=0}^{t} c_t f_{j-t}$$

We then assume that the severity measure, for example CFR, of interest is binomially distributed, in the following way:

$$d_t \sim \mathsf{Binomial}(k_t, \theta_t)$$

We use maximum likelihood estimation to determine the value of θ_t for each t, where θ represents the severity measure of interest.

The epidemiological delay distribution passed to `delay_density` is used to obtain a probability mass function parameterised by time; i.e. $f(t)$ which gives the probability of the binary outcome of a case (survival or death) being known by time t. The delay distribution is parameterised with disease-specific parameters before it is supplied here.

Note that the function arguments `burn_in` and `smoothing_window` are not explicitly used in this calculation. `burn_in` controls how many estimates at the beginning of the outbreak are replaced with `NA`s — the calculation above is not applied to the first `burn_in` data points. The calculation is applied to the smoothed data, if a `smoothing_window` is specified.

References

Nishiura, H., Klinkenberg, D., Roberts, M., & Heesterbeek, J. A. P. (2009). Early Epidemiological Assessment of the Virulence of Emerging Infectious Diseases: A Case Study of an Influenza Pandemic. PLOS ONE, 4(8), e6852. doi:10.1371/journal.pone.0006852

Examples

```
# get data pre-loaded with the package
data("covid_data")
df_covid_uk <- covid_data[covid_data$country == "United Kingdom" &
covid_data$date <= as.Date("2020-09-01"), ]

# estimate time varying severity without correcting for delays
cfr_time_varying <- cfr_time_varying(
  data = df_covid_uk,
  burn_in = 7L
)
# View
tail(cfr_time_varying)

# estimate time varying severity while correcting for delays
# obtain onset-to-death delay distribution parameters from Linton et al. 2020
# J. Clinical Medicine: <https://doi.org/10.3390/jcm9020538>
# view only the first values
cfr_time_varying <- cfr_time_varying(
  data = df_covid_uk,
  delay_density = function(x) dlnorm(x, meanlog = 2.577, sdlog = 0.440),
  burn_in = 7L
)
tail(cfr_time_varying)
```

---

covid_data *Daily Covid-19 case and death data for countries with 100,000 or more deaths*

Description

Data adapted from the {covidregionaldata} package of daily cases and deaths from the 19 countries with 100,000 or more deaths over the period 2020-01-01 to 2022-12-31. See the **References** for the publication which links to data sources made available through {covidregionaldata}. Included as {covidregionaldata} is no longer on CRAN. Data are provided as a `<data.frame>`.

Usage

```
covid_data
```

Format

`covid_data`**:**

A `<data.frame>` with 20,786 rows and 4 columns:

date Calendar date in the format %Y-%m-%d

country The country name in simple format, e.g. "United States" rather than "United States of America"

cases Number of cases reported on each date

deaths Number of deaths reported on each date

Source

doi:10.21105/joss.03290.

References

Joseph Palmer, Katharine Sherratt, Richard Martin-Nielsen, Jonnie Bevan, Hamish Gibbs, Sebastian Funk and Sam Abbott (2021). covidregionaldata: Subnational data for COVID-19 epidemiology. doi:10.21105/joss.03290

`ebola1976` *Ebola 1976 outbreak case data*

Description

An example epidemic outbreak dataset for use with the `cfr` package. This dataset comes from the first Ebola outbreak in Zaire in 1976 as analysed in Camacho et al. (2014).

Usage

```
ebola1976
```

Format

`ebola1976`**:**

A `<data.frame>` with 73 rows and 3 columns:

date Calendar date

cases Number of cases reported

deaths Number of deaths reported

Source

doi:10.1016/j.epidem.2014.09.003

References

Camacho, A., Kucharski, A. J., Funk, S., Breman, J., Piot, P., & Edmunds, W. J. (2014). Potential for large outbreaks of Ebola virus disease. Epidemics, 9, 70–78. doi:10.1016/j.epidem.2014.09.003

`estimate_ascertainment` *Estimate the ascertainment ratio of a disease*

Description

Estimates the proportion of cases or infections that have been ascertained, given a time-series of cases and deaths, a delay distribution and a baseline severity estimate. The resulting ascertainment estimate is calculated as the ratio of the baseline severity estimate, which is assumed to be the 'true' disease severity, and the delay-adjusted severity estimate.

Usage

```
estimate_ascertainment(data, severity_baseline, delay_density = NULL)
```

Arguments

`data`	A `<data.frame>` containing the outbreak data. A daily time series with dates or some other absolute indicator of time (e.g. epiday or epiweek) and the numbers of new cases and new deaths at each time point. Note that the required columns are "date" (for the date), "cases" (for the number of reported cases), and "deaths" (for the number of reported deaths) on each day of the outbreak. Note that the `<data.frame>` is required to have an unbroken sequence of dates with no missing dates in between. The "date" column must be of class `Date` (see `as.Date()`). Note also that the total number of cases must be greater than the total number of reported deaths.
`severity_baseline`	A single number in the range 0.0 – 1.0 for the assumed true baseline severity estimate used to estimate the overall ascertainment ratio. Missing by default, which causes the function to error; must be supplied by the user.
`delay_density`	An optional argument that controls whether delay correction is applied in the severity estimation. May be `NULL`, for no delay correction, or a function that returns the density function of a distribution to evaluate density at user-specified values, e.g. `function(x) stats::dgamma(x = x, shape = 5, scale = 1)`.

Details

`estimate_ascertainment()` uses `cfr_static()` internally to obtain a severity estimate that is compared against the user-specified baseline severity. The profile likelihood method used to obtain the severity estimate is decided by the internal function `.estimate_severity()` as used in `cfr_static()`, when delay correction is applied. See the `cfr_static()` documentation for an explanation of the methods used depending on outbreak size and initial severity guess.

Value

A <data.frame> containing the maximum likelihood estimate estimate and 95% confidence interval of the corrected severity, named "ascertainment_estimate" (for the central estimate), and "ascertainment_low" and "ascertainment_high" for the lower and upper interval limits.

Examples

```
# get data pre-loaded with the package
data("covid_data")
df_covid_uk <- covid_data[covid_data$country == "United Kingdom", ]

df_covid_uk_subset <- subset(df_covid_uk, date <= "2020-05-31")

# use a severity baseline of 1.4% (0.014) taken from Verity et al. (2020)
# Lancet Infectious Diseases: <https://doi.org/10.1016/S1473-3099(20)30243-7>

# use onset-to-death distribution from Linton et al. (2020)
# J. Clinical Medicine: <https://doi.org/10.3390/jcm9020538>

# subset data until 30th June 2020
data <- df_covid_uk[df_covid_uk$date <= "2020-06-30", ]
estimate_ascertainment(
  data = data,
  delay_density = function(x) dlnorm(x, meanlog = 2.577, sdlog = 0.440),
  severity_baseline = 0.014
)
```

---

estimate_outcomes *Estimate known outcomes of cases using a delay distribution*

Description

Estimates the expected number of individuals with known outcomes from a case and outcome time series of outbreak data, and an epidemiological delay distribution of symptom onset to outcome. When calculating a case fatality risk, the outcomes must be deaths, the delay distribution must be an onset-to-death distribution, and the function returns estimates of the known death outcomes.

Usage

```
estimate_outcomes(data, delay_density)
```

Arguments

data	A <data.frame> containing the outbreak data. A daily time series with dates or some other absolute indicator of time (e.g. epiday or epiweek) and the numbers of new cases and new deaths at each time point. Note that the required columns are "date" (for the date), "cases" (for the number of reported cases), and "deaths" (for the number of reported deaths) on each day of the outbreak.

Note that the <data.frame> is required to have an unbroken sequence of dates with no missing dates in between. The "date" column must be of class Date (see as.Date()).

Note also that the total number of cases must be greater than the total number of reported deaths.

delay_density — An optional argument that controls whether delay correction is applied in the severity estimation. May be NULL, for no delay correction, or a function that returns the density function of a distribution to evaluate density at user-specified values, e.g. function(x) stats::dgamma(x = x, shape = 5, scale = 1).

Details

The ratio u_t represents, for the outbreak, the overall proportion of cases whose outcomes are expected to be known by each day i. For an ongoing outbreak with relatively long delays between symptom onset and case outcome, a u_t value of 1.0 may indicate that the outbreak has ended, as the outcomes of all cases are expected to be known.

Value

A <data.frame> with the columns in data, and with two additional columns:

- "estimated_outcomes" for the number of cases with an outcome of interest (usually, death) estimated to be known on the dates specified in data, and
- u_t for the ratio of cumulative number of estimated known outcomes and the cumulative number of cases reported until each date specified in data.

Examples

```
# Load Ebola 1976 outbreak data
data("ebola1976")

# estimate severity for each day while correcting for delays
# obtain onset-to-death delay distribution parameters from Barry et al. 2018
# examine the first few rows of the output
estimated_outcomes <- estimate_outcomes(
  data = ebola1976,
  delay_density = function(x) dgamma(x, shape = 2.40, scale = 3.33)
)

head(estimated_outcomes)
```

---

prepare_data *Prepare common epidemiological data formats for CFR estimation*

Description

This S3 generic has methods for classes commonly used for epidemiological data.

Currently, the only supported data format is <incidence2> from the **incidence2** package. See incidence2::incidence(). Grouped <incidence2> data are supported, see **Details**.

Usage

```
prepare_data(data, ...)

## S3 method for class 'incidence2'
prepare_data(
  data,
  cases_variable = "cases",
  deaths_variable = "deaths",
  fill_NA = TRUE,
  ...
)
```

Arguments

data	A <data.frame>-like object. Currently, only <incidence2> objects are supported. These may be grouped.
...	Currently unused. Passing extra arguments will throw a warning.
cases_variable	A string for the name of the cases variable in the "count_variable" column of data.
deaths_variable	A string for the name of the deaths variable in the "count_variable" column of data.
fill_NA	A logical indicating whether NAs in the cases and deaths data should be replaced by 0s. The default value is TRUE, with a message to make users aware of the replacement.

Details

The method for <incidence2> data can replace NAs in the case and death data with 0s using the fill_NA argument, which is TRUE by default, meaning that NAs are replaced.

Keeping NAs will cause downstream issues when calling functions such as cfr_static() on the data, as they cannot handle NAs. Setting fill_NA = TRUE resolves this issue.

Passing a grouped <incidence2> object to data will result in the function respecting the grouping and returning grouping variables in separate columns.

Value

A <data.frame> suitable for disease severity estimation functions provided in **cfr**, with the columns "date", "cases", and "deaths".

Additionally, for grouped <incidence2> data, columns representing the grouping variables will also be present.

The result has a continuous sequence of dates between the start and end date of data; this is required if the data is to be passed to functions such as cfr_static().

Examples

```
#### For <incidence2> data ####
# load Covid-19 data from incidence2
covid_uk <- incidence2::covidregionaldataUK

# convert to incidence2 object
covid_uk_incidence <- incidence2::incidence(
  covid_uk,
  date_index = "date",
  counts = c("cases_new", "deaths_new"),
  count_names_to = "count_variable"
)

# View tail of prepared data
data <- prepare_data(
  covid_uk_incidence,
  cases_variable = "cases_new",
  deaths_variable = "deaths_new"
)

tail(data)

#### For grouped <incidence2> data ####
# convert data to incidence2 object grouped by region
covid_uk_incidence <- incidence2::incidence(
  covid_uk,
  date_index = "date",
  counts = c("cases_new", "deaths_new"),
  count_names_to = "count_variable",
  groups = "region"
)

# View tail of prepared data
data <- prepare_data(
  covid_uk_incidence,
  cases_variable = "cases_new",
  deaths_variable = "deaths_new"
)

tail(data)
```

Index