

Package ‘i2extras’

October 13, 2022

Type Package

Title Functions to Work with 'incidence2' Objects

Version 0.1.2

Description Provides functions to work with 'incidence2' objects, including a simplified interface for trend fitting and peak estimation. This package is part of the RECON (<<https://www.repidemicsconsortium.org/>>) toolkit for outbreak analysis (<<https://www.reconverse.org/>>).

URL <https://www.reconverse.org/i2extras/>

BugReports <https://github.com/reconverse/i2extras/issues>

Encoding UTF-8

License MIT + file LICENSE

RoxygenNote 7.1.1

Imports utils, stats, incidence2 (>= 1.2.0), trending, dplyr, tibble, rlang, tidyr, vctrs, ellipsis, ggplot2, data.table, tidyselect

Suggests testthat, outbreaks, knitr, rmarkdown, covr

VignetteBuilder knitr

NeedsCompilation no

Author Tim Taylor [aut, cre] (<<https://orcid.org/0000-0002-8587-7113>>),
Thibaut Jombart [aut]

Maintainer Tim Taylor <tim.taylor@hiddenelephants.co.uk>

Repository CRAN

Date/Publication 2021-07-08 09:30:02 UTC

R topics documented:

add_rolling_average	2
bootstrap	3
estimate_peak	4
find_peak	6
fit_curve	7

fit_model	7
flag_low_counts	8
growth_rate	9
is_okerr	10
NA_counts_	11
plot.incidence2_fit	12
plot.incidence2_rolling	13

Index	14
--------------	-----------

add_rolling_average *Add a rolling average*

Description

`add_rolling_average()` adds a rolling average to an `incidence2::incidence()` object. If `x` is a grouped this will be a `dplyr::rowwise()` type object. If `x` is not grouped this will be a subclass of tibble.

Usage

```
add_rolling_average(x, ...)

## Default S3 method:
add_rolling_average(x, ...)

## S3 method for class 'incidence2'
add_rolling_average(x, before = 2, ...)
```

Arguments

`x` An `incidence2::incidence` object.

`...` Not currently used.

`before` how many prior dates to group the current observation with. Default is 2 days.

Value

An object of class `incidence2_rolling`.

Note

If groups are present the average will be calculated across each grouping, therefore care is required when plotting.

Author(s)

Tim Taylor

Examples

```

if (requireNamespace("outbreaks", quietly = TRUE) &&
    requireNamespace("incidence2", quietly = TRUE)) {

  data(ebola_sim_clean, package = "outbreaks")
  dat <- ebola_sim_clean$linelist
  dat <- subset(dat, date_of_onset <= as.Date("2014-10-05"))

  inci <- incidence2::incidence(dat,
                                date_index = date_of_onset,
                                interval = "week",
                                groups = gender)

  ra <- add_rolling_average(inci, before = 2)
  plot(ra, color = "white")

  inci2 <- incidence2::regroup(inci)
  ra2 <- add_rolling_average(inci2, before = 2)
  plot(ra, color = "white")

}

```

bootstrap

*Bootstrap incidence time series***Description**

This function can be used to bootstrap `incidence2::incidence` objects. Bootstrapping is done by sampling with replacement the original input dates. See details for more information on how this is implemented.

Usage

```
bootstrap(x, randomise_groups = FALSE)
```

Arguments

`x` An `incidence2::incidence` object.

`randomise_groups`

A logical indicating whether groups should be randomised as well in the re-sampling procedure; respective group sizes will be preserved, but this can be used to remove any group-specific temporal dynamics. If `FALSE` (default), data are resampled within groups.

Details

As original data are not stored in `incidence2::incidence` objects, the bootstrapping is achieved by multinomial sampling of date bins weighted by their relative incidence.

Value

An incidence2 object.

Author(s)

Thibaut Jombart, Tim Taylor

See Also

[find_peak\(\)](#) to use estimate peak date using bootstrap

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {
  data(fluH7N9_china_2013, package = "outbreaks")
  i <- incidence2::incidence(fluH7N9_china_2013, date_index = date_of_onset)
  bootstrap(i)
  bootstrap(i, randomise_groups = TRUE)
}
```

estimate_peak

Estimate the peak date of an incidence curve using bootstrap

Description

This function can be used to estimate the peak of an epidemic curve stored as [incidence2::incidence](#) object, using bootstrapped samples of the available data. See [bootstrap\(\)](#) for more information on the resampling.

Usage

```
estimate_peak(x, n = 100, alpha = 0.05, progress = TRUE)
```

Arguments

x	An incidence2::incidence object.
n	The number of bootstrap datasets to be generated; defaults to 100.
alpha	The type 1 error chosen for the confidence interval; defaults to 0.05.
progress	Should a progress bar be displayed (default = TRUE)

Details

Input dates are resampled with replacement to form bootstrapped datasets; the peak is reported for each, resulting in a distribution of peak times. When there are ties for peak incidence, only the first date is reported.

Note that the bootstrapping approach used for estimating the peak time makes the following assumptions:

- the total number of event is known (no uncertainty on total incidence)
- dates with no events (zero incidence) will never be in bootstrapped datasets
- the reporting is assumed to be constant over time, i.e. every case is equally likely to be reported

Value

A tibble with the the following columns:

- `observed_date`: the date of peak incidence of the original dataset.
- `observed_count`: the peak incidence of the original dataset.
- `estimated`: the median peak time of the bootstrap datasets.
- `lower_ci/upper_ci`: the confidence interval based on bootstrap datasets.
- `peaks`: a nested tibble containing the the peak times of the bootstrapped datasets.

Author(s)

Thibaut Jombart and Tim Taylor, with inputs on caveats from Michael Höhle.

See Also

[bootstrap\(\)](#) for the bootstrapping underlying this approach and [find_peak\(\)](#) to find the peak in a single `incidence2::incidence` object.

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {  
  
  # load data and create incidence  
  data(fluH7N9_china_2013, package = "outbreaks")  
  i <- incidence2::incidence(fluH7N9_china_2013, date_index = date_of_onset)  
  
  # find 95% CI for peak time using bootstrap  
  peak_data <- estimate_peak(i)  
  peak_data  
  summary(peak_data$peaks)  
}
```

`find_peak`*Find the peak date of an incidence curve*

Description

This function can be used to find the peak of an epidemic curve stored as an [incidence2::incidence](#) object.

Usage

```
find_peak(x)
```

Arguments

`x` An [incidence2::incidence](#) object.

Value

A tibble containing the date of the (first) highest incidence in the data along with the count. If `x` is grouped object then the output will have the peak calculated for each grouping.

Author(s)

Tim Taylor

See Also

[estimate_peak\(\)](#) for bootstrap estimates of the peak time.

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {  
  # load data and create incidence  
  data(fluH7N9_china_2013, package = "outbreaks")  
  i <- incidence2::incidence(fluH7N9_china_2013, date_index = date_of_onset)  
  find_peak(i)  
}
```

fit_curve	<i>Fit an epi curve</i>
-----------	-------------------------

Description

Fit an epi curve

Usage

```
fit_curve(x, model, ...)
```

```
## Default S3 method:
```

```
fit_curve(x, model, ...)
```

```
## S3 method for class 'incidence2'
```

```
fit_curve(x, model = "poisson", alpha = 0.05, ...)
```

Arguments

x	An incidence2::incidence object.
model	The regression model to fit (can be "poisson" or "negbin").
...	Additional arguments to pass to stats::glm() for model = "poisson" or MASS::glm.nb() for model = "negbin".
alpha	Value of alpha used to calculate confidence intervals; defaults to 0.05 which corresponds to a 95% confidence interval.

Value

An object of class `incidence2_fit`.

Author(s)

Tim Taylor

fit_model	<i>Fit a model to an incidence object</i>
-----------	---

Description

Fit a model to an incidence object

Usage

```

fit_model(x, model, ...)

## Default S3 method:
fit_model(x, model, ...)

## S3 method for class 'trending_model'
fit_model(x, model, ...)

## S3 method for class 'list'
fit_model(x, model, ...)

```

Arguments

x	An <code>incidence2::incidence</code> object.
model	Either an individual <code>trending_model</code> object generated by <code>lm_model</code> , <code>glm_model</code> , <code>glm_nb_model</code> , <code>brms_model</code> or a list of these models (see <code>?trending::trending_model()</code> for further details).
...	Not currently used.

Value

An object of class `incidence2_fit` or `incidence2_fit_list` if a list of models is given as input.

Author(s)

Tim Taylor
Tim Taylor

flag_low_counts	<i>Flag low counts and set them to NAs</i>
-----------------	--

Description

Low counts may be genuine, but they can also reflect actually missing data or strong under-reporting. This function aims to detect the latter by flagging any count below a certain threshold, expressed as a fraction of the median count. Setting low values to NAs can be useful to help fitting temporal trends to the data, as zeros / low counts can throw off some models (e.g. Negative Binomial GLMs).

Usage

```
flag_low_counts(x, counts = NULL, threshold = 0.001, set_missing = TRUE)
```


Arguments

x	An <code>incidence2::incidence</code> object.
counts	A tidyselect compliant indication of the counts to be used.
threshold	A numeric multiplier of the median count to be used as threshold. Defaults to 0.001, in which case any count strictly lower than 0.1% of the mean count is flagged as low count.
set_missing	A logical indicating if the low counts identified should be replaced with NAs (TRUE, default). If FALSE, new logical columns with the <code>flag_low</code> suffix will be added, indicating which entries are below the threshold.

Value

An `incidence2::incidence` object.

Author(s)

Tim Taylor and Thibaut Jombart

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE) &&
    requireNamespace("incidence2", quietly = TRUE)) {
  data(covid19_england_nhscalls_2020, package = "outbreaks")
  dat <- covid19_england_nhscalls_2020
  i <- incidence(dat, date, interval = "monday week", count = count)
  plot(i)
  plot(flag_low_counts(i, threshold = 0.1))
  plot(flag_low_counts(i, threshold = 1), title = "removing counts below the median")
}
```

growth_rate

Calculate growth/decay rate

Description

Calculate growth/decay rate

Usage

```
growth_rate(x, ...)

## Default S3 method:
growth_rate(x, ...)

## S3 method for class 'incidence2_fit'
growth_rate(
```

```

    x,
    alpha = 0.05,
    growth_decay_time = TRUE,
    include_warnings = FALSE,
    ...
  )

```

Arguments

x	The output of function <code>fit_curve.incidence2()</code> .
...	Not currently used.
alpha	Value of alpha used to calculate confidence intervals; defaults to 0.05 which corresponds to a 95% confidence interval.
growth_decay_time	Should a doubling/halving time and corresponding confidence intervals be added to the output. Default TRUE.
include_warnings	Include models in output that triggered warnings but not errors. Defaults to FALSE.

Author(s)

Tim Taylor

is_okerr

Error handling for incidence2_fit objects

Description

These functions are used to filter succesful model fits from those that errored or gave warnings.

Usage

```

is_ok(x, ...)

## Default S3 method:
is_ok(x, ...)

## S3 method for class 'incidence2_fit'
is_ok(x, include_warnings = FALSE, ...)

is_error(x, ...)

## Default S3 method:
is_error(x, ...)

```

```
## S3 method for class 'incidence2_fit'
is_error(x, ...)

is_warning(x, ...)

## Default S3 method:
is_warning(x, ...)

## S3 method for class 'incidence2_fit'
is_warning(x, ...)
```

Arguments

`x` The output of function `fit_curve.incidence2()`.

`...` Not currently used.

`include_warnings` Include results in output that triggered warnings but not errors. Defaults to FALSE.

Value

- `is_ok()`: returns rows from an `incidence2_fit` object that did not error (and optionally produce a warning).
- `is_error()`: returns rows from an `incidence2_fit` object that errored.
- `is_warning()`: returns rows from an `incidence2_fit` object that produced warnings.

Author(s)

Tim Taylor

NA_counts_ *Generate NAs of the right type for counts*

Description

Counts can be of type integer or double. When setting up NAs to counts, this needs to be reflected by using the right type of NAs. This function addresses this need.

Usage

```
NA_counts_(x)
```

Arguments

`x` A count vector.

Value

A NA of the type matching the input.

Author(s)

Thibaut

plot.incidence2_fit *Plot a fitted epicurve*

Description

Plot a fitted epicurve

Usage

```
## S3 method for class 'incidence2_fit'  
plot(x, cnt = NULL, include_warnings = FALSE, ci = TRUE, pi = FALSE, ...)
```

Arguments

x	An incidence2_fit object created by <code>fit_curve()</code> .
cnt	The count variable to print. If NULL defaults to the first value from <code>attr(x, "counts")</code> .
include_warnings	Include results in plot that triggered warnings but not errors. Defaults to FALSE.
ci	Plot confidence intervals (defaults to TRUE).
pi	Plot prediction intervals (defaults to FALSE).
...	Additional arguments to be passed to <code>incidence2::plot.incidence2()</code> or <code>incidence2::facet_plot()</code> .

Value

An incidence plot with the addition of a fitted curve. This will be faceted if the object is grouped.

Author(s)

Tim Taylor

```
plot.incidence2_rolling
```

Plot a rolling average incidence object

Description

Plot a rolling average incidence object

Usage

```
## S3 method for class 'incidence2_rolling'  
plot(x, cnt = NULL, ...)
```

Arguments

x	An incidence2_ra object created by add_rolling_average() .
cnt	The count variable to print. If NULL defaults to the first value from <code>attr(x, "counts")</code> .
...	Additional arguments to be passed to incidence2::plot.incidence2() or incidence2::facet_plot() .

Value

An incidence plot with the addition of a rolling average. This will be faceted if the object is grouped.

Author(s)

Tim Taylor

Index

`add_rolling_average`, 2
`add_rolling_average()`, 2, 13

`bootstrap`, 3
`bootstrap()`, 4, 5

`dplyr::rowwise()`, 2

`estimate_peak`, 4
`estimate_peak()`, 6

`find_peak`, 6
`find_peak()`, 4, 5
`fit_curve`, 7
`fit_curve()`, 12
`fit_curve.incidence2()`, 10, 11
`fit_model`, 7
`flag_low_counts`, 8

`growth_rate`, 9

`incidence2::facet_plot()`, 12, 13
`incidence2::incidence`, 2–9
`incidence2::incidence()`, 2
`incidence2::plot.incidence2()`, 12, 13
`incidence2_fit`, 11
`incidence2_fit(fit_curve)`, 7
`is_err.incidence2_fit(is_okerr)`, 10
`is_error(is_okerr)`, 10
`is_ok(is_okerr)`, 10
`is_okerr`, 10
`is_warning(is_okerr)`, 10

`MASS::glm.nb()`, 7

`NA_counts_`, 11

`plot.incidence2_fit`, 12
`plot.incidence2_rolling`, 13

`stats::glm()`, 7

`trending_model`, 8