

# Package: i2extras (via r-universe)

November 5, 2024

**Type** Package

**Title** Functions to Work with 'incidence2' Objects

**Version** 0.2.1.9000

**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/>,  
<https://github.com/reconverse/i2extras/>

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

**Encoding** UTF-8

**License** MIT + file LICENSE

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Depends** incidence2 (>= 2.0.0),

**Imports** utils, stats, MASS, ciTools, dplyr, tibble, rlang, tidyr, vctrs, ggplot2, data.table, tidyselect

**Suggests** testthat, outbreaks, knitr, rmarkdown, covr

**VignetteBuilder** knitr

**Config/pak/sysreqs** cmake make libicu-dev

**Repository** <https://reconverse.r-universe.dev>

**RemoteUrl** <https://github.com/reconverse/i2extras>

**RemoteRef** HEAD

**RemoteSha** 10fea678e96bde442324df85272b3d641f32933c

## Contents

|                               |    |
|-------------------------------|----|
| add_rolling_average . . . . . | 2  |
| bootstrap . . . . .           | 3  |
| estimate_peak . . . . .       | 4  |
| find_peak . . . . .           | 6  |
| fit_curve . . . . .           | 7  |
| flag_low_counts . . . . .     | 8  |
| growth_rate . . . . .         | 9  |
| is_okerr . . . . .            | 10 |
| plot.incidence2_fit . . . . . | 11 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>12</b> |
|--------------|-----------|

---

|                     |                              |
|---------------------|------------------------------|
| add_rolling_average | <i>Add a rolling average</i> |
|---------------------|------------------------------|

---

## Description

add\_rolling\_average() adds a rolling average to an <incidence2> object. If multiple groupings or count variables are present then the average will be calculated for each.

## Usage

```
add_rolling_average(
  x,
  n = 3L,
  complete_dates = TRUE,
  align = c("right", "center"),
  colname = "rolling_average",
  ...
)
```

## Arguments

|                |  |
|----------------|--|
| x              | [incidence2] object  |
| n              | [integer]<br>How many date groupings to consider in each window?<br>double vectors will be converted via as.integer(n).  |
| complete_dates | [bool]<br>Should incidence2::complete_dates() be called on the data prior to adding the rolling average.<br>Defaults to TRUE.  |
| align          | Character, specifying the "alignment" of the rolling window, defaulting to "right".<br>"right" covers preceding rows (the window <i>ends</i> on the current value); "left" covers following rows (the window <i>starts</i> on the current value); "center" is halfway in between (the window is <i>centered</i> on the current value, biased towards "left" when n is even). |

colname [character]  
The name of the column to contain the rolling average.  
... Other arguments passed to `incidence2::complete_dates()`

**Value**

The input object with an additional column for the rolling average.

**Examples**

```
if (requireNamespace("outbreaks", 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",
    groups = "gender",
    interval = "isoweek"
  )

  add_rolling_average(inci, n = 3L)
  inci2 <- incidence2::regroup(inci)
  add_rolling_average(inci2, n = 7L)

}
```

---

 bootstrap

*Bootstrap incidence time series*


---

**Description**

This function can be used to bootstrap `[incidence2]` objects. Bootstrapping is done by sampling with replacement the original input dates.

**Usage**

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

**Arguments**

x An `[incidence2]` object.

randomise\_groups

[bool]

Should groups be randomised as well in the resampling 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

### Examples

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

---

estimate\_peak

*Estimate the peak date of an incidence curve*

---

### Description

This function can be used to estimate the peak of an epidemic curve using bootstrapped samples of the available data.

### Usage

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

**Arguments**

|            |  |
|------------|--|
| x          | An <incidence2> object.  |
| n          | [integer]<br>The number of bootstrap datasets to be generated; defaults to 100.<br>[double] vectors will be converted via <code>as.integer(n)</code> . |
| alpha      | [numeric]<br>The type 1 error chosen for the confidence interval; defaults to 0.05.  |
| first_only | [bool]<br>Should only the first peak (by date) be kept.<br>Defaults to TRUE.   |
| progress   | [bool]<br>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 data frame 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.
- `bootstrap_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] object.

## Examples

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

---

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] object.

## Usage

```
find_peak(x, complete_dates = TRUE, ...)
```

## Arguments

x [incidence2](#) object.

complete\_dates bool.  
Should complete\_dates() be called on the data prior to keeping the first entries.  
Defaults to TRUE.

... Other arguments passed to complete\_dates().

## Value

An [incidence2] object 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.

## 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 <- incidence(fluH7N9_china_2013, date_index = "date_of_onset")  
  find_peak(i)  
}
```

---

fit\_curve

*Fit an epi curve*

---

## Description

Fit an epi curve

## Usage

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

## Arguments

|       |  |
|-------|--|
| x     | An <a href="#">incidence2::incidence</a> object.   |
| model | [character]<br>The regression model to fit (can be "poisson" or "negbin").   |
| alpha | [numeric]<br>Value of alpha used to calculate confidence intervals; defaults to 0.05 which corresponds to a 95% confidence interval.       |
| ...   | Additional arguments to pass to <a href="#">stats::glm()</a> for model = "poisson" or <a href="#">MASS::glm.nb()</a> for model = "negbin". |

## Value

An object of class `incidence2_fit`.

---

flag\_low\_counts      *Flag low counts and set them to NAs*

---

### 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 <a href="#">incidence2::incidence</a> 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 flag_low 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 = "isoweek", counts = "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 | <i>Calculate growth/decay rate</i> |
|-------------|------------------------------------|

---

**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 <code>fit_curve()</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().  |
| ...              | 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

---

plot.incidence2\_fit    *Plot a fitted epicurve*

---

**Description**

Plot a fitted epicurve

**Usage**

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

**Arguments**

|                  |   |
|------------------|---|
| x                | An incidence2_fit object created by <a href="#">fit_curve()</a> .                     |
| include_warnings | Include results in plot that triggered warnings but not errors.<br>Defaults to FALSE. |
| ci               | Plot confidence intervals.<br>Defaults to TRUE.                                       |
| pi               | Plot prediction intervals.<br>Defaults to FALSE.                                      |
| ...              | Additional arguments to be passed to <a href="#">incidence2::plot.incidence2()</a> .  |

**Value**

An incidence plot with the addition of a fitted curve.

**Author(s)**

Tim Taylor

# Index

`add_rolling_average`, 2

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

`estimate_peak`, 4  
`estimate_peaks (estimate_peak)`, 4

`find_peak`, 6  
`find_peak()`, 5  
`fit_curve`, 7  
`fit_curve()`, 11  
`flag_low_counts`, 8

`growth_rate`, 9

`incidence2`, 6  
`incidence2::incidence`, 4, 7, 8  
`incidence2::plot.incidence2()`, 11  
`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

`plot.incidence2_fit`, 11

`stats::glm()`, 7