

Package ‘incidence’

June 11, 2018

Type Package

Title Compute, Handle, Plot and Model Incidence of Dated Events

Version 1.3.1

Description Provides functions and classes to compute, handle and visualise incidence from dated events for a defined time interval. Dates can be provided in various standard formats. The class 'incidence' is used to store computed incidence and can be easily manipulated, subsetted, and plotted. In addition, log-linear models can be fitted to 'incidence' objects using 'fit'. This package is part of the RECON (<<http://www.repidemicsconsortium.org/>>) toolkit for outbreak analysis.

License MIT + file LICENSE

URL <http://www.repidemicsconsortium.org/incidence/>

BugReports <http://github.com/reconhub/incidence/issues>

RoxygenNote 6.0.1

Imports ggplot2, ISOweek

Suggests magrittr, outbreaks, testthat, vdiff, covr, knitr, rmarkdown, scales

VignetteBuilder knitr

NeedsCompilation no

Author Thibaut Jombart [aut, cre],
Rich FitzJohn [aut],
Jun Cai [ctb],
Sangeeta Bhatia [ctb]

Maintainer Thibaut Jombart <thibautjombart@gmail.com>

Repository CRAN

Date/Publication 2018-06-11 13:34:57 UTC

R topics documented:

<code>as.data.frame.incidence</code>	2
<code>cumulate</code>	3

fit	5
incidence	7
incidence_pall	10
plot.incidence	11
pool	12
subset.incidence	13

Index 15

as.data.frame.incidence

Conversion of incidence objects

Description

These functions convert incidence objects into other classes.

Usage

```
## S3 method for class 'incidence'
as.data.frame(x, ..., long = FALSE)

as.incidence(x, ...)

## S3 method for class 'matrix'
as.incidence(x, dates = NULL, interval = NULL,
             isoweeks = TRUE, ...)

## S3 method for class 'data.frame'
as.incidence(x, dates = NULL, interval = NULL,
             isoweeks = TRUE, ...)

## S3 method for class 'numeric'
as.incidence(x, dates = NULL, interval = NULL,
             isoweeks = TRUE, ...)
```

Arguments

x	An incidence object, or an object to be converted as incidence (see details).
...	Further arguments passed to other functions (no used).
long	A logical indicating if the output data.frame should be 'long', i.e. where a single column containing 'groups' is added in case of data computed on several groups.
dates	A vector of dates, each corresponding to the (inclusive) lower limit of the bins.
interval	An integer indicating the time interval used in the computation of the incidence. If NULL, it will be determined from the first time interval between provided dates. If only one date is provided, it will trigger an error.
isoweeks	A logical indicating whether isoweeks should be used in the case of weekly incidence; defaults to TRUE.

Details

Conversion to incidence objects should only be done when the original dates are not available. In such case, the argument `x` should be a matrix corresponding to the `$counts` element of an incidence object, i.e. giving counts with time intervals in rows and named groups in columns. In the absence of groups, a single unnamed columns should be given. `data.frame` and vectors will be coerced to a matrix.

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>, Rich Fitzjohn

See Also

the [incidence](#) function to generate the 'incidence' objects.

Examples

```
## create fake data
data <- c(0,1,1,2,1,3,4,5,5,5,5,4,4,26,6,7,9)
sex <- sample(c("m","f"), length(data), replace=TRUE)

## get incidence per group (sex)
i <- incidence(data, groups = sex)
i
plot(i)

## convert to data.frame
as.data.frame(i)

## same, 'long format'
as.data.frame(i, long = TRUE)

## conversion from a matrix of counts to an incidence object
i$counts
new_i <- as.incidence(i$counts, i$dates)
new_i
all.equal(i, new_i)
```

cumulate

Compute cumulative 'incidence'

Description

`cumulate` is an S3 generic to compute cumulative numbers, with methods for different types of objects:

Usage

```
cumulate(x)

## Default S3 method:
cumulate(x)

## S3 method for class 'incidence'
cumulate(x)
```

Arguments

x An incidence object.

Details

- default method is a wrapper for cumsum
- incidence objects: computes cumulative incidence over time
- projections objects: same, for projections objects, implemented in the similarly named package; see ?cumulate.projections for more information, after loading the package

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

The [incidence](#) function to generate the 'incidence' objects.

Examples

```
dat <- as.integer(c(0,1,2,2,3,5,7))
group <- factor(c(1, 2, 3, 3, 3, 3, 1))
i <- incidence(dat, groups = group)
i
plot(i)

i_cum <- cumulate(i)
i_cum
plot(i_cum)
```

Description

The function `fit` fits two exponential models to incidence data, of the form: $\log(y) = r * t + b$ where 'y' is the incidence, 't' is time (in days), 'r' is the growth rate, and 'b' is the origin. The function `fit` will fit one model by default, but will fit two models on either side of a splitting date (typically the peak of the epidemic) if the argument `split` is provided. The function `fit_optim_split` can be used to find the optimal 'splitting' date, defined as the one for which the best average R2 of the two models is obtained. Plotting can be done using `plot`, or added to an existing incidence plot by the piping-friendly function `add_incidence_fit`.

Usage

```
fit(x, split = NULL, level = 0.95, quiet = FALSE)

fit_optim_split(x, window = x$timespan/4, plot = TRUE, quiet = TRUE)

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

add_incidence_fit(p, x, col_pal = incidence_pal1)

## S3 method for class 'incidence_fit'
plot(x, ...)
```

Arguments

<code>x</code>	An incidence object, generated by the function <code>incidence</code> . For the plotting function, an <code>incidence_fit</code> object.
<code>split</code>	An optional time point identifying the separation between the two models. If <code>NULL</code> , a single model is fitted. If provided, two models would be fitted on the time periods on either side of the split.
<code>level</code>	The confidence interval to be used for predictions; defaults to 95%.
<code>quiet</code>	A logical indicating if warnings from <code>fit</code> should be hidden; <code>FALSE</code> by default. Warnings typically indicate some zero incidence, which are removed before performing the log-linear regression.
<code>window</code>	The size, in days, of the time window either side of the split.
<code>plot</code>	A logical indicating whether a plot should be added to the output, showing the mean R2 for various splits.
<code>...</code>	further arguments passed to other methods (not used)
<code>p</code>	An existing incidence plot.
<code>col_pal</code>	A color palette, defaulting

Value

For `fit`, a list with the class `incidence_fit` (for a single model), or a list containing two `incidence_fit` objects (when fitting two models). `incidence_fit` objects contain:

- `lm`: the fitted linear model
- `info`: a list containing various information extracted from the model (detailed further)
- `origin`: the date corresponding to day '0'

The `$info` item is a list containing:

- `r`: the growth rate
- `r.conf`: the confidence interval of 'r'
- `pred`: a `data.frame` containing predictions of the model, including the true dates (`dates`), their numeric version used in the model (`dates.x`), the predicted value (`fit`), and the lower (`lwr`) and upper (`upr`) bounds of the associated confidence interval.
- `doubling`: the predicted doubling time in days; exists only if 'r' is positive
- `doubling.conf`: the confidence interval of the doubling time
- `halving`: the predicted halving time in days; exists only if 'r' is negative
- `halving.conf`: the confidence interval of the halving time

For `fit_optim_split`, a list containing:

- `df`: a `data.frame` of dates that were used in the optimization procedure, and the corresponding average R2 of the resulting models.
- `split`: the optimal splitting date
- `fit`: the resulting `incidence_fit` objects
- `plot`: a plot showing the content of `df` (`ggplot2` object)

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

the [incidence](#) function to generate the 'incidence' objects.

Examples

```
if (require(outbreaks)) {
  dat <- ebola_sim$linelist$date_of_onset

  ## EXAMPLE WITH A SINGLE MODEL

  ## compute weekly incidence
  i.7 <- incidence(dat, interval=7)
  plot(i.7)
```

```

plot(i.7[1:20])

## fit a model on the first 20 weeks
f <- fit(i.7[1:20])
f
names(f)
head(f$pred)

## plot model alone (not recommended)
plot(f)

## plot data and model (recommended)
plot(i.7, fit = f)
plot(i.7[1:25], fit = f)

## piping versions
if (require(magrittr)) {
  plot(i.7) %>% add_incidence_fit(f)

  ## EXAMPLE WITH 2 PHASES
  ## specifying the peak manually
  f2 <- fit(i.7, split = as.Date("2014-10-15"))
  f2
  plot(i.7) %>% add_incidence_fit(f2)

  ## finding the best 'peak' date
  f3 <- fit_optim_split(i.7)
  f3
  plot(i.7) %>% add_incidence_fit(f3$fit)
}
}

```

incidence

Compute incidence of events from a vector of dates.

Description

This function computes incidence based on dates of events provided in various formats. A fixed interval, provided as numbers of days, is used to define time intervals. Counts within an interval always include the first date, after which they are labelled, and exclude the second. For instance, intervals labelled as 0, 3, 6, ... mean that the first bin includes days 0, 1 and 2, the second interval includes 3, 4 and 5 etc.

Usage

```

incidence(dates, interval = 1L, ...)

## S3 method for class 'integer'

```

```

incidence(dates, interval = 1L, groups = NULL,
  na_as_group = TRUE, last_date = NULL, ...)

## Default S3 method:
incidence(dates, interval = 1L, groups = NULL,
  na_as_group = TRUE, last_date = NULL, ...)

## S3 method for class 'numeric'
incidence(dates, interval = 1L, ...)

## S3 method for class 'Date'
incidence(dates, interval = 1L, iso_week = TRUE,
  last_date = NULL, ...)

## S3 method for class 'POSIXt'
incidence(dates, interval = 1L, ...)

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

```

Arguments

dates	A vector of dates, which can be provided as objects of the class: integer, numeric, Date, POSIXct. Note that decimal numbers will be floored with a warning.
interval	An integer indicating the (fixed) size of the time interval used for computing the incidence; defaults to 1 day.
...	Additional arguments passed to other methods (none are used).
groups	An optional factor defining groups of observations for which incidence should be computed separately.
na_as_group	A logical value indicating if missing group (NA) should be treated as a separate group.
last_date	The last date to be included in the produced epicurve. If NULL (default), the last date will be the most recent provided in dates.
iso_week	A logical value indicating if dates of the weekly incidence should be using ISO week. Only applies when interval = 7. Defaults to be TRUE.
x	An 'incidence' object.

Details

For details about the `incidence` class, see the dedicated vignette:
`vignette("incidence_class", package = "incidence")`

Value

An list with the class `incidence`, which contains the following items:

- **dates**: The dates marking the left side of the bins used for counting events. When ISO week-based weekly incidence is computed, the dates are the first days of corresponding isoweeks.
- **counts**: A matrix of incidence counts, which one column per group (and a single column if no groups were used).
- **timespan**: The length of the period for which incidence is computed, in days.
- **interval**: The bin size, in number of days; e.g. 7 indicates weekly incidence.
- **n**: The total number of cases.
- **isoweeks**: ISO 8601 week format yyyy-Www, which is returned only when ISO week-based weekly incidence is computed.

Author(s)

Thibaut Jombart, Rich Fitzjohn

See Also

The main other functions of the package include:

- **fit**: Fit log-linear model to computed incidence.
- **fit_optim_split**: Find the optimal peak of the epidemic and fits log-linear models on either side of the peak.
- **subset**: Handling of incidence objects.
- **pool**: Sum incidence over groups.
- **as.data.frame**: Convert an incidence object to a `data.frame`.

The following vignettes are also available:

- **overview**: Provides an overview of the package's features.
- **customize_plot**: Provides some tips on finer plot customization.
- **incidence_class**: Details the content of the incidence class.

Examples

```
## toy example
incidence(c(1, 5, 8, 3, 7, 2, 4, 6, 9, 2))
incidence(c(1, 5, 8, 3, 7, 2, 4, 6, 9, 2), 2)

## example using simulated dataset
if(require(outbreaks)) {
  onset <- ebola_sim$linelist$date_of_onset

  ## daily incidence
  inc <- incidence(onset)
  inc
  plot(inc)

  ## weekly incidence
  inc.week <- incidence(onset, interval = 7, iso_week = FALSE)
```

```
inc.week
plot(inc.week)
plot(inc.week, border = "white") # with visible border
inc.isoweek <- incidence(onset, interval = 7, iso_week = TRUE)
inc.isoweek
## use group information
sex <- ebola_sim$linelist$gender
inc.week.gender <- incidence(onset, interval = 7, groups = sex, iso_week = FALSE)
inc.week.gender
head(inc.week.gender$counts)
plot(inc.week.gender)
inc.isoweek.gender <- incidence(onset, interval = 7, groups = sex, iso_week = TRUE)
inc.isoweek.gender
}
```

incidence_pal1

Color palettes used in incidence

Description

These functions are color palettes used in incidence.

Usage

```
incidence_pal1(n)
```

```
incidence_pal1_light(n)
```

```
incidence_pal1_dark(n)
```

Arguments

n a number of colors

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Examples

```
plot(1:4, cex=8, pch=20, col = incidence_pal1(4),
     main = "palette: incidence_pal1")
plot(1:100, cex=8, pch=20, col = incidence_pal1(100),
     main = "palette: incidence_pal1")
plot(1:100, cex=8, pch=20, col = incidence_pal1_light(100),
     main = "palette: incidence_pal1_light")
```

```
plot(1:100, cex=8, pch=20, col = incidence_pal1_dark(100),
     main="palette: incidence_pal1_dark")
```

plot.incidence	<i>Plot function for incidence objects</i>
----------------	--

Description

This function is used to visualise the output of the [incidence](#) function, using the package `ggplot2`.

Usage

```
## S3 method for class 'incidence'
plot(x, ..., fit = NULL, stack = is.null(fit),
     color = "black", border = NA, col_pal = incidence_pal1, alpha = 0.7,
     xlab = "", ylab = NULL, labels_iso_week = !is.null(x$isoweeks),
     n_breaks = 6)
```

Arguments

<code>x</code>	An incidence object, generated by the function incidence .
<code>...</code>	Further arguments passed to other methods (currently not used).
<code>fit</code>	An 'incidence_fit' object as returned by fit .
<code>stack</code>	A logical indicating if bars of multiple groups should be stacked, or displayed side-by-side.
<code>color</code>	The color to be used for the filling of the bars; NA for invisible bars; defaults to "black".
<code>border</code>	The color to be used for the borders of the bars; NA for invisible borders; defaults to NA.
<code>col_pal</code>	The color palette to be used for the groups; defaults to <code>incidence_pal1</code> . See incidence_pal1 for other palettes implemented in <code>incidence</code> .
<code>alpha</code>	The alpha level for color transparency, with 1 being fully opaque and 0 fully transparent; defaults to 0.7.
<code>xlab</code>	The label to be used for the x-axis; empty by default.
<code>ylab</code>	The label to be used for the y-axis; by default, a label will be generated automatically according to the time interval used in <code>incidence</code> computation.
<code>labels_iso_week</code>	a logical value indicating whether labels x axis tick marks are in ISO 8601 week format <code>yyyy-Www</code> when plotting ISO week-based weekly incidence; defaults to be TRUE.
<code>n_breaks</code>	the ideal number of breaks to be used for the x-axis labelling

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

The [incidence](#) function to generate the 'incidence' objects.

Examples

```
if(require(outbreaks)) {
  onset <- ebola_sim$linelist$date_of_onset

  ## daily incidence
  inc <- incidence(onset)
  inc
  plot(inc)

  ## weekly incidence
  inc.week <- incidence(onset, interval = 7)
  inc.week
  plot(inc.week) # default to label x axis tick marks with isoweeks
  plot(inc.week, labels_iso_week = FALSE) # label x axis tick marks with dates
  plot(inc.week, border = "white") # with visible border

  ## use group information
  sex <- ebola_sim$linelist$gender
  inc.week.gender <- incidence(onset, interval = 7, groups = sex)
  plot(inc.week.gender)
  plot(inc.week.gender, labels_iso_week = FALSE)

  ## adding fit
  fit <- fit_optim_split(inc.week.gender)$fit
  plot(inc.week.gender, fit = fit)
  plot(inc.week.gender, fit = fit, labels_iso_week = FALSE)
}
```

pool

Pool 'incidence' across groups

Description

This function pools incidence across all groups of an incidence object. The resulting [incidence](#) object will contains counts summed over all groups present in the input.

Usage

pool(x)

Arguments

x An 'incidence' object.

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

The [incidence](#) function to generate the 'incidence' objects.

Examples

```
dat <- as.integer(c(0,1,2,2,3,5,7))
group <- factor(c(1, 2, 3, 3, 3, 3, 1))
i <- incidence(dat, groups = group)
i
i$counts

## pool all groups
pool(i)
pool(i)$counts

## pool only groups 1 and 3
pool(i[,c(1,3)])
pool(i[,c(1,3)])$counts
```

subset.incidence *Subsetting 'incidence' objects*

Description

Two functions can be used to subset incidence objects. The function `subset` permits to retain dates within a specified range and, optionally, specific groups. The operator `"["` can be used as for matrices, using the syntax `x[i, j]` where `'i'` is a subset of dates, and `'j'` is a subset of groups.

Usage

```
## S3 method for class 'incidence'
subset(x, ..., from = min(x$dates), to = max(x$dates),
       groups = TRUE)

## S3 method for class 'incidence'
x[i, j]
```

Arguments

x	An incidence object, generated by the function incidence .
...	Further arguments passed to other methods (not used).
from	The starting date; data strictly before this date are discarded.
to	The ending date; data strictly after this date are discarded.
groups	(optional) The groups to retained, indicated as subsets of the columns of x\$counts.
i	a subset of dates to retain
j	a subset of groups to retain

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

The [incidence](#) function to generate the 'incidence' objects.

Examples

```
## example using simulated dataset
if(require(outbreaks)) {
  onset <- ebola_sim$linelist$date_of_onset

  ## weekly incidence
  inc <- incidence(onset, interval = 7)
  inc
  inc[1:10] # first 10 weeks
  plot(inc[1:10])
  inc[-c(11:15)] # remove weeks 11-15
  plot(inc[-c(11:15)])
}
```

Index

[.incidence (subset.incidence), 13

add_incidence_fit (fit), 5
as.data.frame, 9
as.data.frame.incidence, 2
as.incidence (as.data.frame.incidence),
2

cumulate, 3

fit, 5, 9, 11
fit_optim_split, 9
fit_optim_split (fit), 5

incidence, 3–6, 7, 11–14
incidence_pal1, 10, 11
incidence_pal1_dark (incidence_pal1), 10
incidence_pal1_light (incidence_pal1),
10

palettes (incidence_pal1), 10
plot.incidence, 11
plot.incidence_fit (fit), 5
pool, 9, 12
print.incidence (incidence), 7
print.incidence_fit (fit), 5

subset, 9
subset.incidence, 13