

Package ‘incidence2’

September 22, 2020

Type Package

Title Compute, Handle and Plot Incidence of Dated Events

Version 0.2.0

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 'incidence2' is used to store computed incidence and can be easily manipulated, subsetted, and plotted. This package is part of the RECON (<<https://www.repidemicsconsortium.org/>>) toolkit for outbreak analysis.

Encoding UTF-8

License MIT + file LICENSE

URL <https://github.com/reconhub/incidence2>

BugReports <https://github.com/reconhub/incidence2/issues>

RoxygenNote 7.1.1

Imports ggplot2, aweek (>= 0.2.0), rlang, dplyr (>= 1.0.0), tibble, ellipsis, vctrs, magrittr, pillar, data.table

Suggests outbreaks, testthat, knitr, rmarkdown, covr

VignetteBuilder knitr

NeedsCompilation no

Author Tim Taylor [aut, cre] (<<https://orcid.org/0000-0002-8587-7113>>),
Thibaut Jombart [aut],
Zhian N. Kamvar [aut] (<<https://orcid.org/0000-0003-1458-7108>>),
Rich FitzJohn [aut],
Jun Cai [ctb] (<<https://orcid.org/0000-0001-9495-1226>>),
Sangeeta Bhatia [ctb],
Jakob Schumacher [ctb],
Juliet R.C. Pulliam [ctb] (<<https://orcid.org/0000-0003-3314-8223>>)

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

Repository CRAN

Date/Publication 2020-09-22 09:50:03 UTC

R topics documented:

accessors	2
as.data.frame.incidence2	5
as_tibble	5
cumulate	6
incidence	7
plot.incidence2	12
print.incidence2	15
regroup	15
summary.incidence2	16
vibrant	17
Index	18

accessors	<i>Access various elements of an incidence object</i>
-----------	---

Description

Access various elements of an incidence object

Usage

```
get_counts(x, ...)  
  
## Default S3 method:  
get_counts(x, ...)  
  
## S3 method for class 'incidence2'  
get_counts(x, ...)  
  
get_counts_name(x, ...)  
  
## Default S3 method:  
get_counts_name(x, ...)  
  
## S3 method for class 'incidence2'  
get_counts_name(x, ...)  
  
get_date_group_names(x, ...)  
  
## Default S3 method:  
get_date_group_names(x, ...)  
  
## S3 method for class 'incidence2'  
get_date_group_names(x, ...)
```

```
get_dates(x, ...)

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

## S3 method for class 'incidence2'
get_dates(x, ...)

get_dates_name(x, ...)

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

## S3 method for class 'incidence2'
get_dates_name(x, ...)

get_group_names(x, ...)

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

## S3 method for class 'incidence2'
get_group_names(x, ...)

get_interval(x, ...)

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

## S3 method for class 'incidence2'
get_interval(x, integer = FALSE, ...)

get_n(x)

## Default S3 method:
get_n(x)

## S3 method for class 'incidence2'
get_n(x)

get_timespan(x, ...)

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

## S3 method for class 'incidence2'
get_timespan(x, ...)
```

Arguments

<code>x</code>	An <code>incidence()</code> object.
<code>...</code>	Not used.
<code>integer</code>	When TRUE, the interval will be converted to an integer vector if it is stored as a character in the incidence object.

Value

- `get_counts()`: The count vector from `x`.
- `get_counts_name()`: The name of the count variable of `x`.
- `get_date_group_names()`: The names of the date group variables of `x`.
- `get_dates()`: The date vector from `x`.
- `get_dates_name()`: The name of the date variable of `x`.
- `get_group_names()`: a character vector of the group variables of `x` or NULL if none are present.
- `get_interval()`: if `integer = TRUE`, an integer vector, otherwise the character value of the interval
- `get_n()` The total number of cases stored in the object
- `get_timespan()`: an integer denoting the timespan in days represented by the incidence object.

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {
  withAutoprint({
    data(ebola_sim_clean, package = "outbreaks")
    dat <- ebola_sim_clean$linelist
    i <- incidence(dat,
                  date_index = date_of_onset,
                  groups = c(gender, hospital))

    get_counts(i)
    get_counts_name(i)

    get_group_names(i)

    get_dates(i)
    get_dates_name(i)

    get_date_group_names(i)

    get_interval(i)
  })
}
```

```

    get_n(i)
    get_timespan(i)
  })
}

```

as.data.frame.incidence2

Convert incident object to dataframe

Description

Convert incident object to dataframe

Usage

```

## S3 method for class 'incidence2'
as.data.frame(x, ...)

```

Arguments

x	An incidence() object.
...	Not used.

Examples

```

dat <- data.frame(dates = Sys.Date() + 1:100,
                  names = rep(c("Jo", "John"), 5))

dat <- incidence(dat, date_index = dates, groups = names)
as.data.frame(dat)

```

as_tibble

Convert incident2 object to a tibble

Description

Convert incident2 object to a tibble

Usage

```

## S3 method for class 'incidence2'
as_tibble(x, ...)

```

Arguments

`x` An `incidence()` object.

`...` Not used.

Examples

```
dat <- data.frame(dates = Sys.Date() + 1:100,
                 names = rep(c("Jo", "John"), 5))

dat <- incidence(dat, date_index = dates, groups = names)
as_tibble(dat)
```

cumulate	<i>Compute cumulative 'incidence'</i>
----------	---------------------------------------

Description

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

- default method is a wrapper for `cumsum`
- incidence objects: computes cumulative incidence over time

Usage

```
cumulate(x)

## Default S3 method:
cumulate(x)

## S3 method for class 'incidence2'
cumulate(x)
```

Arguments

`x` An incidence object.

Author(s)

Thibaut Jombart

Examples

```

dat <- data.frame(
  dates = as.integer(c(0,1,2,2,3,5,7)),
  groups = factor(c(1, 2, 3, 3, 3, 3, 1))
)

i <- incidence(dat, date_index = dates, groups = groups)
i

cumulative_i <- cumulate(i)
cumulative_i

```

incidence	<i>Compute the incidence of events</i>
-----------	--

Description

Compute the incidence of events

Usage

```

incidence(x, date_index, interval = 1L, ...)

## Default S3 method:
incidence(x, date_index, interval = 1L, ...)

## S3 method for class 'Date'
incidence(
  x,
  date_index,
  interval = 1L,
  standard = TRUE,
  groups = NULL,
  na_as_group = TRUE,
  first_date = NULL,
  last_date = NULL,
  count = NULL,
  ...
)

## S3 method for class 'character'
incidence(
  x,
  date_index,
  interval = 1L,
  standard = TRUE,
  groups = NULL,

```

```
    na_as_group = TRUE,
    first_date = NULL,
    last_date = NULL,
    count = NULL,
    ...
)

## S3 method for class 'integer'
incidence(
  x,
  date_index,
  interval = 1L,
  groups = NULL,
  na_as_group = TRUE,
  first_date = NULL,
  last_date = NULL,
  count = NULL,
  ...
)

## S3 method for class 'numeric'
incidence(
  x,
  date_index,
  interval = 1L,
  groups = NULL,
  na_as_group = TRUE,
  first_date = NULL,
  last_date = NULL,
  count = NULL,
  ...
)

## S3 method for class 'POSIXt'
incidence(
  x,
  date_index,
  interval = 1L,
  standard = TRUE,
  groups = NULL,
  na_as_group = TRUE,
  first_date = NULL,
  last_date = NULL,
  count = NULL,
  ...
)
```


Arguments

<code>x</code>	A tibble or a data frame (see Note) representing a linelist.
<code>date_index</code>	The time index of the given data. This should be the name, with or without quotation, corresponding to a date column in <code>x</code> of the class: integer, numeric, Date, POSIXct, POSIXlt, and character. (See Note about numeric and character formats)
<code>interval</code>	An integer or character indicating the (fixed) size of the time interval used for computing the incidence; defaults to 1 day. This can also be a text string that corresponds to a valid date interval: day, week, month, quarter, or year. (See Note).
<code>...</code>	Additional arguments used by other methods.
<code>standard</code>	(Only applicable where <code>date_index</code> references a Date object) When TRUE (default) and the <code>interval</code> one of "week", "month", "quarter", or "year", then this will cause the bins for the counts to start at the beginning of the interval (See Note).
<code>groups</code>	An optional vector giving the names of the groups of observations for which incidence should be grouped. This can be given with or without quotation.
<code>na_as_group</code>	A logical value indicating if missing group values (NA) should be treated as a separate category (TRUE) or removed from consideration (FALSE).
<code>first_date, last_date</code>	optional first/last dates to be used. When these are NULL (default), the dates from the first/last dates are taken from the observations. If these dates are provided, the observations will be trimmed to the range of <code>[first_date, last_date]</code> .
<code>count</code>	The count variable of the given data. If NULL (default) the data is taken to be a linelist of individual observations.

Value

An `incidence2` object. This is a subclass of tibble that represents and aggregated count of observations grouped according to the specified interval and, optionally, the given groups. By default it will contain the following columns:

- **bin_date**: The dates marking the left side of the bins used for counting events. When `standard = TRUE` and the `interval` represents weeks, months, quarters, or years, the first date will represent the first standard date (See Interval specification, below).
- **-groups-**: If specified, column(s) containing the categories of the given groups.
- **count**: The aggregated observation count.

If a "week" interval is specified then the object may also contain additional columns:

- **weeks**: Dates in week format (YYYY-Www), where YYYY corresponds to the year of the given week and ww represents the numeric week of the year. This will be produced from the function `aweek::date2week()`. Note that these will have a special "week_start" attribute indicating which day of the ISO week the week starts on (see Weeks, below).

Note**Input data (dates):**

- **Decimal (numeric) dates:** will be truncated with a warning
- **Character dates** should be in the unambiguous yyyy-mm-dd (ISO 8601) format. Any other format will trigger an error.

Interval specification (interval): If interval is a valid character (e.g. "week" or "1 month"), then the bin will start at the beginning of the interval just before the first observation by default. For example, if the first case was recorded on Wednesday, 2018-05-09:

- "week" : first day of the week (i.e. Monday, 2018-05-07) (defaults to ISO weeks, see "Week intervals", below)
- "month" : first day of the month (i.e. 2018-05-01)
- "quarter" : first day of the quarter (i.e. 2018-04-01)
- "year" : first day of the calendar year (i.e. 2018-01-01)

These default intervals can be overridden with `standard = FALSE`, which sets the interval to begin at the first observed case.

Week intervals:

It is possible to construct standardized incidence objects standardized to any day of the week thanks to the `awEEK::date2week()` function from the **awEEK** package. The default state is to use ISO 8601 definition of weeks, which start on Monday. You can specify the day of the week an incidence object should be standardised to by using the pattern "n W weeks" where "W" represents the weekday in an English or current locale and "n" represents the duration, but this can be omitted. Below are examples of specifying weeks starting on different days assuming we had data that started on 2016-09-05, which is ISO week 36 of 2016:

- `interval = "2 monday weeks"` (Monday 2016-09-05)
- `interval = "1 tue week"` (Tuesday 2016-08-30)
- `interval = "1 Wed week"` (Wednesday 2016-08-31)
- `interval = "1 Thursday week"` (Thursday 2016-09-01)
- `interval = "1 F week"` (Friday 2016-09-02)
- `interval = "1 Saturday week"` (Saturday 2016-09-03)
- `interval = "Sunday week"` (Sunday 2016-09-04)

It's also possible to use something like "3 weeks: Saturday"; In addition, there are keywords reserved for specific days of the week:

- `interval = "week", standard = TRUE` (Default, Monday)
- `interval = "ISOweek"` (Monday)
- `interval = "EPIweek"` (Sunday)
- `interval = "MMWRweek"` (Sunday)

The "EPIweek" specification is not strictly reserved for CDC epiweeks, but can be prefixed (or posfixed) by a day of the week: "1 epiweek: Saturday".

The intervals for "month", "quarter", and "year" will necessarily vary in the number of days they encompass and warnings will be generated when the first date falls outside of a calendar date that is easily represented across the interval.

Author(s)

Thibaut Jombart, Rich Fitzjohn, Zhian Kamvar, Tim Taylor

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {
  withAutoprint({
    data(ebola_sim_clean, package = "outbreaks")
    dat <- ebola_sim_clean$linelist

    # daily incidence
    dat %>%
      incidence(date_of_onset)

    # weekly incidence
    dat %>%
      incidence(date_of_onset, interval = "week", standard = FALSE)

    # starting on a Monday
    dat %>%
      incidence(date_of_onset, interval = "isoweek")

    # starting on a Sunday
    dat %>%
      incidence(date_of_onset, interval = "epiweek")

    # starting on a Saturday
    dat %>%
      incidence(date_of_onset, interval = "saturday epiweek")

    # group by gender
    dat %>%
      incidence(date_of_onset, interval = 7, groups = gender)

    # group by gender and hospital
    dat %>%
      incidence(date_of_onset,
                interval = "2 weeks",
                groups = c(gender, hospital))
  })
}

# use of first_date
dat <- data.frame(dates = Sys.Date() + sample(-3:10, 10, replace = TRUE))
dat %>% incidence(dates,
                  interval = "week",
                  first_date = Sys.Date() + 1)
```

plot.incidence2	<i>Plotting functions</i>
-----------------	---------------------------

Description

incidence2 includes two plotting functions to simplify graph creation.

Usage

```
## S3 method for class 'incidence2'
plot(
  x,
  fill = NULL,
  stack = TRUE,
  title = NULL,
  col_pal = vibrant,
  alpha = 0.7,
  color = NA,
  xlab = "",
  ylab = NULL,
  n_breaks = 6,
  show_cases = FALSE,
  border = "white",
  na_color = "grey",
  group_labels = TRUE,
  centre_ticks = FALSE,
  legend = c("right", "left", "bottom", "top", "none"),
  angle = 0,
  format = NULL,
  ...
)

facet_plot(x, ...)

## S3 method for class 'incidence2'
facet_plot(
  x,
  facets = NULL,
  stack = TRUE,
  fill = NULL,
  title = NULL,
  col_pal = vibrant,
  alpha = 0.7,
  color = NA,
  xlab = "",
  ylab = NULL,
  n_breaks = 3,
```

```

    show_cases = FALSE,
    border = "white",
    na_color = "grey",
    group_labels = TRUE,
    centre_ticks = FALSE,
    legend = c("bottom", "top", "left", "right", "none"),
    angle = 0,
    format = NULL,
    nrow = NULL,
    ...
)

scale_x_incidence(
  x,
  n_breaks = 6,
  group_labels = TRUE,
  format = NULL,
  angle = 0,
  size = NULL,
  coord_equal = FALSE,
  ...
)

```

Arguments

<code>x</code>	An <code>incidence()</code> object.
<code>fill</code>	Which variable to color plots by. If NULL no distinction is made for plot colors.
<code>stack</code>	A logical indicating if bars of multiple groups should be stacked, or displayed side-by-side. Only used if <code>fill</code> is not NULL.
<code>title</code>	Optional title for the graph.
<code>col_pal</code>	<code>col_pal</code> The color palette to be used for the groups; defaults to vibrant (see <code>?palettes</code>).
<code>alpha</code>	The alpha level for color transparency, with 1 being fully opaque and 0 fully transparent; defaults to 0.7.
<code>color</code>	The color to be used for the borders of the bars; NA for invisible borders; defaults to NA.
<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 incidence computation.
<code>n_breaks</code>	<code>n_breaks</code> the ideal number of breaks to be used for the x-axis labeling
<code>show_cases</code>	if TRUE (default: FALSE), then each observation will be colored by a border. The border defaults to a white border unless specified otherwise. This is normally used outbreaks with a small number of cases. Note: this can only be used if <code>stack = TRUE</code>
<code>border</code>	If <code>show_cases</code> is TRUE this represents the color used for the borders of the individual squares plotted (defaults to "white").


```

    plot(inci)
    plot(inci, fill = hospital)
    plot(inci, fill = hospital, stack = FALSE)

    facet_plot(inci)
    facet_plot(inci2)
    facet_plot(inci2, facets = gender)
    facet_plot(inci2, facets = hospital, fill = gender)
  })
}

```

print.incidence2	<i>Print an incidence object.</i>
------------------	-----------------------------------

Description

Print an incidence object.

Usage

```
## S3 method for class 'incidence2'
print(x, ...)
```

Arguments

x	An 'incidence2' object.
...	Not used.

regroup	<i>Regroup 'incidence' objects</i>
---------	------------------------------------

Description

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

Usage

```
regroup(x, groups = NULL)
```

Arguments

x	An incidence() object.
groups	The groups to sum over. If NULL (default) then the function ignores all groups.

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {  
  withAutoprint({  
    data(ebola_sim_clean, package = "outbreaks")  
    dat <- ebola_sim_clean$linelist  
    i <- incidence(dat,  
                   date_index = date_of_onset,  
                   groups = c(gender, hospital))  
  
    i %>% regroup()  
  
    i %>% regroup(hospital)  
  })  
}
```

summary.incidence2	<i>Summary of a given incidence object</i>
--------------------	--

Description

Summary of a given incidence object

Usage

```
## S3 method for class 'incidence2'  
summary(object, ...)
```

Arguments

object	An 'incidence' object.
...	Not used.

Value

object (invisibly).

`vibrant`*Color palettes used in incidence*

Description

These functions are color palettes used in incidence. The palettes come from <https://personal.sron.nl/~pault/#sec:qualitative> and exclude grey, which is reserved for missing data.

Usage`vibrant(n)``muted(n)`**Arguments**

<code>n</code>	a number of colors
----------------	--------------------

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Examples

```
vibrant(5)
muted(10)
```

Index

accessors, [2](#)
as.data.frame.incidence2, [5](#)
as_tibble, [5](#)
awEEK::date2week(), [9](#), [10](#)

cumulate, [6](#)

facet_plot (plot.incidence2), [12](#)

get_counts (accessors), [2](#)
get_counts_name (accessors), [2](#)
get_date_group_names (accessors), [2](#)
get_dates (accessors), [2](#)
get_dates_name (accessors), [2](#)
get_group_names (accessors), [2](#)
get_interval (accessors), [2](#)
get_n (accessors), [2](#)
get_timespan (accessors), [2](#)
ggplot2::ggplot(), [14](#)
ggplot2::scale_x_continuous(), [14](#)
ggplot2::scale_x_date(), [14](#)
ggplot2::scale_x_datetime(), [14](#)

incidence, [7](#)
incidence(), [4–6](#), [13–15](#)

muted (vibrant), [17](#)

palettes (vibrant), [17](#)
plot(), [14](#)
plot.incidence2, [12](#)
print.incidence2, [15](#)

regroup, [15](#)

scale_x_incidence (plot.incidence2), [12](#)
summary.incidence2, [16](#)

vibrant, [17](#)