

# Package ‘incidence2’

March 17, 2023

**Type** Package

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

**Version** 2.0.0

**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 (<<https://www.reconverse.org/>>).

**Encoding** UTF-8

**License** MIT + file LICENSE

**URL** <https://github.com/reconverse/incidence2>

**BugReports** <https://github.com/reconverse/incidence2/issues>

**Depends** grates (>= 1.0.0), R (>= 3.5.0)

**Imports** grDevices, data.table, pillar, utils

**RoxygenNote** 7.2.3

**Suggests** outbreaks, dplyr, ggplot2, knitr, rmarkdown, rlang, clock, testthat (>= 3.0.0), tibble

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**LazyData** true

**NeedsCompilation** no

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

**Maintainer** Tim Taylor <[tim.taylor@hiddenelephants.co.uk](mailto:tim.taylor@hiddenelephants.co.uk)>

**Repository** CRAN

**Date/Publication** 2023-03-17 15:00:02 UTC

## R topics documented:

accessors	2
as.data.frame.incidence2	4
as_incidence	5
complete_dates	6
covidregionaldataUK	7
cumulate	8
incidence	8
keep	11
plot.incidence2	12
print.incidence2	14
regroup	14
summary.incidence2	15
vibrant	16
<b>Index</b>	<b>17</b>

---

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

---

### Description

Access various elements of an incidence object

### Usage

```
get_date_index_name(x, ...)
```

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

```
get_date_index_name(x, ...)
```

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

```
get_date_index_name(x, ...)
```

```
get_dates_name(x, ...)
```

```
get_count_variable_name(x, ...)
```

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

```
get_count_variable_name(x, ...)
```

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

```
get_count_variable_name(x, ...)
```

```
get_count_value_name(x, ...)
```

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

```
get_count_value_name(x, ...)

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

get_group_names(x, ...)

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

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

get_date_index(x, ...)

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

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

get_dates(x, ...)

get_count_variable(x, ...)

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

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

get_count_value(x, ...)

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

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

get_groups(x, ...)

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

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

**Arguments**

x                    An R object.  
 ...                  Not currently used.

**Value**

- `get_date_index_name()`: The name of the `date_index` variable of `x`.
- `get_dates_name()`: Alias for `get_date_index_name()`.
- `get_count_variable_name()`: The name of the count variable of `x`.
- `get_count_value_name()`: The name of the count value of `x`.
- `get_group_names()`: The name(s) of the group variable(s) of `x`.
- `get_date_index()`: The `date_index` variable of `x`.
- `get_dates()`: Alias for `get_date_index()`.
- `get_count_variable()`: The count variable of `x`.
- `get_count_value()`: The count value of `x`.
- `get_groups()`: List of the group variable(s) of `x`.

**Examples**

```
if (requireNamespace("outbreaks", quietly = TRUE)) {

  data(ebola_sim_clean, package = "outbreaks")
  dat <- ebola_sim_clean$linelist
  i <- incidence(dat, date_index = "date_of_onset",
                groups = c("gender", "hospital"))

  get_count_variable_name(i)
  get_group_names(i)
  get_dates_name(i)

}
```

---

as.data.frame.incidence2

*Convert incident object to a data frame*

---

**Description**

Convert incident object to a data frame

**Usage**

```
## S3 method for class 'incidence2'
as.data.frame(x, row.names, optional, ...)
```

**Arguments**

x	<incidence2> object.
row.names	Not used.
optional	Not used.
...	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_incidence	<i>Coerce to an incidence object</i>
--------------	--------------------------------------

---

**Description**

Generic for coercion to an <incidence2> object.

**Usage**

```
as_incidence(x, ...)
```

## Default S3 method:  
as\_incidence(x, ...)

## S3 method for class 'incidence2'  
as\_incidence(x, ...)

**Arguments**

x	An R object.
...	Additional arguments to be passed to or from other methods.

**Value**

An <incidence2> object.

---

complete_dates	<i>Complete dates for all group combinations</i>
----------------	--

---

### Description

This function ensures that an incidence object has the same range of dates for each grouping. By default missing counts will be filled with 0L.

### Usage

```
complete_dates(x, expand = TRUE, fill = 0L, by = 1L)
```

### Arguments

x	<incidence2> object.
expand	[logical] Should a range of dates from the minimum to maximum value of the date index also be created.
fill	[numeric] The value to replace missing counts by. Defaults to 0L.
by	Passed as the by argument to seq. If expand is TRUE (default) then complete_dates will attempt to use function(x) seq(min(x), max(x), by = by) to generate a complete sequence of dates. Defaults to 1L.

### Value

An <incidence2> object.

### Examples

```
x <- data.frame(  
  dates = Sys.Date() + c(1,3,4),  
  groups = c("grp1", "grp2", "grp1"),  
  counts = 1:3  
)  
  
i <- incidence(x, date_index = "dates", groups = "groups", counts = "counts")  
complete_dates(i)
```

---

covidregionaldataUK *Regional data for COVID-19 cases in the UK*

---

## Description

A dataset containing the daily time-series of cases, tests, hospitalisations, and deaths for UK.

## Usage

```
covidregionaldataUK
```

## Format

A data frame with 6370 rows and 26 variables:

**date** the date that the counts were reported (YYYY-MM-DD)

**region** the region name

**region\_code** the region code

**cases\_new** new reported cases for that day

**cases\_total** total reported cases up to and including that day

**deaths\_new** new reported deaths for that day

**deaths\_total** total reported deaths up to and including that day

**recovered\_new** new reported recoveries for that day

**recovered\_total** total reported recoveries up to and including that day

**hosp\_new** new reported hospitalisations for that day

**hosp\_total** total reported hospitalisations up to and including that day (note this is cumulative total of new reported, not total currently in hospital).

**tested\_new** tests for that day

**tested\_total** total tests completed up to and including that day

## Details

Extracted using the `covidregionaldata` package on 2021-06-03.

## Source

<https://CRAN.R-project.org/package=covidregionaldata>

---

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

---

**Description**

cumulate() computes the cumulative incidence over time for an <incidence2> object.

**Usage**

```
cumulate(x)
```

**Arguments**

x [incidence2] object.

**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")
cumulate(i)
```

---

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

---

**Description**

incidence() calculates event the *incidence* of different events across specified time periods and groupings.

**Usage**

```
incidence(
  x,
  date_index,
  groups = NULL,
  counts = NULL,
  count_names_to = "count_variable",
  count_values_to = "count",
  date_names_to = "date_index",
  rm_na_dates = TRUE,
```

```

    interval = NULL,
    offset = NULL,
    ...
)

```

### Arguments

<code>x</code>	A data frame object representing a linelist or pre-aggregated dataset.
<code>date_index</code>	[character] The time index(es) of the given data. This should be the name(s) corresponding to the desired date column(s) in <code>x</code> . A name vector can be used for convenient relabelling of the resultant output. Multiple indices only make sense when <code>x</code> is a linelist.
<code>groups</code>	[character] An optional vector giving the names of the groups of observations for which incidence should be grouped.
<code>counts</code>	[character] The count variables of the given data. If NULL (default) the data is taken to be a linelist of individual observations.
<code>count_names_to</code>	[character] The column to create which will store the counts column names provided that counts is not NULL.
<code>count_values_to</code>	[character] The name of the column to store the resultant count values in.
<code>date_names_to</code>	[character] The name of the column to store the date variables in.
<code>rm_na_dates</code>	[logical] Should NA dates be removed prior to aggregation?
<code>interval</code>	An optional scalar integer or string indicating the (fixed) size of the desired time interval you wish to use for computing the incidence. Defaults to NULL in which case the <code>date_index</code> columns are left unchanged. Numeric values are coerced to integer and treated as a number of days to group. Text strings can be one of:  <ul style="list-style-type: none"> <li>* week(s) or weekly</li> <li>* epiweek(s)</li> <li>* isoweek(s)</li> <li>* month(s) or monthly</li> <li>* yearmonth(s)</li> <li>* quarter(s) or quarterly</li> <li>* yearquarter(s)</li> <li>* year(s) or yearly</li> </ul> More details can be found in the "Interval specification" section.

offset	<p>Only applicable when interval is not NULL.</p> <p>An optional scalar integer or date indicating the value you wish to start counting periods from relative to the Unix Epoch:</p> <ul style="list-style-type: none"> <li>• Default value of NULL corresponds to 0L.</li> <li>• For other integer values this is stored scaled by n (<code>offset &lt;- as.integer(offset) %% n</code>).</li> <li>• For date values this is first converted to an integer offset (<code>offset &lt;- floor(as.numeric(offset))</code>) and then scaled via n as above.</li> </ul>
...	Not currently used.

### Details

<incidence2> objects are a sub class of data frame with some additional invariants. That is, an <incidence2> object must:

- have one column representing the date index (this does not need to be a date object but must have an inherent ordering over time);
- have one column representing the count variable (i.e. what is being counted) and one variable representing the associated count;
- have zero or more columns representing groups;
- not have duplicated rows with regards to the date and group variables.

### Value

An object of class <incidence2, data.frame>.

### Interval specification

Where interval is specified, incidence() uses the `grates` package to generate appropriate date groupings. The grouping used depends on the value of interval. This can be specified as either an integer value or a string corresponding to one of the grates classes:

- integer values: <grates\_period> object, grouped by the specified number of days.
- week(s), weekly, isoweek: <grates\_isoweek> objects.
- epiweek(s): <grates\_epiweek> objects.
- month(s), monthly, yearmonth: <grates\_yearmonth> objects.
- quarter(s), quarterly, yearquarter: <grates\_yearquarter> objects.
- year(s) and yearly: <grates\_year> objects.

### See Also

browseVignettes("grates") for more details on the grate object classes.

**Examples**

```

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

  data(ebolavirus_clean, package = "outbreaks")
  dat <- ebolavirus_clean$linelist
  incidence(dat, "date_of_onset")
  incidence(dat, "date_of_onset", groups = c("gender", "hospital"))

}

```

---

keep	<i>Keep first, last and peak occurrences</i>
------	--

---

**Description**

keep\_first() and keep\_last() keep the first and last n rows to occur for each grouping when in ascending date order. keep\_peaks() keeps the rows with the maximum count value for each group.

**Usage**

```
keep_first(x, n, complete_dates = TRUE, ...)
```

```
keep_last(x, n, complete_dates = TRUE, ...)
```

```
keep_peaks(x, complete_dates = TRUE, first_only = FALSE, ...)
```

**Arguments**

x	<incidence2> object.
n	[integer] Number of entries to keep. double vectors will be converted via as.integer(n).
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().
first_only	[bool] Should only the first peak (by date) be kept. Defaults to TRUE.

**Value**

Incidence object with the chosen entries.

## Examples

```
if (requireNamespace("outbreaks", quietly = TRUE)) {  
  
  data(ebola_sim_clean, package = "outbreaks")  
  dat <- ebola_sim_clean$linelist  
  inci <- incidence(dat, "date_of_onset")  
  keep_first(inci, 3)  
  keep_last(inci, 3)  
  
}
```

---

plot.incidence2      *Plot an incidence object*

---

## Description

plot() can be used to provide a quick bar plot of an incidence object.

## Usage

```
## S3 method for class 'incidence2'  
plot(  
  x,  
  y,  
  width = 1,  
  colour_palette = vibrant,  
  border_colour = NA,  
  na_colour = "grey",  
  alpha = 0.7,  
  title = NULL,  
  angle = 0,  
  size = NULL,  
  nrow = NULL,  
  ...  
)
```

## Arguments

x	<incidence2> object.
y	Not used. Required for compatibility with the plot() generic.
width	[numeric] Value between 0 and 1 indicating the relative size of the bars to the interval. Default 1.

colour_palette	[function]	The color palette to be used for the different count variables. Defaults to vibrant (see ?palettes).
border_colour	[character]	The color to be used for the borders of the bars. Use NA (default) for invisible borders.
na_colour	[character]	The colour to plot NA values in graphs. Defaults to grey.
alpha	[numeric]	The alpha level for color transparency, with 1 being fully opaque and 0 fully transparent Defaults to 0.7.
title	[character]	Optional title for the graph.
angle	[numeric]	Rotation angle for text.
size	[numeric]	text size in pts.
nrow	[integer]	Number of rows used for facetting if there are group variables present and just one count in the incidence object.
...		Not currently used.

### Details

- Facetting will occur automatically if either grouping variables or multiple counts are present.
- If there are multiple count variables, each count will occupy a different row of the resulting plot.
- Utilises ggplot2 so this must be installed to use.

### Value

- A [ggplot2::ggplot()] object.

### Examples

```
if (requireNamespace("outbreaks", quietly = TRUE) && requireNamespace("ggplot2", quietly = TRUE)) {
  withAutoprint({
    data(ebola_sim_clean, package = "outbreaks")
    dat <- ebola_sim_clean$linelist

    inci <- incidence(dat, date_index = "date_of_onset", groups = "hospital")
    plot(inci, angle = 45)

    inci2 <- regroup(inci)
  })
}
```

```

    plot(inci2)
  })
}

```

---

```
print.incidence2      Print an incidence object.
```

---

### Description

Printing of <incidence2> objects is handled via the **pillar** package.

### Usage

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

### Arguments

```
x          <incidence2> object.
...        Additional arguments passed through to pillar::tbl_format_setup().
```

### Examples

```

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

  data(ebola_sim_clean, package = "outbreaks")
  dat <- ebola_sim_clean$linelist

  (out <- incidence(dat, "date_of_onset"))

  # use `n` to print more lines
  print(out, n = 20L)

}

```

---

```
regroup      Regroup 'incidence' objects
```

---

### Description

This function regroups an <incidence2> object across the specified groups. The resulting <incidence2> object will contain counts summed over the groups present in the input.

### Usage

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

**Arguments**

x	<incidence2> object.
groups	[character] The groups to sum over. If NULL (default) then the function returns the corresponding object with no groupings.

**Examples**

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

---

summary.incidence2      *Summary of an incidence object*

---

**Description**

Summary of an incidence object

**Usage**

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

**Arguments**

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

**Value**

object (invisibly).

## Examples

```
data(ebola_sim_clean, package = "outbreaks")
dat <- ebola_sim_clean$linelist
inci <- incidence(dat, "date_of_onset", groups = c("gender", "hospital"))
summary(inci)
```

---

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

n [integer]  
Number of colours.  
double vectors will be converted via `as.integer(n)`.

## Examples

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

# Index

## \* datasets

covidregionaldataUK, [7](#)  
<grates\_epiweek>, [10](#)  
<grates\_isoweek>, [10](#)  
<grates\_period>, [10](#)  
<grates\_year>, [10](#)  
<grates\_yearmonth>, [10](#)  
<grates\_yearquarter>, [10](#)

accessors, [2](#)

as.data.frame.incidence2, [4](#)

as\_incidence, [5](#)

complete\_dates, [6](#)

covidregionaldataUK, [7](#)

cumulate, [8](#)

get\_count\_value (accessors), [2](#)

get\_count\_value\_name (accessors), [2](#)

get\_count\_variable (accessors), [2](#)

get\_count\_variable\_name (accessors), [2](#)

get\_date\_index (accessors), [2](#)

get\_date\_index\_name (accessors), [2](#)

get\_dates (accessors), [2](#)

get\_dates\_name (accessors), [2](#)

get\_group\_names (accessors), [2](#)

get\_groups (accessors), [2](#)

incidence, [8](#)

keep, [11](#)

keep\_first (keep), [11](#)

keep\_last (keep), [11](#)

keep\_peaks (keep), [11](#)

muted (vibrant), [16](#)

palettes (vibrant), [16](#)

plot.incidence2, [12](#)

print.incidence2, [14](#)

regroup, [14](#)

summary.incidence2, [15](#)

vibrant, [16](#)