Package: juncture (via r-universe)

September 4, 2024

Type Package

Title Fast Census Tables for Point in Time Reporting

Version 0.1.0

Author John MacKintosh

Maintainer John MacKintosh <johnmackintosh.jm@gmail.com>

Description Fast, flexible census tables without 'SQL'. Count individuals or resources as they move or change status, at each interval within a specified date range. For example, count the number of patients in hospital accounting for moves between departments, or determine the number of occupants in a hotel between 2 dates.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports checkmate, data.table, lubridate

Suggests covr, tinytest

URL https://github.com/johnmackintosh/juncture

BugReports https://github.com/johnmackintosh/juncture/issues

Depends R (>= 2.10)

Repository https://johnmackintosh.r-universe.dev

RemoteUrl https://github.com/johnmackintosh/juncture

RemoteRef HEAD

RemoteSha f30be1ac55d19f3c2f1b519f1461ac5c8787900f

Contents

beds *#' 10 observations showing patient time in and out.*

Description

A dataset containing ten sets of time in and out by bed. This simulates patients moving in and out of departments or hospitals.

Usage

```
beds
```

Format

A data frame with 10 rows and 4 variables:

bed a grouping variable, representing a specific bed

patient individual patients

start_time time each patient moved into their current bed

end_time time each patient moved out of their current bed

juncture *Count the number of resources by interval*

Description

Counts the number of resources in each location, by the specified interval, for the duration of the resources admission. Results can be returned as grand totals, grouped totals, or at individual resource level per interval.

Usage

```
juncture(
  df,
  identifier,
  time_in,
  time_out,
  group_var = NULL,
  time_unit = "1 hour",
  time_adjust_period = NULL,
  time_adjust_value = NULL,
  results = c("individual", "group", "total"),
  uniques = TRUE
)
```

Arguments

df	dataframe, tibble or data.table.
identifier	Unique row / resource/ person identifier.
time_in	Datetime of time_in as POSIXct.
time_out	Datetime of time_out as POSIXct.
group_var	Optional unique character vector to identify specific resources location or responsible clinician at each interval, or at time of a change in location / responsible clinician during the interval.
time_unit	Character string to denote time intervals to count by e.g. "1 hour", "15 mins".
time_adjust_period	Optional argument which allows the user to obtain a snapshot at a specific time of day by making slight adjustments to the specified interval. Possible values are "start_sec","start_min","end_sec", or "end_min". For example, you may specify hourly intervals, but adjust these to 1 minute past the hour with "start_min", or several seconds before the end with "end_sec".
time_adjust_value	Optional. An integer to adjust the start or end of each period in minutes or seconds, depending on the chosen time_adjust_period (if specified).
results	A character string specifying the granularity of the results. 'individual' returns one row per resource or person, group_var and interval. The results can be input to external tools for further analysis or visualisation. 'group' provides an overall grouped count of resources by the specified time interval. 'total' returns the grand total of resources by each unique time interval.
uniques	Logical. Specifies how to deal with resources which move during an interval, and subsequently have two or more records per interval. Set "uniques" to TRUE to get a distinct count of resources per interval. To be clear, TRUE will count resources only once per interval. Setting "uniques" to FALSE will count each resource entry per interval. If a resource moves during an interval then at least two rows will be returned for that resource for that particular interval. This is useful if you want to count occupied beds, or track moves or transfers between departments. In general, if you use a grouping variable, set "uniques" to FALSE.

Value

data.table showing the resource identifier, the specified group variable , and the count by the relevant unit of time. Also included are the start and end of the interval, plus the date and base hour for convenient interactive filtering of the results.

Examples

```
juncture(beds, identifier ="patient", time_in = "start_time",
time_out = "end_time", time_unit = "1 hour", results = "total")
```

```
juncture(beds, identifier ="patient", time_in = "start_time",
time_out = "end_time", time_unit = "1 hour", results = "individual")

juncture(beds, identifier ="patient", time_in = "start_time",
time_out = "end_time", group_var = "bed",
time_unit = "1 hour", results = "group", uniques = FALSE)
```

Index