

Package: tvtools (via r-universe)

December 8, 2024

Type Package

Title Comprehensive Tools for Panel Data Analysis - 'tvtools'

Version 0.0.3

Depends R (>= 3.1.0)

Description Longitudinal data offers insights into population changes over time but often requires a flexible structure, especially with varying follow-up intervals. Panel data is one way to store such records, though it adds complexity to analysis. The 'tvtools' package for R simplifies exploring and analyzing panel data.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

Suggests knitr, rmarkdown, markdown, covr, devtools, testthat (>= 3.0.0)

VignetteBuilder knitr

Imports data.table, DTwrappers

Config/testthat/edition 3

NeedsCompilation no

Author David Shilane [aut], Mayur Bansal [ctb], Mehak Khara [ctb], Srivastav Budugutta [ctb, cre]

Maintainer Srivastav Budugutta <sb4788@columbia.edu>

Repository CRAN

Date/Publication 2024-10-08 10:10:03 UTC

Contents

calculate.utilization	2
count.events	3
create.baseline	4

cross.sectional.data	5
crude.rates	6
era.splits	7
event.time	8
first.event	9
first.panel.gap	10
followup.time	11
last.event	12
last.panel.gap	13
measurement.rate	15
panel.gaps	16
panel.overlaps	17
structure.panel	17
summarize.panel	18
unusual.duration	19

Index	20
--------------	-----------

calculate.utilization *calculate.utilization*

Description

Calculates the amount or proportion of time that a condition (e.g., use of a medication) is met for each subject over a specified observation period.

Usage

```
calculate.utilization(
  dat,
  outcome.names,
  begin,
  end,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  type = "rate",
  full.followup = FALSE,
  na.rm = TRUE
)
```

Arguments

<code>dat</code>	A data frame structured as a panel data set.
<code>outcome.names</code>	A character vector of variable names from <code>dat</code> representing binary conditions (1/0, TRUE/FALSE) whose utilization is to be calculated.
<code>begin</code>	The numeric starting time for the interval of interest.

end	The numeric ending time for the interval of interest.
id.name	The character name of the identifying variable within dat, used to track subjects across multiple rows.
t1.name	The character name of the time-tracking variable within dat representing the start (left endpoint) of each observation interval.
t2.name	The character name of the time-tracking variable within dat representing the end (right endpoint) of each observation interval.
type	A character string specifying the type of utilization to calculate: "rate" for proportion of time, or "total"/"count" for the count of days. Defaults to "rate".
full.followup	A logical value indicating whether to include all subjects or only those with fully observed records in the specified interval. Defaults to FALSE.
na.rm	A logical value indicating whether missing values should be excluded from the calculation. Defaults to TRUE.

Value

Returns a `data.table` object that contains the calculated utilization information for each subject specified by `id.name`. If `type` is "rate", it returns the proportion of the interval [begin, end) during which the conditions specified in `outcome.names` are met. If `type` is "total" or "count", it returns the total count of days within the interval where the conditions are met. The result is grouped by the `id.name` variable, ensuring each subject's data is aggregated separately.

count.events	<i>count.events</i>
--------------	---------------------

Description

Creates a count of the number of events that occurred within each group from a panel data structure, based on specified binary outcome variables.

Usage

```
count.events(
  dat,
  outcome.names,
  grouping.variables = NULL,
  type = "overall",
  na.rm = TRUE
)
```

Arguments

`dat` A data frame structured as a panel data set.

outcome.names	A character vector of variable names from dat that are expected to be binary (1/0, TRUE/FALSE). The function calculates the count of these variables being TRUE/1 in the specified interval. Variables not found in dat or not binary will be disregarded.
grouping.variables	A character vector of variable names from dat to group the resulting counts. If NULL, the function computes the overall count without grouping.
type	Specifies the counting method: "distinct" for counting only new occurrences separated by zeros (useful for events like hospitalizations spanning multiple records), or "overall" (default) for counting all records with the value of TRUE/1.
na.rm	A logical indicating whether missing values should be ignored in the calculations. Defaults to TRUE.

Value

Returns a data.table object containing the counts of events. The counts are aggregated based on the specified 'grouping.variables'. Each row corresponds to a group defined by 'grouping.variables' and contains counts of the specified 'outcome.names'. If 'type' is "distinct", the count reflects distinct occurrences of events; if 'type' is "overall", it reflects the total count of records with TRUE/1 for the 'outcome.names'. The output structure makes it easy to understand the distribution of events across the different groups or categories defined in the data set.

create.baseline	<i>create.baseline</i>
-----------------	------------------------

Description

Creates a baseline cohort from panel data at the initial time point (t=0). This function is tailored for scenarios where the baseline information is critical, particularly when the outcomes are considered in terms of the time elapsed since the baseline. For time points other than the baseline, consider using the function 'cross.sectional.data()'.

Usage

```
create.baseline(
  dat,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  outcome.names = NULL
)
```

Arguments

dat	A data frame or data table structured as panel data.
id.name	The character name of the identifying variable within dat, used to track subjects across multiple rows of data.

t1.name	The character name of the time variable within dat that represents the left endpoints (start of the observation interval).
t2.name	The character name of the time variable within dat that represents the right endpoints (end of the observation interval).
outcome.names	A character vector of outcome variable names from dat, expected to be binary. The function identifies the first occurrence of each outcome being 1 for each unique id.

Value

Returns a data frame or data table (depending on the input type) representing the baseline cohort of the provided panel data. This output is essentially a snapshot of the data at the initial time point (t=0). The function extracts and formats this baseline information based on the specified id, time, and outcome variables. The resulting dataset provides a foundation for subsequent analyses, particularly for tracking the onset or occurrence of specified outcomes from the start of the observation period.

cross.sectional.data *cross.sectional.data*

Description

Creates a cross-sectional cohort from a panel data structure at a specified time point. It focuses on outcome variables, indicating the time elapsed after the specified point until an event occurred.

Usage

```
cross.sectional.data(
  dat,
  time.point = 0,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  outcome.names = NULL,
  relative.followup = FALSE
)
```

Arguments

dat	A data frame structured as panel data.
time.point	The numeric time at which to create the cross-sectional data. This represents the point in time for which the data snapshot is taken. Subjects are included in the snapshot if they have data recorded at this time point.
id.name	The character name of the identifying variable within dat, used to track subjects across multiple rows of data.

<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>outcome.names</code>	A character vector of outcome variable names from <code>dat</code> , which should be binary. The function calculates the time since the <code>time.point</code> that each outcome first becomes 1 for each unique <code>id</code> .
<code>relative.followup</code>	A logical indicating whether to return the outcomes in absolute time (FALSE) or relative to the <code>time.point</code> (TRUE). Outcomes before the <code>time.point</code> are disregarded when TRUE.

Value

Returns a data frame or data table, depending on the input, containing the cross-sectional data extracted at the specified `time.point`. The dataset includes each subject observed at the `time.point`, with the relevant outcomes and other variables adjusted based on the specified parameters. If `outcome.names` are provided, it includes the calculated times from the specified `time.point` to the first occurrence of the outcomes for each subject. If `relative.followup` is TRUE, these times are relative to the `time.point`; otherwise, they are in absolute terms. The structure of the returned data is ideal for analyses focusing on the status of subjects at a specific moment in the study period.

<code>crude.rates</code>	<i>crude.rates</i>
--------------------------	--------------------

Description

Calculates the rate of specified events relative to the amount of follow-up time, based on panel data. It allows for the segmentation of data into different time periods (eras) and computes the event rates for these periods.

Usage

```
crude.rates(
  dat,
  outcome.names,
  cut.points = NULL,
  time.multiplier = 1,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  grouping.variables = NULL,
  type = "overall",
  na.rm = TRUE,
  era.name = "period"
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>outcome.names</code>	A character vector of variable names from <code>dat</code> , expected to be binary, representing the events of interest. The function calculates the rates of these events within the specified time intervals. Variables not in <code>dat</code> or non-binary will be ignored.
<code>cut.points</code>	A numeric vector specifying the end points of time intervals for rate calculation. The data is split into eras based on these points, and rates are computed for each interval.
<code>time.multiplier</code>	A numeric value that scales the computed rates, useful for converting rates to a standard time unit (e.g., per 1000 person-years).
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used for tracking individuals across multiple data rows.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>grouping.variables</code>	A character vector of variables in <code>dat</code> to group the results by.
<code>type</code>	Specifies the counting method: "distinct" for counting only new occurrences separated by zeros, or "overall" (default) for counting all records with the event.
<code>na.rm</code>	A logical indicating whether to exclude missing values from the calculations.
<code>era.name</code>	The character string used to name the time period column in the resulting table.

Value

Returns a data table containing the calculated rates of the specified events for each group and time period (era). The rates are presented alongside the grouping variables and the specified era. Each row corresponds to a unique combination of the grouping variables and time period, with the event rates adjusted according to the specified `time.multiplier`. The output facilitates the comparison of event rates across different segments of follow-up time and subgroups within the data.

<code>era.splits</code>	<i>era.splits</i>
-------------------------	-------------------

Description

Restructures panel data by dividing records into distinct eras based on specified cut points. Rows that span multiple eras are split into separate rows for each era they encompass. For example, a record spanning from 0 to 2 years could be split into two records: one for the first year (0-1) and another for the second year (1-2), if the cut points are set at each year mark. This process is useful for analyzing data within specific time intervals.

Usage

```
era.splits(dat, cut.points, id.name = "id", t1.name = "t1", t2.name = "t2")
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>cut.points</code>	A numeric vector specifying the endpoints of each era. Each value defines the end of one era and the beginning of the next, allowing records to be split into intervals such as $[\min(x), 10)$, $[10, 20)$, and $[20, \max(x))$. A row with $t1 = 0$ and $t2 = 30$, and cut points at 10 and 20, would be divided into intervals of $[0,10)$, $[10,20)$, and $[20,30)$.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used for tracking subjects across multiple rows.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.

Value

Returns a data table that has been restructured to reflect the specified era splits. Original rows overlapping multiple eras are divided into multiple rows, each representing a discrete interval within the specified eras. The function ensures each subject's observation period is accurately represented according to the specified time intervals. The output is sorted by the identifying variable and the start time of each interval, facilitating further analysis or processing that depends on temporal segmentation.

event.time

event.time

Description

Calculates the time to an event in a panel data structure, based on binary outcome variables. It can determine the time of the first, last, or other statistical measures (like mean or median) of event occurrences for each subject.

Usage

```
event.time(
  dat,
  id.name,
  outcome.names,
  t1.name,
  time.function = "min",
  append.to.table = FALSE,
  event.name = "first.event"
)
```


Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used for tracking subjects across multiple rows.
<code>outcome.names</code>	A character vector of variable names from <code>dat</code> that are expected to be binary, representing the events of interest. The function calculates the time to these events based on the specified function (e.g., first occurrence).
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>time.function</code>	The function to apply to the event times for each subject and outcome. Options include "min" for the time of the first event, "max" for the last event, and others like "mean" or "median" for average or median event times.
<code>append.to.table</code>	A logical indicating whether to append the calculated event times as new columns to the original data.frame (TRUE) or return them as a separate data frame (FALSE).
<code>event.name</code>	The name to give the event time columns when they are appended to the data.

Value

If `append.to.table` is FALSE, returns a data table with the calculated event times for each subject and specified outcome, keyed by the `id.name`. Each outcome will have its own column named according to the original outcome name with the specified `event.name` appended (e.g., "outcome.first.event" for the first event times). If `append.to.table` is TRUE, the original data table is returned with these new columns appended. This facilitates analyses focused on the timing of events relative to the subjects' observation periods in the panel data.

<code>first.event</code>	<i>first.event</i>
--------------------------	--------------------

Description

Calculates the time to the first occurrence of specified events for each subject in a panel data structure. This function is particularly useful for longitudinal data analysis where the timing of first events is crucial for subsequent analyses.

Usage

```
first.event(
  dat,
  id.name,
  outcome.names,
  t1.name,
  append.to.table = FALSE,
  event.name = "first.event"
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows.
<code>outcome.names</code>	A character vector of variable names from <code>dat</code> , expected to be binary, representing the events of interest. The function determines the first time each outcome becomes true (1) for each unique <code>id</code> .
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>append.to.table</code>	A logical indicating whether to append the calculated first event times as new columns to the original data.frame (TRUE) or return them as a separate data frame (FALSE).
<code>event.name</code>	The name to give the event time columns when they are appended to the data, specifically for the first event times.

Value

If `append.to.table` is FALSE, the function returns a data table with the calculated times to the first event for each subject and specified outcome, keyed by the `id.name`. Each outcome will have its own column named according to the `event.name` parameter, appended with the outcome name (e.g., "outcome.first.event" for first event times). If `append.to.table` is TRUE, the original data table is returned with these new columns appended. This facilitates detailed analysis on the timing of first events in relation to the subjects' overall observation periods within the panel data structure.

<code>first.panel.gap</code>	<i>first.panel.gap</i>
------------------------------	------------------------

Description

Identifies the first occurrence of a gap in observation periods for each unique subject within a panel data structure. A gap is defined as a period where no data were recorded for an expected interval between observations.

Usage

```
first.panel.gap(
  dat,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  gap.name = "gap_before",
  first.value = 0,
  expected.gap.between = 0,
  append.to.table = FALSE
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows of data.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>gap.name</code>	A character value for the name of the variable to be used or created that specifies whether a gap is observed before the record.
<code>first.value</code>	The numeric value indicating the expected beginning time of the observation period for each subject.
<code>expected.gap.between</code>	The numeric value indicating the expected amount of time between the end of one record and the start of the next; the default is zero, assuming continuous observation.
<code>append.to.table</code>	A logical value indicating whether the identified first gap times should be appended as a new column to the existing data.frame (TRUE) or returned as a separate data frame (FALSE, default).

Value

If `append.to.table` is FALSE, the function returns a data table with the identified first gap time for each subject, keyed by the `id.name`. Each subject will have a corresponding gap time, indicating the first observed gap in their data. If `append.to.table` is TRUE, the original data table is returned with a new column appended, containing the first gap times for each subject. This functionality is critical for longitudinal studies where maintaining continuous observation of subjects is necessary, and identifying gaps can highlight data collection issues or subject attrition.

<code>followup.time</code>	<i>followup.time</i>
----------------------------	----------------------

Description

Computes the total or maximum follow-up time for each subject in a panel data structure, accounting for observation endpoints like death, loss to follow-up, or study conclusion.

Usage

```
followup.time(
  dat,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
```

```

  followup.name = "followup.time",
  calculate.as = "total",
  append.to.data = FALSE
)

```

Arguments

<code>dat</code>	A data.frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>followup.name</code>	The character name for the new variable to display the calculated follow-up time for each subject.
<code>calculate.as</code>	A character value specifying the calculation method for follow-up time: "max" for the maximum observed time point for each subject, or "total" for the total observed time across all records for each subject (default). Note: "max" and "total" yield the same result when records start at time 0, have no gaps, and don't overlap.
<code>append.to.data</code>	A logical indicating whether to append the calculated follow-up time as a new column to the original data.frame (TRUE), or return it as a separate data frame (FALSE, default).

Value

Returns a modified version of the input data frame or a new data frame based on the `append.to.data` parameter. If `append.to.data` is TRUE, the original data frame is returned with an additional column named as specified by `followup.name`, containing the calculated follow-up times for each subject. If FALSE, a new data frame is returned containing the `id.name` and the calculated follow-up times under the `followup.name` column. This functionality is essential for analyses that require understanding the duration of subject participation or observation within the study period.

<code>last.event</code>	<i>last.event</i>
-------------------------	-------------------

Description

Calculates the time to the last occurrence of specified binary events for each subject in a panel data structure. This is particularly useful for understanding the timing of the last event in longitudinal analyses.

Usage

```
last.event(
  dat,
  id.name,
  outcome.names,
  t1.name,
  append.to.table = FALSE,
  event.name = "last.event"
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows of data.
<code>outcome.names</code>	A character vector of variable names from <code>dat</code> , expected to be binary, indicating the events of interest. The function determines the last time each outcome becomes true (1) for each unique <code>id</code> .
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>append.to.table</code>	A logical indicating whether to append the calculated last event times as new columns to the original data frame (TRUE) or return them as a separate data frame (FALSE).
<code>event.name</code>	The name to give the event time columns when they are appended to the data, specifically for the last event times.

Value

If `append.to.table` is FALSE, the function returns a data table with the calculated times to the last event for each subject and specified outcome, keyed by the `id.name`. Each outcome will have its own column named according to the `event.name` parameter, appended with the outcome name (e.g., "outcome.last.event" for last event times). If `append.to.table` is TRUE, the original data table is returned with these new columns appended. This enables detailed analysis on the timing of last events relative to the subjects' overall observation periods within the panel data.

<code>last.panel.gap</code>	<i>last.panel.gap</i>
-----------------------------	-----------------------

Description

Identifies the time point of the last observed gap in observation for each unique subject in a panel data structure. A gap represents a missing period between recorded observations, important for assessing data completeness and continuity.

Usage

```
last.panel.gap(
  dat,
  id.name,
  t1.name,
  t2.name,
  gap.name = "gap_before",
  first.value = 0,
  expected.gap.between = 0,
  append.to.table = FALSE
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows of data.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>gap.name</code>	A character value for the name of the variable indicating whether a gap is observed before the record.
<code>first.value</code>	The numeric value indicating the expected beginning time of the observation period for each subject.
<code>expected.gap.between</code>	The numeric value specifying the expected time between the end of one record and the start of the next; defaults to zero for continuous observation.
<code>append.to.table</code>	A logical indicating whether to append the identified last gap times as a new column to the existing data.frame (TRUE) or return them as a separate data frame (FALSE, default).

Value

If `append.to.table` is FALSE, returns a data table with the time points of the last gap for each subject, keyed by the `id.name`. Each row will correspond to a unique subject, including the time point of their last observed gap, if any. If `append.to.table` is TRUE, the original data table is returned with an additional column containing these time points for each subject. This functionality aids in analyzing and understanding the patterns of missing observations or breaks in data collection within the study period.

measurement.rate	<i>measurement.rate</i>
------------------	-------------------------

Description

Calculates the proportion of unique subjects remaining under observation and those no longer observed at a specified time point during the follow-up period within a panel data structure. This metric is crucial for evaluating the coverage and retention of subjects in longitudinal studies.

Usage

```
measurement.rate(  
  dat,  
  id.name = "id",  
  t1.name = "t1",  
  t2.name = "t2",  
  time.point = 0,  
  grouping.variables = NULL  
)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>time.point</code>	A numeric value specifying the point in time at which the measurement rate should be calculated. Subjects observed at this point are considered active.
<code>grouping.variables</code>	A character vector of variable names from <code>dat</code> used to group the results. Proportions are calculated within these groups.

Value

Returns a data table that includes the specified grouping variables, the number of subjects observed at the given `time.point`, the total number of unique subjects, and two calculated rates: `'rate.observed'` and `'rate.not.observed'`. `'rate.observed'` is the proportion of subjects active at the specified time point, and `'rate.not.observed'` is the proportion no longer observed. This output is instrumental for analyzing subject retention and attrition over the course of the study.

 panel.gaps

panel.gaps

Description

Identifies gaps in observation periods within panel data, marking each record with a flag indicating whether it was preceded by a gap. A gap reflects a missing period of observation between this and the previous record for a subject, which is crucial for assessing data integrity and continuity.

Usage

```
panel.gaps(
  dat,
  id.name = "id",
  t1.name = "t1",
  t2.name = "t2",
  gap.name = "gap_before",
  first.value = 0,
  expected.gap.between = 0
)
```

Arguments

dat	A data frame structured as panel data.
id.name	The character name of the identifying variable within dat, used to track subjects across multiple rows of data.
t1.name	The character name of the time variable within dat representing the start (left endpoint) of observation intervals.
t2.name	The character name of the time variable within dat representing the end (right endpoint) of observation intervals.
gap.name	A character value specifying the name of the new variable to be created that indicates whether a gap is observed before the record.
first.value	The numeric expected beginning time of the observation period for each subject.
expected.gap.between	The numeric amount of time expected between the end of one record and the start of the next; defaults to zero for continuous observation without expected gaps.

Value

Returns the original data frame with an additional column (named according to the gap.name parameter) for each record, indicating whether a gap in observation was detected before that record. The gap flag is determined based on the specified expected beginning time and the expected gap between records. This enhanced data frame is instrumental for subsequent analyses that require understanding of observation continuity and identifying subjects with missing data periods.

panel.overlaps	<i>panel.overlaps</i>
----------------	-----------------------

Description

Identifies records within a panel data set that have overlapping observation periods for the same subject. Overlaps can indicate data entry errors or issues with data collection protocols.

Usage

```
panel.overlaps(dat, id.name = "id", t1.name = "t1", t2.name = "t2")
```

Arguments

dat	A data frame structured as panel data.
id.name	The character name of the identifying variable within dat, used to track subjects across multiple rows.
t1.name	The character name of the time variable within dat representing the start (left endpoint) of observation intervals.
t2.name	The character name of the time variable within dat representing the end (right endpoint) of observation intervals.

Value

Returns a data table listing subjects (identified by id.name) who have at least one instance of overlapping observation periods. Each row corresponds to a unique subject identified as having overlaps, with additional information or indicators related to the nature or extent of these overlaps. This output is crucial for data cleaning and ensuring the temporal accuracy of the panel data, allowing researchers to identify and rectify anomalies before conducting further analysis.

structure.panel	<i>structure.panel</i>
-----------------	------------------------

Description

Sorts the panel data by subject identifier (id) and the beginning of each observation period (time), ensuring the data is organized sequentially for each subject. This is a crucial step in preparing panel data for time-series or longitudinal analysis, where the order of records affects the analysis outcome.

Usage

```
structure.panel(dat, id.name = "id", t1.name = "t1")
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows of data.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.

Value

Returns the input data frame sorted by the specified `id.name` and `t1.name`, ensuring that the data for each subject is in chronological order based on the start time of each observation period. This structured panel data is essential for any subsequent analyses that depend on the temporal sequence of observations, such as time-to-event analysis, longitudinal modeling, or any study of changes over time within subjects.

<code>summarize.panel</code>	<i>summarize.panel</i>
------------------------------	------------------------

Description

Provides summary statistics for panel data, including the total number of records, the count of unique subjects, average records per subject, total and maximum follow-up time. This summary is essential for understanding the dataset's structure and the extent of data available for each subject.

Usage

```
summarize.panel(dat, id.name, t1.name, t2.name, grouping.variables = NULL)
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>id.name</code>	The character name of the identifying variable within <code>dat</code> , used to track subjects across multiple rows of data.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.
<code>grouping.variables</code>	A character vector of variable names from <code>dat</code> for grouping the resulting summary statistics. If <code>NULL</code> , summary statistics are computed for the entire dataset.

Value

Returns a `data.table` with summary statistics for the panel data, grouped by the specified grouping variables (if provided). The summary includes the total number of records (`total.records`), the number of unique subjects (`unique.ids`), average number of records per subject (`mean.records.per.id`), total follow-up time (`total.followup`), and the maximum follow-up time (`max.followup`) for each group. This summary provides a comprehensive overview of the data's coverage and depth, aiding in its interpretation and the planning of subsequent analyses.

<code>unusual.duration</code>	<i>unusual.duration</i>
-------------------------------	-------------------------

Description

Identifies records within panel data where specified events occur for an unusually long duration, exceeding a predefined maximum length. This is critical for detecting outliers or anomalous data points in longitudinal studies.

Usage

```
unusual.duration(dat, outcome.name, max.length, t1.name = "t1", t2.name = "t2")
```

Arguments

<code>dat</code>	A data frame structured as panel data.
<code>outcome.name</code>	The character name of a binary variable within <code>dat</code> , representing the event of interest. Only records where this event is true (1) are considered for analysis.
<code>max.length</code>	A numeric value specifying the maximum allowed duration for the event. Records where the event duration exceeds this threshold are identified as unusual.
<code>t1.name</code>	The character name of the time variable within <code>dat</code> representing the start (left endpoint) of observation intervals.
<code>t2.name</code>	The character name of the time variable within <code>dat</code> representing the end (right endpoint) of observation intervals.

Value

Returns a subset of the original data frame containing only those records where the specified event occurs for a duration longer than the `max.length` parameter. Each row in this subset corresponds to an event considered unusually long, allowing for easy identification and further examination of these cases. This filtered dataset is instrumental in quality control and ensuring the accuracy and reliability of longitudinal data analyses.

Index

`calculate.utilization`, 2
`count.events`, 3
`create.baseline`, 4
`cross.sectional.data`, 5
`crude.rates`, 6

`era.splits`, 7
`event.time`, 8

`first.event`, 9
`first.panel.gap`, 10
`followup.time`, 11

`last.event`, 12
`last.panel.gap`, 13

`measurement.rate`, 15

`panel.gaps`, 16
`panel.overlaps`, 17

`structure.panel`, 17
`summarize.panel`, 18

`unusual.duration`, 19