Package 'expstudies'

June 14, 2019

Title Calculate Exposures, Assign Records to Intervals

Version 0.0.5

Description Creation of an exposure table with rows for policy-intervals from a table with a unique policy number key and beginning and ending dates for each policy. Methods for assigning supplemental data containing dates and policy numbers to the corresponding interval from the created exposures table.

Depends R (>= 3.1.0)

Imports magrittr, dplyr, lubridate, Rcpp

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

LinkingTo Rcpp

Suggests pander, knitr, rmarkdown, testthat, tidyr

VignetteBuilder knitr

NeedsCompilation yes

Author Matthew Caseres [aut, cre]

Maintainer Matthew Caseres <matthewcaseres@outlook.com>

Repository CRAN

Date/Publication 2019-06-14 11:20:03 UTC

R topics documented:

`addDays` *Create daily exposures*

Description

This function takes a records file and creates a day for each day the policyholder is active

Usage

```
addDays(records, min_date = NULL, max_date = NULL)
```

Arguments

`records`	File containing a unique policy key with start and end dates.
`min_date`	Left truncation date.
`max_date`	Right truncation date.

Examples

```
addDays(records)
```

`addExposures` *Create an exposure data frame*

Description

This function takes a records file with unique policy identifiers in the column "key", exposure start dates in column "start" and exposure end dates in column "end". The output is a data frame with exposures and durations for time intervals.

Usage

```
addExposures(records, type = "PY", lower_year = NULL,
  upper_year = NULL)
```

Arguments

records	File containing a unique policy key with start and end dates.
type	Creates policy year rows for the default type = "PY". Creates policy month rows for type = "PM".
lower_year	A lower year for truncation to reduce calculation time and output size.
upper_year	An upper year for truncation to reduce calculation time and output size.

Value

A data frame with multiple rows for each unique policy key. Each row represents a policy interval.

Examples

```
addExposures(records)
```

---

addStart *Map transactions to exposure intervals*

Description

This function takes in exposures and transactions and returns all transactions with a matching "key" and "trans_date" within an exposure interval and attaches the start date of the corresponding exposure interval. This is useful for grouping by start date and key, aggregating, and joining to the exposure rows. In this way premium pattern analysis can be performed.

Usage

```
addStart(trans, exposures)
```

Arguments

trans	Information we wish to assign to exposure intervals.
exposures	Output from addExposures(records).

Value

Modified transaction records that corresponding to an exposure interval, the interval start date is included.

Examples

```
addStart(trans, exposures)
```

daySize *Estimate the size of daily exposures*

Description

This function takes a records file and creates a day for each day the policyholder is active

Usage

```
daySize(records, min_date = NULL, max_date = NULL)
```

Arguments

records	File containing a unique policy key with start and end dates.
min_date	Left truncation date.
max_date	Right truncation date.

Examples

```
daySize(records)
```

exposures *Example exposure records to demonstrate calculations*

Description

A dataset containing the output of addExposures(records). Used to demonstrate calculations.

Usage

```
exposures
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 24 rows and 5 columns.

expSize *Estimate size of exposure data frame*

Description

This function takes a records file and the same arguments as the addExposures function to estimate the size of the output created. The upper bound is pretty reasonable. The idea with this function is that it enables users to determine if they can reasonably perform an operation on their computer.

Usage

```
expSize(records, type = "PY", lower_year = NULL, upper_year = NULL)
```

Arguments

records	File containing a unique policy key with start and end dates.
type	Creates policy year rows for the default type = "PY". Creates policy month rows for type = "PM". Many other variations on this.
lower_year	A lower year for truncation to reduce calculation time and output size.
upper_year	An upper year for truncation to reduce calculation time and output size.

Value

An upper bound for the number of rows used in the calculation of an exposure frame.

Examples

```
addExposures(records)
```

expstudies *expstudies: package for life experience data*

Description

Creation of an exposure table with rows for policy-intervals from a table with a unique policy number key and beginning and ending dates for each policy. Methods for assigning supplemental data containing dates and policy numbers to the corresponding interval from the created exposures table.

expstudies functions

- addExposures - Creation of an exposure table with either policy years or policy months.
- addStart - Allocate transactions to intervals.

`findStart` *Find the correct interval/key combination for start intervals.*

Description

Find the correct interval/key combination for start intervals.

Usage

```
findStart(exp_key, start_int, end_int, trans_key, trans_date)
```

Arguments

`exp_key`	sorted key column.
`start_int`	start_intervals that are are sorted within their keys
`end_int`	end_intervals that are are sorted because start_int is sorted.
`trans_key`	that must have a matching interval. This is guaranteed by the R code before calling findStart.
`trans_date`	sorted date column.

`makeDays` *Create a DataFrame with all dates in the exposure period for each key*

Description

Create a DataFrame with all dates in the exposure period for each key

Usage

```
makeDays(record_key, record_start, record_days, final_size)
```

Arguments

`record_key`	Vector of keys from the input record
`record_start`	Vector of start dates from the input record
`record_days`	Vector containing the number of days in the interval for a row in a record
`final_size`	Sum of days in all intervals, used to initialize the DataFrame that is returned.

mortality_tables *Mortality tables for A/E analysis*

Description

A list containing several tidy mortality tables that can be joined to an exposure data frame. Find more tables here - github.com/ActuarialAnalyst/LongMortalityTables. Tables are from the SOA website.

Usage

```
mortality_tables
```

Format

An object of class `list` of length 6.

records *Example policy records to demonstrate calculations*

Description

A dataset containing 2 policyholders (identified by "key") as well as their issue date ("start") their termination date ("end") as well as some additional information. Used to demonstrate calculations.

Usage

```
records
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 2 rows and 5 columns.

trans *Example transaction records to demonstrate calculations*

Description

A dataset containing transactions. Contains fields for the policyholder identifier ("key"), the transaction date ("trans_date") and the amount of the premium paid ("amt"). Used to demonstrate calculations.

Usage

```
trans
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 205 rows and 3 columns.

Index