Package: mobilityIndexR (via r-universe)

September 2, 2024

Type Package

Title Calculates Transition Matrices and Mobility Indices

Version 0.2.1

Author Brett Mullins and Trevor Harkreader

Maintainer Brett Mullins <brettcmullins@gmail.com>

Description Measures mobility in a population through transition matrices and mobility indices. Relative, mixed, and absolute transition matrices are supported. The Prais-Bibby, Absolute Movement, Origin Specific, and Weighted Group Mobility indices are supported. Example income and grade data are included.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports stats

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown

VignetteBuilder knitr

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

RemoteUrl https://github.com/bcmullins/mobilityindexr

RemoteRef HEAD

RemoteSha 6fdd657051aa95ff667a839a89e4a440abcbc52f

Contents

getHypothesisTest *Hypothesis Test for Two Mobility Datasets*

Description

Calculates hypothesis tests of mobility indices from two datasets. Specifically, for datasets A and B, this function performs one-sided nonparametric hypothesis tests that the index value for A is greater than the corresponding index value for B. Supports Prais-Bibby, Absolute Movement, Origin Specific, and Weighted Group Mobility indices and relative, mixed, and absolute types of rankings in the calculation these indices.

Usage

```
getHypothesisTest(
  dat_A,
  dat_B,
  cols_A,
  cols_B,
  type,
  indices = "all",
  num_ranks,
  exclude_value,
  bounds,
  rerank_exclude_value = FALSE,
  strict = TRUE,
  bootstrap_iter = 100
)
```

Arguments

`dat_A` a dataframe with an "id" column

`dat_B` a dataframe with an "id" column

`cols_A` a list of character strings denoting the first and second column to be used in the index calculations for dat_A

`cols_B` a list of character strings denoting the first and second column to be used in the index calculations for dat_B

`type` a character string indicating the type of ranking; accepts ’relative’, ’mixed’, and ’absolute’

`indices` a vector of character strings indicating which mobility indices are desired; currently support ’prais_bibby’, ’average_movement’, ’wgm’, and ’origin_specific’. The default value is ’all’.

`num_ranks` an integer specifying the number of ranks for a relative or mixed ranking

`exclude_value` a single numeric value that is excluded in calculating the transition matrix; see the rerank_exclude_value parameter to specify how the exclude value is handled `bounds` a sequence of numeric bounds for defining absolute ranks

`rerank_exclude_value`
a character string indicating how the exclude value is handled when present; accepts 'as_new_rank', 'as_existing_rank', and 'exclude'

`strict` logical. If TRUE, indices are calculated from the given values. If FALSE, indices are calculated by jittering the values to ensure uniqueness of bounds of ranks. Only used with relative and mixed types. The default value is TRUE.

`bootstrap_iter` the number of bootstrap iterations used to estimate hypothesis tests. The default value is 100.

Value

Returns a named vector containing the estimated probabilities that index value for dataset A is greater than the corresponding index value for dataset B

Examples

```
getHypothesisTest(dat_A = incomeMobility,
                  dat_B = incomeMobility,
                  cols_A = c("t0", "t3"),
                  cols_B = c("t5", "t8"),
                  type = "relative",
                  num_ranks = 5)
```

`getMobilityIndices` *Calculates Mobility Indices at Two Points in Time*

Description

Calculates mobility indices from two columns in dataset. Supports Prais-Bibby, Absolute Movement, Origin Specific, and Weighted Group Mobility indices and relative, mixed, and absolute types of rankings in the calculation these indices.

Usage

```
getMobilityIndices(
  dat,
  col_x,
  col_y,
  type,
  indices = "all",
  num_ranks,
  exclude_value,
  bounds,
  rerank_exclude_value,
  strict = TRUE,
  intervals = FALSE,
  interval_pct = 0.95,
  bootstrap_iter = 100
)
```

Arguments

dat a dataframe with an "id" column

col_x a character string denoting the first column to be used in the index calculation

col_y a character string denoting the second column to be used in the index calculation

type a character string indicating the type of ranking; accepts 'relative', 'mixed', and 'absolute'

indices a vector of character strings indicating which mobility indices are desired; currently support 'prais_bibby', 'average_movement', 'wgm', and 'origin_specific'. The default value is 'all'.

num_ranks an integer specifying the number of ranks for a relative or mixed ranking

exclude_value a single numeric value that is excluded in calculating the transition matrix; see the rerank_exclude_value parameter to specify how the exclude value is handled

bounds a sequence of numeric bounds for defining absolute ranks

rerank_exclude_value
a character string indicating how the exclude value is handled when present; accepts 'as_new_rank', 'as_existing_rank', and 'exclude'

strict logical. If TRUE, indices are calculated from the given values. If FALSE, indices are calculated by jittering the values to ensure uniqueness of bounds of ranks. Only used with relative and mixed types. The default value is TRUE.

intervals logical. If TRUE, will calculate bootstrapped confidence intervals using the percentile method. The default value is FALSE.

interval_pct a number between zero and one indicating the size of the bootstrapped intervals. The default value is 0.95.

bootstrap_iter the number of bootstrap iterations used to estimate intervals. The default value is 100.

Value

Returns a named vector containing the desired index values

Examples

```
data(incomeMobility)
getMobilityIndices(dat = incomeMobility,
                   col_x = 't0',
                   col_y = 't2',
                   type = 'relative',
                   num_ranks = 5)
```

getTMatrix *Calculates Transition Matrix*

Description

Returns transition matrix from two columns in dataset. Supports relative, mixed, and absolute transition matrices as well as handling an excluded value.

Usage

```
getTMatrix(
  dat,
  col_x,
  col_y,
  type,
  probs = TRUE,
  num_ranks,
  exclude_value,
  bounds,
  rerank_exclude_value,
  strict = TRUE
)
```

Arguments

`dat`	a dataframe with an "id" column
`col_x`	a character string denoting the first column from dat to be used in the transition matrix
`col_y`	a character string denoting the second column from dat to be used in the transition matrix
`type`	a character string indicating the type of transition matrix; accepts 'relative', 'mixed', and 'absolute'
`probs`	logical. If TRUE, values in transition matrix are probabilities; if FALSE, values in transition matrix are counts
`num_ranks`	an integer specifying the number of ranks for a relative or mixed transition matrix
`exclude_value`	a single numeric value that is excluded in calculating the transition matrix; see the rerank_exclude_value parameter to specify how the exclude value is handled
`bounds`	a sequence of numeric bounds for defining absolute transition matrix ranks
`rerank_exclude_value`	a character string indicating how the exclude value is handled when present; accepts 'as_new_rank', 'as_existing_rank', and 'exclude'
`strict`	logical. If TRUE, transition matrix is calculated from the given values. If FALSE, transition matrix is calculated by jittering the values to ensure uniqueness of bounds. Only used with relative and mixed types.

Value

Returns a list with a transition matrix as a Matrix and vectors of the the x and y bounds corresponding to the ranks in the matrix

Examples

```
data(incomeMobility)
getTMatrix(dat = incomeMobility,
           col_x = 't0',
           col_y = 't9',
           type = 'relative',
           num_ranks = 5)
```

`gradeMobility` *Example Grades Dataset*

Description

This is an example grade dataset for measuring grade mobility.

Usage

```
gradeMobility
```

Format

A `data.frame` with 13 columns:

id A unique integer id for each row representing a student

cohort Class of student

t0 Grade at time 0

t1 Grade at time 1

t2 Grade at time 2

t3 Grade at time 3

t4 Grade at time 4

t5 Grade at time 5

t6 Grade at time 6

t7 Grade at time 7

t8 Grade at time 8

t9 Grade at time 9

t10 Grade at time 10

`incomeMobility` *Example Income Dataset*

Description

This is an example income dataset to measure income mobility.

Usage

```
incomeMobility
```

Format

A `data.frame` with 13 columns:

id A unique integer id for each row representing an individual

cohort Year when first income is observed

t0 Income at time 0

t1 Income at time 1

t2 Income at time 2

t3 Income at time 3

t4 Income at time 4

t5 Income at time 5

t6 Income at time 6

t7 Income at time 7

t8 Income at time 8

t9 Income at time 9

t10 Income at time 10

`incomeZeroInfMobility` *Example Zero-Inflated Income Dataset*

Description

These is an example income dataset to measure income mobility with an inflated number of zeros. Zero-inflated data is often observed in practice and will cause difficulties for some transition matrix and mobility index approaches.

Usage

```
incomeZeroInfMobility
```

Format

A `data.frame` with 13 columns:

id A unique integer id for each row representing an individual

cohort Year when first income is observed

t0 Income at time 0

t1 Income at time 1

t2 Income at time 2

t3 Income at time 3

t4 Income at time 4

t5 Income at time 5

t6 Income at time 6

t7 Income at time 7

t8 Income at time 8

t9 Income at time 9

t10 Income at time 10

`mobilityIndexR` *mobilityIndexR: A package for calculating transition matrices and mobility indices*

Description

This package measures mobility in a population through transition matrices and mobility indices. Relative, mixed, and absolute transition matrices are supported for various use cases. With respect to indices, the Prais-Bibby, Absolute Movement, Origin Specific, and Shorrocks indices are supported. Example income and grade data are included for demos.

mobilityIndexR functions

getTMatrix getMobilityIndices getHypothesisTest

Index