Package 'akmedoids'

October 12, 2022

Type Package

Title Anchored Kmedoids for Longitudinal Data Clustering

Version 1.3.0

Author Monsuru Adepeju [cre, aut],
Samuel Langton [aut],
Jon Bannister [aut]

Maintainer Monsuru Adepeju <monsuur2010@yahoo.com>

Description Advances a novel adaptation of longitudinal k-means clustering technique (Genolini et al. (2015) <doi:10.18637/jss.v065.i04>) for grouping trajectories based on the similarities of their long-term trends and determines the optimal solution based on either the average silhouette width (Rousseeuw P. J. 1987) or the Calinski-Harabatz criterion (Calinski and Harabatz (1974) <doi:10.1080/03610927408827101>). Includes functions to extract descriptive statistics and generate a visualisation of the resulting groups, drawing methods from the 'ggplot2' library (Wickham H. (2016) <doi:10.1007/978-3-319-24277-4>). The package also includes a number of other useful functions for exploring and manipulating longitudinal data prior to the clustering process.

Language en-US

License GPL-3

URL https://cran.r-project.org/package=akmedoids

BugReports https://github.com/MAnalytics/akmedoids/issues

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

Imports kml, stats, utils, dplyr, signal, Hmisc, grDevices, ggplot2, clusterCrit

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, digest, gdtools, kableExtra

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-13 04:50:02 UTC

R topics documented:

`akclustr` *Anchored k-medoids clustering*

Description

Given a list of trajectories and a functional method, this function clusters the trajectories into a k number of groups. If a vector of two numbers is given, the function determines the best solution from those options based on the Caliński-Harabasz criterion.

Usage

```
akclustr(traj, id_field = FALSE, method = "linear",
k = c(3,6), crit="Silhouette", verbose = TRUE, quality_plot=FALSE)
```

Arguments

traj	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
id_field	[numeric or character] Whether the first column of the traj is a unique (id) field. Default: FALSE. If TRUE the function recognizes the second column as the first time points.
method	[character] The parametric initialization strategy. Currently, the only available method is a linear method, set as "linear". This uses the time-dependent linear regression lines and the resulting groups are order in the order on increasing slopes.
k	[integer or vector (numeric)] either an exact integer number of clusters, or a vector of length two specifying the minimum and maximum numbers of clusters to be examined from which the best solution will be determined. In either case, the minimum number of clusters is 3. The default is c(3,6).
crit	[character] a string specifying the type of the criterion to use for assessing the quality of the cluster solutions, when k is a vector of two values (as above). Default: crit="Silhouette", use the average Silhouette width (Rousseeuw P. J. 1987). Using the "Silhouette" criterion, the optimal value of k can be determined as the elbow point of the curve. Other valid criterion is the "Calinski_Harabasz" (Caliński T. & Harabasz J. 1974) in which the maximum score represents the point of optimality. Having determined the optimal k, the function can then be re-run, using the exact (optimal) value of k.
verbose	to suppress output messages (to the console) during clustering. Default: TRUE.
quality_plot	Whether to show plot of quality criteria across different values of k. Default: FALSE.

Details

This function works by first approximating the trajectories based on the chosen parametric forms (e.g. linear), and then partitions the original trajectories based on the form groupings, in similar fashion to k-means clustering (Genolini et al. 2015). The key distinction of akmedoids compared with existing longitudinal approaches is that both the initial starting points as well as the subsequent cluster centers (as the iteration progresses) are based the selection of observations (medoids) as oppose to centroids.

Value

generates an akobject consisting of the cluster solutions at the specified values of k. Also, the graphical plot of the quality scores of the cluster solutions.

References

1. Genolini, C. et al. (2015) kml and kml3d: R Packages to Cluster Longitudinal Data. Journal of Statistical Software, 65(4), 1-34. URL http://www.jstatsoft.org/v65/i04/.

2. Rousseeuw P. J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math 20:53–65.

3. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun. Stat. 3:1-27.

Examples

```
data(traj)

trajectry <- data_imputation(traj, id_field = TRUE, method = 2,
replace_with = 1, fill_zeros = FALSE)

trajectry <- props(trajectry$CompleteData, id_field = TRUE)

print(trajectry)

output <- akclustr(trajectry, id_field = TRUE,
method = "linear", k = c(3,7), crit='Calinski_Harabasz',
verbose = FALSE, quality_plot=FALSE)

print(output)
```

---

alpha_label *Numerics ids to alphabetical ids*

Description

Function to transform a list of numeric ids to alphabetic ids

Usage

```
alpha_label(x)
```

Arguments

x	A vector of numeric ids

Details

Given a vector of numeric cluster ids, ‘alpha_label‘ converts each id to its corresponding alphabets. It combines alphabets for ids greater than 26.

Value

A vector of alphabetical ids.

Examples

```
data(TO1Risk)

set.seed(1000)
#pick 4 random clusters
center <- TO1Risk[runif(4,1,nrow(TO1Risk)), ]

#Assigning each individual to nearest centre
numeric_Labels <- kml::affectIndivC(TO1Risk, center)

mode(numeric_Labels)

#transform numeric cluster labels to alphabets
alphab_Labels <- alpha_label(numeric_Labels)

mode(alphab_Labels)
```

---

clustr *Sample labels of cluster groups*

Description

A dataframe of alphabetical labels representing the optimal solution of ‘traj‘ dataset based on ‘akClust‘ function

Usage

```
clustr
```

Format

A dataframe containing one variable:

- label: alphabetical label by clusters

---

data_imputation *Data imputation for longitudinal data*

Description

This function fills any missing entries (`NA`, `Inf`, `null`) in a matrix or dataframe, according to a specified method. By default, `'0'` is considered a value.

Usage

```
data_imputation(traj, id_field = FALSE, method = 2,
replace_with = 1, fill_zeros = FALSE, verbose=TRUE)
```

Arguments

traj	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time points.
id_field	[numeric or character] Whether the first column of the traj is a unique (id) field. Default: FALSE. If TRUE the function recognises the second column as the first time step.
method	[an integer] indicating a method for calculating the missing values. Options are: '1': arithmetic method, and '2': regression method. The default is '1': arithmetic method
replace_with	[an integer from 1 to 6] indicating the technique, based on a specified method, for calculating the missing entries. '1': arithmetic method, replace_with options are: '1': Mean value of the corresponding column; '2': Minimum value of corresponding column; '3': Maximum value of corresponding column; '4': Mean value of corresponding row; '5': Minimum value of corresponding row, or '6': Maximum value of corresponding row. For '2': regression method: the available option for the replace_with is: '1': linear. The regression method fits a linear regression line to a trajectory with missing entry(s) and estimates the missing data values from the regression line. Note: only the missing data points derive their new values from the regression line while the rest of the data points retain their original values. The function terminates if there are trajectories with only one observation. The default is '1': Mean value of the corresponding column
fill_zeros	[TRUE or FALSE] whether to consider zeros 0 as missing values when 2: regression method is used. The default is FALSE.
verbose	to suppress printing output messages (to the console). Default: TRUE.

Details

Given a matrix or data.frame with some missing values indicated by (NA, Inf, null), this function impute the missing value by using either an estimation from the corresponding rows or columns, or to use a regression method to estimate the missing values.

Value

A data.frame with missing values (NA, Inf, null) imputed according to the a specified technique.

Examples

```
# Using the example 'traj' datasets

imp_data <- data_imputation(traj, id_field = TRUE, method = 2,
```

```
replace_with = 1,
fill_zeros = FALSE, verbose=FALSE)
```

`elbow_point` *Determine the elbow point on a curve*

Description

Given a list of x, y coordinates on a curve, function determines the elbow point of the curve.

Usage

```
elbow_point(x, y)
```

Arguments

x	vector of x coordinates of points on the curve
y	vector of y coordinates of points on the curve

Details

highlight the maximum curvature to identify the elbow point (credit: 'github.com/agentlans')

Value

indicate the optimal k value determined by the elbow point point.

Examples

```
# Generate some curve
x <- runif(100, min=-2, max=3)
y <- -exp(-x) * (1+rnorm(100)/3)
plot(x, y)
#Plot elbow points
abline(v=elbow_point(x,y)$y, col="blue", pch=20, cex=3)
```

`outlier_detect` *Outlier detection and replacement*

Description

This function identifies outlier observations in the trajectories, and allows users to replace the observations or remove trajectories entirely.

Usage

```
outlier_detect(traj, id_field = FALSE, method = 1, threshold = 0.95,
count = 1, replace_with = 1, verbose=TRUE)
```

Arguments

`traj`	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time points.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognizes the second column as the first time step.
`method`	[integer (numeric)] indicating the method for identifying the outlier. Options are: `'1'`: quantile method (`default`), and `'2'`: manual method. The `manual` method requires a user-defined value.
`threshold`	[numeric] A cut-off value for outliers. If the `method` parameter is set as `'1'`:quantile, the `threshold` should be a numeric vector of probability between `[0,1]`, whilst if the `method` is set as `'2'`: `manual`, the `threshold` could be any numeric vector.
`count`	[integer (numeric)] indicating the number of observations (in a trajectory) that must exceed the `threshold` in order for the trajectory to be considered an `outlier`. Default is `1`.
`replace_with`	[integer (numeric)] indicating the technique to use for calculating a replacement for an outlier observation. The remaining observations on the row or the column in which the outlier observation is located are used to calculate the replacement. The replacement options are: `'1'`: Mean value of the column, `'2'`: Mean value of the row and `'3'`: remove the row (trajectory) completely from the data. Default value is the `'1'` option.
`verbose`	to suppress output messages (to the console). Default: `TRUE`.

Details

Given a matrix, this function identifies outliers that exceed the threshold and replaces the outliers with an estimate calculated using the other observations either the rows or the columns in which the outlier observation is located. Option is also provided to remove the trajectories (containing the outlier) from the data.

Value

A dataframe with outlier observations replaced or removed.

Examples

```
data(traj)

trajectry <- data_imputation(traj, id_field=TRUE, method = 1,
   replace_with = 1, verbose=FALSE)

trajectry <- props(trajectry$CompleteData, id_field=TRUE)

outp <- outlier_detect(trajectry, id_field = TRUE, method = 1,
threshold = 0.95, count = 1, replace_with = 1, verbose=TRUE)

outp <- outlier_detect(trajectry, id_field = TRUE, method = 2, threshold = 15,
  count = 4, replace_with = 3, verbose=TRUE)
```

---

plot_akstats *Plot of cluster groups.*

Description

Takes the 'ak_object' from the `'akclustr'` as input and produce either the 'line' plot or 'stacked' histogram.

Usage

```
plot_akstats(
  ak_object,
  k = 3,
  reference = 1,
  n_quant = 4,
  type = "lines",
  y_scaling = "fixed"
)
```

Arguments

`ak_object`	An output of `akclustr` function. The object contains individual trajectories and their cluster solution(s) at the specified values of `k`. Also, includes the optimal value of `k` based on the criterion specified. at (different) values of `k` the `traj`.
`k`	[integer] `k` cluster to generate its solution.
`reference`	[numeric] Specifying the reference line from which the direction of each group is measured. Options are: `1`: slope of mean trajectory, `2`: slope of medoid trajectory, `3`: slope of a horizontal line (i.e. slope = 0). Default: `1`.

n_quant	[numeric] Number of equal intervals (quantiles) to create between the reference line (R) and the medoids (M) of the most-diverging groups of both sides of (R). Default is 4 - meaning quartile subdivisions on each side of (R). In this scenario, the function returns the quartile in which the medoid of each group falls. This result can be used to further categorize the groups into 'classes'. For example, groups that fall within the 1st quartile may be classified as 'Stable' groups (Adepeju et al. 2021).
type	[character] plot type. Available options are: "lines" and "stacked".
y_scaling	[character] works only if type="lines". y_scaling set the vertical scales of the cluster panels. Options are: "fixed": uses uniform scale for all panels, "free": uses variable scales for panels.

Details

Generates the plots of cluster groups - same plots generated by the 'show_plots' argument of print_akstats. The function draw from the functionalities of the ggplot2 library. For a more customized visualisation, we recommend that users deploy ggplot2 directly (Wickham H. (2016)).

Value

A plot showing group membership or sizes (proportion) and statistics.

References

1. Adepeju, M. et al. (2021). Anchored k-medoids: A novel adaptation of k-medoids further refined to measure inequality in the exposure to crime across micro places, doi: 10.1007/s42001-021-00103-1.

2. Wickham H. (2016). Elegant graphics for Data Analysis. Spring-Verlag New York (2016).

Examples

```
data(traj)

trajectry <- data_imputation(traj, id_field = TRUE, method = 1,
replace_with = 1, fill_zeros = FALSE)

print(trajectry$CompleteData)

trajectry <- props(trajectry$CompleteData, id_field = TRUE)

aksolution <- akclustr(trajectry, id_field = TRUE,
method = "linear", k = c(3,5), crit='Calinski_Harabasz')

plot_akstats(aksolution, k = 4, type="lines",
y_scaling="fixed")

plot_akstats(aksolution, k = 4, reference = 1,
n_quant = 4, type="stacked")
```

popl *Simulated population data.*

Description

Sample simulated population data to be used as the denominator variable against 'traj' dataset. Contains data for two consecutive census years

Usage

```
popl
```

Format

A dataframe with the following variables:

- location_id: Character id of sample census unit at which the population is obtained.
- census_2003: Population estimates at the sample locations for the census year 2003.
- census_2007: Population estimates at the sample locations for the census year 2007.

print_akstats *Descriptive (Change) statistics*

Description

This function perform two tasks: (i) it generate the descriptive and change statistics of groups, particularly suited for the outputs form the `akclustr` function, and (ii) generates the plots of the groups (performances).

Usage

```
print_akstats(ak_object, k = 3, reference = 1, n_quant = 4, show_plots = FALSE)
```

Arguments

`ak_object`	An output of `akclustr` function. The object contains individual trajectories and their cluster solution(s) at the specified values of `k`. Also, includes the optimal value of `k` based on the criterion specified. at (different) values of `k` the `traj`.
`k`	[integer] `k` cluster to generate its solution.
`reference`	[numeric] Specifying the reference line from which the direction of each group is measured. Options are: `1`: slope of mean trajectory, `2`: slope of medoid trajectory, `3`: slope of a horizontal line (i.e. slope = 0). Default: `1`.

n_quant	[numeric] Number of equal intervals (quantiles) to create between the reference line (R) and the medoids (M) of the most-diverging groups of both sides of (R). Default is 4 - meaning quartile subdivisions on each side of (R). In this scenario, the function returns the quartile in which the medoid of each group falls. This result can be used to further categorize the groups into 'classes'. For example, groups that fall within the 1st quartile may be classified as 'Stable' groups (Adepeju et al. 2021).
show_plots	[TRUE or FALSE] Provides the trajectory group plot. Please, see plot_akstats function for more plot options. Defaults FALSE

Details

Generates the plot of trajectory groupings. Given an ak_object class (from the akclustr function), this function show the plots of cluster groups. The plot component draws from plot_akstats function.

Value

A plot showing group membership or sizes (proportion) and statistics.

References

1. Adepeju, M. et al. (2021). Anchored k-medoids: A novel adaptation of k-medoids further refined to measure inequality in the exposure to crime across micro places, doi: 10.1007/s42001-021-00103-1.

2. Wickham H. (2016). Elegant graphics for Data Analysis. Spring-Verlag New York (2016).

Examples

```
data(traj)

trajectry <- data_imputation(traj, id_field = TRUE, method = 1,
replace_with = 1, fill_zeros = FALSE)

print(trajectry$CompleteData)

trajectry <- props(trajectry$CompleteData, id_field = TRUE)

aksolution <- akclustr(trajectry, id_field = TRUE,
    method = "linear", k = c(3,5), crit='Calinski_Harabasz')

print_akstats(aksolution, k = 4, show_plots=FALSE)
```

`props` *Conversion of counts (or rates) to 'Proportion'*

Description

This function converts counts or rates to proportions.

Usage

```
props(traj, id_field = TRUE, scale = 1, digits = 4)
```

Arguments

`traj`	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time points.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognizes the second column as the first time step.
`scale`	[numeric] To scale the 'proportion' measures. Default: `1`
`digits`	[numeric] Specifying number of digits to approximate the output to. Default: `4`.

Details

Given a matrix of observations (counts or rates), this function converts each observation to a proportion equivalent to the sum of each column. In other words, each observation is divided by the sum of the column where it is located, i.e. `prop = [a cell value] / sum[corresponding column]`

Value

A dataframe of proportion measures

Examples

```
trajectry <- data_imputation(traj, id_field = TRUE, method = 2,
replace_with = 1, fill_zeros = FALSE) #filling the missing values

trajectry <- props(trajectry$CompleteData, id_field = TRUE,
scale=1, digits=4)

print(trajectry)
```

rates *Conversion of counts to rates*

Description

Calculates rates from 'observed' count and a denominator data

Usage

```
rates(traj, denomin, id_field, multiplier)
```

Arguments

traj	[matrix (numeric)] longitudinal (e.g. observed count) data (m x n). Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time steps.
denomin	[matrix (numeric)] longitudinal (denominator) data of the same column as 'traj' (n).
id_field	[numeric or character] Default is TRUE. The first column of both the 'traj' and the 'denomin' object must be the unique (id) field. If FALSE, the function will terminate. The assumption is that columns of both the traj and denominat corresponds. That is, column2, column3, ... represent time points 2, 3, ..., respectively, in each object.
multiplier	[numeric] A quantify by which to the ratio traj/denomin is expressed. Default is 100.

Value

An object which comprised of four output variables, namely: (i) '$common_ids' - individual ids present in both 'traj' (trajectory data) and 'denomin' (denominator data); (ii) '$ids_unique_to_traj_data' - individual ids unique to trajectory data (i.e. not present in the denominator data); (iii) '$ids_unique_to_denom_data' - individual ids unique to denominator data (i.e. not present in the trajectory data); (iv) '' - a dataframe of rates estimates. Note: only the individual ids in '$rates_estimates' are used in the 'rates' estimation.

Examples

```
traj2 <- data_imputation(traj, id_field = TRUE, method = 2,
replace_with = 1, fill_zeros = FALSE)

pop <- popl #read denominator data

pop2 <- as.data.frame(matrix(0, nrow(popl), ncol(traj)))

colnames(pop2) <- names(traj2$CompleteData)
```

```
pop2[,1] <- as.vector(as.character(pop[,1]))

pop2[,4] <- as.vector(as.character(pop[,2]))

pop2[,8] <- as.vector(as.character(pop[,3]))

list_ <- c(2, 3, 5, 6, 7, 9, 10) #vector of missing years

#fill the missing fields with 'NA'
for(u_ in seq_len(length(list_))){
    pop2[,list_[u_]] <- "NA"
}

#estimate missing fields
pop_imp_result <- data_imputation(pop2, id_field = TRUE, method = 2,
replace_with = 1, fill_zeros = FALSE)

#calculate rates i.e. crimes per 200 population
crime_rates <- rates(traj2$CompleteData, denomin=pop_imp_result$CompleteData,
id_field=TRUE, multiplier = 200)
```

`remove_rows_n` *Removes rows that contain 'NA' and/or 'Inf' entries*

Description

This function removes any rows in which an 'NA' or an 'Inf' entry is found. The function is also able to remove records with 'Inf' entries, distinguishing it from the popular 'na.omit()' function in R.

Usage

```
remove_rows_n(traj, id_field=TRUE, remove=1)
```

Arguments

`traj`	[data.frame (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time points.
`id_field`	[numeric or character] Whether the first column of the `traj` is a unique (`id`) field. Default: `FALSE`. If `TRUE` the function recognises the second column as the first time step.
`remove`	[integer] Type of missing entries to remove. `1` for 'NA', `2` for 'Inf', and `3` for both. Default:`1`.

Details

Given a matrix (or a dataframe) containing an 'NA' or an 'Inf' entry, the function returns only rows with complete observations.

Value

A matrix with complete observations

Examples

```
data(traj)

remove_rows_n(traj, id_field=TRUE, remove=3)
```

`simulated` *Simulated longitudinal dataset*

Description

Contains simulated trajectories belonging to one of the three pre-defined groups, namely (a) decreasing, (b) stable and (c) increasing groups.

Usage

```
simulated
```

Format

A dataframe with the following variables:

- X1: Values across locations at time step 1
- X2: Values across locations at time step 2
- X3: Values across locations at time step 3
- X4: Values across locations at time step 4
- X5: Values across locations at time step 5
- X6: Values across locations at time step 6
- X7: Values across locations at time step 7
- X8: Values across locations at time step 8
- X9: Values across locations at time step 9
- X10: Values across locations at time step 10
- X11: Values across locations at time step 11
- X12: Values across locations at time step 12
- X13: Values across locations at time step 13
- X14: Values across locations at time step 14
- X15: Values across locations at time step 15
- X16: Values across locations at time step 16
- X17: Values across locations at time step 17

- X18: Values across locations at time step 18
- X19: Values across locations at time step 19
- X20: Values across locations at time step 20
- X21: Values across locations at time step 21

`TO1Risk` *Time-at-risk for the Adjudicated Toronto Youth Data (Sample 1)*

Description

Real-life time-at-risk per year for 378 individuals from the age of 8 to 38 in the Toronto, Ontario, Canada. The data is obtained through the R package 'crimCV'. For further information, please see: Nielsen, J. (2018) crimCV: Group-Based Modelling of Longitudinal Data. R package version 0.9.6. URL https://CRAN.R-project.org/package=crimCV.

Usage

```
TO1Risk
```

Format

A dataframe with the following variables:

- 8: Time-at-risk per year at age 8
- 9: Time-at-risk per year at age 9
- 10: Time-at-risk per year at age 10
- 11: Time-at-risk per year at age 11
- 12: Time-at-risk per year at age 12
- 13: Time-at-risk per year at age 13
- 14: Time-at-risk per year at age 14
- 15: Time-at-risk per year at age 15
- 16: Time-at-risk per year at age 16
- 17: Time-at-risk per year at age 17
- 18: Time-at-risk per year at age 18
- 19: Time-at-risk per year at age 19
- 20: Time-at-risk per year at age 20
- 21: Time-at-risk per year at age 21
- 22: Time-at-risk per year at age 22
- 23: Time-at-risk per year at age 23
- 24: Time-at-risk per year at age 24
- 25: Time-at-risk per year at age 25

- 26: Time-at-risk per year at age 26
- 27: Time-at-risk per year at age 27
- 28: Time-at-risk per year at age 28
- 29: Time-at-risk per year at age 29
- 30: Time-at-risk per year at age 30
- 31: Time-at-risk per year at age 31
- 32: Time-at-risk per year at age 32
- 33: Time-at-risk per year at age 33
- 34: Time-at-risk per year at age 34
- 35: Time-at-risk per year at age 35
- 36: Time-at-risk per year at age 36
- 37: Time-at-risk per year at age 37
- 38: Time-at-risk per year at age 38

`traj` *Sample longitudinal dataset*

Description

Simulated longitudinal datasets containing trajectories with missing values (`NA`, `Inf`, `null`)

Usage

```
traj
```

Format

A dataframe with the following variables:

- location_ids: Character id of sample locations at which values are obtained.
- X2001: Values at time step 1 (i.e. year 2001)
- X2002: Values at time step 2 (i.e. year 2002)
- X2003: Values at time step 3 (i.e. year 2003)
- X2004: Values at time step 4 (i.e. year 2004)
- X2005: Values at time step 5 (i.e. year 2005)
- X2006: Values at time step 6 (i.e. year 2006)
- X2007: Values at time step 7 (i.e. year 2007)
- X2008: Values at time step 8 (i.e. year 2008)
- X2009: Values at time step 9 (i.e. year 2009)

traj_w_spaces *Sample longitudinal dataset containing whitespaces*

Description

Longitudinal dataset with both trailing and leading whitespaces. For example, there is a trailing whitespace at cell [3, 6], while there is a leading whitespace at cell [9, 4].

Usage

```
traj_w_spaces
```

Format

A dataframe with the following variables:

- location_ids: Character id of sample locations at which values are obtained.
- X2001: Values at time step 1 (i.e. year 2001)
- X2002: Values at time step 2 (i.e. year 2002)
- X2003: Values at time step 3 (i.e. year 2003)
- X2004: Values at time step 4 (i.e. year 2004)
- X2005: Values at time step 5 (i.e. year 2005)
- X2006: Values at time step 6 (i.e. year 2006)
- X2007: Values at time step 7 (i.e. year 2007)
- X2008: Values at time step 8 (i.e. year 2008)
- X2009: Values at time step 9 (i.e. year 2009)

w_spaces *Whitespaces removal*

Description

This function removes all the leading and the trailing whitespaces in data

Usage

```
w_spaces(traj, remove="Both", verbose=TRUE)
```

Arguments

traj	[matrix (numeric)]: longitudinal data. Each row represents an individual trajectory (of observations). The columns show the observations at consecutive time points.
remove	[character]: Type of whitespace to remove. That is, "Left" (leading), (2) "Right" (trailing), or "Both" (both leading and trailing whitespaces). Default: "Both".
verbose	to suppress output messages (to the console). Default: TRUE.

Details

Given a matrix suspected to contain whitespaces, this function removes the type of the whitespaces specified and returns a cleaned data. 'Whitespaces' are white characters often introduced into data during data entry, for instance by wrongly pressing the spacebar. For example, neither " A" nor "A " is the same as "A" because of the whitespaces that exist in them. They can also result from systematic errors in data recording devices.

Value

A matrix with all whitespaces (if any) removed.

References

https://en.wikipedia.org/wiki/Whitespace_character

Examples

```
data(traj_w_spaces)

w_spaces(traj_w_spaces, remove="Both", verbose=TRUE)
```

Index