Package 'dataprep'

January 15, 2022

Type Package

Title Efficient and Flexible Data Preprocessing Tools

Version 0.1.5

Author Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>, Hao Wu, Hai-Yan Li, Qiang Zhang, Zhanqing Li, Ke-Bin He, Lanzhou University, Tsinghua University

Maintainer Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

Description Efficiently and flexibly preprocess data using a set of data filtering, deletion, and interpolation tools.
These data preprocessing methods are developed based on the principles of completeness, accuracy, threshold method, and linear interpolation and through the setting of constraint conditions, time completion & recovery, and fast & efficient calculation and grouping.
Key preprocessing steps include deletions of variables and observations, outlier removal, and missing values (NA) interpolation, which are dependent on the incomplete and dispersed degrees of raw data.
They clean data more accurately, keep more samples, and add no outliers after interpolation, compared with ordinary methods.
Auto-identification of consecutive NA via run-length based grouping is used in observation deletion, outlier removal, and NA interpolation; thus, new outliers are not generated in interpolation. Conditional extremum is proposed to realize point-by-point weighed outlier removal that saves non-outliers from being removed.
Plus, time series interpolation with values to refer to within short periods further ensures reliable interpolation.
These methods are based on and improved from the reference: Liang, C.-S., Wu, H., Li, H.-Y., Zhang, Q., Li, Z. & He, K.-B. (2020) <doi:10.1016/j.scitotenv.2020.140923>.

Depends R (>= 3.5.0)

Imports ggplot2, scales, foreach, doParallel, dplyr, reshape2, data.table, zoo

License GPL (>= 2)

Suggests knitr, rmarkdown

VignetteBuilder knitr, rmarkdown

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-15 13:32:42 UTC

R topics documented:

`condextr` *Remove outliers using point-by-point weighed outlier removal by conditional extremum*

Description

Care is needed when dealing with outliers that are common real-life phenomena besides missing values in data. Unfortunately, many non-outliers may be removed by a one-for-all threshold method, which will be largely avoided if a one-by-one considered way is developed and applied. The `condextr` proposed here considers every value (point) that will potentially be removed, combining constraint conditions and extremum (maximum and minimum). Therefore, it is a function of point-by-point weighed outlier removal by conditional extremum. Observation deletion is combined in the process of outlier removal since large gaps consisted of excessive missing values may be formed in time series after removing certain outliers.

Usage

```
condextr(data, start = NULL, end = NULL, group = NULL, top = 0.995,
top.error = 0.1, top.magnitude = 0.2, bottom = 0.0025, bottom.error = 0.2,
bottom.magnitude = 0.4, interval = 10, by = "min", half = 30,
times = 10, cores = NULL)
```

Arguments

`data` A data frame containing outliers (and missing values). Its columns from `start` to `end` will be checked.

`start` The column number of the first selected variable.

`end` The column number of the last selected variable.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`top` The top percentile is 0.995 by default.

`top.error` The top allowable error coefficient is 0.1 by default.

`top.magnitude` The order of magnitude coefficient of the top error is 0.2 by default.

`bottom` The bottom percentile is 0.0025 by default.

`bottom.error` The bottom allowable error coefficient is 0.2 by default.

`bottom.magnitude`
The order of magnitude coefficient of the bottom error is 0.4 by default.

`interval` The interval of observation deletion, i.e. the number of outlier deletions before each observation deletion, is 10 by default.

`by` The time extension unit by is a minute ("min") by default. The user can specify other time units. For example, "5 min" means that the time extension unit is 5 minutes.

`half` Half window size of hourly moving average. It is 30 (minutes) by default, which is determined by the time expansion unit minute ("min").

`times` The number of observation deletions in outlier removal is 10 by default.

`cores` The number of CPU cores.

Details

A point-by-point constraint (consideration) outlier removal method based on conditional extremum is proposed, which is more advantageous than the traditional "one size fits all" percentile deletion method in deleting outliers. Moreover, it emphasizes that the outlier removal should be grouped if there are groups such as month because of the value difference among different groups.

Value

A data frame after removing outliers.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H., Francois, R., Henry, L. & Muller, K. 2017. dplyr: A Grammar of Data Manipulation. 0.7.4 ed. http://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

3. Wickham, H., Francois, R., Henry, L. & Muller, K. 2019. dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr.

4. Dowle, M., Srinivasan, A., Gorecki, J., Short, T., Lianoglou, S., Antonyan, E., 2017. data.table: Extension of 'data.frame', 1.10.4-3 ed, http://r-datatable.com.

5. Dowle, M., Srinivasan, A., 2021. data.table: Extension of ‘data.frame‘. R package version 1.14.0. https://CRAN.R-project.org/package=data.table.

6. Wallig, M., Microsoft & Weston, S. 2020. foreach: Provides Foreach Looping Construct. R package version 1.5.0. https://CRAN.R-project.org/package=foreach.

7. Ooi, H., Corporation, M. & Weston, S. 2019. doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.15. https://CRAN.R-project.org/package=doParallel.

Examples

```
# Remove outliers by condextr after deleting observations by obsedele
# 337 observations will be deleted in obsedele(data[,c(1:4,27:61)],5,39,4).
# Further, 362 observations will be deleted in condextr by obsedele
# Here, for executing time reason, a smaller example is used to show.
# Besides, only 2 cores are used for submission test.
condextr(obsedele(data[1:500,c(1,4,17:19)],3,5,2,cores=2),3,5,2,cores=2)
```

data *Example data (particle number concentrations in SMEAR I Varrio forest)*

Description

The raw data is downloaded from https://smear.avaa.csc.fi/download.

Usage

```
data
```

Format

A data frame with 7640 observations on the following 65 variables.

`date` a POSIXct

`tconc` a numeric vector

`TPNC` a numeric vector

`monthyear` a character vector

‘`1`’ a numeric vector

‘`1.12`’ a numeric vector

‘`1.26`’ a numeric vector

‘`1.41`’ a numeric vector

‘`1.58`’ a numeric vector

‘`1.78`’ a numeric vector

‘`2`’ a numeric vector

‘`2.24`’ a numeric vector

‘`2.51`’ a numeric vector

‘`2.82`’ a numeric vector

‘`3.16`’ a numeric vector

‘`3.55`’ a numeric vector

‘`3.98`’ a numeric vector

‘`4.47`’ a numeric vector

‘`5.01`’ a numeric vector

‘`5.62`’ a numeric vector

‘`6.31`’ a numeric vector

‘`7.08`’ a numeric vector

‘`7.94`’ a numeric vector

‘`8.91`’ a numeric vector

‘`10`’ a numeric vector

‘`11.2`’ a numeric vector

‘`12.6`’ a numeric vector

‘`14.1`’ a numeric vector

‘`15.8`’ a numeric vector

‘`17.8`’ a numeric vector

‘`20`’ a numeric vector

‘`22.4`’ a numeric vector

‘`25.1`’ a numeric vector

‘`28.2`’ a numeric vector

‘`31.6`’ a numeric vector

‘`35.5`’ a numeric vector

‘`39.8`’ a numeric vector

‘`44.7`’ a numeric vector

‘`50.1`’ a numeric vector

‘`56.2`’ a numeric vector

‘`63.1`’ a numeric vector

‘70.8’ a numeric vector

‘79.4’ a numeric vector

‘89.1’ a numeric vector

‘100’ a numeric vector

‘112’ a numeric vector

‘126’ a numeric vector

‘141’ a numeric vector

‘158’ a numeric vector

‘178’ a numeric vector

‘200’ a numeric vector

‘224’ a numeric vector

‘251’ a numeric vector

‘282’ a numeric vector

‘316’ a numeric vector

‘355’ a numeric vector

‘398’ a numeric vector

‘447’ a numeric vector

‘501’ a numeric vector

‘562’ a numeric vector

‘631’ a numeric vector

‘708’ a numeric vector

‘794’ a numeric vector

‘891’ a numeric vector

‘1000’ a numeric vector

Source

https://smear.avaa.csc.fi/download

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

Examples

```
data
## maybe str(data)
```

data1 — *Example data (data1, particle number concentrations in SMEAR I Varrio forest)*

Description

Calculated from the raw data that is downloaded from https://smear.avaa.csc.fi/download.

Usage

```
data1
```

Format

A data frame with 7640 observations on the following 7 variables.

`date` a POSIXct

`monthyear` a character vector

`Nucleation` a numeric vector

`Aitken` a numeric vector

`Accumulation` a numeric vector

`tconc` a numeric vector

`TPNC` a numeric vector

Source

https://smear.avaa.csc.fi/download

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

Examples

```
data1
## maybe str(data1)
```

dataprep *Data preprocessing with multiple steps in one function*

Description

The four steps, i.e., variable deletion by `varidele`, observation deletion by `obsedele`, outlier removal by `condextr`, and missing value interpolation by `shorvalu` can be finished in `dataprep`.

Usage

```
dataprep(data, start = NULL, end = NULL, group = NULL, optimal = FALSE,
interval = 10, times = 10, fraction = 0.25,
top = 0.995, top.error = 0.1, top.magnitude = 0.2,
bottom = 0.0025, bottom.error = 0.2, bottom.magnitude = 0.4, by = "min",
half = 30, intervals = 30, cores = NULL)
```

Arguments

`data` A data frame containing outliers (and missing values). Its columns from `start` to `end` will be checked.

`start` The column number of the first selected variable.

`end` The column number of the last selected variable.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`optimal` A Boolean to decide whether the `optisolu` should be used to find optimal `interval` and `times` for `condextr`.

`interval` The interval of observation deletion, i.e. the number of outlier deletions before each observation deletion, is 10 by default.

`times` The number of observation deletions in outlier removal is 10 by default.

`fraction` The proportion of missing values of variables. Default is 0.25.

`top` The top percentile is 0.995 by default.

`top.error` The top allowable error coefficient is 0.1 by default.

`top.magnitude` The order of magnitude coefficient of the top error is 0.2 by default.

`bottom` The bottom percentile is 0.0025 by default.

`bottom.error` The bottom allowable error coefficient is 0.2 by default.

`bottom.magnitude`
The order of magnitude coefficient of the bottom error is 0.4 by default.

`by` The time extension unit by is a minute ("min") by default. The user can specify other time units. For example, "5 min" means that the time extension unit is 5 minutes.

`half` Half window size of hourly moving average. It is 30 (minutes) by default, which is determined by the time expansion unit minute ("min").

`intervals` The time gap of dividing periods as groups, is 30 (minutes) by default. This confines the interpolation inside short periods so that each interpolation has observed value(s) to refer to within every half an hour.

`cores` The number of CPU cores.

Details

If optimal = T, relatively much more time will be needed to finish the data preprocessing, but it gets better final result.

Value

A preprocessed data frame after variable deletion, observation deletion, outlier removal, and missing value interpolation.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H., Francois, R., Henry, L. & Muller, K. 2017. dplyr: A Grammar of Data Manipulation. 0.7.4 ed. http://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

3. Wickham, H., Francois, R., Henry, L. & Muller, K. 2019. dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr.

4. Dowle, M., Srinivasan, A., Gorecki, J., Short, T., Lianoglou, S., Antonyan, E., 2017. data.table: Extension of 'data.frame', 1.10.4-3 ed, http://r-datatable.com.

5. Dowle, M., Srinivasan, A., 2021. data.table: Extension of ‘data.frame‘. R package version 1.14.0. https://CRAN.R-project.org/package=data.table.

6. Wallig, M., Microsoft & Weston, S. 2020. foreach: Provides Foreach Looping Construct. R package version 1.5.0. https://CRAN.R-project.org/package=foreach.

7. Ooi, H., Corporation, M. & Weston, S. 2019. doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.15. https://CRAN.R-project.org/package=doParallel.

8. Zeileis, A. & Grothendieck, G. 2005. zoo: S3 infrastructure for regular and irregular time series. Journal of Statistical Software, 14(6):1-27.

9. Zeileis, A., Grothendieck, G. & Ryan, J.A. 2019. zoo: S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations). R package version 1.8-6. https://cran.r-project.org/web/packages/zoo/.

See Also

`dataprep::varidele`, `dataprep::obsedele`, `dataprep::condextr`, `dataprep::shorvalu` and `dataprep::optisolu`

Examples

```
# Combine 4 steps in one function
# In dataprep(data,5,65,4), 26 variables, 1097 outliers and 699 observations will be deleted
# Besides, 6012 missing values will be replaced
# Setting optimal=T will get optimized result
# Here, for executing time reason, a smaller example is used to show
dataprep(data[1:60,c(1,4,18:19)],3,4,2,
interval=2,times=1,cores=2)

# Check if results are the same
identical(shorvalu(condextr(obsedele(varidele(
data[1:60,c(1,4,18:19)],3,4),3,4,2,cores=2),3,4,2,
interval=2,times=1,cores=2),3,4),
dataprep(data[1:60,c(1,4,18:19)],3,4,2,
interval=2,times=1,cores=2))
```

`descdata` *Fast descriptive statistics*

Description

It describes data using R basic functions, without calling other packages to avoid redundant calculations, which is faster.

Usage

```
descdata(data, start = NULL, end = NULL, stats= 1:9, first = "variables")
```

Arguments

`data` A data frame to describe, from the column `start` to the column end.

`start` The column number of the first variable to describe.

`end` The column number of the last variable to describe.

`stats` Selecting or rearranging the items from the 9 statistics, i.e., n, na, mean, sd, median, trimmed, min, max, and IQR. It can be a vector or a single value, in 'character' or 'numeric' class.

`first` The name of the first column of the output. It is the general name of the items (variables).

Details

This function can be used for different types of data, as long as the variables are numeric. Because it describes the data frame from the column `start` to the column `end`, the variables need to be linked together instead of being scattered.

Value

A data frame of descriptive statistics:

`size`	default general name of items (variables). Users can define it via the parameter first.
`n`	number of valid cases
`na`	number of invalid cases
`mean`	mean of each item
`sd`	standard deviation
`median`	median of each item
`trimmed`	trimmed mean (with trim defaulting to .1)
`min`	minimum of each item
`max`	maximum of each item
`IQR`	interquartile range of each item

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

See Also

`dataprep::descplot`

Examples

```
# Variable names are essentially numeric
descdata(data,5,65)
# Use numbers to select statistics
descdata(data,5,65,c(2,7:9))
# Use characters to select statistics
descdata(data,5,65,c('na','min','max','IQR'))

# When type of variable names is character
descdata(data1,3,7)
# Use numbers to select statistics
descdata(data1,3,7,c(2,7:9))
# Use characters to select statistics
descdata(data1,3,7,c('na','min','max','IQR'))
```

descplot *View the descriptive statistics via plot*

Description

It applies to an original (a raw) data and produces a plot to describe the data with 9 statistics including n, na, mean, sd, median, trimmed, min, max, and IQR.

Usage

```
descplot(data, start = NULL, end = NULL, stats= 1:9, first = "variables")
```

Arguments

`data` A data frame to describe, from the column `start` to the column `end`.

`start` The column number of the first variable to describe.

`end` The column number of the last variable to describe.

`stats` Selecting or rearranging the items from the 9 statistics, i.e., n, na, mean, sd, median, trimmed, min, max, and IQR. It can be a vector or a single value, in 'character' or 'numeric' class.

`first` The name of the first column of the output. It is the general name of the items (variables).

Details

This function will describe the data first using descdata. Then, A plot to show the result will be produced using the package `ggplot2` (coupled with self-defined `melt` or `reshape2::melt` to melt the intermediate data). The variables from `start` to `end` need to be linked together instead of being scattered.

Value

A plot to show the descriptive result of the data, including:

`size` default general name of items (variables). Users can define it via the parameter `first`.

`n` number of valid cases

`na` number of invalid cases

`mean` mean of each item

`sd` standard deviation

`median` median of each item

`trimmed` trimmed mean (with trim defaulting to .1)

`min` minimum of each item

`max` maximum of each item

`IQR` interquartile range of each item

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H. 2007. Reshaping data with the reshape package. Journal of Statistical Software, 21(12):1-20.

3. Wickham, H. 2009. ggplot2: Elegant Graphics for Data Analysis. http://ggplot2.org: Springer-Verlag New York.

4. Wickham, H. 2016. ggplot2: elegant graphics for data analysis. Springer-Verlag New York.

See Also

`dataprep::descdata` and `dataprep::melt`

Examples

```
# Line plots for variable names that are essentially numeric
descplot(data,5,65)
# Use numbers to select statistics
descplot(data,5,65,c(2,7:9))
# Use characters to select statistics
descplot(data,5,65,c('na','min','max','IQR'))

# Bar charts for type of variable names that is character
descplot(data1,3,7)
# Use numbers to select statistics
descplot(data1,3,7,7:9)
# Use characters to select statistics
descplot(data1,3,7,c('min','max','IQR'))
```

`melt` *Turn variable names and values into two columns*

Description

Turn the names and values of all pending variables into two columns. These variables are inversely selected inside function (`cols`) and waiting to be melted. After melting, the data format changes from wide to long.

Usage

```
melt(data, cols = NULL)
```

Arguments

`data`	A data frame to melt, from the column `start` to the column end.
`cols`	Inversely selected columns inside function, except which are columns waiting to be melted.

Details

This function (`dataprep::melt`) will be used when `reshape2` is not installed.

Value

A long-format data frame from its original wide format.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H. 2007. Reshaping data with the reshape package. Journal of Statistical Software, 21(12):1-20.

Examples

```
# The first pending variable contains only NA
melt(data,1:4)

# Number concentrations of modes and total particles are not NA
melt(data1,1:2)
```

`obsedele` *Delete observations with variable(s) containing too many consecutive missing values (NA) in time series*

Description

The description of varidele mentions that missing values are common in real data but excessive missing values would led to inaccurate information and conclusions about the data. To control and improve the quality of original data, besides deleting the variables with too many missing values, the observations with one or more variables containing excessive consecutive missing values in time series should further be deleted.

Usage

```
obsedele(data, start = NULL, end = NULL, group = NULL, by = "min",
half = 30, cores = NULL)
```

Arguments

`data` A data frame containing variables with too many consecutive missing values (NA) in time series. Its columns from `start` to `end` will be checked.

`start` The column number of the first selected variable.

`end` The column number of the last selected variable.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`by` The time extension unit `by` is a minute ("min") by default. The user can specify other time units. For example, "5 min" means that the time extension unit is 5 minutes.

`half` Half window size of hourly moving average. It is 30 (minutes) by default, which is determined by the time expansion unit minute ("min"). Users can set its value as required.

`cores` The number of CPU cores.

Details

How to delete observations based on consecutive missing value? The idea here is to remove the observations with incomplete half-hour averages, that is, the observations with at least one variable missing more than half an hour. Besides the design of flexible constraints, fast and efficient algorithm is also used, which saves much more time. Using basic functions such `merge` and `seq` (in loop) temporarily to extend the full time period and using the fast and efficient `data.table::rleid`, `data.table::rowid`, and `data.table::setorder` to realize run-length based grouping (even faster than calculating moving average by C++ optimized algorithm) are very important to quickly detect consecutive missing values. For the loop, parallel computing can be conducted using packages `parallel`, `doParallel`, and `foreach`. Further, this method will also be used to delete outliers. In this way, it ensures that the observations with excessive consecutive missing values are deleted completely and the interpolation in time series is reasonable.

Value

A data frame after deleting observations with too many consecutive missing values in time series.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Dowle, M., Srinivasan, A., Gorecki, J., Short, T., Lianoglou, S., Antonyan, E., 2017. data.table: Extension of 'data.frame', 1.10.4-3 ed, http://r-datatable.com.

3. Dowle, M., Srinivasan, A., 2021. data.table: Extension of ‘data.frame‘. R package version 1.14.0. https://CRAN.R-project.org/package=data.table.

4. Wallig, M., Microsoft & Weston, S. 2020. foreach: Provides Foreach Looping Construct. R package version 1.5.0. https://CRAN.R-project.org/package=foreach.

5. Ooi, H., Corporation, M. & Weston, S. 2019. doParallel: Foreach Parallel Adaptor for the ’parallel’ Package. R package version 1.0.15. https://CRAN.R-project.org/package=doParallel.

Examples

```
# Select start as 27 and end as 61 according to varidele
# This selection ignores the first 22 and the last 4 variables
# Not show the first 22 variables dropped by varidele
# A total of 39 variables left (65 - 22 - 4)
# Here, a smaller example is used for saving time.
# Besides, only 2 cores are used for submission test.

obsedele(data[c(1:200,3255:3454),c(1:4,27:61)],5,39,4,cores=2)
```

`optisolu` *Find an optimal combination of* `interval` *and* `times` *for* `condextr`

Description

Optimal values of `interval` and `times` in the proposed conditional extremum based outlier removal method, i.e., `condextr` can be searched out after comparing with the traditional "one size fits all" percentile deletion method in deleting outliers. Three parameters are used for this comparison, including sample deletion ratio (SDR), outlier removal ratio (ORR), and signal-to-noise ratio (SNR).

Usage

```
optisolu(data, start = NULL, end = NULL, group = NULL, interval = 35, times = 10,
top = 0.995, top.error = 0.1, top.magnitude = 0.2,
bottom = 0.0025, bottom.error = 0.2, bottom.magnitude = 0.4,
by = "min", half = 30, cores = NULL)
```

Arguments

`data` A data frame containing outliers (and missing values). Its columns from `start` to `end` will be checked.

`start` The column number of the first selected variable.

`end` The column number of the last selected variable.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`interval` The interval of observation deletion, i.e., the number of outlier deletions before each observation deletion, is 35 by default. Its values from 1 to `interval` will be tested for optimal solution.

`times` The number of observation deletions in outlier removal is 10 by default. The values from 1 to `times` will be tested for optimal solution.

`top` The top percentile is 0.995 by default.

`top.error` The top allowable error coefficient is 0.1 by default.

`top.magnitude` The order of magnitude coefficient of the top error is 0.2 by default.

`bottom` The bottom percentile is 0.0025 by default.

`bottom.error` The bottom allowable error coefficient is 0.2 by default.

`bottom.magnitude`
The order of magnitude coefficient of the bottom error is 0.4 by default.

`by` The time extension unit by is a minute ("min") by default. The user can specify other time units. For example, "5 min" means that the time extension unit is 5 minutes.

`half` Half window size of hourly moving average. It is 30 (minutes) by default, which is determined by the time expansion unit minute ("min").

`cores` The number of CPU cores.

Details

The three ratios offer indices to show the quality of outlier removal methods. Besides, other parameters such as new outlier production (NOP) are also important. Since the preprocessing roadmap is based on the ideas of grouping, both flexible and strict constraints for outliers, and interpolation within short period and with effective observed values, the new outlier production is greatly restricted.

Value

A data frame indicating the quality of outlier remove by `condextr` with different values of `interval` and `times`. A total of 9 columns are listed in it.

`case` Order of combination of `interval` and `times`

`interval` The interval of observation deletion, i.e., the number of outlier deletions before each observation deletion

`times` The number of observation deletions in outlier removal

`sdr` Sample deletion ratio (SDR)

`orr` Outlier removal ratio (ORR)

`snr` Signal-to-noise ratio (SNR)

`index` Quality level of outlier removal based on the three parameters

`relaindex` A relative form of the index

`optimal` A Boolean variable to show if the result of conditional extremum is better, in terms of all the three parameters, than the traditional "one size fits all" percentile deletion method in deleting outliers.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H., Francois, R., Henry, L. & Muller, K. 2017. dplyr: A Grammar of Data Manipulation. 0.7.4 ed. http://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

3. Wickham, H., Francois, R., Henry, L. & Muller, K. 2019. dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr.

4. Dowle, M., Srinivasan, A., Gorecki, J., Short, T., Lianoglou, S., Antonyan, E., 2017. data.table: Extension of 'data.frame', 1.10.4-3 ed, http://r-datatable.com.

5. Dowle, M., Srinivasan, A., 2021. data.table: Extension of ‘data.frame‘. R package version 1.14.0. https://CRAN.R-project.org/package=data.table.

6. Wallig, M., Microsoft & Weston, S. 2020. foreach: Provides Foreach Looping Construct. R package version 1.5.0. https://CRAN.R-project.org/package=foreach.

7. Ooi, H., Corporation, M. & Weston, S. 2019. doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.15. https://CRAN.R-project.org/package=doParallel.

Examples

```
# Setting interval as 35 and times as 10 can find optimal solutions
# optisolu(obsedele(data[c(1:4,27:61)],5,39,4),5,39,4,35,10)
# Here, for executing time reason, a smaller example is used to show
# But too small interval and times will not get optimal solutions

optisolu(data[1:50,c(1,4,18:19)],3,4,2,2,1,cores=2)
```

percdata *Calculate the top and bottom percentiles of each selected variable*

Description

Outliers can be preliminarily checked by the calculated top and bottom percentiles. Basic R functions in packages from system library are used to get these percentiles of selected variables in data frames, instead of calling other packages. It saves time.

Usage

```
percdata(data, start = NULL, end = NULL, group = NULL, diff = 0.1, part = 'both')
```

Arguments

`data` A data frame to calculate percentiles, from the column `start` to the column `end`.

`start` The column number of the first variable to calculate percentiles for.

`end` The column number of the last variable to calculate percentiles for.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`diff` The common difference between `quantile`'s `probs`. Default is 0.1.

`part` The option of calculating bottom and/or top percentiles (parts). Default is 'both', or 2 for both bottom and top parts. Setting it as 'bottom' or 0 for bottom part and 'top' or 1 for top part.

Details

The data to be processed ranges from the column `start` to the last column `end`. The column numbers of these two columns are needed for the arguments. This requires that the variables of the data to be processed are arranged continuously in the database or table. Or else, it is necessary to move the columns in advance to make a continuous arrangement.

Value

Top (highest or greatest) and bottom (lowest or smallest) percentiles are calculated. According to the default `diff` (=0.1), the calculated values are as follows.

`0th`	Quantile with `probs` = `0`
`0.1th`	Quantile with `probs` = `0.001`
`0.2th`	Quantile with `probs` = `0.002`
`0.3th`	Quantile with `probs` = `0.003`
`0.4th`	Quantile with `probs` = `0.004`
`0.5th`	Quantile with `probs` = `0.005`
`99.5th`	Quantile with `probs` = `0.995`
`99.6th`	Quantile with `probs` = `0.996`
`99.7th`	Quantile with `probs` = `0.997`
`99.8th`	Quantile with `probs` = `0.998`
`99.9th`	Quantile with `probs` = `0.999`
`100th`	Quantile with `probs` = `1`

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

See Also

`dataprep::percplot`

Examples

```
# Select the grouping variable and remaining variables after deletion by varidele.
# Column 4 ('monthyear') is the group and the fraction for varidele is 0.25.
# After extracting according to the result by varidele, the group is in the first column.
percdata(data[,c(4,27:61)],2,36,1)
```

`percoutl` *Traditional percentile-based outlier removal*

Description

The percentile-based outlier removal methods usually take a quantile as a threshold and values above it will be deleted. Here, two quantiles are used for both the top and the bottom. For the bottom, accordingly, values below the quantile threshold will be removed.

Usage

```
percoutl(data, start = NULL, end = NULL, group = NULL,
top = 0.995, bottom = 0.0025, by = "min", half = 30, cores = NULL)
```

Arguments

`data`	A data frame containing outliers (and missing values). Its columns from `start` to `end` will be checked.
`start`	The column number of the first selected variable.
`end`	The column number of the last selected variable.
`group`	The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.
`top`	The top percentile is 0.995 by default.
`bottom`	The bottom percentile is 0.0025 by default.
`by`	The time extension unit by is a minute ("min") by default. The user can specify other time units. For example, "5 min" means that the time extension unit is 5 minutes.

`half`	Half window size of hourly moving average. It is 30 (minutes) by default, which is determined by the time expansion unit minute ("min").
`cores`	The number of CPU cores.

Details

Unlike `condextr`, a point-by-point considered outlier removal method, the traditional percentile-based `percoutl` is a "one size fits all" outlier deletion method. It may delete too many or too few values that are non-outliers or outliers respectively.

Value

A data frame after deleting outliers.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H., Francois, R., Henry, L. & Muller, K. 2017. dplyr: A Grammar of Data Manipulation. 0.7.4 ed. http://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

3. Wickham, H., Francois, R., Henry, L. & Muller, K. 2019. dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr.

4. Dowle, M., Srinivasan, A., Gorecki, J., Short, T., Lianoglou, S., Antonyan, E., 2017. data.table: Extension of 'data.frame', 1.10.4-3 ed, http://r-datatable.com.

5. Dowle, M., Srinivasan, A., 2021. data.table: Extension of ‘data.frame‘. R package version 1.14.0. https://CRAN.R-project.org/package=data.table.

6. Wallig, M., Microsoft & Weston, S. 2020. foreach: Provides Foreach Looping Construct. R package version 1.5.0. https://CRAN.R-project.org/package=foreach.

7. Ooi, H., Corporation, M. & Weston, S. 2019. doParallel: Foreach Parallel Adaptor for the 'parallel' Package. R package version 1.0.15. https://CRAN.R-project.org/package=doParallel.

See Also

`dataprep::condextr`

Examples

```
percoutl(obsedele(data[c(1:200,3255:3454),c(1:4,27:61)],5,39,4,cores=2),5,39,4,cores=2)
# Result
```

percplot *Plot the top and bottom percentiles of each selected variable*

Description

The top and bottom percentiles of selected variables calculated by `percdata` can be plotted by `percplot` that offers a vivid check of possible outliers. It uses `reshape2::melt` or `dataprep::melt` to melt the data and uses `ggplot2` and `scales` to plot the data.

Usage

```
percplot(data, start = NULL, end = NULL, group = NULL, ncol = NULL,
diff = 0.1, part = 'both')
```

Arguments

`data` A data frame to calculate percentiles, from the column `start` to the column `end`.

`start` The column number of the first variable to calculate percentiles for.

`end` The column number of the last variable to calculate percentiles for.

`group` The column number of the grouping variable. It can be selected according to whether the data needs to be processed in groups. If grouping is not required, leave it default (NULL); if grouping is required, set `group` as the column number (position) where the grouping variable is located. If there are more than one grouping variable, it can be turned into a longer group through combination and transformation in advance.

`ncol` The total columns of the plot.

`diff` The common difference between `quantile`'s `probs`. Default is 0.1.

`part` The option of plotting bottom and/or top percentiles (parts). Default is 'both', or 2 for both bottom and top parts. Setting it as 'bottom' or 0 for bottom part and 'top' or 1 for top part.

Details

Four scenes are considered according to the scales of x and y axes, namely the ranges of x and y values. For example, the code, `sd(diff(log(as.numeric(as.character(names(data[,start:end])))))) / mean(diff(log(as.numeric(as.character(names(data[,start:end])))))) < 0.1 & max(data[,start:end],na.r = T) / min(data[,start:end],na.rm = T) > = 10^3`, means that the coefficient of variation of the lagged differences of `log(x)` is below 0.1 and meanwhile the maximum y is 1000 times greater than or equal to the minimum y.

Value

Top (highest or greatest) and bottom (lowest or smallest) percentiles are plotted.

`0th` Quantile with `probs = 0`

`0.1th` Quantile with `probs = 0.001`

`0.2th`	Quantile with `probs` = `0.002`
`0.3th`	Quantile with `probs` = `0.003`
`0.4th`	Quantile with `probs` = `0.004`
`0.5th`	Quantile with `probs` = `0.005`
`99.5th`	Quantile with `probs` = `0.995`
`99.6th`	Quantile with `probs` = `0.996`
`99.7th`	Quantile with `probs` = `0.997`
`99.8th`	Quantile with `probs` = `0.998`
`99.9th`	Quantile with `probs` = `0.999`
`100th`	Quantile with `probs` = `1`

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H. 2007. Reshaping data with the reshape package. Journal of Statistical Software, 21(12):1-20.

3. Wickham, H. 2009. ggplot2: Elegant Graphics for Data Analysis. http://ggplot2.org: Springer-Verlag New York.

4. Wickham, H. 2016. ggplot2: elegant graphics for data analysis. Springer-Verlag New York.

5. Wickham, H. 2017. scales: Scale Functions for Visualization. 0.5.0 ed. https://github.com/hadley/scales.

6. Wickham, H. & Seidel, D. 2019. scales: Scale Functions for Visualization. R package version 1.1.0. https://CRAN.R-project.org/package=scales.

See Also

`dataprep::percdata` and `dataprep::melt`

Examples

```
# Plot
percplot(data,5,65,4)

# Plot
percplot(data1,3,7,2)
```

shorvalu *Interpolation with values to refer to within short periods*

Description

Time gaps and available values are considered in NA interpolation by `shorvalu`. Thus, more reliable interpolation is realized with these constraints and the successive using of `obsedele` in the preceding outlier removal.

Usage

```
shorvalu(data, start, end, intervals = 30, units = 'mins')
```

Arguments

`data`	A data frame containing outliers. Its columns from `start` to `end` will be checked.
`start`	The column number of the first selected variable.
`end`	The column number of the last selected variable.
`intervals`	The time gap of dividing periods as groups, is 30 (minutes) by default. This confines the interpolation inside short periods so that each interpolation has observed value(s) to refer to within every half an hour.
`units`	Units in time intervals/differences. It can be one of "secs", "mins", "hours", "days", or "weeks". The default is 'mins'.

Details

It offers a robust interpolation method based on considering time gaps and available values.

Value

A data frame with missing values being replaced linearly within short periods and with values to refer to.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

2. Wickham, H., Francois, R., Henry, L. & Muller, K. 2017. dplyr: A Grammar of Data Manipulation. 0.7.4 ed. http://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

3. Wickham, H., Francois, R., Henry, L. & Muller, K. 2019. dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr.

4. Zeileis, A. & Grothendieck, G. 2005. zoo: S3 infrastructure for regular and irregular time series. Journal of Statistical Software, 14(6):1-27.

5. Zeileis, A., Grothendieck, G. & Ryan, J.A. 2019. zoo: S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations). R package version 1.8-6. https://cran.r-project.org/web/packages/zoo/.

Examples

```
shorvalu(condextr(obsedele(data[1:250,c(1,4,17:19)],3,5,2,cores=2),
3,5,2,cores=2),3,5)
```

varidele *Delete variables containing too many missing values (NA)*

Description

Missing values often exist in real data. However, excessive missing values would lead to information distortion and inappropriate handling of them may end up coming to inaccurate conclusions about the data. Therefore, to control and improve the quality of original data that have already been produced by instruments in the first beginning, the data preprocessing method in this package introduces the deletion of variables to filter out and delete the variables with too many missing values.

Usage

```
varidele(data, start = NULL, end = NULL, fraction = 0.25)
```

Arguments

`data`	A data frame containing variables with excessive missing values. Its columns from `start` to end will be checked.
`start`	The column number of the first selected variable.
`end`	The column number of the last selected variable.
`fraction`	The proportion of missing values of variables. Default is 0.25.

Details

It operates only at the beginning and the end variables, so as to ensure that the remaining variables after deleting are continuous without breaks. The deletion of variables with excessive missing values is mainly based on the proportion of missing values of variables, excluding blank observations. The default proportion is 0.25, which can be adjusted according to practical needs.

Value

A data frame after deleting variables with too many missing values.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

References

1. Example data is from https://smear.avaa.csc.fi/download. It includes particle number concentrations in SMEAR I Varrio forest.

Examples

```
# Show the first 5 and last 5 rows and columns besides the date column.
varidele(data,5,65)[c(1:5,(nrow(data)-4):nrow(data)),c(1,5:9,35:39)]
# The first 22 variables and last 4 variables are deleted with the NA proportion of 0.25.


# Increasing the NA proportion can keep more variables.
# Proportions 0.4 and 0.5 have the same result for this example data.
# Namely 2 more variables in the beginning and 1 more variable in the end are kept.
varidele(data,5,65,.5)[c(1:5,(nrow(data)-4):nrow(data)),c(1,5:9,38:42)]

# Setting proportion as 0.6, then 3 more variables in the beginning are kept
# than that of proportions 0.4 and 0.5.
varidele(data,5,65,.6)[c(1:5,(nrow(data)-4):nrow(data)),c(1,5:9,41:45)]
```

`zerona` *Turn zeros to missing values*

Description

Zeros are suitable in logarithmic scale and should be removed for plots.

Usage

```
zerona(x)
```

Arguments

x	A dataframe, matrix, or vector containing zeros.

Value

A dataframe, matrix, or vector with zeros being turned into missing values.

Author(s)

Chun-Sheng Liang <liangchunsheng@lzu.edu.cn>

Examples

```
zerona(0:5)
zerona(cbind(a=0:5,b=c(6:10,0)))
```

Index