Package 'envoutliers'

October 13, 2022

Type Package

Title Methods for Identification of Outliers in Environmental Data

Version 1.1.0

Description

Three semi-parametric methods for detection of outliers in environmental data based on kernel regression and subsequent analysis of smoothing residuals. The first method (Campulova, Michalek, Mikuska and Bokal (2018) <DOI:10.1002/cem.2997>) analyzes the residuals using changepoint analysis, the second method is based on control charts (Campulova, Veselik and Michalek (2017) <DOI:10.1016/j.apr.2017.01.004>) and the third method (Holesovsky, Campulova and Michalek (2018) <DOI:10.1016/j.apr.2017.06.005>) analyzes the residuals using extreme value theory (Holesovsky, Campulova and Michalek (2018) <DOI:10.1016/j.apr.2017.06.005>).

License GPL-2

Encoding UTF-8

LazyData true

Imports MASS, car, changepoint, ecp, graphics, ismev, lokern, robustbase, stats

Suggests openair

RoxygenNote 6.1.1

NeedsCompilation no

Author Martina Campulova [cre],
Martina Campulova [aut],
Roman Campula [ctb]

Maintainer Martina Campulova <martina.campulova@mendelu.cz>

Repository CRAN

Date/Publication 2020-05-07 15:20:02 UTC

R topics documented:

---

boxcoxTransform   *Box-Cox transformation of data - Only intended for developer use*

Description

Performs Box-Cox power transformation of the data. The optimal value of power parameter is selected based on profile log-likelihoods. The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Usage

```
boxcoxTransform(x)
```

Arguments

x   a numeric vector of data values.

Details

This function computes the Box-Cox power transformation of the data. The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for a transformation of data to normality. The optimal value of a power parameter is estimated based on profile log-likelihoods calculated using boxcox function implemented in **MASS** package.

Value

A list is returned with elements:

lambda   a numeric value giving power parameter

x   a numeric vector of data values

x.transformed   a numeric vector of transformed data

References

Box G, Cox D (1964). An analysis of transformations. Journal of the Royal Statistical Society: Series B, 26, 211–234.

Venables WN, Ripley BD (2002). Modern Applied Statistics with S. New York, fourth edition. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

---

changepoint   *Changepoint analysis - Only intended for developer use*

Description

Performs changepoint analysis using PELT algorithm or A Nonparametric Approach for Multiple Changepoints. The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Usage

```
changepoint(x, cp.analysis.type, pen.value, alpha.edivisive)
```

Arguments

x   a numeric vector of data values.

cp.analysis.type

               a character string specifying the type of changepoint analysis

               Possible options are

- "parametric" to perform changepoint analysis using PELT algorithm (Killick et al., 2012)
- "nonparametric" to perform a nonparametric approach for multiple changepoins (Matteson and James, 2014)

pen.value   A character string giving the formula for manual penalty used in PELT algorithm. Only required for cp.analysis.type = "parametric".

alpha.edivisive

               a numeric value giving the moment index used for determining the distance between and within segments in the nonparametric changepoint model.

Details

This function performs changepoint analysis using parametric or nonparametric approach. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for partitioning smoothing residuals into homogeneous segments.

Value

A list is returned with elements:

x   a numeric vector of data values

cp.segmet   an estimated integer membership vector for individual segments

References

Killick R, Fearnhead P, Eckley IA (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590–1598.

Matteson D, James N (2014). A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data. Journal of the American Statistical Association, 109(505), 334–345.

Nicholas A. James, David S. Matteson (2014). ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data. Journal of Statistical Software, 62(7), 1-25, URL "http://www.jstatsoft.org/v62/i07/".

Killick R, Haynes K, Eckley IA (2016). changepoint: An R package for changepoint analysis. R package version 2.2.2, <URL: https://CRAN.R-project.org/package=changepoint>.

---

changepoint.plot	*Changepoint outlier detection plot - Only intended for developer use*

Description

Plot of results obtained using function `KRDetect.outliers.changepoint` for identification of outliers using changepoint analysis. The function is called by `plot.KRDetect` and is not intended for use by regular users of the package.

Usage

```
changepoint.plot(x, show.segments, ...)
```

Arguments

x	a list obtained as an output of function `KRDetect.outliers.changepoint` for identification of outliers using changepoint analysis.
show.segments	a logical variable specifying if vertical lines representing individual segments are plotted.
...	further arguments to be passed to the `plot` function.

Details

This function plots the results obtained using function `KRDetect.outliers.changepoint` identificating outliers using changepoint analysis based method. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for plotting results obtained using functions implemented in package **envoutliers**.

---

chebyshev.inequality.detect

Chebyshev inequality based identification of outliers on segments - Only intended for developer use

---

Description

Identification of outlier data values on individual homogeneous segments using Chebyshev inequality. The function is called by `KRDetect.outliers.changepoint` and is not intended for use by regular users of the package.

Usage

```
chebyshev.inequality.detect(x, cp.segment, L.default)
```

Arguments

x	a numeric vector of data.
cp.segment	an integer membership vector for individual segments.
L.default	a numeric value of L parameter determining the criterion for outlier detection: the limits for outlier observations on individual segments are set as $+/-L*$ $sample\,standard\,deviation\,of\,data\,on\,the\,corresponding\,segment$ If L.default = NULL, its value on individual segments is estimated using Algorithm A1 (Campulova et al., 2018).

Details

This function detects outlier observations on individual segments using Chebyshev inequality. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for identification of outlier residuals.

Value

A list is returned with elements:

L	a numeric vector of L parameters used for outlier identification on individual segments
outlier	a logical vector specifing the identified outliers, TRUE means that corresponding data value from vector x is detected as outlier

References

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

---

control.limits.R	*Limits for control chart* R *- Only intended for developer use*

Description

Estimation of limits of control chart *R*. The function is called by KRDetect.outliers.controlchart and is not intended for use by regular users of the package.

Usage

```
control.limits.R(x, group.size, L)
```

Arguments

x	a numeric vector of data values.
group.size	a positive integer giving the number of observations in individual segments used for computation of control chart limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis.
L	a positive numeric value giving parameter L specifying the width of control limits.

Details

This function computes parameters based on which control chart *R* can be constructed. The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for identification limits based on control chart *R*.

Value

A list is returned with elements:

x	a numeric vector of data
range.est	a numeric value giving an estimate of range parameter
groups.count	a numeric value giving a number of segments used for estimating parameters of control chart
groups.range	a numeric vector giving sample ranges in individual segments used for estimating parameters of control chart
LCL	a numeric value giving lower control limit of control chart *R*
UCL	a numeric value giving upper control limit of control chart *R*

References

Shewhart W (1931). Quality control chart. Bell System Technical Journal, 5, 593–603.

SAS/QC User's Guide, Version 8, 1999. SAS Institute, Cary, N.C.

Wild C, Seber G (2000). Chance encounters: A first course in data analysis and inference. New York: John Wiley.

---

control.limits.s	*Limits for control chart* s *- Only intended for developer use*

Description

Estimation of limits of control chart *s*. The function is called by KRDetect.outliers.controlchart and is not intended for use by regular users of the package.

Usage

```
control.limits.s(x, group.size, L)
```

Arguments

`x`	a numeric vector of data values.
`group.size`	a positive integer giving the number of observations in individual segments used for computation of control chart limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis.
`L`	a positive numeric value giving parameter `L` specifying the width of control limits.

Details

This function computes parameters based on which control chart *s* can be constructed. The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for identification limits based on control chart *s*.

Value

A list is returned with elements:

`x`	a numeric vector of data
`sd.est`	a numeric value giving an estimate of standard deviation parameter
`groups.count`	a numeric value giving a number of segments used for estimating parameters of control chart
`groups.sd`	a numeric vector giving sample standard deviations in individual segments used for estimating parameters of control chart
`LCL`	a numeric value giving lower control limit of control chart *s*
`UCL`	a numeric value giving upper control limit of control chart *s*

References

Shewhart W (1931). Quality control chart. Bell System Technical Journal, 5, 593–603.

SAS/QC User's Guide, Version 8, 1999. SAS Institute, Cary, N.C.

Wild C, Seber G (2000). Chance encounters: A first course in data analysis and inference. New York: John Wiley.

---

control.limits.x	*Limits for control chart* x *- Only intended for developer use*

Description

Estimation of limits of control chart *x*. The function is called by KRDetect.outliers.controlchart and is not intended for use by regular users of the package.

Usage

```
control.limits.x(x, method = "range", group.size, L)
```

Arguments

x	a numeric vector of data values.
method	a character string specifying the preferred estimate of standard deviation parameter.

Possible options are

- "range" for estimation based on sample ranges
- "sd" for estimation based on sample standard deviations

group.size	a positive integer giving the number of observations in individual segments used for computation of control chart limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis.
L	a positive numeric value giving parameter L specifying the width of control limits.

Details

This function computes parameters based on which control chart *x* can be constructed. The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for identification limits based on control chart *x*.

Value

A list is returned with elements:

x	a numeric vector of data
mean	a numeric value giving sample mean of vector x
groups.count	a numeric value giving a number of segments used for estimating parameters of control chart
groups.mean	a numeric vector giving sample means in individual segments used for estimating parameters of control chart
LCL	numeric value giving lower control limit of control chart *x*
UCL	numeric value giving upper control limit of control chart *s*

References

Shewhart W (1931). Quality control chart. Bell System Technical Journal, 5, 593–603.

SAS/QC User's Guide, Version 8, 1999. SAS Institute, Cary, N.C.

Wild C, Seber G (2000). Chance encounters: A first course in data analysis and inference. New York: John Wiley.

---

controlchart.plot   *Control chart outliers detection plot - Only intended for developer use*

Description

Plot of results obtained using function `KRDetect.outliers.controlchart` for identification of outliers using control charts. The function is called by `plot.KRDetect` and is not intended for use by regular users of the package.

Usage

```
controlchart.plot(x, plot.type = "all", ...)
```

Arguments

x	a list obtained as an output of function `KRDetect.outliers.controlchart` for identification of outliers using control charts.
plot.type	a type of plot with outliers displayed.
	Possible options are

- `"all"` to show outliers detected using control chart *x*, *R* and *s*
- `"x"` to show outliers detected using control chart *x*
- `"R"` to show outliers detected using control chart *R*
- `"s"` to show outliers detected using control chart *s*

...	further arguments to be passed to the `plot` function.

Details

This function plots the results obtained using function `KRDetect.outliers.controlchart` identificating outliers using control charts. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for plotting results obtained using functions implemented in package **envoutliers**.

EV.plot   *Extreme value outlier detection plot - Only intended for developer use*

Description

Plot of results obtained using function `KRDetect.outliers.EV` for identification of outliers using extreme value theory. The function is called by `plot.KRDetect` and is not intended for use by regular users of the package.

Usage

```
EV.plot(x, plot.type = "all", ...)
```

Arguments

x	a list obtained as an output of function `KRDetect.outliers.EV` for identification of outliers using extreme value theory.
plot.type	a type of plot with outliers displayed.
	Possible options are
	• `"all"` to show outliers with both extremely low and high value
	• `"min"` to show outliers with extremely low value
	• `"max"` to show outliers with extremely high value
...	further arguments to be passed to the `plot` function.

Details

This function plots the results obtained using function `KRDetect.outliers.EV` identificating outliers using changepoint analysis based method. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for plotting results obtained using functions implemented in package **envoutliers**.

---

extremal.index.censored

Extremal index estimation (Holesovsky and Fusek, 2020) - Only intended for developer use

---

Description

Estimation of an extremal index using the censored estimator suggested in (Holesovsky and Fusek, 2020). The function is called by `KRDetect.outliers.EV` and is not intended for use by regular users of the package.

Usage

```
extremal.index.censored(x, u, D)
```

Arguments

x	a numeric vector of observations.
u	a numeric value giving threshold.
D	a nonnegative integer giving the value of D parameter (Holesovsky and Fusek, 2020)

Details

This function computes the censored estimate of extremal index suggested in (Holesovsky and Fusek, 2020). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within `KRDetect.outliers.EV`.

Value

a numeric value of an extremal index estimate

References

Holesovsky, J, Fusek, M (2020). Estimation of the Extremal Index Using Censored Distributions. Extremes, DOI: 10.1007/s10687-020-00374-3.

---

extremal.index.gomes   *Extremal index estimation (Gomes, 1993) - Only intended for developer use*

Description

Estimation of an extremal index using the block maxima approach suggested by (Gomes, 1993). The function is called by KRDetect.outliers.EV and is not intended for use by regular users of the package.

Usage

```
extremal.index.gomes(x, block.length)
```

Arguments

x   a numeric vector of observations.

block.length   a numeric value giving the length of blocks.

Details

This function computes the estimate of extremal index suggested by (Gomes, 1993). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within KRDetect.outliers.EV.

Value

A numeric value of an extremal index estimate

References

Gomes M (1993). On the estimation of parameter of rare events in environmental time series. In Statistics for the Environment, volume 2 of Water Related Issues, pp. 225-241. Wiley.

Heffernan JE, Stephenson AG (2016). ismev: An Introduction to Statistical Modeling of Extreme Values. R package version 1.41, URL http://CRAN.R-project.org/package=ismev.

---

extremal.index.intervals

Extremal index estimation (Ferro and Segers, 2003) - Only intended for developer use

---

Description

Estimation of an extremal index using the Intervals estimator suggested in (Ferro and Segers, 2003). The function is called by `KRDetect.outliers.EV` and is not intended for use by regular users of the package.

Usage

```
extremal.index.intervals(x, u)
```

Arguments

x	a numeric vector of observations.
u	a numeric value giving threshold

Details

This function computes the estimate of extremal index suggested in (Ferro and Segers, 2003). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within `KRDetect.outliers.EV`.

Value

a numeric value of an extremal index estimate

References

Ferro, CAT, Segers, J (2003). Inference for Cluster of Extreme Values. Journal of Royal Statistical Society, Series B, 65(2), 545-556.

---

extremal.index.Kgaps   *Extremal index estimation (Suveges and Davison, 2010) - Only intended for developer use*

Description

Estimation of an extremal index using the K-gaps estimator suggested in (Suveges and Davison, 2010). The function is called by `KRDetect.outliers.EV` and is not intended for use by regular users of the package.

Usage

```
extremal.index.Kgaps(x, u, K)
```

Arguments

x	a numeric vector of observations.
u	a numeric value giving threshold.
K	a nonnegative integer giving the value of K parameter (Suveges and Davison, 2010).

Details

This function computes the K-gaps estimate of extremal index suggested in (Suveges and Davison, 2010). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within KRDetect.outliers.EV.

Value

a numeric value of an extremal index estimate

References

Suveges, M, Davison, AC (2010). Model Misspecification in Peaks Over Threshold Analysis. The Annals of Applied Statistics, 4(1), 203-221.

---

extremal.index.runs *Extremal index estimation (Smith and Weissman, 1994) - Only intended for developer use*

Description

Estimation of an extremal index using the runs estimator suggested in (Smith and Weissman, 1994). The function is called by KRDetect.outliers.EV and is not intended for use by regular users of the package.

Usage

```
extremal.index.runs(x, u, r)
```

Arguments

x	a numeric vector of observations.
u	a numeric value giving threshold.
r	a positive integer giving the value of runs parameter (Smith and Weissman, 1994).

Details

This function computes the runs estimate of extremal index suggested in (Smith and Weissman, 1994). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within `KRDetect.outliers.EV`.

Value

a numeric value of an extremal index estimate

References

Smith, RL, Weissman, I (1994). Estimating the Extremal Index. Journal of the Royal Statistical Society, Series B, 56, 515-529.

---

extremal.index.sliding.blocks

Extremal index estimation (Northrop, 2015) - Only intended for developer use

---

Description

Estimation of an extremal index using the sliding blocks estimator suggested in (Northrop, 2015). The function is called by `KRDetect.outliers.EV` and is not intended for use by regular users of the package.

Usage

```
extremal.index.sliding.blocks(x, b = round(sqrt(length(x))))
```

Arguments

x	a numeric vector of observations.
b	a numeric value giving the length of blocks. Default is b = round(sqrt(n)).

Details

This function computes the sliding blocks estimate of extremal index suggested in (Northrop, 2015). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within `KRDetect.outliers.EV`.

Value

a numeric value of an extremal index estimate

References

Northrop, PJ (2015). An Efficient Semiparametric Maxima Estimator of the Extremal Index. Extremes, 18, 585-603.

find.alpha	*Parameter* alpha *for Quantiles of normal distribution based outlier detection - Only intended for developer use*

Description

Finds the value of parameter *alpha* defining the criterion for outlier identification using quantiles of normal distribution. The parameter is found using Modified algorithm A1 (Campulova et al., 2018.) The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Usage

```
find.alpha(x, alpha.start = 0.05, eps = NULL)
```

Arguments

x	a numeric vector of data values.
alpha.start	a numeric value giving the largest reasonable value of parameter *alpha*.
eps	a numeric value of the epsilon parameter. If eps = NULL, the value is calculated as recommended in Modified Algorithm A1 (Campulova et al., 2018).

Details

This function finds the value of parameter *alpha* defining the criterion for outlier identification using quantiles of normal distribution. The algorithm is based on Modified Algorithm A1 described in (Campulova et al., 2018). Nonoutliers are characterised as a homogeneous set of data randomly distributed around zero value. The differences between the data correspond to random fluctuations in the measurements. The algorithm finds the value of the parameter by scanning possible values of *alpha* and investigating differences of the corresponding nonoutliers. The idea is to choose alpha corresponding to the maximum change of the maximum difference found among the ordered nonoutlier data. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for finding outlier residuals based on quantiles of normal distribution.

Value

A numeric value giving the parameter *alpha*

References

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

---

find.L   *Parameter L for Chebyshev inequality based outlier detection - Only*
                                intended for developer use

Description

Finds the value of parameter L defining the criterion for outlier identification using Chebyshev inequality. The parameter is found using Algorithm A1 (Campulova et al., 2018). The function is called by `KRDetect.outliers.changepoint` and is not intended for use by regular users of the package.

Usage

```
find.L(x, L.start = 2.5, eps = NULL)
```

Arguments

x   a numeric vector of data values.

L.start   a numeric value giving the smallest reasonable value of parameter L.

eps   A numeric value of the *epsilon* parameter. If eps = NULL, the value is calculated
                as recommended in Modified Algorithm A1 (Campulova et al., 2018).

Details

This function finds the value of parameter L defining the criterion for outlier identification using Chebyshev inequality. The algorithm is based on Algorithm A1 described in (Campulova et al., 2018). Nonoutliers are characterised as a homogeneous set of data randomly distributed around zero value. The differences between the data correspond to random fluctuations in the measurements. The algorithm finds the value of the parameter by scanning possible values of L and investigating differences of the corresponding nonoutliers. The idea is to choose L corresponding to the maximum change of the maximum difference found among the ordered nonoutlier data. The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for finding outlier residuals based on Chebyshev inequality.

Value

A numeric value giving the parameter L

References

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

---

get.norm   *Table of Control Charts Constants - Only intended for developer use*

Description

Creation of Table of Control Chart Constants. The function is called by control.limits.x, control.limits.R and control.limits.s. This function is not intended for use by regular users of the package.

Usage

```
get.norm()
```

Details

This function creates a table with columns giving constants for computation limits of control charts. The function is exported for developer use only. It does not have any input parameters and does not perform any checks on inputs since it is only a convenience function for computing control chart limits.

Value

data.frame whose columns are numeric vectors giving constants for control charts limits computation

References

JOGLEKAR, Anand M. Statistical methods for six sigma: in R&D and manufacturing. Hoboken, NJ: Wiley-Interscience. ISBN sbn0-471-20342-4.

---

grubbs.detect   *Grubbs test based identification of outliers on segments - Only intended for developer use*

Description

Identification of outlier data values on individual homogeneous segments using Grubbs test. The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Usage

```
grubbs.detect(x, cp.segment)
```

Arguments

x	a numeric vector of data.
cp.segment	an integer membership vector for individual segments.

Details

This function detects outlier observations on individual segments using Grubbs test. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for identification of outlier residuals.

Value

A logical vector specifing the identified outliers, TRUE means that corresponding data value from vector x is detected as outlier.

References

Grubbs F (1950). Sample criteria for testing outlying observations. The Annals of Mathematical Statistics, 21(1), 27-58.

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

---

grubbs.test	*Outlier detection using Grubbs test - Only intended for developer use*

Description

Sequential identification of outliers using Grubbs' test. The algorithm first considers the data value with the highest absolute value. If the null hypothesis that such a value is not an outlier is rejected, the considered value is detected as an outlier and excluded from further analysis. Subsequently, a value with the second-highest absolute value is considered, and its quality is again evaluated using the Grubbs test. This procedure is repeated until no outlier is detected. The function is called by KRDetect.outliers.changepoint and grubbs.detect. The function is not intended for use by regular users of the package.

Usage

```
grubbs.test(x, alpha = 0.05)
```

Arguments

x	a numeric vector of data values.
alpha	a numeric value giving test significance level.

Details

This function sequentially identifies outlier data using Grubbs test. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for identification of outlier residuals using Grubbs test.

Value

A list is returned with elements:

result   A table containing information about identified outliers. The number of rows of the table corresponds to the number of identified outliers. The table has follow-ing columns: `index` - a numeric value giving the index of detected outlier in the original data, `value` - a numeric value of the identified outlier, `test.value` - a numeric value of the test statistics, `critical.value` - a numeric value giving the test statistics, `p.value` - a numeric value giving *p.value* of the test

outliers.exists
                    A logical value. `TRUE` means that at least one outlier was detected.

References

Grubbs F (1950). Sample criteria for testing outlying observations. The Annals of Mathematical Statistics, 21(1), 27-58.

KRDetect.outliers.changepoint
Identification of outliers using changepoint analysis

Description

Identification of outliers in environmental data using method based on kernel smoothing, change-point analysis of smoothing residuals and subsequent analysis of residuals on homogeneous seg-ments (Campulova et al., 2018).

Usage

```
KRDetect.outliers.changepoint(x, perform.smoothing = TRUE,
  perform.cp.analysis = TRUE, bandwidth.type = "local",
  bandwidth.value = NULL, kernel.order = 2,
  cp.analysis.type = "parametric", pen.value = "5*log(n)",
  alpha.edivisive = 0.3, min.segment.length = 30,
  segment.length.for.merge = 15, method = "auto",
  prefer.grubbs = TRUE, alpha.default = NULL, L.default = NULL)
```

Arguments

x	data values. Supported data types

- a numeric vector
- a time series object `ts`
- a time series object `xts`
- a time series object `zoo`

perform.smoothing	
	a logical value specifying if data smoothing is performed. If `TRUE` (default), data are smoothed.
perform.cp.analysis	
	a logical value specifying if changepoint analysis is performed. If `TRUE` (default), smoothing residuals are partitioned into homogeneous segments.
bandwidth.type	a character string specifying the type of bandwidth.
	Possible options are

- `"local"` (default) to use local bandwidth
- `"global"` to use global bandwidth

bandwidth.value	
	a local bandwidth array (for `bandwidth.type = "local"`) or global bandwidth value (for `bandwidth.type = "global"`) for kernel regression estimation. If `bandwidth.type = "NULL"` (default) a data-adaptive local plug-in (Herrmann, 1997) (for `bandwidth.type = "local"`) or data-adaptive global plug-in (Gasser et al., 1991) (for `bandwidth.type = "global"`) bandwidth is used instead.
kernel.order	a nonnegative integer giving the order of the optimal kernel (Gasser et al., 1985) used for smoothing.
	Possible options are

- `kernel.order = 2` (default)
- `kernel.order = 4`

cp.analysis.type	
	a character string specifying the type of changepoint analysis.
	Possible options are

- `"parametric"` (default) to perform changepoint analysis using PELT algorithm (Killick et al., 2012)
- `"nonparametric"` to perform a nonparametric approach for multiple changepoins (Matteson and James, 2014)

pen.value	a character string giving the formula for manual penalty used in PELT algorithm. Only required for `cp.analysis.type = "parametric"`. Default is `pen.value = "5*log(n)"`.
alpha.edivisive	
	a numeric value giving the moment index used for determining the distance between and within segments in nonparametric changepoint model. Default is `alpha.edivisive = 0.3`.
min.segment.length	
	a numeric value giving minimal required number of observations on segments from changepoint analysis. If a segment contains less than `min.segment.length`

observations and the variances of data on the segment and the previous one are supposed to be equal (based on Levene´s test (Fox, 2016) for homogeneity of variances), the segment is merged with previous one. Analogous, the first segment can be merged with the second one. Default is `min.segment.length = 30`.

segment.length.for.merge

a numeric value giving minimal required number of observations on segments for performing the homogeneity test within changepoint split control. A segment with less data than `segment.length.for.merge` is merged with the previous one without testing the homogeneity of variances (the first segment is merged with the second one). Default is `min.segment.length.for.merge = 15`.

method   a character string specifying the method for identification of outlier residuals. Possible options are

- "auto" (default) for automatic selection based on the structure of the residuals
- "grubbs.test" for Grubbs test
- "normal.distribution" for quantiles of normal distribution
- "chebyshev.inequality" for chebyshev inequality

prefer.grubbs   a logical variable specyfing if Grubbs test for identification of outlier residuals is preferred to quantiles of normal distribution. `TRUE` (default) means that Grubbs test is preferred. Only required for `method = "auto"`.

alpha.default   a numeric value from interval (0,1) of *alpha* parameter determining the criterion for (residual) outlier detection: the limits for outlier residuals on individual segments are set as $+/-(alpha/2 - quantile of normal distribution with parameters corresponding to re$ $(sample standard deviation of residuals on corresponding segment)$. If `alpha.default = NULL` (default), its value on individual segments is estimated using Modified Algorithm A1 (Campulova et al., 2018).

L.default   a numeric value of *L* parameter determining the criterion for outlier (residual) detection: the limits for outlier residuals on individual segments are set as $+/- L * sample standard deviation of residuals on corresponding segment$. If `L.default = NULL` (default), its value on individual segments is estimated using Algorithm A1 (Campulova et al., 2018).

Details

This function identifies outliers in time series using procedure based on kernel smoothing, changepoint analysis of smoothing residuals and subsequent analysis of residuals on homogeneous segments (Campulova et al., 2018). Three different approaches (Grubbs test, quantiles of normal distribution, Chebyshev inequality), that can be selected automatically based on data structure or specified by the user, can be used to detect outlier residuals. Crucial for the method is the choice of parameters alpha and *L* for quantiles of normal distribution and Chebyshev inequality approach, that define the criterion for outlier detection. These values can be specified by the user or estimated automatically using data driven algorithms (Campulova et al., 2018).

Value

A "KRDetect" object which contains a list with elements:

`method.type`	a character string giving the type of method used for outlier idetification
`x`	a numeric vector of observations
`index`	a numeric vector of index design points assigned to individual observations
`smoothed`	a numeric vector of estimates of the kernel regression function (smoothed data)
`changepoints`	an integer membership vector for individual segments
`normality.results`	
	a data.frame of normality results of residuals on individual segments
`detection.method`	
	a character string giving the type of method used for identification of outlier residuals
`alpha`	a numeric vector of alpha parameters used for outlier identification on individual segments
`L`	a numeric vector of L parameters used for outlier identification on individual segments
`outlier`	a logical vector specyfing the identified outliers, `TRUE` means that corresponding observation from vector `x` is detected as outlier

References

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

Gasser T, Kneip A, Kohler W (1991). A flexible and fast method for automatic smoothing. Journal of the American Statistical Association, 86, 643–652.

Herrmann E (1997). Local bandwidth choice in kernel regression estimation. Journal of Computational and Graphical Statistics, 6(1), 35–54.

Eva Herrmann; Packaged for R and enhanced by Martin Maechler (2016). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-8. https://CRAN.R-project.org/package=lokern.

Killick R, Fearnhead P, Eckley IA (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500), 1590–1598.

Killick R, Haynes K, Eckley IA (2016). changepoint: An R package for changepoint analysis. R package version 2.2.2, <URL: https://CRAN.R-project.org/package=changepoint>.

Matteson D, James N (2014). A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data. Journal of the American Statistical Association, 109(505), 334–345.

Nicholas A. James, David S. Matteson (2014). ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data. Journal of Statistical Software, 62(7), 1-25, URL "http://www.jstatsoft.org/v62/i07/".

Brys G, Hubert M, Struyf A (2008). Goodness-of-fit tests based on a robust measure of skewness. Computational Statistics, 23(3), 429–442.

Todorov V, Filzmoser P (2009). An Object-Oriented Framework for Robust Multivariate Analysis. Journal of Statistical Software, 32(3), 1-47. URL http://www.jstatsoft.org/v32/i03/.

Box G, Cox D (1964). An analysis of transformations. Journal of the Royal Statistical Society: Series B, 26, 211–234.

Venables WN, Ripley BD (2002). Modern Applied Statistics with S. New York, fourth edition. ISBN 0-387-95457-0, URL http://www.stats.ox.ac.uk/pub/MASS4.

Grubbs F (1950). Sample criteria for testing outlying observations. The Annals of Mathematical Statistics, 21(1), 27-58.

Fox J (2016). Applied regression analysis and generalized linear models. 3 edition. Los Angeles: SAGE. ISBN 9781452205663.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
result = KRDetect.outliers.changepoint(x)
summary(result)
plot(result)
plot(result, show.segments = FALSE)
```

KRDetect.outliers.controlchart

Identification of outliers using control charts

Description

Identification of outliers in environmental data using two-step method based on kernel smoothing and control charts (Campulova et al., 2017). The outliers are identified as observations corresponding to segments of smoothing residuals exceeding control charts limits.

Usage

```
KRDetect.outliers.controlchart(x, perform.smoothing = TRUE,
  bandwidth.type = "local", bandwidth.value = NULL, kernel.order = 2,
  method = "range", group.size.x = 3, group.size.R = 3,
  group.size.s = 3, L.x = 3, L.R = 3, L.s = 3)
```

Arguments

x   data values. Supported data types

- a numeric vector
- a time series object `ts`
- a time series object `xts`
- a time series object `zoo`

perform.smoothing

a logical value specifying if data smoothing is performed. If `TRUE` (default), data are smoothed.

bandwidth.type   a character string specifying the type of bandwidth.

Possible options are

- "local" (default) to use local bandwidth
- "global" to use global bandwidth

bandwidth.value

a local bandwidth array (for bandwidth.type = "local") or global bandwidth value (for bandwidth.type = "global") for kernel regression estimation. If bandwidth.type = "NULL" (default) a data-adaptive local plug-in (Herrmann, 1997) (for bandwidth.type = "local") or data-adaptive global plug-in (Gasser et al., 1991) (for bandwidth.type = "global") bandwidth is used instead.

kernel.order

a nonnegative integer giving the order of the optimal kernel (Gasser et al., 1985) used for smoothing.

Possible options are

- kernel.order = 2 (default)
- kernel.order = 4

method

a character string specifying the preferred estimate of standard deviation parameter.

Possible options are

- "range" (default) for estimation based on sample ranges
- "sd" for estimation based on sample standard deviations

group.size.x

a positive integer giving the number of observations in individual segments used for computation of *x* chart control limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis. Default is group.size.x = 3.

group.size.R

a positive integer giving the number of observations in individual segments used for computation of *R* chart control limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis. Default is group.size.R = 3.

group.size.s

a positive integer giving the number of observations in individual segments used for computation of *s* chart control limits. If the data can not be equidistantly divided, the first extra values will be excluded from the analysis. Default is group.size.s = 3.

L.x

a positive numeric value giving parameter L specifying the width of *x* chart control limits. Default is L.x = 3.

L.R

a positive numeric value giving parameter L specifying the width of *R* chart control limits. Default is L.R = 3.

L.s

a positive numeric value giving parameter L specifying the width of *s* chart control limits. Default is L.s = 3.

Details

This function identifies outliers in environmental data using two-step procedure (Campulova et al., 2017). The procedure consists of kernel smoothing and subsequent identification of observations corresponding to segments of smoothing residuals exceeding control charts limits. This way the method does not identify individual outliers but segments of observations, where the outliers occur. The output of the method are three logical vectors specyfing the outliers identified based on each of the three control charts. Beside that logical vector specyfing the outliers identified based on at

least one type of control limits is returned. Crucial for the method is the choice of paramaters `L.x`, `L.R` and `L.s` specifying the width of control limits. Different values of the parameters determine different criteria for outlier detection. For more information see (Campulova et al., 2017).

Value

A `"KRDetect"` object which contains a list with elements:

`method.type`	a character string giving the type of method used for outlier idetification
`x`	a numeric vector of observations
`index`	a numeric vector of index design points assigned to individual observations
`smoothed`	a numeric vector of estimates of the kernel regression function (smoothed data)
`outlier.x`	a logical vector specyfing the identified outliers based on limits of control chart *x*, TRUE means that corresponding observation from vector x is detected as outlier
`outlier.R`	a logical vector specyfing the identified outliers based on limits of control chart *R*, TRUE means that corresponding observation from vector x is detected as outlier
`outlier.s`	a logical vector specyfing the identified outliers based on limits of control chart *s*, TRUE means that corresponding observation from vector x is detected as outlier
`outlier`	a logical vector specyfing the identified outliers based on at least one type of control limits. TRUE means that corresponding observation from vector x is detected as outlier
`LCL.x`	a numeric value giving lower control limit of control chart *x*
`UCL.x`	a numeric value giving upper control limit of control chart *x*
`LCL.s`	a numeric value giving lower control limit of control chart *s*
`UCL.s`	a numeric value giving upper control limit of control chart *s*
`LCL.R`	a numeric value giving lower control limit of control chart *R*
`UCL.R`	a numeric value giving upper control limit of control chart *R*

References

Campulova M, Veselik P, Michalek J (2017). Control chart and Six sigma based algorithms for identification of outliers in experimental data, with an application to particulate matter PM10. Atmospheric Pollution Research. Doi=10.1016/j.apr.2017.01.004.

Shewhart W (1931). Quality control chart. Bell System Technical Journal, 5, 593–603.

SAS/QC User's Guide, Version 8, 1999. SAS Institute, Cary, N.C.

Wild C, Seber G (2000). Chance encounters: A first course in data analysis and inference. New York: John Wiley.

Joglekar, Anand M. Statistical methods for six sigma: in R&D and manufacturing. Hoboken, NJ: Wiley-Interscience. ISBN sbn0-471-20342-4.

Gasser T, Kneip A, Kohler W (1991). A flexible and fast method for automatic smoothing. Journal of the American Statistical Association, 86, 643–652.

Herrmann E (1997). Local bandwidth choice in kernel regression estimation. Journal of Computational and Graphical Statistics, 6(1), 35–54.

Eva Herrmann; Packaged for R and enhanced by Martin Maechler (2016). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-8. https://CRAN.R-project.org/package=lokern

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
result = KRDetect.outliers.controlchart(x)
summary(result)
plot(result)
plot(result, plot.type = "x")
plot(result, plot.type = "R")
plot(result, plot.type = "s")
```

KRDetect.outliers.EV   *Identification of outliers using extreme value theory*

Description

Identification of outliers in environmental data using semiparametric method based on kernel smoothing and extreme value theory (Holesovsky et al., 2018). The outliers are identified as observations whose values are exceeded on average once a given period that is specified by the user.

Usage

```
KRDetect.outliers.EV(x, perform.smoothing = TRUE,
  bandwidth.type = "local", bandwidth.value = NULL, kernel.order = 2,
  gpd.fit.method = "mle", threshold.min = NULL, threshold.max = NULL,
  k.min = round(length(na.omit(x)) * 0.1),
  k.max = round(length(na.omit(x)) * 0.1), extremal.index.min = NULL,
  extremal.index.max = NULL, extremal.index.type = "block.maxima",
  block.length.min = round(sqrt(length(na.omit(x)))),
  block.length.max = round(sqrt(length(na.omit(x)))), D.min = NULL,
  D.max = NULL, K.min = NULL, K.max = NULL, r.min = NULL,
  r.max = NULL, return.period = 120)
```

Arguments

x   data values. Supported data types

- a numeric vector
- a time series object ts
- a time series object xts
- a time series object zoo

perform.smoothing

a logical value specifying if data smoothing is performed. If TRUE (default), data are smoothed.

bandwidth.type   a character string specifying the type of bandwidth.

Possible options are

- "local" (default) to use local bandwidth
- "global" to use global bandwidth

bandwidth.value

a local bandwidth array (for bandwidth.type = "local") or global bandwidth value (for bandwidth.type = "global") for kernel regression estimation. If bandwidth.type = "NULL" (default) a data-adaptive local plug-in (Herrmann, 1997) (for bandwidth.type = "local") or data-adaptive global plug-in (Gasser et al., 1991) (for bandwidth.type = "global") bandwidth is used instead.

kernel.order   a nonnegative integer giving the order of the optimal kernel (Gasser et al., 1985) used for smoothing.

Possible options are

- kernel.order = 2 (default)
- kernel.order = 4

gpd.fit.method   a character string specifying the method used for the estimate of the scale and shape parameters of GP distribution.

Possible options are

- "mle" (default) for maximum likelihood estimates (Coles, 2001)
- "moment" for moment estimates (de Haan and Ferreira2006)

threshold.min   a threshold value for residuals with low values, that is used to find the maximum likelihood estimates of shape and scale parameters of GP distribution and selected types of extremal index estimates (specifically: Intervals estimator (Ferro and Segers, 2003), censored estimator, (Holesovsky and Fusek, 2020), K-gaps estimator (Suveges and Davison, 2010), runs estimator (Smith and Weissman, 1994)). If threshold.min = NULL (default), threshold is estimated as 90% quantile of smoothing residuals.

threshold.max   a threshold value for residuals with high values, that is used to find the maximum likelihood estimates of shape and scale parameters of GP distribution and selected types of extremal index estimates (specifically: Intervals estimator (Ferro and Segers, 2003), censored estimator, (Holesovsky and Fusek, 2020), K-gaps estimator (Suveges and Davison, 2010), runs estimator (Smith and Weissman, 1994)). If threshold.max = NULL (default), threshold is estimated as 90% quantile of smoothing residuals.

k.min   a positive integer for residuals with low values giving the number of largest order statistics used to find the moment estimates (de Haan and Ferreira, 2006) of shape and scale parameters of GP distribution. Default is k.min = round(length(x) * 0.1).

k.max   a positive integer for residuals with high values giving the number of largest order statistics used to find the moment estimates (de Haan and Ferreira, 2006) of shape and scale parameters of GP distribution. Default is k.max = round(length(x) * 0.1).

extremal.index.min

a numeric value giving the extremal index for identification of outliers with extremely low value. If extremal.index.min = NULL (default), the extremal index is estimated using the method specified by the parameter extremal.index.type.

extremal.index.max

> a numeric value giving the extremal index for identification of outliers with extremely high value. If extremal.index.max = NULL (default), the extremal index is estimated using the method specified by the parameter extremal.index.type.

extremal.index.type

> a character string specifying the type of extremal index estimate.
>
> Possible options are
>
> - "block.maxima" (default) for block maxima estimator (Gomes, 1993).
> - "intervals" for intervals estimator (Ferro and Segers, 2003).
> - "censored" for censored estimator (Holesovsky and Fusek, 2020).
> - "Kgaps" for K-gaps estimator (Suveges and Davison, 2010).
> - "sliding.blocks" for sliding blocks estimator (Northrop, 2015).
> - "runs" for runs estimator (Smith and Weissman, 1994).

block.length.min

> a numeric value for residuals with low values giving the length of blocks for estimation of extremal index. Only required for extremal.index.type = "block.maxima" and extremal.index.type = "sliding.blocks". Default is block.length.min = round(sqrt(length(x))).

block.length.max

> a numeric value for residuals with high values giving the length of blocks for estimation of extremal index. Only required for extremal.index.type = "block.maxima" and extremal.index.type = "sliding.blocks". Default is block.length.max = round(sqrt(length(x))).

D.min

> a nonnegative integer for residuals with low values giving the value of D parameter used for censored extremal index estimate (Holesovsky and Fusek, 2020). Only required for extremal.index.type = "censored".

D.max

> a nonnegative integer for residuals with high values giving the value of D parameter used for censored extremal index estimate (Holesovsky and Fusek, 2020). Only required for extremal.index.type = "censored".

K.min

> a nonnegative integer for residuals with low values giving the value of K parameter used for K-gaps extremal index estimate (Suveges and Davison, 2010). Only required for extremal.index.type = "Kgaps".

K.max

> a nonnegative integer for residuals with high values giving the value of K parameter used for K-gaps extremal index estimate (Suveges and Davison, 2010). Only required for extremal.index.type = "Kgaps".

r.min

> a positive integer for residuals with low values giving the value of runs parameter of runs extremal index estimate (Smith and Weissman, 1994). Only required for extremal.index.type = "runs".

r.max

> a positive integer for residuals with high values giving the value of runs parameter of runs extremal index estimate (Smith and Weissman, 1994). Only required for extremal.index.type = "runs".

return.period

> a positive numeric value giving return period. Default is r = 120, which means that observations whose values are exceeded on average once every 120 observations are detected as outliers.

Details

This function identifies outliers in time series using two-step procedure (Holesovsky et al., 2018). The procedure consists of kernel smoothing and extreme value estimation of high threshold exceedances for smoothing residuals. Outliers with both extremely high and extremely low values are identified. Crucial for the method is the choice of return period - parameter defining the criterion for outliers detection. The outliers with extremely high values are detected as observations whose values are exceeded on average once a given return.period of observations. Analogous, the outliers with extremely low values are identified.

Value

A `"KRDetect"` object which contains a list with elements:

`method.type`	a character string giving the type of method used for outlier idetification
`x`	a numeric vector of observations
`index`	a numeric vector of index design points assigned to individual observations
`smoothed`	a numeric vector of estimates of the kernel regression function (smoothed data)
`GPD.fit.method`	the method used for the estimate of the scale and shape parameters of GP distribution
`extremal.index.type`	
	the type of extremal index estimate used for the identification of outliers
`sigma.min`	a numeric value giving scale parameter of Generalised Pareto distribution used for identification of outliers with extremely low value
`sigma.max`	a numeric value giving scale parameter of Generalised Pareto distribution used for identification of outliers with extremely high value
`xi.min`	a numeric value giving shape parameter of Generalised Pareto distribution used for identification of outliers with extremely low value
`xi.max`	a numeric value giving shape parameter of Generalised Pareto distribution used for identification of outliers with extremely high value
`lambda_u.min`	a numeric value giving relative frequency of the number of threshold value exceedances and identification of outliers with extremely low value. The value of the parameter is returned only for `gpd.fit.method = "mle"`.
`lambda_u.max`	a numeric value giving relative frequency of the number of threshold value exceedances and identification of outliers with extremely high value. The value of the parameter is returned only for `gpd.fit.method = "mle"`.
`extremal.index.min`	
	a numeric value giving extremal index used for identification of outliers with extremely low value
`extremal.index.max`	
	a numeric value giving extremal index used for identification of outliers with extremely high value
`threshold.min`	a numeric value giving threshold value used for identification of outliers with extremely low value.
`threshold.max`	a numeric value giving threshold value used for identification of outliers with extremely high value.

```
return.level.min
```
a numeric value giving return level used for identification of outliers with extremely low value

```
return.level.max
```
a numeric value giving return level used for identification of outliers with extremely high value

`outlier.min`   a logical vector specyfing the identified outliers with extremely low value. `TRUE` means that corresponding observation from vector x is detected as outlier

`outlier.max`   a logical vector specyfing the identified outliers with extremely high value. `TRUE` means that corresponding observation from vector x is detected as outlier

`outlier`   a logical vector specyfing the identified outliers with both extremely low and extremely high value. `TRUE` means that corresponding observation from vector x is detected as outlier

References

Holesovsky J, Campulova M, Michalek J (2018). Semiparametric Outlier Detection in Nonstationary Times Series: Case Study for Atmospheric Pollution in Brno, Czech Republic. Atmospheric Pollution Research, 9(1).

Theo Gasser, Alois Kneip & Walter Koehler (1991) A flexible and fast method for automatic smoothing. Journal of the American Statistical Association 86, 643-652. https://doi.org/10.2307/2290393

E. Herrmann (1997) Local bandwidth choice in kernel regression estimation. Journal of Graphical and Computational Statistics 6, 35-54.

Herrmann E, Maechler M (2013). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-5, URL http://CRAN.R-project.org/package=lokern.

Gasser, T, Muller, H-G, Mammitzsch, V (1985). Kernels for nonparametric curve estimation. Journal of the Royal Statistical Society, B Met., 47(2), 238-252.

Gomes M (1993). On the estimation of parameter of rare events in environmental time series. In Statistics for the Environment, volume 2 of Water Related Issues, pp. 225-241. Wiley.

Ferro, CAT, Segers, J (2003). Inference for Cluster of Extreme Values. Journal of Royal Statistical Society, Series B, 65(2), 545-556.

Holesovsky, J, Fusek, M (2020). Estimation of the Extremal Index Using Censored Distributions. Extremes, In Press.

Suveges, M, Davison, AC (2010). Model Misspecification in Peaks Over Threshold Analysis. The Annals of Applied Statistics, 4(1), 203-221.

Northrop, PJ (2015). An Efficient Semiparametric Maxima Estimator of the Extremal Index. Extremes, 18, 585-603.

Smith, RL, Weissman, I (1994). Estimating the Extremal Index. Journal of the Royal Statistical Society, Series B, 56, 515-529.

Heffernan JE, Stephenson AG (2016). ismev: An Introduction to Statistical Modeling of Extreme Values. R package version 1.41, URL http://CRAN.R-project.org/package=ismev.

Coles S (2001). An Introduction to Statistical Modeling of Extreme Values. 3 edition. London: Springer. ISBN 1-85233-459-2.

de Haan, L, Ferreira, A (2006). Extreme Value Theory: An Introduction. Springer.

Pickands J (1975). Statistical inference using extreme order statistics. The Annals of Statistics, 3(1), 119-131.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
result = KRDetect.outliers.EV(x)
summary(result)
plot(result)
plot(result, plot.type = "min")
plot(result, plot.type = "max")
```

KRDetect.outliers.plot

Outlier detection plot

Description

This function is deprecated. Use [plot.KRDetect](#) instead.

Usage

```
KRDetect.outliers.plot(x, all, segments)
```

Arguments

x	a list obtained as the output of function KRDetect.outliers.changepoint, KRDetect.outliers.controlchart or KRDetect.outliers.EV for identification of outliers.
all	a logical variable, in case of results obtained using function KRDetect.outliers.controlchart specifying if individual graphs for outliers detected using control chart x, R and s are plotted together with graph visualising outliers detected based on at least 1 control chart. If all = FALSE, only one graph visualising outliers detected based on at least 1 control chart is plotted. in case of results obtained using function KRDetect.outliers.EV specifying if individual graphs for outliers with extremely low and extremely high value are plotted together with graph visualising outliers with both extremely low and extremely high value. If all = FALSE, only one graph visualising outliers with both extremely low and extremely high value is plotted. Only required for results obtained using functions KRDetect.outliers.controlchart and KRDetect.outliers.EV. Default is all = TRUE.
segments	a logical variable specifying if vertical lines representing individual segments are plotted. Only required for results obtained using KRDetect.outliers.changepoint function. Default is segments = TRUE.

Details

Plot of results obrained using functions KRDetect.outliers.changepoint, KRDetect.outliers.controlchart
and KRDetect.outliers.EV for identification of outliers. The function graphically visualizes results
obtained using functions for outlier detection implemented in package envoutliers.

This function plots the results obtained using function KRDetect.outliers.changepoint, KRDetect.outliers.controlchart
or KRDetect.outliers.EV implemented in package envoutliers and identificating outliers.

---

mc.left   *Left medcouple (LMC) - Only intended for developer use*

Description

Calculates left medcouple (MLC). The function is called by `KRDetect.outliers.changepoint`
and is not intended for use by regular users of the package.

Usage

```
mc.left(x)
```

Arguments

x   a numeric vector of data values.

Details

This function computes left medcouple (LMC). The function is exported for developer use only. It
does not perform any checks on inputs since it is only a convenience function.

Value

A numeric value giving left medcouple

References

Brys G, Hubert M, Struyf A (2008). Goodness-of-fit tests based on a robust measure of skewness.
Computational Statistics, 23(3), 429–442.

Todorov V, Filzmoser P (2009). An Object-Oriented Framework for Robust Multivariate Analysis.
Journal of Statistical Software, 32(3), 1-47. URL http://www.jstatsoft.org/v32/i03/.

---

mc.right   *Right medcouple (RMC) - Only intended for developer use*

Description

Calculates right medcouple (RMC). The function is called by `KRDetect.outliers.changepoint` and is not intended for use by regular users of the package.

Usage

```
mc.right(x)
```

Arguments

x   a numeric vector of data values.

Details

This function computes right medcouple (RMC). The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function.

Value

A numeric value giving right medcouple

References

Brys G, Hubert M, Struyf A (2008). Goodness-of-fit tests based on a robust measure of skewness. Computational Statistics, 23(3), 429–442.

Todorov V, Filzmoser P (2009). An Object-Oriented Framework for Robust Multivariate Analysis. Journal of Statistical Software, 32(3), 1-47. URL http://www.jstatsoft.org/v32/i03/.

---

mc.test   *Robust medcouple MC-LR test - Only intended for developer use*

Description

Performs robust medcouple test to evaluate the fit of the data to normal distribution. The function is called by `KRDetect.outliers.changepoint` and is not intended for use by regular users of the package.

Usage

```
mc.test(x, alpha = 0.05)
```

Arguments

x	a numeric vector of data values.
alpha	numeric value giving test significance level.

Details

This function performs robust medcouple test based on the left and right medcouple (LMC and LRC). The function is exported for developer use only. It does not perform any checks on inputs since it is only a convenience function for robust testing of the normality.

Value

A list is returned with elements:

test.stat	a numeric value giving the value of test statistics
crit numeric	a vector of critical values defining rejection region of the test

References

Brys G, Hubert M, Struyf A (2008). Goodness-of-fit tests based on a robust measure of skewness. Computational Statistics, 23(3), 429–442.

Todorov V, Filzmoser P (2009). An Object-Oriented Framework for Robust Multivariate Analysis. Journal of Statistical Software, 32(3), 1-47. URL http://www.jstatsoft.org/v32/i03/.

Moment.gpd.fit	*Moment estimates of GP distribution parameters - Only intended for developer use*

Description

Moment estimates of shape and scale parameters of GP distribution using the approach presented in (de Haan and Ferreira, 2006). The function is called by KRDetect.outliers.EV and is not intended for use by regular users of the package.

Usage

```
Moment.gpd.fit(x, k = round(length(x) * 0.1))
```

Arguments

x	a numeric vector of observations.
k	a positive integer giving the number of top rank statistics (de Haan and Ferreira, 2006). Default is k = round(length(x) * 0.1).

Details

This function computes the moment estimates of shape and scale parameters of GP distribution (de Haan and Ferreira, 2006). The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within `KRDetect.outliers.EV`.

Value

a numeric vector giving the moment estimates for the scale and shape parameters, resp. a numeric vector giving the standard deviations for the scale and shape parameter estimates, resp.

References

de Haan, L, Ferreira, A (2006). Extreme Value Theory: An Introduction. Springer.

MRL.plot   *Mean residual life (MRL) plot*

Description

An empirical mean residual life plot (Coles, 2001), including confidence intervals, is produced based on maximum likelihood or moment estimates.

Usage

```
MRL.plot(x, umin = quantile(na.omit(x), probs = 0.8),
  umax = quantile(na.omit(x), probs = 0.95),
  kmin = round(length(na.omit(x)) * 0.05),
  kmax = round(length(na.omit(x)) * 0.2), nint = 100, conf = 0.95,
  est.method = "mle", u0 = NULL, k0 = NULL)
```

Arguments

x
: data values. Supported data types

  - a numeric vector
  - a time series object `ts`
  - a time series object `xts`
  - a time series object `zoo`

umin
: the minimum threshold at which the mean residual life function is calculated based on maximum likelihood estimates. Default is `umin = quantile(na.omit(x), probs = 0.8)`.

umax
: the maximum threshold at which the mean residual life function is calculated based on maximum likelihood estimates. Default is `umin = quantile(na.omit(x), probs = 0.95)`.

kmin
: the minimum number of largest order statistics for which the mean residual life function is calculated based on moment estimates. Default is `kmin = round(length(na.omit(x)) * 0.05)`.

kmax	the maximum number of largest order statistics for which the mean residual life function is calculated based on moment estimates. Default is kmax = round(length(na.omit(x)) * 0.2).
nint	the number of points at which the mean residual life function is calculated. Default is nint = 100.
conf	the confidence coefficient for the confidence intervals depicted in the plot. Default is conf = 0.95.
est.method	a character string specifying the type of estimates for the scale and shape parameters of GP distribution. Possible options are • "mle" (default) to use maximum likelihood estimates (Coles, 2001) • "moment" to use moment estimates (de Haan and Ferreira2006).
u0	a numeric value giving the threshold meant for a GP approximation of the threshold exceedances. Default is u0 = NULL.
k0	a numeric value giving the number (k0-1) of largest observations meant for a GP approximation. Default is k0 = NULL.

Details

The function constructs MRL plot (Coles, 2001) based on maximum likelihood or moment estimates for parameters of GP distribution. The MRL, i.e. the estimates of the mean excess, are expected to change linearly with threshold levels at which the GP model is appropriate. If u0 (or k0, respectively) is given, a GP mean-threshold dependency line is plotted in addition to the MRL plot (Coles, 2001; Eq. 4.9). Each of the lines provide the user an option to assess the suitability of u0 or k0 as a lower bound for the threshold exceedances (for u0) or the number of upper order statistics (for k0) to fit the GP distribution. In case est.method = "mle" and u0 takes a value, the theoretical GP mean is estimated by the MLE estimates of the GP parameters. For the case est.method = "moment" and k0 is given, the theoretical GP mean is estimated using the moment estimates. In case est.method = "moment" the value x(n-k) on the x-axis of MRL plot denotes the (k + 1)-th largest observation of the total number of n observations.

References

Theo Gasser, Alois Kneip & Walter Koehler (1991) A flexible and fast method for automatic smoothing. Journal of the American Statistical Association 86, 643-652. https://doi.org/10.2307/2290393

E. Herrmann (1997) Local bandwidth choice in kernel regression estimation. Journal of Graphical and Computational Statistics 6, 35-54.

Herrmann E, Maechler M (2013). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-5, URL http://CRAN.R-project.org/package=lokern.

Gasser, T, Muller, H-G, Mammitzsch, V (1985). Kernels for nonparametric curve estimation. Journal of the Royal Statistical Society, B Met., 47(2), 238-252.

Coles, S (2001). An Introduction to Statistical Modeling of Extreme Values. Springer-Verlag, London, U.K., 208pp.

de Haan, L, Ferreira, A (2006). Extreme Value Theory: An Introduction. Springer.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
res = smoothing(y = x)$residuals
MRL.plot(res)
```

---

normal.distr.quantiles.detect

Normal distribution based identification of outliers on segments - Only intended for developer use

---

Description

Identification of outlier data values on individual homogeneous segments using quantiles of normal distribution. The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Usage

```
normal.distr.quantiles.detect(x, cp.segment, alpha.default)
```

Arguments

x	a numeric vector of data.
cp.segment	an integer membership vector for individual segments.
alpha.default	a numeric value from interval (0,1) of alpha parameter determining the criterion for outlier detection: the limits for outlier observations on individual segments are set as *+/- (alpha/2-quantile of normal distribution with parameters corresponding to data on studied segment) * (sample standard deviation of data on corresponding segment)* If alpha.default = NULL, its value on individual segments is estimated using Modified Algorithm A1 (Campulova et al., 2018).

Details

This function detects outlier observations on individual segments using quantiles of normal distribution. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function for identification of outlier residuals.

Value

A list is returned with elements:

alpha	a numeric vector of alpha parameters used for outlier identification on individual segments
outlier	a logical vector specyfing the identified outliers, TRUE means that corresponding data value from vector x is detected as an outlier

References

Campulova M, Michalek J, Mikuska P, Bokal D (2018). Nonparametric algorithm for identification of outliers in environmental data. Journal of Chemometrics, 32, 453-463.

---

plot.KRDetect *Outlier detection plot*

Description

Plot of results obtained using functions KRDetect.outliers.changepoint, KRDetect.outliers.controlchart and KRDetect.outliers.EV for identification of outliers. The function graphically visualizes results obtained using functions for outlier detection implemented in package **envoutliers**.

Usage

```
## S3 method for class 'KRDetect'
plot(x, show.segments = TRUE, plot.type = "all",
  xlab = "index", ylab = "data values", ...)
```

Arguments

x	a KRDetect object obtained as an output of function KRDetect.outliers.changepoint, KRDetect.outliers.controlchart or KRDetect.outliers.EV for identification of outliers.
show.segments	a logical variable specifying if vertical lines representing individual segments are plotted. Only required for results obtained using KRDetect.outliers.changepoint function.
plot.type	a type of plot with outliers displayed.
	Possible options for KRDetect.outliers.controlchart are
	• "all" to show outliers detected using control chart x, R and s
	• "x" to show outliers detected using control chart x
	• "R" to show outliers detected using control chart R
	• "s" to show outliers detected using control chart s
	Possible options for KRDetect.outliers.EV are
	• "all" to show outliers with both extremely low and high value
	• "min" to show outliers with extremely low value
	• "max" to show outliers with extremely high value
xlab	a title for the x axis
ylab	a title for the y axis
...	further arguments to be passed to the plot function.

Details

This function plots the results obtained using function KRDetect.outliers.changepoint, KRDetect.outliers.controlchart or KRDetect.outliers.EV implemented in package **envoutliers** and identificating outliers.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
result = KRDetect.outliers.EV(x)
plot(result)
```

---

return.level.est *Return level estimation - Only intended for developer use*

Description

Estimation of return level for a given threshold value using Peaks Over Threshold model. The function is called by KRDetect.outliers.EV and is not intended for use by regular users of the package.

Usage

```
return.level.est(r, u, sigma_u, xi, lambda_u, theta)
```

Arguments

r	a return period.
u	a threshold value.
sigma_u	a scale parameter of Generalized Pareto distribution.
xi	a shape parameter of Generalized Pareto distribution.
lambda_u	a relative frequency of the number of threshold value exceedances.
theta	an extremal index.

Details

This function computes the estimate of return level for a given threshold value using Peaks Over Threshold model. The function is exported for developer use only. It does not perform any checks on inputs since it is only convenience function used within KRDetect.outliers.EV.

Value

A numeric value of return lever corresponding to return period r

References

Coles S (2001). An Introduction to Statistical Modeling of Extreme Values. 3 edition. London: Springer. ISBN 1-85233-459-2.

Pickands J (1975). Statistical inference using extreme order statistics. The Annals of Statistics, 3(1), 119-131.

segment.length.control

Segment length control - Only intended for developer use

Description

Control of a number of data values on individual segments. In case a number of data values on a segment is too small, the segment is (under the presumption of meeting certain conditions) merged with the previous one. The first segment can be merged with the previous one.

Usage

```
segment.length.control(index, x, cp.segment, min.segment.length,
  segment.length.for.merge)
```

Arguments

index   a numeric vector of design points.

x   a numeric vector of data.

cp.segment   an integer membership vector for individual segments.

min.segment.length

        a numeric value giving minimal required number of observations on segments from changepoint analysis. If a segment contains less than min.segment.length observations and the variances of data on the segment and the previous one are supposed to be equal (based on Levene´s test (Fox, 2016) for homogeneity of variances), the segment is merged with previous one. Analogous, the first segment can be merged with the second one.

segment.length.for.merge

        a numeric value giving giving minimal required number of observations on segments for performing the homogeneity test within changepoint split control. A segment with fewer data than segment.length.for.merge is merged with the previous one without testing the homogeneity of variances (the first segment is merged with the second one).

Details

Control of data splitting into segments. If a segment contains less than a given number of observations specified by the user and the variances of data on the segment and the previous one are equally based on the robust version of Levene's test, the segment is merged with previous one. Analogous, the first segment can be merged with the second one. The user can also specify a minimum length of a segment for performing the homogeneity test. A segment with fewer data than this minimal length is merged with the previous one without testing the homogeneity of variances. The function is called by KRDetect.outliers.changepoint and is not intended for use by regular users of the package.

Value

An integer membership vector for individual segments

References

Fox J (2016). Applied regression analysis and generalized linear models. 3 edition. Los Angeles: SAGE. ISBN 9781452205663.

---

smoothing   *Kernel regression smoothing*

Description

Nonparametric estimation of regression function using kernel regression with local or global data-adaptive plug-in bandwidth and optimal kernels.

Usage

```
smoothing(x = c(1:length(y)), y, bandwidth.type = "local",
  bandwidth.value = NULL, kernel.order = 2)
```

Arguments

x   data values. Supported data types

- a numeric vector
- a time series object `ts`
- a time series object `xts`
- a time series object `zoo`

y   a numeric vector of data values.

bandwidth.type   a character string specifying the type of bandwidth.

Possible options are

- `"local"` (default) to use local bandwidth
- `"global"` to use global bandwidth

bandwidth.value

a local bandwidth array (for `bandwidth.type = "local"`) or global bandwidth value (for `bandwidth.type = "global"`) for kernel regression estimation. If `bandwidth.type = "NULL"` (default), a data-adaptive local plug-in (Herrmann, 1997) (for `bandwidth.type = "local"`) or data-adaptive global plug-in (Gasser et al., 1991) (for `bandwidth.type = "global"`) bandwidth is used instead.

kernel.order   a nonnegative integer giving the order of the optimal kernel (Gasser et al., 1985) used for smoothing.

Possible options are

- `kernel.order = 2` (default)
- `kernel.order = 4`

Details

This function computes the estimate of kernel regression function using a local or global data-adaptive plug-in algorithm and optimal kernels (Gasser et al., 1985).

Value

A list is returned with elements:

data.smoothed   a numeric vector of estimates of the kernel regression function (smoothed data).

residuals   a numeric vector of smoothing residuals

References

Gasser T, Kneip A, Kohler W (1991). A flexible and fast method for automatic smoothing. Journal of the American Statistical Association, 86, 643-652.

Herrmann E (1997). Local bandwidth choice in kernel regression estimation. Journal of Computational and Graphical Statistics, 6(1), 35-54.

Gasser, T, Müller, H-G, Mammitzsch, V (1985). Kernels for nonparametric curve estimation. Journal of the Royal Statistical Society, B Met., 47(2), 238-252.

Eva Herrmann; Packaged for R and enhanced by Martin Maechler (2016). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-8. https://CRAN.R-project.org/package=lokern

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
smoothed = smoothing(y = x)
smoothed$data.smoothed
smoothed$residuals
```

---

stability.plot   *Stability plot*

Description

A stability plot for maximum likelihood or moment estimates of the GP parameters (Coles, 2001), including confidence intervals, at a range of thresholds or number of the largest observations.

Usage

```
stability.plot(x, umin = quantile(na.omit(x), probs = 0.8),
  umax = quantile(na.omit(x), probs = 0.95),
  kmin = round(length(na.omit(x)) * 0.05),
  kmax = round(length(na.omit(x)) * 0.2), nint = 100, conf = 0.95,
  est.method = "mle", u0 = NULL, k0 = NULL)
```

Arguments

x	data values. Supported data types

  - a numeric vector
  - a time series object `ts`
  - a time series object `xts`
  - a time series object `zoo`

umin	the minimum threshold at which the mean residual life function is calculated. Default is `umin = quantile(na.omit(x), probs = 0.8)`.
umax	the maximum threshold at which the mean residual life function is calculated. Default is `ummax = quantile(na.omit(x), probs = 0.95)`.
kmin	the minimum number of largest order statistics for which the mean residual life function is calculated based on moment estimates. Default is `kmin = round(length(na.omit(x)) * 0.05)`.
kmax	the maximum number of largest order statistics for which the mean residual life function is calculated based on moment estimates. Default is `kmax = round(length(na.omit(x)) * 0.2)`.
nint	the number of points at which the mean residual life function is calculated. Default is `nint = 100`.
conf	the confidence coefficient for the confidence intervals depicted in the plot. Default is `conf = 0.95`.
est.method	a character string specifying the type of estimates for the scale and shape parameters of GP distribution.

  Possible options are

  - `"mle"` (default) to use maximum likelihood estimates (Coles, 2001)
  - `"moment"` to use moment estimates (de Haan and Ferreira, 2006).

u0	a numeric value giving the threshold meant for a GP approximation of the threshold exceedances. Default is `u0 = NULL`.
k0	a numeric value giving the number (`k0-1`) of largest observations meant for a GP approximation. Default is `k0 = NULL`.

Details

The function estimates the GP parameters at a range of thresholds (in case `est.method = "mle"`) or a range of upper order statistics (in case of `est.method = "moment"`), and shows the sample paths of the estimates. The estimates of the shape or the scale parameter are expected to be constant or to change linearly, respectively, with threshold levels at which the GP model is appropriate. If `u0` (or `k0`, respectively) is given, a threshold-dependency lines for the particular parameters are plotted in addition. The lines provide the user an option to assess the suitability of `u0` or `k0` as a lower bound for the threshold exceedances (for `u0`) or the number of upper order statistics (for `k0`) to fit the GP distribution. In case `est.method = "mle"` and `u0` takes a value, the theoretical dependency lines for the parameters are evaluated on the basis of MLE estimates. For the case `est.method = "moment"` and `k0` is given, the dependency lines are estimated using the moment estimators. In case `est.method = "moment"` the value `x(n-k)` on the x-axis of MRL plot denotes the $(k + 1)$-th largest observation of the total number of n observations.

References

Theo Gasser, Alois Kneip & Walter Koehler (1991) A flexible and fast method for automatic smoothing. Journal of the American Statistical Association 86, 643-652. https://doi.org/10.2307/2290393

E. Herrmann (1997) Local bandwidth choice in kernel regression estimation. Journal of Graphical and Computational Statistics 6, 35-54.

Herrmann E, Maechler M (2013). lokern: Kernel Regression Smoothing with Local or Global Plug-in Bandwidth. R package version 1.1-5, URL http://CRAN.R-project.org/package=lokern.

Gasser, T, Muller, H-G, Mammitzsch, V (1985). Kernels for nonparametric curve estimation. Journal of the Royal Statistical Society, B Met., 47(2), 238-252.

Coles, S (2001). An Introduction to Statistical Modeling of Extreme Values. Springer-Verlag, London, U.K., 208pp.

de Haan, L, Ferreira, A (2006). Extreme Value Theory: An Introduction. Springer.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
res = smoothing(y = x)$residuals
stability.plot(res)
```

---

summary.KRDetect   *Summary of the outlier detection results*

Description

Summary of results obtained using functions KRDetect.outliers.changepoint, KRDetect.outliers.controlchart and KRDetect.outliers.EV for identification of outliers.

Usage

```
## S3 method for class 'KRDetect'
summary(object, ...)
```

Arguments

object	a KRDetect object obtained as an output of function KRDetect.outliers.changepoint, KRDetect.outliers.controlchart or KRDetect.outliers.EV for identification of outliers.
...	further arguments to be passed to the summary function.

Details

The function summarizes the results obtained using functions KRDetect.outliers.changepoint, KRDetect.outliers.controlchart and KRDetect.outliers.EV for identification of outliers.

Examples

```
data("mydata", package = "openair")
x = mydata$o3[format(mydata$date, "%m %Y") == "12 2002"]
result = KRDetect.outliers.changepoint(x)
summary(result)
result = KRDetect.outliers.controlchart(x)
summary(result)
result = KRDetect.outliers.EV(x)
summary(result)
```

Index