

# Package ‘anomaly’

September 19, 2019

**Type** Package

**Title** Detecting Anomalies in Data

**Version** 2.0.2

**Date** 2019-09-19

**Description** Implements Collective And Point Anomaly (CAPA) <arXiv:1806.01947>, Multi-Variate Collective And Point Anomaly (MVCAPA) <arXiv:1909.01691>, and Proportion Adaptive Segment Selection (PASS) <doi:10.1093/biomet/ass059> methods for the detection of anomalies in time series data.

**License** GPL

**Imports** dplyr,rlang,methods,assertive,Rdpack,ggplot2,reshape2,Rcpp (>= 0.12.18),robust

**LinkingTo** Rcpp,BH

**Suggests** magrittr

**Depends** R (>= 2.10)

**NeedsCompilation** yes

**RoxygenNote** 6.1.1

**RdMacros** Rdpack

**Author** Alex Fisch [aut],  
Daniel Grose [aut, cre],  
Lawrence Bardwell [ctb],  
Idris Eckley [ths],  
Paul Fearnhead [ths]

**Maintainer** Daniel Grose <dan.grose@lancaster.ac.uk>

**Repository** CRAN

**Date/Publication** 2019-09-19 12:00:03 UTC

## R topics documented:

ac_corrected . . . . .	2
anomaly_series . . . . .	3

capa	4
capa.mv	6
capa.uv	7
collective_anomalies	8
Lightcurves	10
machinetemp	10
pass	11
period_average	12
plot	13
point_anomalies	14
robustscale	15
show	16
simulate	17
summary	18

## Index 19

---

ac_corrected	<i>Transforms the data X to account for autocorrelation.</i>
--------------	--

---

### Description

Transforms the data  $X$  to account for autocorrelation by centring and scaling. It uses the transformation  $X'_i = \frac{X_i - \mu_i}{k_i \sigma_i}$ , where  $\mu_i$  and  $\sigma_i$  are robust estimates for the mean and standard deviation of each variate (column),  $X_i$ , of  $X$ . The estimates are calculated using the median and median absolute deviation. The scaling  $k_i = \sqrt{\left(\frac{1+\phi_i}{1-\phi_i}\right)}$ , with  $\phi_i$  a robust estimate for the autocorrelation at lag 1, is used to account for AR(1) structure in the noise.

### Usage

```
ac_corrected(X)
```

### Arguments

$X$	A numeric matrix containing the potentially multivariate data to be transformed. Each column corresponds to a component and each row to an observation.
-----	---

### Value

A numeric matrix of the same dimension as  $X$  containing the transformed data.

### Examples

```
library(anomaly)
# generate some multivariate data
set.seed(0)
X<-simulate(n=1000,p=4,mu=10,locations=c(200,400,600),
            duration=100,proportions=c(0.25,0.5,0.75))
# compare the medians of each variate and transformed variate
```

```

head(apply(X,2,median))
head(apply(ac_corrected(X),2,median))
# compare the variances of each variate and transformed variate
head(apply(X,2,var))
head(apply(ac_corrected(X),2,var))

```

---

anomaly\_series      *A technique for detecting anomalous segments based on CAPA.*

---

### Description

Method for calling [capa.uv](#) in a manner consistent with older versions of the anomaly package.

### Usage

```

anomaly_series(x, penaltywindow = NULL, penaltyanomaly = NULL,
  minimumsegmentlength = 10, warnings = TRUE, method = "meanvar")

```

### Arguments

x	A numeric vector containing the data which is to be inspected.
penaltywindow	A numeric constant indicating the penalty for adding an additional epidemic changepoint. It defaults to a BIC style penalty if no argument is provided.
penaltyanomaly	A numeric constant indicating the penalty for adding an additional point anomaly. It defaults to a BIC style penalty if no argument is provided.
minimumsegmentlength	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
warnings	A logical determining whether the warnings should be displayed. It defaults to TRUE.
method	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies characterised by joint changes in mean and variance (the default) and "mean" for collective anomalies characterised by changes in mean only.

### Value

An anomaly\_series object.

### References

Fisch A, Eckley I, Fearnhead P (2018). "A linear time method for the detection of point and collective anomalies." *ArXiv e-prints*. 1806.01947.

## Examples

```

library(anomaly)
set.seed(2018)
# Generate data typically following a normal distribution with mean 0 and variance 1.
# Then introduce 3 anomaly windows and 4 point outliers.
x = rnorm(5000)
x[401:500] = rnorm(100,4,1)
x[1601:1800] = rnorm(200,0,0.01)
x[3201:3500] = rnorm(300,0,10)
x[c(1000,2000,3000,4000)] = rnorm(4,0,100)
inferred_anomalies = anomaly_series(x)
plot(inferred_anomalies)
summary(inferred_anomalies)
print(inferred_anomalies)

### Repeat with other method

inferred_anomalies = anomaly_series(x,method="mean")
plot(inferred_anomalies)
summary(inferred_anomalies)
print(inferred_anomalies)

### Real data example:

library(anomaly)
data(Lightcurves)
### Plot the data for Kepler 10965588: No transit apparent
plot(Lightcurves$Kepler10965588$Day,Lightcurves$Kepler10965588$Brightness,xlab = "Day",pch=".")
### Examine a period of 62.9 days for Kepler 10965588
binned_data = period_average(Lightcurves$Kepler10965588,62.9)
inferred_anomalies = anomaly_series(binned_data)
plot(inferred_anomalies,xlab="Bin")
# We found a planet!

```

---

capa

*A technique for detecting anomalous segments and points based on CAPA.*

---

## Description

A technique for detecting anomalous segments and points based on CAPA (Collective And Point Anomalies) by Fisch et al. (2018). This is a generic method that can be used for both univariate and multivariate data. The specific method that is used for the analysis is deduced by capa from the dimensions of the data.

## Usage

```

capa(x, beta = NULL, beta_tilde = NULL, type = "meanvar",
     min_seg_len = 10, max_seg_len = NULL, max_lag = 0,
     transform = robustscale)

```

**Arguments**

x	A numeric matrix with n rows and p columns containing the data which is to be inspected.
beta	A numeric vector of length p, giving the marginal penalties. If $p > 1$ , type = "meanvar" or type = "mean" and max_lag > 0 it defaults to the penalty regime 2' described in Fisch, Eckley and Fearnhead (2019). If $p > 1$ , type = "mean" and max_lag = 0 it defaults to the pointwise minimum of the penalty regimes 1, 2, and 3 in Fisch, Eckley and Fearnhead (2019).
beta_tilde	A numeric constant indicating the penalty for adding an additional point anomaly. It defaults to a BIC style penalty if no argument is provided.
type	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies characterised by joint changes in mean and variance (the default) or "mean" for collective anomalies characterised by changes in mean only.
min_seg_len	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
max_seg_len	An integer indicating the maximum length of epidemic changes. It must be at least min_seg_len and defaults to Inf.
max_lag	A non-negative integer indicating the maximum start or end lag. Only useful for multivariate data. Default value is 0.
transform	A function used to centre the data prior to analysis by <code>capa</code> . This can, for example, be used to compensate for the effects of autocorrelation in the data. Importantly, the untransformed data remains available for post processing results obtained using <code>capa</code> . The package includes several methods that are commonly used for the transform, (see <code>robustscale</code> and <code>ac_corrected</code> ), but a user defined function can be specified. The default values is <code>transform=robust_scale</code> .

**Value**

An S4 class of type `capa.class`.

**References**

Fisch A, Eckley I, Fearnhead P (2018). "A linear time method for the detection of point and collective anomalies." *ArXiv e-prints*. 1806.01947.

**Examples**

```
library(anomaly)
# generate some multivariate data
set.seed(0)
sim.data<-simulate(n=500,p=200,mu=2,locations=c(100,200,300),
                  duration=6,proportions=c(0.04,0.06,0.08))
res<-capa(sim.data,type="mean",min_seg_len=2,max_lag=5)
collective_anomalies(res)
```

---

capa.mv	<i>Detection of multivariate anomalous segments and points using MVCAPA.</i>
---------	--

---

### Description

This function implements MVCAPA (Multi-Variate Collective And Point Anomaly) from Fisch et al. (2019). It detects potentially lagged collective anomalies as well as point anomalies in multivariate time series data. The runtime of MVCAPA scales linearly (up to logarithmic factors) in  $n_{col}(x)$  and  $maxlag$ . If  $max\_seg\_len$  is not set, the runtime scales quadratically at worst and linearly at best in  $n_{row}(x)$ . If  $max\_seg\_len$  is set the runtime scales like  $n_{row}(x)*max\_seg\_len$ .

### Usage

```
capa.mv(x, beta = NULL, beta_tilde = NULL, type = "meanvar",
        min_seg_len = 10, max_seg_len = Inf, max_lag = 0,
        transform = robustscale)
```

### Arguments

x	A numeric matrix with n rows and p columns containing the data which is to be inspected.
beta	A numeric vector of length p, giving the marginal penalties. If type = "meanvar" or if type = "mean" and maxlag > 0 it defaults to the penalty regime 2' described in Fisch, Eckley and Fearnhead (2019). If type = "mean" and maxlag = 0 it defaults to the pointwise minimum of the penalty regimes 1, 2, and 3 in Fisch, Eckley and Fearnhead (2019).
beta_tilde	A numeric constant indicating the penalty for adding an additional point anomaly. It defaults to a BIC style penalty if no argument is provided.
type	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies characterised by joint changes in mean and variance (the default) or "mean" for collective anomalies characterised by changes in mean only.
min_seg_len	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
max_seg_len	An integer indicating the maximum length of epidemic changes. It must be at least the min_seg_len and defaults to Inf.
max_lag	A non-negative integer indicating the maximum start or end lag. Default value is 0.
transform	A function used to transform the data prior to analysis by <code>capa.mv</code> . This can, for example, be used to compensate for the effects of autocorrelation in the data. Importantly, the untransformed data remains available for post processing results obtained using <code>capa.mv</code> . The package includes several methods that are commonly used for the transform, (see <code>robustscale</code> and <code>ac_corrected</code> ), but a user defined function can be specified. The default value is <code>transform=robust_scale</code> .

**Value**

An S4 class of type `capa.mv.class`.

**References**

Fisch A, Eckley I, Fearnhead P (2019). “Subset multivariate collective and point anomaly detection.” *ArXiv e-prints*. 1909.01691.

**Examples**

```
library(anomaly)
# generate some multivariate data
set.seed(0)
sim.data<-simulate(n=500,p=200,mu=2,locations=c(100,200,300),
                  duration=6,proportions=c(0.04,0.06,0.08))
res<-capa.mv(sim.data,type="mean",min_seg_len=2,max_lag=5)
collective_anomalies(res)
```

---

 capa.uv

---

*Detection of univariate anomalous segments and points using CAPA.*


---

**Description**

A technique for detecting anomalous segments and points in univariate time series data based on CAPA (Collective And Point Anomalies) by Fisch et al. (2018). CAPA assumes that the data has a certain mean and variance for most time points and detects segments in which the mean and/or variance deviates from the typical mean and variance as collective anomalies. It also detects point outliers and returns a measure of strength for the changes in mean and variance. If the number of anomalous windows scales linearly with the number of data points, CAPA scales linearly with the number of data points. At worst, if there are no anomalies at all and `max_seg_len` is unspecified, the computational cost of CAPA scales quadratically with the number of data points.

**Usage**

```
capa.uv(x, beta = NULL, beta_tilde = NULL, type = "meanvar",
        min_seg_len = 10, max_seg_len = Inf, transform = robustscale)
```

**Arguments**

<code>x</code>	A numeric vector containing the data which is to be inspected.
<code>beta</code>	A numeric constant indicating the penalty for adding an additional epidemic changepoint. It defaults to a BIC style penalty if no argument is provided.
<code>beta_tilde</code>	A numeric constant indicating the penalty for adding an additional point anomaly. It defaults to a BIC style penalty if no argument is provided.

type	A string indicating which type of deviations from the baseline are considered. Can be "meanvar" for collective anomalies characterised by joint changes in mean and variance (the default) or "mean" for collective anomalies characterised by changes in mean only.
min_seg_len	An integer indicating the minimum length of epidemic changes. It must be at least 2 and defaults to 10.
max_seg_len	An integer indicating the maximum length of epidemic changes. It must be at least the min_seg_len and defaults to Inf.
transform	A function used to transform the data prior to analysis by <a href="#">capa.uv</a> . This can, for example, be used to compensate for the effects of autocorrelation in the data. Importantly, the untransformed data remains available for post processing results obtained using <a href="#">capa.uv</a> . The package includes several methods that are commonly used for the transform, (see <a href="#">robustscale</a> and <a href="#">ac_corrected</a> ), but a user defined function can be specified. The default values is transform=robust_scale.

### Value

An S4 class of type capa.uv.class.

### References

Fisch A, Eckley I, Fearnhead P (2018). "A linear time method for the detection of point and collective anomalies." *ArXiv e-prints*. 1806.01947.

### Examples

```
library(anomaly)
data(machinetemp)
attach(machinetemp)
res<-capa.uv(temperature,type="mean")
canoms<-collective_anomalies(res)
dim(canoms)[1] # over fitted due to autocorrelation
psi<-0.98 # computed using covRob
inflated_penalty<-3*(1+psi)/(1-psi)*log(length(temperature))
res<-capa.uv(temperature,type="mean",beta=inflated_penalty,
            beta_tilde=inflated_penalty)
res
```

---

collective\_anomalies    *Collective anomaly location, lags, and mean/variance changes.*

---

### Description

Creates a data frame containing collective anomaly locations, lags and changes in mean and variance as detected by [capa.uv](#), [capa.mv](#), and [capa](#).



For an object produced by [capa.uv](#), `collective_anomalies` returns a data frame with columns containing the start and end position of the anomaly, the change in mean due to the anomaly. When `type="meanvar"`, the change in variance due to the anomaly is also returned in an additional column.

For an object produced by [capa.mv](#), `collective_anomalies` returns a data frame with columns containing the start and end position of the anomaly, the variates affected by the anomaly, as well as their the start and end lags. When `type="mean"` only the change in mean is reported. When `type="meanvar"` both the change in mean and change in variance are included. If `merged=FALSE` (the default), then all the collective anomalies are processed individually even if they are common across multiple variates. If `merged=TRUE`, then the collective anomalies are grouped together across all variates that they appear in.

For an object produced by [capa](#), `collective_anomalies` returns the same results as [capa.uv](#) when the data is univariate, or the same results as [capa.mv](#) when the data is multivariate.

## Usage

```
## S4 method for signature 'capa.class'  
collective_anomalies(object, merged = FALSE)  
  
## S4 method for signature 'capa.uv.class'  
collective_anomalies(object)  
  
## S4 method for signature 'capa.mv.class'  
collective_anomalies(object)
```

## Arguments

<code>object</code>	An S4 class produced by <a href="#">capa</a> , <a href="#">capa.uv</a> and <a href="#">capa.mv</a> .
<code>merged</code>	Boolean value. If <code>merged=TRUE</code> then collective anomalies that are common across multiple variates are merged together. This is useful when comparing the relative strength of multivariate collective anomalies. Default value is <code>merged=FALSE</code> . Note - <code>merged=TRUE</code> is currently only available when <code>type="mean"</code> .

## Value

A data frame.

## See Also

[capa](#), [capa.uv](#), [capa.mv](#).

---

Lightcurves

*Kepler Lightcurve data.*

---

### Description

One of the most successful approaches for the detection of exoplanets is the so called transit method: A star's brightness is continuously measured over time by a powerful telescope. If one or multiple planets orbit this star the recorded luminosity of the star will exhibit periodically recurring dips due to the transits of the planet in front of the telescope's lens – an effect comparable to that of an eclipse. Given how small planets are compared to stars the transit signals are known to be very weak.

The stars included in this file all have known exoplanets with the following periods:

Kepler 1871056: 2 planets with orbital periods of 40.8 and 140.1 days

Kepler 2307415: 2 planets with orbital periods of 4.61 and 12.12 days

Kepler 3102384: 2 planets with orbital periods of 10.57 and 523.9 days

Kepler 3231341: 4 planets with orbital periods of 4.24, 8.15, 12.33, and 19.00 days

Kepler 3447722: 3 planets with orbital periods of 10.30, 16.09, and 35.68 days

Kepler 4139816: 4 planets with orbital periods of 3.34, 7.82, 20.06, and 46.18 days

Kepler 10965588: 1 planet with orbital period of 62.89 days

More information about the exoplanets above and more data can be found at <https://exoplanetarchive.ipac.caltech.edu/index.html>

This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.

### Usage

```
data(Lightcurves)
```

### Format

A list of seven dataframes named "Kepler1871056", "Kepler2307415", "Kepler3102384", "Kepler3231341", "Kepler3447722", "Kepler4139816", and "Kepler10965588". Each dataframe consists of two columns called "Brightness" and "Day", containing measurements of a star's brightness and the measurement's timestamp respectively.

---

machinetemp

*Machine temperature data.*

---

**Description**

Temperature sensor data of an internal component of a large, industrial machine. The data contains three known anomalies. The first anomaly is a planned shutdown of the machine. The second anomaly is difficult to detect and directly led to the third anomaly, a catastrophic failure of the machine. The data consists of 22695 observations of machine temperature recorded at 5 minute intervals along with the date and time of the measurement. The data was obtained from the Numenta Anomaly Benchmark (Ahmad et al. 2017), which can be found at <https://github.com/numenta/NAB>.

**Usage**

```
data(machinetemp)
```

**Format**

A dataframe with 22695 rows and 2 columns. The first column contains the date and time of the temperature measurement. The second column contains the machine temperature.

**References**

Ahmad S, Lavin A, Purdy S, Agha Z (2017). "Unsupervised real-time anomaly detection for streaming data." *Neurocomputing*, **262**, 134 - 147. ISSN 0925-2312, doi: [10.1016/j.neucom.2017.04.070](https://doi.org/10.1016/j.neucom.2017.04.070), Online Real-Time Learning Strategies for Data Streams, <http://www.sciencedirect.com/science/article/pii/S0925231217309864>.

---

pass

*Detection of multivariate anomalous segments using PASS.*

---

**Description**

Implements the PASS (Proportion Adaptive Segment Selection) procedure of Jeng et al. (2012). PASS uses a higher criticism statistic to pool the information about the presence or absence of a collective anomaly across the components. It uses Circular Binary Segmentation to detect multiple collective anomalies.

**Usage**

```
pass(x, alpha = 2, lambda = NULL, Lmax = 10, Lmin = 1,
     transform = robustscale)
```

**Arguments**

x	An n x m real matrix representing n observations of m variates.
alpha	A positive integer > 0. This value is used to stabilise the higher criticism based test statistic used by PASS leading to a better finite sample familywise error rate. Anomalies affecting fewer than alpha components will however in all likelihood escape detection.

lambda	A positive real value setting the threshold value for the familywise Type 1 error. The default value is $(1.1\log(n \times Lmax) + 2\log(\log(m)))/\sqrt{\log(\log(m))}$ .
Lmax	A positive integer ( $Lmax > 0$ ) corresponding to the maximum segment length. The default value is 10.
Lmin	A positive integer ( $Lmax \geq Lmin > 0$ ) corresponding to the minimum segment length. The default value is 1.
transform	A function used to transform the data prior to analysis. The default value is to scale the data using the median and the median absolute deviation.

### Value

An S4 object of type `.pass.class` containing the data `X`, procedure parameter values, and the results.

### References

Jeng XJ, Cai TT, Li H (2012). "Simultaneous discovery of rare and common segment variants." *Biometrika*, **100**(1), 157-172. ISSN 0006-3444, doi: [10.1093/biomet/ass059](https://doi.org/10.1093/biomet/ass059), <http://oup.prod.sis.lan/biomet/article-pdf/100/1/157/479862/ass059.pdf>, <https://doi.org/10.1093/biomet/ass059>.

### Examples

```
library(anomaly)
# generate some multivariate data
set.seed(0)
sim.data<-simulate(n=500,p=200,mu=2,locations=c(100,200,300),
                  duration=6,proportions=c(0.04,0.06,0.08))
res<-pass(sim.data)
summary(res)
plot(res,variate_names=FALSE)
```

---

period_average	<i>A function to search the Kepler data for periodically recurring dips in luminosity.</i>
----------------	--

---

### Description

The signal of transiting planets is very weak, especially if the planet is small. This function amplifies it by exploiting the periodicity of the signal. All observations times are taken modulo the period and then binned. An average is then taken within each bin, those averages then stored as a vector and returned. If the orbital period of an exoplanet (or an integer fraction thereof) is used as argument for "period", the signal to noise ratio of the transit is improved, which can allow for the planet's detection.

### Usage

```
period_average(data, period)
```

**Arguments**

data	A dataframe with one column named "Day" and the other "Brightness", such as Kepler10965588 (included in the package).
period	A numeric which is larger than 0 representing the period (in days) which is to be examined.

**Value**

A vector of numerics.

**Examples**

```
library(anomaly)
data(Lightcurves)
### Plot the data for Kepler 10965588: No transit apparent
plot(Lightcurves$Kepler10965588$Day,Lightcurves$Kepler10965588$Brightness,xlab = "Day",pch=".")
### Examine a period of 62.9 days for Kepler 10965588
binned_data = period_average(Lightcurves$Kepler10965588,62.9)
inferred_anomalies = anomaly_series(binned_data)
plot(inferred_anomalies,xlab="Bin")
# We found a planet!
```

---

plot

*Visualisation of data, collective and point anomalies.*


---

**Description**

Plot methods for S4 objects returned by [capa](#), [capa.uv](#), [capa.mv](#), and [pass](#).

The plot can either be a line plot or a tile plot, the type produced depending on the options provided to the plot function and/or the dimensions of the data associated with the S4 object.

**Usage**

```
## S4 method for signature 'anomaly_series,ANY'
plot(x, xlab = "", ylab = "", ...)

## S4 method for signature 'capa.class,ANY'
plot(x, subset, variate_names, tile_plot)

## S4 method for signature 'capa.uv.class,ANY'
plot(x, variate_name)

## S4 method for signature 'capa.mv.class,ANY'
plot(x, subset, variate_names, tile_plot)

## S4 method for signature 'pass.class,ANY'
plot(x, subset, variate_names, tile_plot)
```

**Arguments**

x	An S4 class produced by <a href="#">capa</a> , <a href="#">capa.uv</a> , <a href="#">capa.mv</a> .
xlab	Character string containing label for the x-axis.
ylab	Character string containing label for the y-axis.
...	Other parameters to be passed to plotting methods.
subset	A numeric vector specifying a subset of the variates to be displayed. Default value is all of the variates present in the data.
variate_names	Logical value indicating if variate names should be displayed on the plot. This is useful when a large number of variates are being displayed as it makes the visualisation easier to interpret. Default value is TRUE.
tile_plot	Logical value. If TRUE then a tile plot of the data is produced. The data displayed in the tile plot is normalised to values in [0,1] for each variate. This type of plot is useful when the data contains are large number of variates. The default value is TRUE if the number of variates is greater than 20.
variate_name	Logical value indicating if the variate name should be displayed. Default value is <code>variate.name=TRUE</code> .

**Value**

A ggplot object.

**See Also**

[capa](#), [capa.uv](#), [capa.mv](#), [pass](#).

---

point\_anomalies      *Point anomaly location and strength.*

---

**Description**

Creates a data frame containing point anomaly locations and strengths as detected by [capa](#), [capa.uv](#), and [capa.mv](#).

For an object produced by [capa.uv](#), the output is a data frame with columns containing the position and strength of the anomaly.

For an object produced by [capa.mv](#), `point_anomalies` returns a data frame with columns containing the position, variate, and strength of the anomaly.

For an object produced by [capa](#), `point_anomalies` returns the same results as [capa.uv](#) when the data is univariate, and the same results as [capa.uv](#) when the data is multivariate.

**Usage**

```
## S4 method for signature 'capa.class'
point_anomalies(object)

## S4 method for signature 'capa.uv.class'
point_anomalies(object)

## S4 method for signature 'capa.mv.class'
point_anomalies(object)
```

**Arguments**

object            An S4 class produced by [capa](#), [capa.uv](#), and [capa.mv](#).

**Value**

A data frame.

**See Also**

[capa](#), [capa.uv](#), [capa.mv](#).

---

robustscale

*robustscale*

---

**Description**

Transforms the data  $X$  by centring and scaling using  $X'_{ij} = \frac{X_i - \mu_i}{\sigma_i}$  where  $\mu_i$  and  $\sigma_i$  are robust estimates for the mean and standard deviation of each variate (column),  $X_i$ , of the multivariate time series  $X$ . The estimates are calculated using the median and median absolute deviation. This method is the default value for the transform argument used by the [capa](#) function, since the [capa](#) method assumes that the typical distribution of the data is standard normal.

**Usage**

```
robustscale(X)
```

**Arguments**

X            A numeric matrix containing the data to be transformed. Each column corresponds to a component and each row to an observation.

**Value**

A numeric matrix containing the transformed data.

## Examples

```
library(anomaly)
# generate some multivariate data
set.seed(0)
X<-simulate(n=1000,p=4,mu=10,locations=c(200,400,600),
            duration=100,proportions=c(0.25,0.5,0.75))
# compare the medians of each variate and transformed variate
head(apply(X,2,median))
head(apply(robustscale(X),2,median))
# compare the variances of each variate and transformed variate
head(apply(X,2,var))
head(apply(robustscale(X),2,var))
```

---

show

*Displays S4 objects produced by capa methods.*

---

## Description

Displays S4 object produced by [capa](#), [capa.uv](#), [capa.mv](#), and [pass](#). The information produced is the same as that provided by the summary method. The method is used by the S4 system for automatic printing.

## Usage

```
## S4 method for signature 'anomaly_series'
show(object)

## S4 method for signature 'capa.class'
show(object)

## S4 method for signature 'pass.class'
show(object)
```

## Arguments

object            An S4 class produced by [capa](#), [capa.uv](#), [capa.mv](#), or [pass](#).

## See Also

[capa](#), [capa.uv](#), [capa.mv](#), [pass](#).



---

`simulate`*A function for generating simulated multivariate data*

---

### Description

Generates multivariate simulated data having  $n$  observations and  $p$  variates. The data have a standard Gaussian distribution except at a specified number of locations where there is a change in mean in a proportion of the variates. The function is useful for generating data to demonstrate and assess multivariate anomaly detection methods such as `capa.mv` and `pass`.

### Usage

```
simulate(n = 100, p = 10, mu = 1, locations = 40, durations = 20,  
         proportions = 0.1)
```

### Arguments

<code>n</code>	The number of observations. The default is $n=100$ .
<code>p</code>	The number of variates. The default is $p=10$ .
<code>mu</code>	The change in mean. Default is $\mu=1$ .
<code>locations</code>	A vector of locations (or scalar for a single location) where the change in mean occurs. The default is <code>locations=20</code> .
<code>durations</code>	A scalar or vector (the same length as <code>locations</code> ) of values indicating the duration for the change in mean. If the durations are all of the same length then a scalar value can be used. The default is <code>durations=20</code> .
<code>proportions</code>	A scalar or vector (the same length as <code>locations</code> ) of values in the range $(0,1]$ indicating the proportion of variates at each location that are affected by the change in mean. If the proportions are all same than a scalar value can be used. The default is <code>proportions=0.1</code> .

### Value

A matrix with  $n$  rows and  $p$  columns

### Examples

```
library(anomaly)  
sim.data<-simulate(500,200,2,c(100,200,300),6,c(0.04,0.06,0.08))
```

---

summary

*Summary of collective and point anomalies.*

---

### Description

Summary methods for S4 objects returned by [capa](#), [capa.uv](#), [capa.mv](#), and [pass](#). The output displayed depends on the type of S4 object passed to `summary`. For all types, the output indicates whether the data is univariate or multivariate, the number of observations in the data, and the type of change being detected.

For an object produced by [capa.uv](#) or [capa.mv](#), `summary` displays a summary of the analysis type used and the results produced by [point\\_anomalies](#) and [collective\\_anomalies](#).

For an object produced by [capa](#) is the same as for an object produced by [capa.uv](#) or [capa.mv](#).

### Usage

```
## S4 method for signature 'anomaly_series'  
summary(object)
```

```
## S4 method for signature 'capa.class'  
summary(object)
```

```
## S4 method for signature 'capa.uv.class'  
summary(object)
```

```
## S4 method for signature 'capa.mv.class'  
summary(object)
```

```
## S4 method for signature 'pass.class'  
summary(object)
```

### Arguments

`object` An S4 class produced by [capa](#), [capa.uv](#), [capa.mv](#), or [pass](#).  
... Other parameters to be passed to `summary` methods.

### See Also

[capa](#), [capa.uv](#), [capa.mv](#), [pass](#), [point\\_anomalies](#), [collective\\_anomalies](#).

# Index

## \*Topic **datasets**

- Lightcurves, [10](#)
- machinetemp, [10](#)
  
- ac\_corrected, [2](#), [5](#), [6](#), [8](#)
- anomaly\_series, [3](#)
  
- capa, [4](#), [5](#), [8](#), [9](#), [13–16](#), [18](#)
- capa.mv, [6](#), [6](#), [8](#), [9](#), [13–16](#), [18](#)
- capa.uv, [3](#), [7](#), [8](#), [9](#), [13–16](#), [18](#)
- collective\_anomalies, [8](#), [18](#)
- collective\_anomalies, capa.class-method (collective\_anomalies), [8](#)
- collective\_anomalies, capa.mv.class-method (collective\_anomalies), [8](#)
- collective\_anomalies, capa.uv.class-method (collective\_anomalies), [8](#)
  
- Lightcurves, [10](#)
  
- machinetemp, [10](#)
  
- pass, [11](#), [13](#), [14](#), [16](#), [18](#)
- period\_average, [12](#)
- plot, [13](#)
- plot, anomaly\_series, ANY-method (plot), [13](#)
- plot, capa.class, ANY-method (plot), [13](#)
- plot, capa.mv.class, ANY-method (plot), [13](#)
- plot, capa.uv.class, ANY-method (plot), [13](#)
- plot, pass.class, ANY-method (plot), [13](#)
- plot-pass.class (plot), [13](#)
- point\_anomalies, [14](#), [18](#)
- point\_anomalies, capa.class-method (point\_anomalies), [14](#)
- point\_anomalies, capa.mv.class-method (point\_anomalies), [14](#)
- point\_anomalies, capa.uv.class-method (point\_anomalies), [14](#)
  
- robustscale, [5](#), [6](#), [8](#), [15](#)
  
- show, [16](#)
- show, anomaly\_series-method (show), [16](#)
- show, capa.class-method (show), [16](#)
- show, pass.class-method (show), [16](#)
- simulate, [17](#)
- summary, [18](#)
- summary, anomaly\_series-method (summary), [18](#)
- summary, capa.class-method (summary), [18](#)
- summary, capa.mv.class-method (summary), [18](#)
- summary, capa.uv.class-method (summary), [18](#)
- summary, pass.class-method (summary), [18](#)