Package ‘revss’

June 3, 2020

Type Package

Title Robust Estimation in Very Small Samples

Version 0.0.2

Date 2020-05-28

Description Implements the estimation techniques described in Rousseeuw & Verboven (2002) <doi:10.1016/S0167-9473(02)00078-6> for the location and scale of very small samples.

License BSD_2_clause + file LICENSE

URL https://github.com/aadler/revss

BugReports https://github.com/aadler/revss/issues

Encoding UTF-8

Suggests testthat, covr

Imports stats

LazyData true

NeedsCompilation no

Author Avraham Adler [aut, cph, cre] (<https://orcid.org/0000-0002-3039-0703>)

Maintainer Avraham Adler <Avraham.Adler@gmail.com>

Repository CRAN

Date/Publication 2020-06-03 15:20:03 UTC

R topics documented:

revss-package *Robust Estimation in Very Small Samples*

Description

Implements the estimation techniques described in Rousseeuw & Verboven (2002) <doi:10.1016/S0167-9473(02)00078-6> for the location and scale of very small samples.

Details

The DESCRIPTION file:

Package:	revss
Type:	Package
Title:	Robust Estimation in Very Small Samples
Version:	0.0.2
Date:	2020-05-28
Authors@R:	c(person(given = "Avraham", family = "Adler", role = c("aut", "cph", "cre"), email = "Avraham.Adler@
Description:	Implements the estimation techniques described in Rousseeuw & Verboven (2002) <doi:10.1016/S0167
License:	BSD_2_clause + file LICENSE
URL:	https://github.com/aadler/revss
BugReports:	https://github.com/aadler/revss/issues
Encoding:	UTF-8
Suggests:	testthat, covr
Imports:	stats
LazyData:	true
NeedsCompilation:	no
Author:	Avraham Adler [aut, cph, cre] (<https://orcid.org/0000-0002-3039-0703>)
Maintainer:	Avraham Adler <Avraham.Adler@gmail.com>

Index of help topics:

```
adm                     Average Distance to the Median
revss-package           Robust Estimation in Very Small Samples
robLoc                  Robust Estimate of Location
robScale                Robust Estimate of Scale
```

Author(s)

NA

Maintainer: NA

`adm` *Average Distance to the Median*

Description

Compute the mean absolute deviation from the median, and (by default) adjust by a factor for asymptotically normal consistency.

Usage

```
adm(x, center = median(x), constant = sqrt(pi / 2), na.rm = FALSE)
```

Arguments

`x`	A numeric vector.
`center`	The central value from which to measure the average distance. Defaults to the median.
`constant`	A scale factor for asymptotic normality. In large samples, the `adm` asymptotically approaches $\sqrt{\frac{2}{\pi}}$ times the sample standard deviation.
`na.rm`	If `TRUE` then `NA` values are stripped from `x` before computation takes place.

Details

Computes the average distance, as an absolute value, between each observation and the central observation—usually the median. In statistical literature this is also called the **mean absolute deviation around the median**. Unfortunately, this shares the same acronym as the median absolute deviation (MAD), which is the median equivalent of this function.

The default is to adjust the factor for asymptotically normal consistency. In large samples this approaches $\sqrt{\frac{2}{\pi}}$, which is the default. In smaller samples, simulation studies suggest for samples of up to approximately length $n = 50$, it behaves more akin to $1.296816e^{n\times-e^{7.647996}} + 0.2757346e^{n\times-e^{-1.664514}}$.

If `na.rm` is `TRUE` then `NA` values are stripped from `x` before computation takes place. If this is not done then an `NA` value in `x` will cause `mad` to return `NA`.

Value

$$ADM = C\frac{1}{n}\sum_{i=1}^{n}|x_i - median(x)$$

where C is the consistency constant.

Author(s)

Avraham Adler `<Avraham.Adler@gmail.com>`

References

Nair, K. R. (1947) A Note on the Mean Deviation from the Median. *Biometrika*, **34**, 3/4, 360–362. doi: 10.2307/2332448

See Also

`mad` for the median absolute deviation from the `median`

Examples

```
adm(c(1:9))
x <- c(1,2,3,5,7,8)
c(adm(x), adm(x, constant = 1))
```

---

robLoc *Robust Estimate of Location*

Description

Compute the robust estimate of location for very small samples.

Usage

```
robLoc(x, scale = NULL, na.rm = FALSE, maxit = 60, tol = sqrt(.Machine$double.eps))
```

Arguments

`x`	A numeric vector.
`scale`	The scale, if known, can be used to enhance the estimate for the location; defaults to unknown.
`na.rm`	If `TRUE` then `NA` values are stripped from `x` before computation takes place.
`maxit`	The maximum number of iterations; defaults to 60.
`tol`	The desired accuracy.

Details

Computes the M-estimator for location using the logistic ψ function of Rousseeuw & Verboven (2002, 4.1). If there are three or fewer entries, the function defaults to the `median`.

If the scale is known and passed through `scale`, the algorithm uses the suggestion in Rousseeuw & Verboven section 5 (2002), substituting the known scale for the `mad`.

If `na.rm` is `TRUE` then `NA` values are stripped from `x` before computation takes place. If this is not done then an `NA` value in `x` will cause `mad` to return `NA`.

The tolerance and number of iterations are similar to those in existing base R functions.

Value

Solves for the robust estimate of location, T_n, which is the solution to

$$\frac{1}{n}\sum_{i=1}^{n}\psi\left(\frac{x_i - T_n}{S_n}\right) = 0$$

where S_n is fixed at `mad(x)`. The ψ-function selected by Rousseeuw & Verboven is

$$\psi_{log}(x) = \frac{e^x - 1}{e^x + 1}$$

This is equivalent to `2 * plogit(x) -1`.

Author(s)

Avraham Adler `<Avraham.Adler@gmail.com>`

References

Rousseeuw, Peter J. and Verboven, Sabine (2002) Robust estimation in very small samples. *Computational Statistics & Data Analysis*, **40**, (4), 741–758. doi: 10.1016/S01679473(02)000786

See Also

`median`

Examples

```
robLoc(c(1:9))
x <- c(1,2,3,5,7,8)
robLoc(x)
```

`robScale` *Robust Estimate of Scale*

Description

Compute the robust estimate of scale for very small samples.

Usage

```
robScale(x, loc = NULL, implbound = 1e-4, na.rm = FALSE, maxit = 60,
tol = sqrt(.Machine$double.eps))
```

Arguments

`x`	A numeric vector.
`loc`	The location, if known, can be used to enhance the estimate for the scale; defaults to unknown.
`implbound`	The smallest value that `mad` is allowed before being considered too close to 0.
`na.rm`	If `TRUE` then `NA` values are stripped from `x` before computation takes place.
`maxit`	The maximum number of iterations; defaults to 60.
`tol`	The desired accuracy.

Details

Computes the M-estimator for scale using a smooth ρ-function defined as the square of the logistic ψ function used in location estimation (Rousseeuw & Verboven, 2002, 4.2). When the sequence of observations is too short for a robust estimate, the scale estimate will default to `mad` so long as `mad` has not "imploded", i.e. it is greater than `implbound` which defaults to 0.0001.

If the location is known and passed through `loc`, the algorithm uses the suggestion in Rousseeuw & Verboven section 5 (2002), converting the observations to distances from 0 and iterating on the adjusted sequence.

If `na.rm` is `TRUE` then `NA` values are stripped from `x` before computation takes place. If this is not done then an `NA` value in `x` will cause `mad` to return `NA`.

The tolerance and number of iterations are similar to those in existing base R functions.

Value

Solves for the robust estimate of scale, S_n, which is the solution to

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{x_i - T_n}{S_n}\right) = \beta$$

where T_n is fixed at `median(x)` and β is fixed at 0.5. The ρ-function selected by Rousseeuw & Verboven is based on the square of the ψ-function used in `robLoc`. Specifically

$$\rho_{log}(x) = \psi_{log}^2\left(\frac{x}{0.3739}\right)$$

The 0.3739 is needed for β to be 0.5.

Author(s)

Avraham Adler <Avraham.Adler@gmail.com>

References

Rousseeuw, Peter J. and Verboven, Sabine (2002) Robust estimation in very small samples. *Computational Statistics & Data Analysis*, **40**, (4), 741–758. doi: 10.1016/S01679473(02)000786

See Also

adm and mad as basic robust estimators of scale.

Qn and Sn in the **robustbase** package which are specialized robust scale estimators for larger samples. The latter two are based on code written by Peter Rousseeuw.

Examples

```
robScale(c(1:9))
x <- c(1,2,3,5,7,8)
c(robScale(x), robScale(x, loc = 5))
```

Index