

Package: rrobot (via r-universe)

May 9, 2026

Type Package

Title Robust Outlier Detection for Diverse Distributions

Version 0.1.3

Maintainer Amanda Mejia <mandy.mejia@gmail.com>

Description Provides robust outlier detection techniques for identifying anomalies in multivariate data, with a focus on methods that remain effective under non-Gaussian distributions. For more details see Saluja, Parlak, and Mejia (2026+) <[doi:10.48550/arXiv.2505.11806](https://doi.org/10.48550/arXiv.2505.11806)>.

License GPL-3

URL <https://github.com/mandymejia/rrobot>

BugReports <https://github.com/mandymejia/rrobot/issues>

Depends R (>= 3.6.0)

Imports MASS, stats, cellWise, expm, robustbase, gamlss, imputeTS, isotree, ggplot2, tidyr, reshape2, rlang

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Language en-US

Config/pak/sysreqs libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository <https://mandymejia.r-universe.dev>

Date/Publication 2026-03-04 21:09:12 UTC

RemoteUrl <https://github.com/mandymejia/rrobot>

RemoteRef HEAD

RemoteSha 40adfa71fd601dd4ea2751eba83185d372e7e8a7

Contents

...	3
alpha	3
B	3
binwidth	3
boot_quant	4
compute_RD	4
cov_mcd	5
cutoff	5
dist	6
imp_data	6
imp_datasets	6
impute_method	6
ind_incl	7
k	7
M	7
method	7
method_univOut	8
mode	8
plot.RD	8
quantile	9
RD_obj	9
RD_org_obj	9
SHASH_out	10
SHASH_transform	12
summary.F_result	13
summary.MI_boot_result	14
summary.MI_result	14
summary.SI_boot_result	15
summary.SI_result	15
thr	16
thresh_result	16
threshold_RD	16
trans	18
verbose	18
w	18
x	18
Index	19

... *Dots parameter documentation*

Description

Dots parameter documentation

Arguments

... Additional arguments to to method-specific functions.

alpha *Alpha parameter documentation*

Description

Alpha parameter documentation

Arguments

alpha Significance level used to compute RD threshold (default = 0.01 for 99th percentile).

B *B parameter documentation*

Description

B parameter documentation

Arguments

B Integer; number of bootstrap samples per imputed dataset (default = 1000).

binwidth *Binwidth parameter documentation*

Description

Binwidth parameter documentation

Arguments

binwidth Histogram bin width (default = 0.1).

boot_quant	<i>Boot_quant parameter documentation</i>
------------	---

Description

Boot_quant parameter documentation

Arguments

boot_quant	Numeric; confidence level for bootstrap confidence intervals (default = 0.95, for 95% CI).
------------	--

compute_RD	<i>Compute Squared robust distance and covariance from a Subset</i>
------------	---

Description

Calculates the robust mean, covariance matrix, and optionally robust distances using either:

- "auto" mode: automatically selects the best robust subset using covMcd
- "manual" mode: uses provided robust covariance matrix and subset indices

Usage

```
compute_RD(
  x,
  mode = c("auto", "manual"),
  cov_mcd = NULL,
  ind_incl = NULL,
  dist = TRUE
)
```

Arguments

x	A numeric matrix or data frame of dimensions $T \times p$ (observations \times variables).
mode	Character string; either "auto" (default) to compute MCD internally or "manual" to use user-supplied values.
cov_mcd	Optional covariance matrix ($p \times p$); required in "manual" mode.
ind_incl	Optional vector of row indices used to compute the robust mean; required in "manual" mode.
dist	Logical; if TRUE, compute squared robust Mahalanobis distances for all observations.

Value

A list with elements:

ind_incl Vector of row indices used to compute the robust mean and covariance.

ind_excl Vector of excluded row indices.

h Number of included observations.

xbar_star Robust mean vector (length p).

S_star Robust covariance matrix ($p \times p$).

invcov_sqrt Matrix square root of the inverse covariance matrix ($p \times p$).

RD Squared robust distances for all observations (length T), or NULL if dist = FALSE.

call The matched function call.

 cov_mcd

Cov_mcd parameter documentation

Description

Cov_mcd parameter documentation

Arguments

cov_mcd Optional covariance matrix ($p \times p$); required in "manual" mode.

 cutoff

Cutoff parameter documentation

Description

Cutoff parameter documentation

Arguments

cutoff A numeric value indicating how many MADs away from the median to flag as outliers. The default value is set to be 4.

dist *Dist parameter documentation*

Description

Dist parameter documentation

Arguments

dist Logical; if TRUE, compute squared robust Mahalanobis distances for all observations.

imp_data *Imp_data parameter documentation*

Description

Imp_data parameter documentation

Arguments

imp_data A numeric matrix ($T \times p$) of single-imputed data.

imp_datasets *Imp_datasets parameter documentation*

Description

Imp_datasets parameter documentation

Arguments

imp_datasets A list of M numeric matrices ($T \times p$); multiply imputed datasets.

impute_method *Impute_method parameter documentation*

Description

Impute_method parameter documentation

Arguments

impute_method Character string; imputation method for univariate outliers.

ind_incl	<i>Ind_incl parameter documentation</i>
----------	---

Description

Ind_incl parameter documentation

Arguments

ind_incl	Optional vector of row indices used to compute the robust mean; required in "manual" mode.
----------	--

k	<i>K parameter documentation</i>
---	----------------------------------

Description

K parameter documentation

Arguments

k	Integer; number of perturbation cycles per imputation (default = 10).
---	---

M	<i>M parameter documentation</i>
---	----------------------------------

Description

M parameter documentation

Arguments

M	Integer; number of multiply imputed datasets (default = 5).
---	---

method	<i>Threshold_method parameter documentation</i>
--------	---

Description

Threshold_method parameter documentation

Arguments

method	Character string; one of "all", "SI", "SI_boot", "MI", "MI_boot", "F", "SHASH".
--------	---

method_univOut	<i>Method_univOut parameter documentation</i>
----------------	---

Description

Method_univOut parameter documentation

Arguments

method	Character string. One of "SHASH" or "robZ".
--------	---

mode	<i>Mode parameter documentation</i>
------	-------------------------------------

Description

Mode parameter documentation

Arguments

mode	Character string; either "auto" (default) to compute MCD internally or "manual" to use user-supplied values.
------	--

plot.RD	<i>Plot Method for RD Analysis Results</i>
---------	--

Description

Creates diagnostic plots for robust distance analysis results.

Usage

```
## S3 method for class 'RD'
plot(x, type = c("histogram", "imputations", "univOut"), method = NULL, ...)
```

Arguments

x	An object of class "RD" from RD() or threshold_RD().
type	Character string specifying plot type: "histogram" (default), "imputations", or "univOut".
method	Character string specifying threshold method. Auto-detected if NULL.
...	Additional arguments passed to plotting functions.

Value

A ggplot object.

quantile	<i>Quantile parameter documentation</i>
----------	---

Description

Quantile parameter documentation

Arguments

quantile	Numeric in (0,1) specifying the upper quantile for thresholding; the expected False Positive Rate for the chosen threshold.
----------	---

RD_obj	<i>RD_obj parameter documentation</i>
--------	---------------------------------------

Description

RD_obj parameter documentation

Arguments

RD_obj	Pre-computed RD_result object from compute_RD .
--------	---

RD_org_obj	<i>RD_org_obj parameter documentation</i>
------------	---

Description

RD_org_obj parameter documentation

Arguments

RD_org_obj	Output list from compute_RD on the original data. Must contain \$RD, \$S_star, and \$ind_incl.
------------	--

SHASH_out

*SHASH-based Outlier Detection (Extended)***Description**

Detects univariate outliers using an iterative SHASH fitting process with optional pre-flagging strategies. A SHASH (Sinh-Arcsinh) distribution is fitted to the data iteratively, each time excluding candidate outliers from the fit, until the set of flagged observations converges or `maxit` is reached.

Usage

```
SHASH_out(
  x,
  thr0 = 2.58,
  thr1 = 2.58,
  thr = 4,
  tail = c("both", "upper", "lower"),
  use_iso = TRUE,
  thr_iso = 0.6,
  maxit = 100,
  weight_init = NULL
)
```

Arguments

<code>x</code>	Numeric vector. May contain NA values; they are excluded from fitting and propagated as NA in all output vectors.
<code>thr0</code>	Positive numeric scalar. Threshold for initial outlier pre-flagging when <code>use_iso = FALSE</code> (default: 2.58).
<code>thr1</code>	Positive numeric scalar. Threshold used to classify observations as inliers during iterative convergence (default: 2.58).
<code>thr</code>	Positive numeric scalar. Final threshold applied to the converged SHASH-normalised scores to declare outliers in the returned output (default: 4).
<code>tail</code>	Character string specifying which tail(s) to check for outliers. Must be one of "both" (default), "upper", or "lower". <ul style="list-style-type: none"> • "upper": detect upper-tail outliers only. • "lower": detect lower-tail outliers only. • "both": detect two-sided outliers.
<code>use_iso</code>	Logical. If TRUE (default), uses an isolation forest (via <code>isotree</code>) to pre-screen candidate outliers before the iterative fitting loop begins.
<code>thr_iso</code>	Numeric scalar in [0, 1]. Isolation forest anomaly score threshold above which observations are treated as candidate outliers during pre-screening (default: 0.6). Only used when <code>use_iso = TRUE</code> .

maxit	Positive integer. Maximum number of fitting iterations before the algorithm stops regardless of convergence (default: 100).
weight_init	Optional logical vector of length $\text{length}(x)$. If supplied, these weights initialise the iterative fit directly, bypassing both the isolation forest and empirical-rule pre-screening. TRUE means the observation is treated as an inlier in the first iteration.

Value

A list of class "SHASH_out" with the following elements:

out_idx	Integer vector. Indices of observations in x that were flagged as outliers at the final threshold thr .
x_norm	Numeric vector. SHASH-normalised scores for every observation (same length as x ; NA where x was NA).
SHASH_coef	Named list with elements μ , σ , ν , and τ : the fitted SHASH parameter estimates from the final iteration (σ and τ are on the log scale, as returned by <code>gam1ssML</code>).
isotree_scores	Numeric vector of isolation forest anomaly scores (same length as x). NA when $\text{use_iso} = \text{FALSE}$ or weight_init was supplied.
initial_weights	Logical vector. Inlier weights used for the very first fitting iteration (same length as x).
indx_iters	Integer matrix of dimensions $\text{length}(x) \times \text{last_iter}$. Each column records which observations were flagged as outliers (value 1) during that iteration.
norm_iters	Numeric matrix of dimensions $\text{length}(x) \times \text{last_iter}$. Each column records the SHASH-normalised scores from that iteration.
last_iter	Integer. The number of iterations completed before convergence or hitting maxit .
converged	Logical. TRUE if the inlier weight vector stabilised before reaching maxit .
params	List. A record of all input parameters, stored for reproducibility.

Examples

```
# --- Example 1: Synthetic data with known injected outliers -----
# Using rnorm lets us inject outliers at known positions so we can verify
# the function finds exactly what we planted.
set.seed(42)
x <- rnorm(200, mean = 10, sd = 2)

# Shift a handful of observations far into the upper tail
outlier_positions <- c(17, 77, seq(190, 200))
x[outlier_positions] <- x[outlier_positions] + 10

result_sim <- SHASH_out(
  x,
  thr0 = 2.58,
  thr1 = 2.58,
  thr = 4,
  tail = "both",
  use_iso = FALSE # skip isolation forest to keep the example fast
```

```

)

result_sim$out_idx    # should recover positions near outlier_positions
result_sim$converged # did the iterative fit stabilise?

# --- Example 2: Real benchmark data (Hawkins-Bradu-Kass) -----
# hbk is a classic outlier detection benchmark shipped with robustbase,
# which this package already imports, so it is always available.
data("hbk", package = "robustbase")

result_hbk <- SHASH_out(
  hbk$X1,
  thr0 = 2.58,
  thr1 = 2.58,
  thr  = 4,
  tail = "both",
  use_iso = FALSE
)

result_hbk$out_idx # flagged observations in the X1 column
result_hbk$SHASH_coef # fitted SHASH parameters; sigma and tau are log-scale

# Which positions were flagged as outliers?
result_hbk$out_idx

# Did the algorithm converge before hitting maxit?
result_hbk$converged

# How many iterations did it take?
result_hbk$last_iter

```

SHASH_transform

SHASH Data Transformation

Description

These two functions form a matched pair for transforming data between the SHASH (Sinh-Arcsinh) distribution and the standard normal distribution. `SHASH_to_normal()` maps SHASH-distributed observations onto an approximately normal scale; `normal_to_SHASH()` is the inverse.

Usage

```
SHASH_to_normal(x, mu, sigma, nu, tau)
```

```
normal_to_SHASH(x, mu, sigma, nu, tau)
```

Arguments

x	Numeric vector of values to transform.
mu	Numeric scalar. Location parameter controlling the mean of the SHASH distribution.
sigma	Numeric scalar. Spread parameter on the log scale. The function applies $\exp(\text{sigma})$ internally, so pass the raw coefficient as returned by <code>gamLssML()</code> . Pass $\text{sigma} = 0$ to get unit spread since $\exp(0) = 1$.
nu	Numeric scalar. Skewness parameter. A value of 0 gives a symmetric distribution.
tau	Numeric scalar. Tail-weight parameter on the log scale. Pass $\text{tau} = 0$ for normal-like tails since $\exp(0) = 1$.

Value

A numeric vector of transformed values, the same length as x.

Functions

- `SHASH_to_normal()`: Transforms SHASH-distributed data to approximately normal data.
- `normal_to_SHASH()`: Transforms standard normal data back to the SHASH-distributed scale.

Examples

```
set.seed(42)
x <- rnorm(200)
x[c(17, 77)] <- x[c(17, 77)] + 5

mu <- 0; sigma <- 0; nu <- 0; tau <- 0

z <- SHASH_to_normal(x, mu = mu, sigma = sigma, nu = nu, tau = tau)
x_recovered <- normal_to_SHASH(z, mu = mu, sigma = sigma, nu = nu, tau = tau)
all.equal(x, x_recovered)
```

summary.F_result

Summary method for Hardin & Rocke F results

Description

Summary method for Hardin & Rocke F results

Usage

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

Arguments

object An object of class "F_result" or "HR_result"
 ... Additional arguments to to method-specific functions.

Value

NULL, invisibly

summary.MI_boot_result *Summary method for MI_boot results*

Description

Summary method for MI_boot results

Usage

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

Arguments

object An object of class "MI_boot_result"
 ... Additional arguments to to method-specific functions.

Value

NULL, invisibly

summary.MI_result *Summary method for MI results*

Description

Summary method for MI results

Usage

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

Arguments

object An object of class "MI_result"
 ... Additional arguments to to method-specific functions.

Value

NULL, invisibly

summary.SI_boot_result

Summary method for SI_boot results

Description

Summary method for SI_boot results

Usage

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

Arguments

object An object of class "SI_boot_result"
... Additional arguments to to method-specific functions.

Value

NULL, invisibly

summary.SI_result

Summary method for SI results

Description

Summary method for SI results

Usage

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

Arguments

object An object of class "SI_result"
... Additional arguments to to method-specific functions.

Value

NULL, invisibly

thr	<i>Thr parameter documentation</i>
-----	------------------------------------

Description

Thr parameter documentation

Arguments

thr	Threshold multiplier for outlier detection (default = 4).
-----	---

thresh_result	<i>Thresh_result parameter documentation</i>
---------------	--

Description

Thresh_result parameter documentation

Arguments

thresh_result	A threshold result object from any threshold method containing threshold information.
---------------	---

threshold_RD	<i>Comprehensive Outlier Detection Using Robust Distance Thresholding</i>
--------------	---

Description

Performs univariate outlier detection + imputation, robust distance, and multiple thresholding methods.

Usage

```
threshold_RD(
  x,
  w = NULL,
  method = c("SI_boot", "MI", "MI_boot", "SI", "F", "SHASH", "all"),
  RD_obj = NULL,
  impute_method = "mean",
  cutoff = 4,
  trans = "SHASH",
  M = 50,
  k = 100,
```

```

alpha = 0.01,
quantile = 0.01,
verbose = FALSE,
boot_quant = 0.95,
B = 1000
)

```

Arguments

x	A numeric matrix or data frame of dimensions $T \times p$ (observations \times variables).
w	A numeric matrix ($n_time \times L$) of low-kurtosis ICA components used as predictors (required for MI).
method	Character string; one of "all", "SI", "SI_boot", "MI", "MI_boot", "F", "SHASH".
RD_obj	Pre-computed RD_result object from <code>compute_RD</code> .
impute_method	Character string; imputation method for univariate outliers.
cutoff	A numeric value indicating how many MADs away from the median to flag as outliers. The default value is set to be 4.
trans	Character string; transformation method, one of "SHASH" or "robZ".
M	Integer; number of multiply imputed datasets (default = 5).
k	Integer; number of perturbation cycles per imputation (default = 10).
alpha	Significance level used to compute RD threshold (default = 0.01 for 99th percentile).
quantile	Numeric in (0,1) specifying the upper quantile for thresholding; the expected False Positive Rate for the chosen threshold.
verbose	Logical; if TRUE, print progress messages.
boot_quant	Numeric; confidence level for bootstrap confidence intervals (default = 0.95, for 95% CI).
B	Integer; number of bootstrap samples per imputed dataset (default = 1000).

Value

A list with:

thresholds Result from the specific threshold method, or list of all methods if "all".

RD_obj The robust distance object from `compute_RD()`.

call The matched function call.

trans	<i>Trans parameter documentation</i>
-------	--------------------------------------

Description

Trans parameter documentation

Arguments

trans	Character string; transformation method, one of "SHASH" or "robZ".
-------	--

verbose	<i>Verbose parameter documentation</i>
---------	--

Description

Verbose parameter documentation

Arguments

verbose	Logical; if TRUE, print progress messages.
---------	--

w	<i>W parameter documentation</i>
---	----------------------------------

Description

W parameter documentation

Arguments

w	A numeric matrix ($n_time \times L$) of low-kurtosis ICA components used as predictors (required for MI).
---	--

x	<i>X parameter documentation</i>
---	----------------------------------

Description

X parameter documentation

Arguments

x	A numeric matrix or data frame of dimensions $T \times p$ (observations \times variables).
---	--

Index

..., 3

alpha, 3

B, 3

binwidth, 3

boot_quant, 4

compute_RD, 4, 9, 17

cov_mcd, 5

cutoff, 5

dist, 6

imp_data, 6

imp_datasets, 6

impute_method, 6

ind_incl, 7

k, 7

M, 7

method, 7

method_univOut, 8

mode, 8

normal_to_SHASH (SHASH_transform), 12

plot_RD, 8

quantile, 9

RD_obj, 9

RD_org_obj, 9

SHASH_out, 10

SHASH_to_normal (SHASH_transform), 12

SHASH_transform, 12

summary.F_result, 13

summary.MI_boot_result, 14

summary.MI_result, 14

summary.SI_boot_result, 15

summary.SI_result, 15

thr, 16

thresh_result, 16

threshold_RD, 16

trans, 18

verbose, 18

w, 18

x, 18