

# Package ‘extremevalues’

February 14, 2012

**Description** Detect extreme values in onedimensional data

**Version** 2.2

**Date** 2009-13-11

**Title** Univariate outlier detection

**Author** Mark van der Loo <mark.vanderloo@gmail.com>

**Maintainer** Mark van der Loo <mark.vanderloo@gmail.com>

**Depends** R (>= 2.8.0)

**Suggests** gWidgets, gWidgetstcltk

**License** GPL-2

**URL** <http://www.r-project.org>, <http://www.markvanderloo.eu>

**Repository** CRAN

**Date/Publication** 2011-01-03 17:24:59

## R topics documented:

|                         |          |
|-------------------------|----------|
| evGui . . . . .         | 2        |
| extremevalues . . . . . | 3        |
| getOutliers . . . . .   | 3        |
| invErf . . . . .        | 5        |
| outlierPlot . . . . .   | 6        |
| pareto . . . . .        | 7        |
| <b>Index</b>            | <b>9</b> |

---

evGui

*GUI to explore options and results of the "extremevalues" package*

---

## Description

Opens a Graphical User Interface and plots results. Options of the extremevalue package functions can be set and results are updated instantly. Includes a code generator button.

## Usage

```
evGui(y)
```

## Arguments

y                    A vector of type numeric

## Note

The GUI is programmed in a very quick and pretty dirty way, but it works fine. It will be replaced by a gtk-version in the future.

## Author(s)

Mark van der Loo

## References

[www.markvanderloo.eu](http://www.markvanderloo.eu)

## See Also

[getOutliers](#)

## Examples

```
## Not run:  
y <- rnorm(100)  
evGui(y)  
  
## End(Not run)
```

---

`extremevalues`*An R package for outlier detection*

---

### Description

This package offers outlier detection and plot functions for univariate data.

The package is the implementation of the outlier detection methods introduced in the reference below. Briefly, the methods work as follows. Using a subset of the data, the parameters for a model distribution are estimated using regression of the sorted data on their QQ-plot positions.

A value in the data is an outlier when it is unlikely to be drawn from the estimated distribution. There are two methods to determine the "unlikelyness". The first, called "Method I", determines the value above which less than  $\rho$  observations are expected, given the total number of observations in the data. Here  $\rho$  is a parameter which should have a value of 1 or less. The second notion of unlikelyness uses the fit residuals. Extremely large or small values are outliers when their residuals are above or below a confidence limit  $\alpha$ , to be determined by the user.

### References

M.P.J. van der Loo, Distribution based outlier detection for univariate data. Discussion paper 10003, Statistics Netherlands, The Hague (2010). Available from [www.markvanderloo.eu](http://www.markvanderloo.eu) or [www.cbs.nl](http://www.cbs.nl).

### See Also

[getOutliers](#), [outlierPlot](#)

---

`getOutliers`*Detect outliers*

---

### Description

`getOutliers` is a wrapper function for `getOutliersI` and `getOutliersII`.

### Usage

```
getOutliers(y, method="I", ...)  
getOutliersI(y, rho=c(1,1), FLim=c(0.1,0.9), distribution="normal")  
getOutliersII(y, alpha=c(0.05, 0.05), FLim=c(0.1, 0.9), distribution="normal", returnResiduals=TRUE)
```

**Arguments**

|                 |  |
|-----------------|--|
| y               | Vector of one-dimensional nonnegative data   |
| method          | "I" or "II"  |
| ...             | Optional arguments to be passed to getOutliersI or getOutliersII   |
| distribution    | Model distribution used to estimate the limit. Choose from "lognormal", "exponential", "pareto", "weibull" or "normal" (default).                              |
| FLim            | c(Fmin,Fmax) quantile limits indicating which data should be used to fit the model distribution. Must obey $0 < Fmin < Fmax < 1$ .                             |
| rho             | (Method I) A value $y_i$ is an outlier if it is below (above) the limit where less than rho[2] (rho[1]) observations are expected. Must be $>0$ .              |
| alpha           | (Method II) A value $y_i$ is an outlier if it has a residual below (above) the alpha[1] (alpha[2]) confidence limit for the residues. Must be between 0 and 1. |
| returnResiduals | (Method II) Whether or not to return a vector of residuals from the fit  |

**Details**

Both methods use the subset of  $y$ -values between the Fmin and Fmax quantiles to fit a model cumulative density distribution. **Method I** detects outliers by checking which are below (above) the limit where according to the model distribution less than rho[1] (rho[2]) observations are expected (given length(y) observations). **Method II** detects outliers by finding the observations (not used in the fit) who's fit residuals are below (above) the estimated confidence limit alpha[1] (alpha[2]) while all lower (higher) observations are outliers too.

**Value**

|              |  |
|--------------|--|
| nOut         | Number of left and right outliers.   |
| iLeft        | Index vector indicating left outliers in y   |
| iRight       | Index vector indicating right outliers in y  |
| limit        | For <b>Method I</b> : $y$ -values below (above) limit[1] (limit[2]) are outliers. For <b>Method II</b> : elements with residuals below (above) limit[1] (limit[2]) are outliers if all smaller (larger) elements are outliers as well. |
| method       | The used method: "method I" or "method II"   |
| distribution | The used model distribution  |
| Fmin         | FLim[1]  |
| Fmax         | FLim[2]  |
| yMin         | Smallest $y$ -value used in fit  |
| yMax         | Largest $y$ -value used in fit   |
| Nfit         | Number of values used in the fit   |
| rho          | <b>Method I</b> , the input rho-values for left and right outliers   |
| alphaConf    | <b>Method II</b> , the input confidence levels for left and right outliers   |

|        |   |
|--------|---|
| R2     | R-squared value for the fit. Note that this is the <i>ordinary least squares</i> value, defined by $R^2 = 1 - SS_{err}/SS_y$ . Where $SS_{err}$ is the squared sum of residuals. For the lognormal, Pareto and Weibull models, the $y$ -variable is transformed before fitting. Since predicted values are transformed back before calculating $SS_{err}$ , this $R^2$ can be negative. |
| lambda | (exponential distribution) Estimated location (and spread) parameter for $f(y) = \lambda \exp(-\lambda y)$  |
| mu     | (lognormal distribution) Estimated $E(\ln(y))$ for lognormal distribution   |
| sigma  | (lognormal distribution) Estimated $Var(\ln(y))$ for lognormal distribution   |
| ym     | (pareto distribution) Estimated location parameter (mode) for pareto distribution   |
| alpha  | (pareto distribution) Estimated spread parameter for pareto distribution  |
| k      | (weibull distribution) estimated shape parameter $k$ for weibull distribution   |
| lambda | (weibull distribution) estimated scale parameter $\lambda$ for weibull distribution   |
| mu     | (normal distribution) Estimated $E(y)$ for normal distribution  |
| sigma  | (normal distribution) Estimated $Var(y)$ for normal distribution  |

**Author(s)**

Mark van der Loo, see [www.markvanderloo.eu](http://www.markvanderloo.eu)

**References**

M.P.J. van der Loo, Distribution based outlier detection for univariate data. Discussion paper 10003, Statistics Netherlands, The Hague. Available from [www.markvanderloo.eu](http://www.markvanderloo.eu) or [www.cbs.nl](http://www.cbs.nl).

The file <your R directory>/R-<version>/library/extremevalues/extremevalues.pdf contains a worked example. It can also be downloaded from my website.

**Examples**

```
y <- rlnorm(100)
y <- c(0.1*min(y), y, 10*max(y))
K <- getOutliers(y, method="I", distribution="lognormal")
L <- getOutliers(y, method="II", distribution="lognormal")
par(mfrow=c(1,2))
outlierPlot(y,K, mode="qq")
outlierPlot(y,L, mode="residual")
```

---

 invErf

*Inverse error function*


---

**Description**

Inverse error function

**Usage**

```
invErf(x)
```

**Arguments**

x (Vector of) real value(s) in the range (-1,1)

**Value**

(vector of) value(s) of the inverse error function

**Author(s)**

Mark van der Loo, [www.markvanderloo.eu](http://www.markvanderloo.eu)

**Examples**

```
x <-seq(-0.99,0.99,0.01);
plot(x,invErf(x),'l');
```

---

outlierPlot

*Plot results of outlierdetection*

---

**Description**

This is a wrapper for two plot functions which can be used to analyse the results of outlier detection with the extremevalues package.

**Usage**

```
outlierPlot(y, L, mode="qq", ...)
qqFitPlot(y, L, title=NA, xlab=NA, ylab=NA, fat=FALSE)
plotMethodII(y, L, title=NA, xlab=NA, ylab=NA, fat=FALSE)
```

**Arguments**

|       |   |
|-------|---|
| y     | A vector of values  |
| L     | The result of <code>L &lt;- getOutliers(y,...)</code>   |
| mode  | Plot type. "qq" for Quantile-quantile plot with indicated outliers, "residual" for plot of fit residuals with indicated outliers (Method II only) |
| ...   | Optional arguments, to be transferred to qqFitPlot or plotMethodII (see below)  |
| title | A custom title (must be a string)   |
| xlab  | A custom label for the x-axis (must be a string)  |
| ylab  | A custom label for the y-axis (must be a string)  |
| fat   | If TRUE, axis, fonts, labels, points and lines are thicker for export and publication   |

## Details

Outliers are marked with a color or special symbol. If **mode="qq"**: observed against predicted y-values are plotted. Points between vertical lines were used in the fit. If `L$method="Method I"`, horizontal lines indicate the limits below (above) which observations are outliers. **mode="residuals"** only works when `L$Method="Method II"`. It generates a residual plot where points between two vertical lines were used in the fit. Horizontal lines indicate the computed confidence limits. The outermost points in the gray areas are outliers.

## Author(s)

Mark van der Loo, [www.markvanderloo.eu](http://www.markvanderloo.eu)

## References

The file `<your R directory>/R-<version>/library/extremevalues/extremevalues.pdf` contains a worked example. It can also be downloaded from my website.

## Examples

```
y <- rlnorm(100)
y <- c(0.1*min(y),y,10*max(y))
K <- getOutliers(y,method="I",distribution="lognormal")
L <- getOutliers(y,method="II",distribution="lognormal")
par(mfrow=c(1,2))
outlierPlot(y,K,mode="qq")
outlierPlot(y,L,mode="residual")
```

---

pareto

*Pareto distribution*

---

## Description

Pareto density distribution, quantile function and random generator.

## Usage

```
dpareto(x, xm=1, alpha=1)
qpareto(p, xm=1, alpha=1)
rpareto(n, xm=1, alpha=1)
```

## Arguments

|       |   |
|-------|---|
| xm    | location parameter (mode of distribution) |
| alpha | spread parameter                          |
| x     | Vector of realizations                    |
| p     | Vector of probabilities                   |
| n     | number of samples to draw                 |

**Value**

|         |   |
|---------|---|
| dpareto | Probability density                     |
| qpareto | Quantile at probability p (inverse cdf) |
| rpareto | Random value                            |

**Author(s)**

Mark van der Loo [www.markvanderloo.eu](http://www.markvanderloo.eu)

**Examples**

```
q <- qpareto(0.5);
```

# Index

## \*Topic **outliers**

- evGui, [2](#)
- dpareto (pareto), [7](#)
- evGui, [2](#)
- extremevalues, [3](#)
- extremevalues-package (extremevalues), [3](#)
- getOutliers, [2](#), [3](#), [3](#)
- getOutliersI (getOutliers), [3](#)
- getOutliersII (getOutliers), [3](#)
- invErf, [5](#)
- outlierPlot, [3](#), [6](#)
- pareto, [7](#)
- plotMethodII (outlierPlot), [6](#)
- qpareto (pareto), [7](#)
- qqFitPlot (outlierPlot), [6](#)
- rpareto (pareto), [7](#)