

# Package: outqrf (via r-universe)

January 9, 2025

**Type** Package

**Title** Find the Outlier by Quantile Random Forests

**Version** 1.0.0

**Date** 2024-08-27

**Author** Tengfei Xu [aut, cre]

**Maintainer** Tengfei Xu <flystar233@gmail.com>

**Description** Provides a method to find the outlier in custom data by quantile random forests method. Introduced by Meinshausen Nicolai (2006) <<https://dl.acm.org/doi/10.5555/1248547.1248582>>. It directly calls the ranger() function of the 'ranger' package to perform data fitting and prediction. We also implement the evaluation of outlier prediction results. Compared with random forest detection of outliers, this method has higher accuracy and stability on large datasets.

**License** MIT + file LICENSE

**Depends** R (>= 4.0.0)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** stats, ranger, dplyr, missRanger, ggpubr, ggplot2, tidyr

**URL** <https://github.com/flystar233/outqrf>

**BugReports** <https://github.com/flystar233/outqrf/issues>

**Suggests** renv, knitr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-09-10 09:10:02 UTC

**Config/pak/sysreqs** cmake make libicu-dev

## Contents

evaluateOutliers . . . . .	2
find_index . . . . .	3
generateOutliers . . . . .	3
get_quantily_value . . . . .	4
get_right_rank . . . . .	4
outqrf . . . . .	5
plot.outqrf . . . . .	6
<b>Index</b>	<b>7</b>

---

evaluateOutliers	<i>Evaluate Outliers</i>
------------------	--------------------------

---

### Description

This function evaluates the performance of the outlier detection algorithm.

### Usage

```
evaluateOutliers(original_data, anomaly_data, anomaly_result)
```

### Arguments

`original_data` A data frame containing the original data.  
`anomaly_data` A data frame containing the anomaly data.  
`anomaly_result` A data frame containing the predicted anomalies.

### Value

A data frame containing the evaluation metrics.

### Examples

```
anomaly_data <- generateOutliers(iris, p = 0.05, sd_factor = 5, seed = 123)
qrf <- outqrf(anomaly_data)
evaluateOutliers(iris, anomaly_data, qrf$outliers)
```

---

find_index	<i>find the closest index</i>
------------	-------------------------------

---

**Description**

This function finds the closest index to a given value in a vector.

**Usage**

```
find_index(x, y)
```

**Arguments**

x	a vector
y	a value

**Value**

the index of the closest value in the vector

**Examples**

```
find_index(c(1, 2, 3, 4, 5), 3.5)
```

---

generateOutliers	<i>Adds Outliers</i>
------------------	----------------------

---

**Description**

Adds Outliers

**Usage**

```
generateOutliers(data, p = 0.05, sd_factor = 5, seed = NULL)
```

**Arguments**

data	data.frame.
p	Proportion of outliers to add to data.
sd_factor	Each outlier is generated by shifting the original value by a realization of a normal random variable with sd_factor times the original sample standard deviation.
seed	An integer seed.

**Value**

data with some outliers.

**Examples**

```
generateOutliers(iris, p = 0.05, sd_factor = 5)
```

---

get\_quantily\_value     *get numeric value from string*

---

**Description**

This function extracts the numeric value from a string.

**Usage**

```
get_quantily_value(name)
```

**Arguments**

name                    a string

**Value**

a numeric value

**Examples**

```
get_quantily_value("quantiles = 0.001")
```

---

get\_right\_rank             *find the right rank*

---

**Description**

This function finds the right rank of a response value in a quantile random forest.

**Usage**

```
get_right_rank(response, outMatrix, median_outMatrix, rmse_)
```

**Arguments**

response                a vector of response values  
 outMatrix               a matrix of out values  
 median\_outMatrix        a vector of median out values  
 rmse\_                    a vector of rmse values

**Value**

a vector of ranks

---

outqrf	<i>find outliers</i>
--------	----------------------

---

**Description**

This function finds outliers in a dataset using quantile random forests.

**Usage**

```
outqrf(
  data,
  quantiles_type = 1000,
  threshold = 0.025,
  impute = TRUE,
  verbose = 1,
  weight = FALSE,
  ...
)
```

**Arguments**

data	a data frame
quantiles_type	'1000':seq(from = 0.001, to = 0.999, by = 0.001), '400':seq(0.0025,0.9975,0.0025)
threshold	a threshold for outlier detection
impute	a boolean value indicating whether to impute missing values
verbose	a boolean value indicating whether to print verbose output
weight	a boolean value indicating whether to use weight. if TRUE, The actual threshold will be threshold*r2.
...	additional arguments passed to the ranger function

**Value**

An object of class "outqrf" and a list with the following elements.

- **Data**: Original data set in unchanged row order
- **outliers**: Compact representation of outliers. Each row corresponds to an outlier and contains the following columns:
  - **row**: Row number of the outlier
  - **col**: Variable name of the outlier
  - **observed**: value of the outlier
  - **predicted**: predicted value of the outlier

- rank: Rank of the outlier
- outMatrix: Predicted value at different quantiles for each observation
- r.squared: R-squared value of the quantile random forest model
- outMatrix: Predicted value at different quantiles for each observation
- r.squared: R-squared value of the quantile random forest model
- oob.error: Out-of-bag error of the quantile random forest model
- rmse: RMSE of the quantile random forest model
- threshold: Threshold for outlier detection

### Examples

```
iris_with_outliers <- generateOutliers(iris, p=0.05)
qrf = outqrf(iris_with_outliers)
qrf$outliers
evaluateOutliers(iris,iris_with_outliers,qrf$outliers)
```

---

plot.outqrf

*Plots outqrf*

---

### Description

This function can plot paired boxplot of an "outqrf" object. It helps us to better observe the relationship between the original and predicted values

### Usage

```
## S3 method for class 'outqrf'
plot(x, ...)
```

### Arguments

x                    An object of class "outqrf".  
 ...                   other param maybe used.

### Value

A ggplot2 object

### Examples

```
irisWithOutliers <- generateOutliers(iris, seed = 2024)
qrf <- outqrf(irisWithOutliers)
plot(qrf)
```

# Index

`evaluateOutliers`, 2

`find_index`, 3

`generateOutliers`, 3

`get_quantily_value`, 4

`get_right_rank`, 4

`outqrf`, 5

`plot.outqrf`, 6