

# Package ‘rfinterval’

July 23, 2025

**Type** Package

**Title** Predictive Inference for Random Forests

**Version** 1.0.0

**Date** 2019-07-14

**Maintainer** Haozhe Zhang <haozhe.stat@gmail.com>

**Description** An integrated package for constructing random forest prediction intervals using a fast implementation package ‘ranger’. This package can apply the following three methods described in Haozhe Zhang, Joshua Zimmerman, Dan Nettleton, and Daniel J. Nordman (2019) <doi:10.1080/00031305.2019.1585288>: the out-of-bag prediction interval, the split conformal method, and the quantile regression forest.

**License** GPL-3

**Imports** ranger, MASS

**Depends** R (>= 3.1)

**URL** <http://github.com/haozhestat/rfinterval>

**BugReports** <http://github.com/haozhestat/rfinterval/issues>

**Suggests** testthat

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**Language** en-US

**NeedsCompilation** no

**Author** Haozhe Zhang [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-7771-4808>>)

**Repository** CRAN

**Date/Publication** 2019-07-18 16:40:04 UTC

## Contents

BeijingPM25 . . . . .	2
rfinterval . . . . .	3
sim_data . . . . .	4

---

BeijingPM25

*Beijing PM2.5 Air Pollution Data*

---

### Description

This hourly data set contains the PM2.5 data of US Embassy in Beijing. Meanwhile, meteorological data from Beijing Capital International Airport are also included.

### Usage

BeijingPM25

### Format

A data frame with 8661 rows and 11 variables:

**pm2.5** PM2.5 concentration (ug/m<sup>3</sup>)

**month** month of observation

**day** day of observation

**hour** hour of observation

**DEWP** dew point

**TEMP** temperature

**PRES** air pressure

**cbwd** combined wind direction

**Iws** cumulated wind speed

**Is** cumulated hours of snow

**Ir** cumulated hours of rain

### Source

Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H. and Chen, S. X. (2015). Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. Proceedings of the Royal Society A, 471, 20150257.

---

 rfinterval *Prediction Intervals for Random forests*


---

**Description**

The `rfinterval` constructs prediction intervals for random forest predictions using a fast implementation package 'ranger'.

**Usage**

```
rfinterval(formula = NULL, train_data = NULL, test_data = NULL,
           method = c("oob", "split-conformal", "quantreg"), alpha = 0.1,
           symmetry = TRUE, seed = NULL, params_ranger = NULL)
```

**Arguments**

<code>formula</code>	Object of class <code>formula</code> or character describing the model to fit. Interaction terms supported only for numerical variables.
<code>train_data</code>	Training data of class <code>data.frame</code> , <code>matrix</code> , or <code>dgCMatrix</code> (Matrix).
<code>test_data</code>	Test data of class <code>data.frame</code> , <code>matrix</code> , or <code>dgCMatrix</code> (Matrix).
<code>method</code>	Method for constructing prediction interval. If <code>method = "oob"</code> , compute the out-of-bag prediction intervals; if <code>method = "split-conformal"</code> , compute the split conformal prediction interval; if <code>method = "quantreg"</code> , use quantile regression forest to compute prediction intervals.
<code>alpha</code>	Confidence level. <code>alpha = 0.05</code> for the 95% prediction interval.
<code>symmetry</code>	True if constructing symmetric out-of-bag prediction intervals, False otherwise. Only for <code>method = "oob"</code>
<code>seed</code>	Seed (only for <code>method = "split-conformal"</code> )
<code>params_ranger</code>	List of further parameters that should be passed to <code>ranger</code> . See <a href="#">ranger</a> for possible parameters.

**Value**

<code>oob_interval</code>	Out-of-bag prediction intervals
<code>sc_interval</code>	Split-conformal prediction intervals
<code>quantreg_interval</code>	Quantile regression forest prediction intervals
<code>alpha</code>	Confidence level for prediction intervals
<code>testPred</code>	Random forest prediction for test set
<code>train_data</code>	Training data
<code>test_data</code>	Test data

## References

Haozhe Zhang, Joshua Zimmerman, Dan Nettleton, and Dan Nordman. (2019). "Random Forest Prediction Intervals." *The American Statistician*. Doi: 10.1080/00031305.2019.1585288.

Haozhe Zhang. (2019). "Topics in Functional Data Analysis and Machine Learning Predictive Inference." Ph.D. Dissertations. Iowa State University Digital Repository. 17929.

Lei, J., Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. "Distribution-free predictive inference for regression." *Journal of the American Statistical Association* 113, no. 523 (2018): 1094-1111.

Meinshausen, Nicolai. "Quantile regression forests." *Journal of Machine Learning Research* 7 (2006): 983-999.

Leo Breiman. (2001). Random Forests. *Machine Learning* 45(1), 5-32.

## Examples

```
train_data <- sim_data(n = 500, p = 8)
test_data <- sim_data(n = 500, p = 8)
output <- rfinterval(y~., train_data = train_data, test_data = test_data,
                    method = c("oob", "split-conformal", "quantreg"),
                    symmetry = TRUE, alpha = 0.1)
y <- test_data$y
mean(output$oob_interval$lo < y & output$oob_interval$up > y)
mean(output$sc_interval$lo < y & output$sc_interval$up > y)
mean(output$quantreg_interval$lo < y & output$quantreg_interval$up > y)
```

---

sim\_data

*Simulate data*

---

## Description

Simulate data for illustrate the performance of prediction intervals for random forests

## Usage

```
sim_data(n = 500, p = 10, rho = 0.6, predictor_dist = "correlated",
         mean_function = "nonlinear-interaction",
         error_dist = "homoscedastic")
```

## Arguments

n	Sample size
p	Number of features
rho	Correlation between predictors
predictor_dist	Distribution of predictor: "uncorrelated", and "correlated"
mean_function	Mean function: "linear", "nonlinear", and "nonlinear-interaction"
error_dist	Distribution of error: "homoscedastic", "heteroscedastic", and "heavy-tailed"

*sim\_data*

5

**Value**

a data.frame of simulated data

**Examples**

```
train_data <- sim_data(n = 500, p = 10)
test_data <- sim_data(n = 500, p = 10)
```

# Index

## \* datasets

BeijingPM25, [2](#)

BeijingPM25, [2](#)

ranger, [3](#)

rfinterval, [3](#)

sim\_data, [4](#)