Package ‘fuzzyreg’

June 6, 2019

Title  Fuzzy Linear Regression
Version 0.5.1
Date 2019-06-05
Author Pavel Skrabanek [aut, cph], Natalia Martinkova [aut, cre, cph]
Maintainer Natalia Martinkova <martinkova@ivb.cz>
Depends R (>= 3.4.0),
License GPL-3
Encoding UTF-8
LazyData true
Imports limSolve, quadprog, stats, graphics, grDevices
RoxygenNote 6.1.1
Suggests testthat, FuzzyNumbers
BuildVignettes true
NeedsCompilation no
Repository CRAN
Date/Publication 2019-06-06 07:50:14 UTC
**R topics documented:**

bats .................................................. 2
coef.fuzzylm ......................................... 3
dom .................................................... 4
flar ..................................................... 4
fis ...................................................... 5
fuzzify ................................................ 6
fuzzydat ............................................. 7
fuzzylm ............................................... 8
GOF ................................................... 10
moflr ................................................ 11
oplr .................................................. 12
plot.fuzzylm ......................................... 13
plr ................................................... 14
plrls ................................................ 15
predict.fuzzylm ..................................... 16
print.fuzzylm ........................................ 17
print.summary.fuzzylm .............................. 17
prodFuzzy ........................................... 18
prodSfuzzy ......................................... 19
sumFuzzy .......................................... 19
summary.fuzzylm .................................... 20
TEF .................................................. 21

**Index**

<table>
<thead>
<tr>
<th>bats</th>
<th>Temperature Data of Hibernating Bats and Climate at Site</th>
</tr>
</thead>
</table>

**Description**

Body surface temperature of multiple species of hibernating bats and mean annual surface temperature at the hibernation site.

**Usage**

data(bats)

**Format**

A data frame with 528 rows and two variables:

MAST numeric Mean annual surface temperature at the site in degrees Celsius
temperature numeric Body surface temperature of hibernating bats in degrees Celsius
**Source**


**Examples**

```r
data(bats)
# remove outlier
dat <- bats[!(bats$Mast < 0 & bats$temperature > 7), ]
# fuzzy linear regression model as published
fit <- fuzzylm(temperature ~ mastL data = dat, method = "plrls", h = 0.01, k1 = 5)
plot(fit, res = 30, col = "orange")
```

**coeff.fuzzylm**

*Extract Model Coefficients from Fuzzy Linear Model*

**Description**

Extracts coefficients of the fuzzy regression model in object `fuzzylm`.

**Usage**

```r
## S3 method for class 'fuzzylm'
coef(object, complete = TRUE, ...)
```

**Arguments**

- `object` a `fuzzylm` object.
- `complete` not used for a `fuzzylm` object.
- `...` other arguments.

**Value**

matrix with coefficients for the central tendency of the model, upper and lower boundary.

**Note**

The function returns real value numbers that define model predictions at $\mu_\beta(x) = 1$ and $\mu_\beta(x) > 0$, not triangular fuzzy numbers. To extract triangular fuzzy number coefficients of the model, use `object$coef`. 


Examples

data(fuzzydat)
f <- fuzzylm(y ~ x, data = fuzzydat$lee)
coef(f)

---

dom

Real Value Degree of Membership to a Triangular Fuzzy Number

Description

Calculates the degree of membership of a real number to a triangular fuzzy number. The fuzzy number is defined by its central value and the left and right spreads.

Usage

dom(x, TFN)

Arguments

x

a numeric vector.

TFN

a numeric vector of length 3.

Value

Returns a numeric in interval $[0, 1]$.

Examples

x <- seq(from = 0, to = 2, length.out = 10)
A <- c(1, 1, 1)
dom(x, A)

---

flar

Fuzzy Linear Regression using the Fuzzy Least Absolute Residual Method

Description

The function calculates fuzzy regression coefficients using the fuzzy least absolute residual (FLAR) method proposed by Zeng et al. (2017) for non-symmetric triangular fuzzy numbers.

Usage

flar(x, y)
Arguments

- **x**: matrix with the second to last columns representing independent variable observations. The first column is related to the intercept, so it consists of ones. Missing values not allowed.

- **y**: matrix of dependent variable observations. The first column contains the central tendency, the second column the left spread and the third column the right spread of non-symmetric triangular fuzzy numbers. Missing values not allowed.

Details

The FLAR method expects real value input for the explanatory variables, and non-symmetric triangular fuzzy numbers for the response variable. The prediction returns non-symmetric triangular fuzzy numbers.

Value

Returns a `fuzzylm` object that includes the model coefficients, limits for data predictions from the model and the input data.

Note

Preferred use is through the `fuzzylm` wrapper function with argument `method = "flar"`.

References


See Also

- `fuzzylm`

Examples

```r
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$dia, "flar", , "yl", "yl")
```

Description

Fuzzy Linear Regression using the Fuzzy Least Squares Method

Usage

`f1s(x, y)`
fuzzify

Arguments

x       two column matrix with the second column representing independent variable
        observations. The first column is related to the intercept, so it consists of ones.
        Missing values not allowed.

y       matrix of dependent variable observations. The first column contains the central
        tendency, the second column the left spread and the third column the right spread
        of non-symmetric triangular fuzzy numbers. Missing values not allowed.

Details

The FLS method for the fuzzy linear regression fits a simple model.

Value

Returns a fuzzylm object that includes the model coefficients, limits for data predictions from the
model and the input data.

Note

Preferred use is through the fuzzylm wrapper function with argument method = "fls".

References


See Also

fuzzylm

Examples

data(fuzzydat)
x <- fuzzydat$dia[, 1, drop = FALSE]
x <- cbind(rep(1, nrow(x)), x)
y <- fuzzydat$dia[, c(2,3,3)]
fls(x = x, y = y)

fuzzify                   Convert Real Value Numbers to Triangular Fuzzy Numbers

Description

Uses naive alternative methods to approximate triangular fuzzy numbers from real value number
input data.

Usage

fuzzify(x, y = NULL, method = "mean", err = 0, dimnames = list("x",
           "y"), ...)
Arguments

- **x**: numeric vector.
- **y**: numeric vector. Error term for the error method.
- **method**: character vector specifying the conversion method. See Details.
- **dimnames**: list of length 2 giving names of the `x` and `y` variables.
- **...**: additional parameters passed to other functions.

Details

Converts crisp numbers in `x` to a triangular fuzzy number (TFN). Optionally, values in `y` can be used as grouping elements and are coerced to a factor.

Method `mean` calculates the central value of a TFN as the mean of `x` given `y`, and the left and right spreads as standard deviations.

Method `median` gives the central values as a median and left and right spreads are calculated as distance of the first and third quartile from the median.

Method `zero` inserts zeros to both spreads.

Method `error` uses a user-defined numeric value or vector for the spreads. The length of the numeric vector in argument `err` must be in (1, length(x), 2 * length(x)).

Value

A data.frame with columns representing the central value, left and right spread of `x` and the values in `y` coerced to a factor. Attempt is made to inherit names from the input data. Methods `mean` and `zero` will return symmetric TFNs, whereas methods `median` and `error` can return non-symmetric TFNs depending on input data and the data or the values in the `err` argument.

Examples

```r
fuzzify(1:5)
fuzzify(1:6, c(1,1,1,2,2), method = "err", err = runif(6) * 1e-3)
```

---

**fuzzydat**

**Data For Fuzzy Linear Regression**

**Description**

Example data reported by the authors of the respective fuzzy linear regression methods for testing model fit performance.

**Usage**

```r
data(fuzzydat)
```
fuzzylm

Fuzzy Linear Regression

Description

A wrapper function that calculates fuzzy regression coefficients using a chosen method.

Usage

```r
fuzzylm(formula, data, method = "plrls", fuzzy.left.x = NULL,
         fuzzy.right.x = NULL, fuzzy.left.y = NULL, fuzzy.right.y = NULL,
         ...)
```

Arguments

- `formula`: a model formula.
- `data`: a data.frame, containing the variables in formula.
- `method`: method for fitting of the fuzzy linear model.
- `fuzzy.left.x`: character string vector specifying column name(s) with the left spread of the fuzzy independent variable(s).
- `fuzzy.right.x`: character string vector specifying column name(s) with the right spread of the fuzzy independent variable(s).
- `fuzzy.left.y`: character string vector specifying column name(s) with the left spread of the fuzzy dependent variable.
fuzzylm

fuzzy.right.y  character string vector specifying column name(s) with the right spread of the
fuzzy dependent variable.

...  additional parameters used by specific methods.

Details

The implemented methods include plrls for fitting the fuzzy linear regression from the crisp input
data (Lee and Tanaka 1999), and fls (Diamond 1988), oplr (Hung and Yang 2006), moflr

Value

Returns a fuzzylm object that includes the model coefficients, limits for data predictions from the
model and the input data.

References


See Also

plot, predict, summary

Examples

data(fuzzydat)
fuzzylm(y ~ x, data = fuzzydat$lee, method = "plrls")
## Not run:
# returns an error due to the incorrect number of spreads
fuzzylm(y ~ x, data = fuzzydat$dia, method = "fls", fuzzy.left.y = "yl")
## End(Not run)
# use the same column name for the left and right spread, when the method requests
# non-symmetric fuzzy numbers, but the data specify symmetric fuzzy numbers
fuzzylm(y ~ x, data = fuzzydat$dia, method = "fls", fuzzy.left.y = "yl", fuzzy.right.y = "yl")
**GOF**

**Goodness of Fit of Fuzzy Regression Model**

**Description**

Calculates mean error rate based on Diamond’s distance of two variables representing triangular fuzzy numbers, where one is the response variable and the other is the prediction from a fuzzy regression model.

**Usage**

`GOF(object, sc = 1e-06)`

**Arguments**

- `object` a `fuzzylm` object.
- `sc` scaling constant used for numerical stability when spreads are equal to zero.

**Details**

The Diamond’s distance of two triangular fuzzy numbers is sum of squared differences of the core and both support values of the fuzzy numbers.

**Value**

A numeric.

**References**


**See Also**

`fuzzylm`

**Examples**

```r
data(bats)
f <- fuzzylm(temperature ~ MAST, data = bats)
GOF(f)
```
Description

This function calculates fuzzy regression coefficients using the multi-objective fuzzy linear regression (MOFLR) method developed by Nasrabadi et al. (2005) that combines the least squares approach (fitting of a central tendency) with the possibilistic approach (fitting of spreads) when approximating an observed linear dependence by a fuzzy linear model.

Usage

moflr(x, y, omega = 0.5, sc = 1e-05)

Arguments

- **x**: matrix of \( n \) independent variable values, followed by \( n \) spreads. First column is expected to consist of ones, representing intercept. Missing values not allowed.
- **y**: two column matrix of dependent variable values and the respective spread. Method assumes symmetric triangular fuzzy input, so the second spread (if present) is ignored. Missing values not allowed.
- **omega**: a scalar that specifies weight that determines trade-off of between outliers penalization and data fitting in interval \([0, 1]\), where high values of omega decrease the penalization of outliers.
- **sc**: scaling constant used to input random spreads for the intercept, necessary for computational stability.

Details

The function input expects both the response and the predictors in form of symmetric fuzzy numbers. The prediction returns symmetric triangular fuzzy number coefficients. The Nasrabadi et al.’s method can process datasets with multiple outliers. Values \( \omega > 0.5 \) decrease weight of outliers on the solution.

Value

Returns a \texttt{fuzzylm} object that includes the model coefficients, limits for data predictions from the model and the input data.

Note

Preferred use is through the \texttt{fuzzylm} wrapper function with argument \texttt{method = "moflr"}.

References

See Also

fuzzylm

Examples

data(fuzzydat)
fuzzylm(y~x, fuzzydat$nas, "moflr", "xl", "yl")

Description

The function calculates fuzzy regression coefficients using the possibilistic linear regression with
an outlier omission approach method (OPLR) developed by Hung and Yang (2006) that combines
the least squares approach (fitting of a central tendency) with the possibilistic approach (fitting of
spreads) when approximating an observed linear dependence by a fuzzy linear model.

Usage

oplr(x, y, h = 0)

Arguments

x matrix with the independent variables observations. The first column is related
to the intercept, so it consists of ones. Missing values not allowed.
y two column matrix of the dependent variable values and the respective spread.
Method assumes symmetric triangular fuzzy input, so the second spread (if
present) is ignored. Missing values not allowed.
h a scalar value in interval [0, 1], specifying the h-level.

Details

The function input expects symmetric fuzzy response and crisp predictors. The prediction returns
symmetric triangular fuzzy number coefficients. The OPLR method can detect one outlier in the
data that is farther than $1.5 \times$ IQR from either quartile.
The h-level is a degree of fitting chosen by the decision maker.

Value

Returns a fuzzylm object that includes the model coefficients, limits for data predictions from the
model and the input data.

Note

Preferred use is through the fuzzylm wrapper function with argument method = "oplr".
References


See Also

fuzzylm

Examples

```r
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$hun, "oplr", , , "yl")
```

## S3 method for class 'fuzzylm'

```r
plot(x, y = NULL, which = 1, res = 2,
     col.fuzzy = NA, length = 0.05, angle = 90, main = "method",
     xlab = NULL, ylab = NULL, ...)
```

### Arguments

- `x`: a fuzzylm object.
- `y`: NULL for plotting a fuzzylm object.
- `which`: an integer or character string specifying which explanatory variable to plot.
- `res`: an integer >= 2 specifying resolution of shading for the regression plot. Minimum resolution for shading the plot is 3.
- `col.fuzzy`: color for shading of the regression plot.
- `length`: length of the edges of the arrow head (in inches).
- `angle`: angle from the shaft of the arrow to the edge of the arrow head.
- `main`: a main title for the plot. Default title specifies method used to fit the model.
- `xlab`: a label for the x axis, defaults to a description of x.
- `ylab`: a label for the y axis, defaults to a description of y.
- `...`: additional graphical parameters.
Details
Silently plots the data. Fuzzy numbers are plotted with points for the central value and arrows specifying spreads.

Examples
```r
data(fuzzydat)
f = fuzzylm(y ~ x, fuzzydat$lee)
plot(f)
plot(f, res = 20, col.fuzzy = "red")
```

Description
The function calculates fuzzy regression coefficients using the possibilistic linear regression method (PLR) developed by Tanaka et al. (1989). Specifically, the min problem is implemented in this function.

Usage
```
plr(x, y, h = 0)
```

Arguments
- `x` matrix of \( n \) independent variable observations. The first column is related to the intercept, so it consists of ones. Missing values not allowed.
- `y` two column matrix of dependent variable values and the respective spread. Method assumes symmetric triangular fuzzy input, so the second spread (if present) is ignored. Missing values not allowed.
- `h` a scalar value in interval \([0, 1]\), specifying the h-level.

Details
The function input expects the response in form of a symmetric fuzzy number and the predictors as crisp numbers. The prediction returns symmetric triangular fuzzy number coefficients. The h-level is a degree of fitting chosen by the decision maker.

Value
Returns a fuzzylm object that includes the model coefficients, limits for data predictions from the model and the input data.

Note
Preferred use is through the fuzzylm wrapper function with argument `method = "plr"`. 
References

See Also
fuzzylm

Examples
```r
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$tan, "plr", , "yl", "yr")
```

---

**plrls**  
*Fuzzy Linear Regression using the Possibilistic Linear Regression with Least Squares Method*

Description
The function calculates fuzzy regression coefficients using the possibilistic linear regression with least squares method developed by Lee and Tanaka (1999) that combines the least squares approach (fitting of a central tendency) with the possibilistic approach (fitting of spreads) when approximating an observed linear dependence by a fuzzy linear model.

Usage
```r
plrls(x, y, h = 0, k1 = 1, k2 = 1, epsilon = 1e-05)
```

Arguments
- **x**: two column matrix with the second column representing independent variable observations. The first column is related to the intercept, so it consists of ones. Missing values not allowed.
- **y**: one column matrix of dependent variable values, missing values not allowed.
- **h**: a scalar value in interval [0, 1], specifying the h-level.
- **k1**: weight coefficient for the central tendency.
- **k2**: weight coefficient for the spreads.
- **epsilon**: small positive number that supports search for the optimal solution.

Details
The function input expects crisp numbers of both the explanatory and response variables, and the prediction returns non-symmetric triangular fuzzy number coefficients.

The h-level is a degree of fitting chosen by the decision maker.
Value

Returns a fuzzylm object that includes the model coefficients, limits for data predictions from the model and the input data.

Note

Preferred use is through the fuzzylm wrapper function with argument method = "plrls".

References


See Also

fuzzylm

Examples

```r
x <- matrix(c(rep(1, 15), rep(1:3, each = 5)), ncol = 2)
y <- matrix(c(rnorm(5, 1), rnorm(5, 2), rnorm(5, 3)), ncol = 1)
plrls(x = x, y = y)
```

---

### predict.fuzzylm

*Predict Method for Fuzzy Linear Regression*

**Description**

Predicts the central tendency and spreads from a fuzzy linear regression model.

**Usage**

```r
## S3 method for class 'fuzzylm'
predict(object, newdata, ...)
```

**Arguments**

- `object` a fuzzylm object.
- `newdata` an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
- `...` further arguments passed to or from other methods.

**Value**

fuzzylm object with newdata replacing the slot x and predictions in triangular fuzzy number format representing the central values and left and right spreads replacing the slot y.
Examples

data(fuzzydat)
f <- fuzzylm(y ~ x, data = fuzzydat$lee)
predict(f)

print.fuzzylm

Prints Fuzzy Linear Regression Result

Description

Prints the call and coefficients from the fuzzylm object.

Usage

## S3 method for class 'fuzzylm'
print(x, ...)

Arguments

x  a fuzzylm object.
...

further arguments passed to or from other methods.

Examples

x <- rep(1:3, each = 5)
y <- c(rnorm(5, 1), rnorm(5, 2), rnorm(5, 3))
dat <- data.frame(x = x, y = y)
f <- fuzzylm(y ~ x, dat)
f

print.summary.fuzzylm

Prints Fuzzy Linear Regression Summary

Description

Prints the models for the central tendency and spreads from the fuzzylm object.

Usage

## S3 method for class 'summary.fuzzylm'
print(x, ...)

Arguments

x  a summary of a fuzzylm object.
...

further arguments passed to or from other methods.
Examples

```r
x <- rep(1:3, each = 5)
y <- c(rnorm(5, 1), rnorm(5, 2), rnorm(5, 3))
dat <- data.frame(x = x, y = y)
f <- fuzzylm(y ~ x, dat)
sum.f <- summary(f)
sum.f
```

---

**prodFuzzy**

*Product of Two Triangular Fuzzy Numbers*

Description

Calculates product of two triangular fuzzy numbers defined as a central value, left and right spread.

Usage

```r
prodFuzzy(x, y)
```

Arguments

- `x`:
  - a numeric vector of length three, specifying a triangular fuzzy number as its central value, left and right spread.

- `y`:
  - a numeric vector of length three, specifying a triangular fuzzy number as its central value, left and right spread.

Value

Returns a numeric vector, representing a triangular fuzzy number.

Examples

```r
x <- c(1, 0.2, 0.2)
y <- c(2, 0.2, 0.2)
prodFuzzy(x = x, y = y)
```
**prodSfuzzy**

*Product of a Scalar and a Triangular Fuzzy Number*

**Description**
Calculates product of a real number scalar and a triangular fuzzy number defined as a central value, left and right spread.

**Usage**

```r
prodSfuzzy(x, y)
```

**Arguments**
- `x` numeric vector of length one.
- `y` a numeric vector of length three, specifying a triangular fuzzy number as its central value, left and right spread.

**Details**
Note that if `x < 0` the left and right spread will be reversed.

**Value**
Returns a numeric vector, representing a triangular fuzzy number.

**Examples**

```r
x <- 2
y <- c(2, 0.2, 0.2)
prodSfuzzy(x = x, y = y)
x <- -2
prodSfuzzy(x = x, y = y)
```

---

**sumFuzzy**

*Sum of Two Triangular Fuzzy Numbers*

**Description**
Calculates a sum of two triangular fuzzy numbers defined as a central value, left and right spread.

**Usage**

```r
sumFuzzy(x, y)
```
Arguments

- **x**: a numeric vector of length three, specifying a triangular fuzzy number as its central value, left and right spread.
- **y**: a numeric vector of length three, specifying a triangular fuzzy number as its central value, left and right spread.

Value

Returns a numeric vector, representing a triangular fuzzy number.

Examples

```r
x <- c(1, 0.1, 0.2)
y <- c(2, 0.2, 0.2)
sumFuzzy(x = x, y = y)
```

---

**summary.fuzzylm**

**Summarizes Fuzzy Linear Regression**

Description

Calculates the summary from the fuzzylm object.

Usage

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

Arguments

- **object**: a fuzzylm object.
- **...**: additional parameters passed to and from other methods.

Value

Returns a list with models for the central tendency and spreads from the fuzzy linear regression, total error of fit of the model and a goodness-of-fit measure.

See Also

TEF, GOF

Examples

```r
data(fuzzydat)
f <- fuzzylm(y ~ x, fuzzydat$lee)
sum.f <- summary(f)
sum.f
```
Description

Calculates total error of fit of a fuzzy regression model based on the concept of difference in membership functions of triangular fuzzy numbers between the estimated and observed fuzzy dependent variables.

Usage

```r
TEF(object, sc = 1e-06, ...)
```

Arguments

- `object` a `fuzzylm` object.
- `sc` scaling constant used for numerical stability when spreads are equal to zero.
- `...` additional arguments passed to the `integrate` function.

Details

Calculates \( \sum E \), where \( E \) is the difference in membership functions between two triangular fuzzy numbers. Here, between the observation and the prediction from a fuzzy regression model `fuzzylm`.

Value

A numeric with sum of pairwise differences between the triangular fuzzy numbers.

Note

`TEF` is not suitable for assessing fuzzy linear regression models that were fitted from crisp input data. Such data will result in division by zero. The scaling constant `sc` numerically allows the calculation to proceed, but it is not advisable. Use `GOF` instead.

References


See Also

`fuzzylm`, `GOF`

Examples

```r
data(fuzzydat)
f <- fuzzylm(y ~ x, fuzzydat$lee)
TEF(f)
```
Index

*Topic **datasets**
  bats, 2
  fuzzydat, 7
*Topic **fuzzy**
  flar, 4
  fls, 5
  moflr, 11
  oplr, 12
  plot.fuzzylm, 13
  plr, 14
  plrls, 15
  predict.fuzzylm, 16
  print.fuzzylm, 17
  print.summary.fuzzylm, 17
  summary.fuzzylm, 20
  bats, 2
  coef.fuzzylm, 3
  dom, 4
  flar, 4
  fls, 5, 9
  fuzzify, 6
  fuzzydat, 7
  fuzzylm, 5, 6, 8, 10–16, 21
  GOF, 10, 20, 21
  moflr, 9, 11
  oplr, 9, 12
  plot, 9
  plot.fuzzylm, 13
  plr, 9, 14
  plrls, 9, 15
  predict, 9
  predict.fuzzylm, 16
  print.fuzzylm, 17
  print.summary.fuzzylm, 17
  prodFuzzy, 18
  prodSfuzzy, 19
  sumFuzzy, 19
  summary, 9
  summary.fuzzylm, 20
  TEF, 20, 21

22