

# 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>

**Description** Estimators for fuzzy linear regression. The functions estimate parameters of fuzzy linear regression models with crisp or fuzzy independent variables (triangular fuzzy numbers are supported). Implements multiple methods for parameter estimation and algebraic operations with triangular fuzzy numbers. Includes functions for summarising, printing and plotting the model fit. Calculates predictions from the model and total error of fit.

Diamond (1988) <doi:10.1016/0020-0255(88)90047-3>,

Hung & Yang (2006) <doi:10.1016/j.fss.2006.08.004>,

Lee & Tanaka (1999) <doi:10.15807/jorsj.42.98>,

Nasrabadi, Nasrabadi & Nasrabad (2005) <doi:10.1016/j.amc.2004.02.008>,

Tanaka, Hayashi & Watada (1989) <doi:10.1016/0377-2217(89)90431-1>,

Zeng, Feng & Li (2017) <doi:10.1016/j.asoc.2016.09.029>.

**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
fls	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
<b>Index</b>	<b>22</b>

---

bats

*Temperature Data of Hibernating Bats and Climate at Site*


---

### 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**

Martinkova, N., Pikula, J., Zukal, J., Kovacova, V., Bandouchova, H., Bartonicka, T., Botvinkin, A.D., Brichta, J., Dundarova, H., Kokurewicz, T., Irwin, N.R., Linhart, P., Orlov, O.L., Piacek, V., Skrabanek, P., Tiunov, M.P. and Zahradnikova, A., Jr. (2018) Hibernation temperature-dependent *Pseudogymnoascus destructans* infection intensity in Palearctic bats. *Virulence* 9: 1734-1750.

Hijmans, R.J., Cameron, S.E., Parra, J.L., Jones, P.G. and Jarvis, A. (2005) Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology* 25: 1965-1978.

**Examples**

```
data(bats)
# remove outlier
dat <- bats[!(bats$MAST < 0 & bats$temperature > 7), ]
# fuzzy linear regression model as published
fit <- fuzzylm(temperature ~ MAST, data = dat, method = "plr1s", h = 0.01, k1 = 5)
plot(fit, res = 30, col = "orange")
```

coef.fuzzylm

*Extract Model Coefficients from Fuzzy Linear Model***Description**

Extracts coefficients of the fuzzy regression model in object fuzzylm.

**Usage**

```
## 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  $\mu_{\bar{y}}(x) = 1$  and  $\mu_{\bar{y}}(x) > 0$ , 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	<i>Real Value Degree of Membership to a Triangular Fuzzy Number</i>
-----	---

---

**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	<i>Fuzzy Linear Regression using the Fuzzy Least Absolute Residual Method</i>
------	---

---

**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**

Zeng, W., Feng, Q. and Li, J. (2017) Fuzzy least absolute linear regression. *Applied Soft Computing* 52: 1009-1019.

**See Also**

[fuzzylm](#)

**Examples**

```
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$dia, "flar", , , "y1", "y1")
```

---

fls

*Fuzzy Linear Regression using the Fuzzy Least Squares Method*


---

**Description**

The function calculates fuzzy regression coefficients using the fuzzy least squares (FLS) method proposed by Diamond (1988) for non-symmetric triangular fuzzy numbers.

**Usage**

```
fls(x, y)
```

**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**

Diamond, P. (1988) Fuzzy least squares. *Information Sciences* 46(3): 141-157.

**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**

<code>x</code>	numeric vector.
<code>y</code>	vector that can be coerced to factor (optional).
<code>method</code>	character vector specifying the conversion method. See Details.
<code>err</code>	numeric vector. Error term for the error method.
<code>dimnames</code>	list of length 2 giving names of the <code>x</code> and <code>y</code> variables.
<code>...</code>	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  $(1, \text{length}(x), 2 * \text{length}(x))$ .

**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**

```
fuzzify(1:5)
fuzzify(1:6, c(1,1,1,2,2,2), 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**

```
data(fuzzydat)
```

**Format**

A list of data.frames.

**Source**

Diamond, P. (1988) Fuzzy least squares. *Information Sciences* 46(3): 141-157.

Hung, W.-L. and Yang, M.-S. (2006) An omission approach for detecting outliers in fuzzy regression models. *Fuzzy Sets and Systems* 157: 3109-3122.

Lee, H. and Tanaka, H. (1999) Fuzzy approximations with non-symmetric fuzzy parameters in fuzzy regression analysis. *Journal of the Operations Research Society Japan* 42: 98-112.

Nasrabadi, M. M., Nasrabadi, E. and Nasrabad, A. R. (2005) Fuzzy linear regression analysis: a multi-objective programming approach. *Applied Mathematics and Computation* 163: 245-251.

Tanaka H., Hayashi I. and Watada J. (1989) Possibilistic linear regression analysis for fuzzy data. *European Journal of Operational Research* 40: 389-396.

**Examples**

```
data(fuzzydat)
fuzzylm(y ~ x, data = fuzzydat$lee)
fuzzylm(y ~ x, data = fuzzydat$dia, method = "fls", fuzzy.left.y = "y1", fuzzy.right.y = "y1")
```

---

fuzzylm

*Fuzzy Linear Regression*

---

**Description**

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

**Usage**

```
fuzzylm(formula, data, method = "plr1s", 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.



`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` (Nasrabadi et al. 2005) and `plr` (Tanaka et al. 1989) methods for triangular fuzzy numbers.

### Value

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

### References

- Diamond, P. (1988) Fuzzy least squares. *Information Sciences* 46(3): 141-157.
- Hung, W.-L. and Yang, M.-S. (2006) An omission approach for detecting outliers in fuzzy regression models. *Fuzzy Sets and Systems* 157: 3109-3122.
- Lee, H. and Tanaka, H. (1999) Fuzzy approximations with non-symmetric fuzzy parameters in fuzzy regression analysis. *Journal of the Operations Research Society Japan* 42: 98-112.
- Nasrabadi, M. M., Nasrabadi, E. and Nasrabady, A. R. (2005) Fuzzy linear regression analysis: a multi-objective programming approach. *Applied Mathematics and Computation* 163: 245-251.
- Tanaka H., Hayashi I. and Watada J. (1989) Possibilistic linear regression analysis for fuzzy data. *European Journal of Operational Research* 40: 389-396.
- Zeng, W., Feng, Q. and Li, J. (2017) Fuzzy least absolute linear regression. *Applied Soft Computing* 52: 1009-1019.

### 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 = "y1")
## 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 = "y1", fuzzy.right.y = "y1")
```

---

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 <code>fuzzylm</code> 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

Diamond P. (1988) Fuzzy least squares. *Information Sciences* 46(3): 141-157.  
Wang N., Zhang W.-X., Mei C.-L. (2007) Fuzzy nonparametric regression based on local linear smoothing technique. *Information Sciences* 177: 3882-3900.

### See Also

[fuzzylm](#)

### Examples

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

---

moflr	<i>Fuzzy Linear Regression Using the Multi-Objective Fuzzy Linear Regression Method</i>
-------	---

---

### 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 `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 = "moflr"`.

### References

Nasrabadi, M. M., Nasrabadi, E. and Nasrabad, A. R. (2005) Fuzzy linear regression analysis: a multi-objective programming approach. *Applied Mathematics and Computation* 163: 245-251.

**See Also**[fuzzylm](#)**Examples**

```
data(fuzzydat)
fuzzylm(y~x, fuzzydat$nas, "moflr", "x1", , "y1")
```

oplr

---

*Fuzzy Linear Regression Using the Possibilistic Linear Regression with Omission Method*

---

**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  $1.5 * IQR$  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**

Hung, W.-L. and Yang, M.-S. (2006) An omission approach for detecting outliers in fuzzy regression models. *Fuzzy Sets and Systems* 157: 3109-3122.

**See Also**

[fuzzylm](#)

**Examples**

```
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$hun, "oplr", , , "y1")
```

---

plot.fuzzylm

*Plot Fuzzy Linear Regression*

---

**Description**

Plots the data and the central tendency with spreads of a fuzzy linear regression. For multiple regression, allows choice of which variable to display. Optionally colors the polygon for the regression.

**Usage**

```
## S3 method for class 'fuzzylm'
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 $\geq 2$ 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**

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

---

 plr

*Fuzzy Linear Regression Using the Possibilistic Linear Regression Method*

---

**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**

Tanaka H., Hayashi I. and Watada J. (1989) Possibilistic linear regression analysis for fuzzy data. *European Journal of Operational Research* 40: 389-396.

**See Also**

[fuzzylm](#)

**Examples**

```
data(fuzzydat)
fuzzylm(y ~ x, fuzzydat$tan, "plr", , , "y1", "yr")
```

---

plrIs	<i>Fuzzy Linear Regression using the Possibilistic Linear Regression with Least Squares Method</i>
-------	--

---

**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**

```
plrIs(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 = "plr1s"`.

**References**

Lee, H. and Tanaka, H. (1999) Fuzzy approximations with non-symmetric fuzzy parameters in fuzzy regression analysis. *Journal of the Operations Research Society Japan* 42: 98-112.

**See Also**

[fuzzylm](#)

**Examples**

```
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)
plr1s(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**

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

**Arguments**

<code>object</code>	a <code>fuzzylm</code> object.
<code>newdata</code>	an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
<code>...</code>	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**

```
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**

```
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**

```
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**

```
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**

```
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**

```
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**

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

---

summary.fuzzyIm      *Summarizes Fuzzy Linear Regression*

---

**Description**

Calculates the summary from the fuzzyIm object.

**Usage**

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

**Arguments**

- `object` a fuzzyIm 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**

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

---

TEF	<i>Total Error of Fit of Fuzzy Regression Model</i>
-----	---

---

**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**

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

**Arguments**

object	a <code>fuzzylm</code> object.
sc	scaling constant used for numerical stability when spreads are equal to zero.
...	additional arguments passed to the <code>integrate</code> 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**

Kim B. and Bishu R. R. (1998) Evaluation of fuzzy linear regression models by comparing membership functions. *Fuzzy Sets and Systems* 100: 343-352.

**See Also**

[fuzzylm](#), [GOF](#)

**Examples**

```
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](#)  
plrIs, [15](#)  
predict.fuzzylm, [16](#)  
print.fuzzylm, [17](#)  
print.summary.fuzzylm, [17](#)  
summary.fuzzylm, [20](#)

print.summary.fuzzylm, [17](#)  
prodFuzzy, [18](#)  
prodSfuzzy, [19](#)

sumFuzzy, [19](#)  
summary, [9](#)  
summary.fuzzylm, [20](#)

TEF, [20](#), [21](#)

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](#)  
plrIs, [9](#), [15](#)  
predict, [9](#)  
predict.fuzzylm, [16](#)  
print.fuzzylm, [17](#)