



Calculator for fuzzy numbers

Abbas Parchami¹

Received: 26 February 2017 / Accepted: 4 February 2019 / Published online: 21 March 2019
© The Author(s) 2019

Abstract

Usually, the statistical estimators (or mathematical functions) are the base of scientific decision making. In applied situations, at least one of the parameters or variables of the decision function may be fuzzy valued, instead of real valued. In such vague situations, one way to perform the calculations is using extension principle approach which has a complex form. Recently, two software packages have been freely available to perform some facilities in calculation and computation based on fuzzy numbers. In other words, these two software packages have the role of the first *calculator* on fuzzy numbers. This paper discussed and compared two software packages “FuzzyNumbers” and “Calculator.LR.FNs” which were recently published on CRAN by Gagolewski and Parchami, respectively. These packages have the ability of installation on R software, and in fact they propose some useful instruments and functions to the users for drawing and easily using arithmetic operators on the set of fuzzy numbers. For the convenience of the readers, the proposed methods and functions have been presented with several numerical examples to help in better understanding.

Keywords R package · LR fuzzy number · Arithmetic operators · Extension principle

A brief introduction to R

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R foundation for statistical computing. The R language is widely used among statisticians for developing their statistical computations which show that R’s popularity has increased substantially in recent years. The source code for the R software environment is written primarily in C, Fortran, and R. R and its packages are freely available under the GNU general public license, and pre-compiled binary versions are provided for various operating systems [18].

Available fuzzy packages on CRAN

The capabilities of R are extended through user-created packages, which allow specialized statistical techniques,

graphical devices, import/export capabilities, reporting tools, etc. These packages are developed primarily in R, and sometimes in Fortran, C, C++, and Java. A core set of packages is included with the installation of R, with more than 10,162 additional packages (till February 2017) available at the Comprehensive R Archive Network (CRAN). The “Task Views” page (subject list) on the CRAN website lists a wide range of tasks (in fields such as agriculture, finance, genetics, high-performance computing, machine learning, data mining, medical imaging, social sciences, mathematics and spatial statistics) to which R has been applied and for which packages are available [18]. Among the listed packages on CRAN, there exists 36 fuzzy packages till now. These packages cover a wide range of tasks and are listed by name below:

AnalyzeTS (Analyze Fuzzy Time Series)
anfis (Adaptive Neuro Fuzzy Inference System in R)
Calculator.LR.FNs (Calculator for LR Fuzzy Numbers)
coppeCosenzaR (COPPE-Cosenza Fuzzy Hierarchy Model)
fclust (Fuzzy Clustering)
fcm (Inference of Fuzzy Cognitive Maps (FCMs))
FCMapper (Fuzzy Cognitive Mapping)
FLR (Fuzzy Logic Rule Classifier)

✉ Abbas Parchami
parchami@uk.ac.ir

¹ Department of Statistics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

frbs (Fuzzy Rule-Based Systems for Classification and Regression Tasks)
fso (Fuzzy Set Ordination)
fugeR (Fuzzy Genetic, a machine learning algorithm to construct prediction model based on fuzzy logic)
fuzzyFDR (Exact calculation of fuzzy decision rules for multiple testing)
Fuzzy.p.value (Computing Fuzzy p-Value)
FuzzyAHP ((Fuzzy) AHP Calculation)
fuzzyFDR (Exact calculation of fuzzy decision rules for multiple testing)
fuzzyforest (Fuzzy Forests)
FuzzyLP (Fuzzy Linear Programming)
FuzzyMCDM (Multi-Criteria Decision Making Methods for Fuzzy Data)
FuzzyNumbers (Tools to Deal with Fuzzy Numbers)
FuzzyNumbers.Ext.2 (Apply Two Fuzzy Numbers on a Monotone Function)
FuzzyR (Fuzzy Logic Toolkit for R)
fuzzyRankTests (Fuzzy Rank Tests and Confidence Intervals)
FuzzyStatProb (Fuzzy Stationary Probabilities from a Sequence of Observations of an Unknown Markov Chain)
FuzzyStatTra (Statistical Methods for Trapezoidal Fuzzy Numbers)
FuzzyToolkitUoN (Type 1 Fuzzy Logic Toolkit)
fuzzywuzzyR (Fuzzy String Matching)
lfl (Linguistic Fuzzy Logic)
RcmdrPlugin.FuzzyClust (R Commander Plug-in for Fuzzy Clustering Methods (Fuzzy C-Means and Gustafson Kessel))
rdd (Regression Discontinuity Estimation)
Rfmtree (Fuzzy Measure Tools for R)
RoughSets (Data Analysis Using Rough Set and Fuzzy Rough Set Theories)
SAFD (Statistical Analysis of Fuzzy Data)
SDEFSR (Subgroup Discovery with Evolutionary Fuzzy Systems in R)
spatialClust (Spatial Clustering using Fuzzy Geographically Weighted Clustering)
vegclust (Fuzzy clustering of vegetation data)
Weighted.Desc.Stat (Weighted Descriptive Statistics)

This paper is organized as follows. Section 3 discusses the preliminary reasons for the need of a calculator in a fuzzy environment. Among the above R packages, two packages “FuzzyNumbers” and “Calculator.LR.FNs” are introduced and discussed in Sects. 4 and 5, respectively. Finally, a comparison between packages “FuzzyNumbers” and “Calculator.LR.FNs” is briefly provided in the last section.

Why do you need a calculator in fuzzy environment?

Although classical arithmetic operations can be extended by the extension principle approach, the complexity of this principle causes some computational challenges/difficulties. To explain some of such difficulties, a numerical example is given in this section for calculating the variance of several fuzzy numbers, which has been known as an important field in research during the past two decades. There are several approaches to solve this problem by defuzzification, e.g., see [1,3,4,6,7,12,15,16] and [21]. But in these works, the original “fuzzy” problem looks like a “crisp” problem after defuzzification with crisp components. Such approaches are debatable in the community of fuzzy researchers, as according to these people the essence of fuzziness has been lost. They claim that if we formulate an imprecise/fuzzy data, the result of the final answer to the problem should be initially also imprecise. According to the critics, open to criticisms from the opponents of fuzzy sets, the notion of “fuzziness” seems to be not necessary for the description of the data in all the above-mentioned works.

Example 1 Suppose that we are going to compute a fuzzy variance for fuzzy numbers using the extension principle to retain the notion of “fuzziness” in the problem. In other words, unlike the above-mentioned studies, we wish to calculate/achieve the membership function of fuzzy (and not precise) variance for fuzzy data based on the extension principle. For this, consider three triangular fuzzy numbers: $x_1 = T(7.6, 8, 9)$, $x_2 = T(4.3, 5, 5.6)$ and $x_3 = T(6, 7, 8)$. It must be mentioned that this is one of the simplest versions of the problem to show the computational challenges, since here it is assumed that: (1) the shape functions of all *LR* fuzzy numbers are the same, (2) the triangular fuzzy data is considered, and (3) the sample size is small. First, one can easily compute the mean of triangular fuzzy numbers x_1 , x_2 and x_3 based on the extension principle as follows:

$$\begin{aligned}\bar{x} &= \frac{1}{3} \odot (x_1 \oplus x_2 \oplus x_3) \\ &= \frac{1}{3} \odot [T(7.6, 8, 9) \oplus T(4.3, 5, 5.6) \oplus T(6, 7, 8)] \\ &= T\left(\frac{7.6 + 4.3 + 6}{3}, \frac{8 + 5 + 7}{3}, \frac{9 + 5.6 + 8}{3}\right) \\ &= T(5.97, 6.67, 7.53).\end{aligned}$$

It must be noted that if the left and the right shape functions of *LR* fuzzy numbers are not the same, then the summation (and hence the mean) of *LR* fuzzy numbers is not easily computable.

Fuzzy variance computation is followed below, step by step, based on the arithmetic operations of LR fuzzy numbers:

$$\begin{aligned}
 s^2 &= \frac{1}{3} \odot \left\{ \oplus_{i=1}^3 (x_i \ominus \bar{x})^2 \right\} \\
 &= \frac{1}{3} \odot \left\{ (x_1 \ominus \bar{x})^2 \oplus (x_2 \ominus \bar{x})^2 \oplus (x_3 \ominus \bar{x})^2 \right\} \\
 &= \frac{1}{3} \odot \left\{ [T(7.6, 8, 9) \ominus T(5.97, 6.67, 7.53)]^2 \right. \\
 &\quad \oplus [T(4.3, 5, 5.6) \ominus T(5.97, 6.67, 7.53)]^2 \\
 &\quad \left. \oplus [T(6, 7, 8) \ominus T(5.97, 6.67, 7.53)]^2 \right\} \\
 &= \frac{1}{3} \odot \{ T(0.07, 1.33, 3.03)^2 \\
 &\quad \oplus T(-3.23, -1.67, -0.37)^2 \\
 &\quad \oplus T(-1.53, 0.33, 2.03)^2 \}.
 \end{aligned}$$

Till now, the computation is exactly based on the extension principle, but the results for the square of triangular fuzzy numbers $T(0.07, 1.33, 3.03)^2$, $T(-3.23, -1.67, -0.37)^2$ and $T(-1.53, 0.33, 2.03)^2$ are not triangular fuzzy numbers. Meanwhile, as a matter of fact, the shape functions of these squares are not same and therefore their summation is not easily computable. To avoid this difficulty, we discuss three possible approaches/strategies as follows:

1. Defuzzification is the first strategy which leads the user to a crisp value for variance. As presented at the beginning of this section, defuzzification approach is the simplest method to confront the problem and this is not our goal, since defuzzification causes elimination of the notion of “fuzziness” in the problem.
2. As another approach (second strategy), one can use a triangular approximation for the square of a triangular fuzzy number, i.e., use an approximation for the multiplication of two triangular fuzzy numbers (see, Chapter 2 from [2]). There exist two criticisms on this approach: (1) the calculation is not exactly based on the extension principle and the result will be an approximation, (2) the approximation formula for multiplication is available just for positive or negative fuzzy numbers. Here, note that the support of the difference of the third fuzzy number from the mean contains a zero point, i.e., $0 \in \text{supp}(x_3 \ominus \bar{x}) = [-1.53, 2.03]$, and hence the second approach failed to compute the fuzzy variance in this example. In other words, although one can approximate $(x_1 \ominus \bar{x})^2$ and $(x_2 \ominus \bar{x})^2$, respectively, by triangular fuzzy numbers $T(-1.59, 1.77, 6.29)$ and $T(-2.42, 2.79, 7.13)$, he/she has not any approximation for x_3^2 and therefore one cannot have any approximation for the summation based on the second strategy.

3. The third approach considers α -cuts of fuzzy numbers, for some $\alpha \in (0, 1]$, and working with the arithmetic operations for intervals, instead of arithmetic operations on fuzzy numbers. We refer the readers to [11] in which an algorithm is proposed to calculate the fuzzy variance based on the extension principle and it coincides with the considered method in package “FuzzyNumbers”. Meanwhile, we will return to a such problem to solve it easily by package “FuzzyNumbers” in the next section (see, Example 11).

See Fig. 1 and note that the membership functions of $(x_i \ominus \bar{x})^2$'s and the fuzzy variance are drawn by package “FuzzyNumbers”, which will be discussed in the next section. This numerical example can be developed for computing the membership function of fuzzy standard deviation, and also computing the membership function of fuzzy covariance between two vectors of fuzzy numbers.

To avoid such computational difficulties/challenges in the fuzzy environment, we propose using two R packages “FuzzyNumbers” and “Calculator.LR.FNs” which are the topics of the next sections, respectively.

“FuzzyNumbers” package

The title of *FuzzyNumbers* is “Tools to Deal with Fuzzy Numbers” and the version of 0.4-6 published on CRAN by Gagolewski and Caha in 2019 [8]. In this section, some basic functions of package “FuzzyNumbers” has been presented and reviewed from [9] with several numerical examples.

Introducing fuzzy number

The user can introduce a fuzzy number by several methods in “FuzzyNumbers” package.

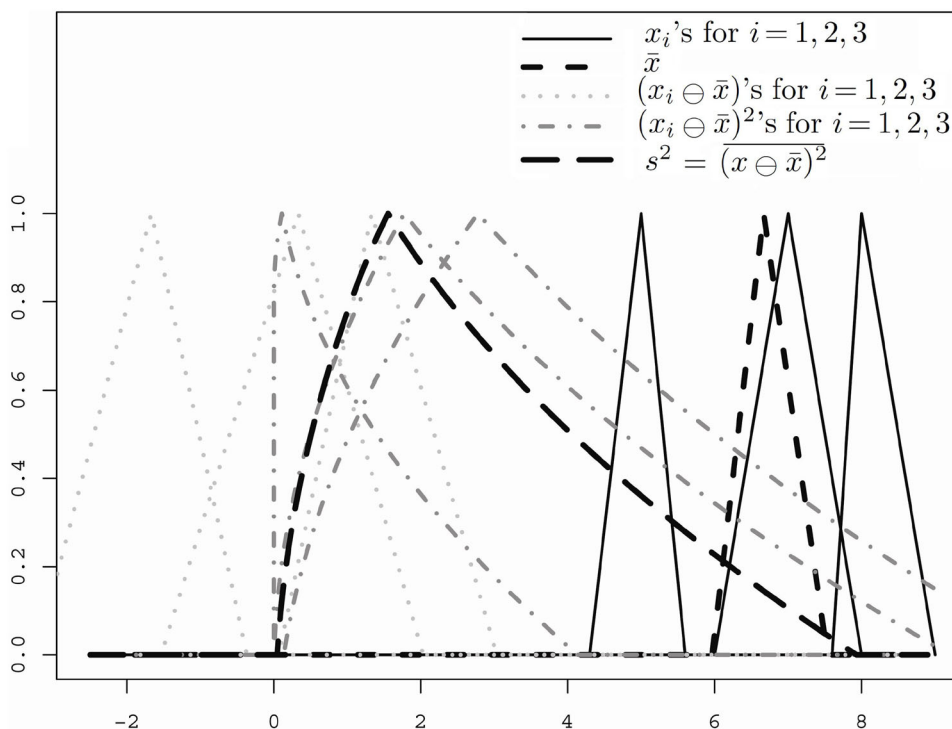
LR fuzzy number

Definition 1 (Fuzzy number in “FuzzyNumbers” package) Fuzzy number A is a fuzzy set from R with membership function

$$\mu_A(x) = \begin{cases} 0 & \text{if } x < a_1, \\ \text{left} \left(\frac{x-a_1}{a_2-a_1} \right) & \text{if } a_1 \leq x < a_2, \\ 1 & \text{if } a_2 \leq x < a_3, \\ \text{right} \left(\frac{x-a_3}{a_4-a_3} \right) & \text{if } a_3 \leq x < a_4, \\ 0 & \text{if } a_4 < x, \end{cases} \quad (1)$$

in which $a_1, a_2, a_3, a_4 \in R$, $\text{left} : [0, 1] \rightarrow [0, 1]$ is a non-increasing function and $\text{right} : [0, 1] \rightarrow [0, 1]$ is a non-increasing function.

Fig. 1 Membership functions of triangular fuzzy numbers and the elements needed for calculating their mean and variance in Example 1



Example 2 Let us introduce and plot fuzzy number A_1 with core $(a_2, a_3) = (2, 4)$, support $(a_1, a_4) = (1, 7)$ and left and right shape functions $\text{left}(x) = x$ and $\text{right}(x) = 1 - x^2$. To this goal, we can run the following comments, after installation and loading “FuzzyNumbers” package in R (see Fig. 2):

```
A1 <- FuzzyNumber(1, 2, 4, 7,
  left=function(x) x,
  right=function(x) 1-x^2
)
plot(A1)
```

Triangular and trapezoidal fuzzy numbers

Example 3 Introduction and plotting of triangular and trapezoidal fuzzy numbers (and also indicator functions) are presented in this example by several methods for “FuzzyNumbers” package.

```
T1 <- TrapezoidalFuzzyNumber(1, 1.5, 4, 7)
plot(T1)
T2 <- TrapezoidalFuzzyNumber(1.4,3,3,7) #Or: T2=TriangularFuzzyNumber(1.4,3,7)
plot(T2, add=TRUE, col=2, lty=3)
TrapezoidalFuzzyNumber(2,2,3,3) # crisp interval
as.TrapezoidalFuzzyNumber(c(2,3)) # is the same
TrapezoidalFuzzyNumber(5,5,5,5) # crisp real
as.TrapezoidalFuzzyNumber(5) # is the same
```

Piecewise linear fuzzy number

Another kind of linear fuzzy numbers, named “piecewise linear fuzzy number”, can be created in “FuzzyNumbers” package by function `PiecewiseLinearFuzzyNumber`. `knot.n` is one of its arguments which controls the number of knots. Using arguments `knot.alpha`, `knot.left` and `knot.right`, the user can be easily control the height, first and end points of knots, respectively.

Example 4 (Creating a piecewise linear fuzzy number)

```
P1 <- PiecewiseLinearFuzzyNumber(1, 2, 3, 4,
  knot.n=1, knot.alpha=0.25, knot.left=1.5,
  knot.right=3.25)
plot(P1, xlim=c(0.5,4.5))
P1
P2 <- PiecewiseLinearFuzzyNumber(1, 2, 3, 4,
  knot.n=2, knot.alpha=c(0.25,0.6),
  knot.left=c(1.5,1.8), knot.right=c(3.25, 3.5))
plot(P2, col=2, lty=2, pch=2, add=TRUE)
P2
```

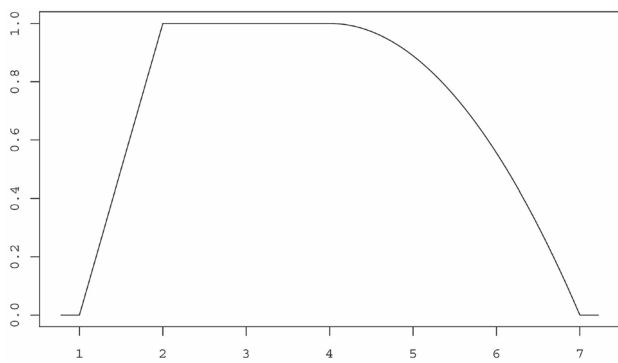


Fig. 2 The membership function of the fuzzy number in Example 2

Power fuzzy number

Considering the shape functions $left(x) = x^{p.left}$ and $right(x) = (1 - x)^{p.right}$ for the introduced fuzzy number in Eq. (1), create a special kind of fuzzy non-linear number called “power fuzzy number”. This kind of fuzzy number can easily be created in package “FuzzyNumbers” by function `PowerFuzzyNumber`. It is obvious that any trapezoidal fuzzy number is a special case of “power fuzzy number” with $p.left = p.right = 1$.

Core, support and cuts of fuzzy numbers

Computing the core, support and α -cuts of a fuzzy number is possible in “FuzzyNumbers” package by using functions `core`, `supp`, and `alphacut`, respectively. Meanwhile, for evaluation of a real point membership degree into a fuzzy number use from function `evaluate`. For more clarify, see the next example [9].

Example 5

```
A <- TrapezoidalFuzzyNumber(1, 1.5, 4, 7)
plot(A)
cuts <- alphacut(A, c(0, 0.5, 1))
cuts
cuts[, 1] # or equivalently cuts[, "L"]
supp(A)
core(A)
evaluate(A, 0)
evaluate(A, c(-3,0,3))
evaluate(A, seq(-1, 2, by=0.5))
```

Arithmetic operations on fuzzy numbers

Two-dimensional operators’ addition, subtraction, multiplication and division are, respectively, introduced by `+`, `-`, `.` and `/` in “FuzzyNumbers” package, which can be considered between two fuzzy numbers. Moreover, `.` can be considered

between a real number and a fuzzy number for scalar multiplication.

Example 6 (Addition and subtraction of two triangular/trapezoidal fuzzy numbers)

```
A <- TrapezoidalFuzzyNumber(0, 1, 1, 2)
B <- TriangularFuzzyNumber(1, 2, 3)
plot(A, xlim=c(0,6))
plot(B, add=TRUE, col=2, lty=2)
plot(A+B, add=TRUE, col=4, lty=4)
```

Although the addition and subtraction of two triangular/trapezoidal fuzzy numbers can be easily computed directly, before considering other operations or using other kinds of fuzzy numbers in “FuzzyNumbers” package, first the fuzzy numbers must be piecewise by function `PiecewiseLinearFuzzyNumber` and then the arithmetic operation is applicable on the piecewised/sliced fuzzy numbers.

Remark 1 The “piecewised linear fuzzy numbers” can be generated by “FuzzyNumbers” package via one of the following three methods:

- (1) **creation** (by function `PiecewiseLinearFuzzyNumber`),
- (2) **conversion** (by function `as.PiecewiseLinearFuzzyNumber`), and
- (3) **approximation** (by function `PiecewiseLinearApproximation`).

The first function is applied for directly creating the triangular/trapezoidal piecewise fuzzy numbers. The second function is applied to exactly convert (and not approximate) a triangular/trapezoidal fuzzy number into a piecewise fuzzy number. The third function is applied to approximate an introduced fuzzy number with a piecewise fuzzy number by one of three methods, `NearestEuclidean`, `SupportCorePreserving` and `Naive`, which are defined via the argument `method` in function

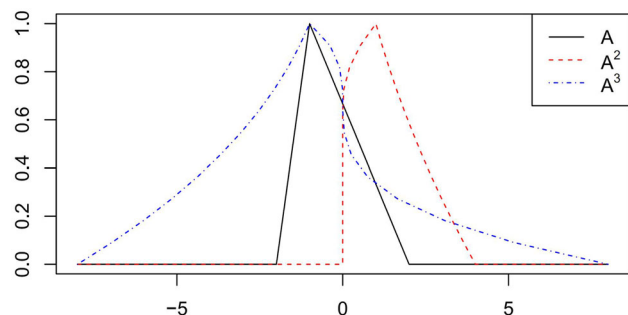


Fig. 3 The membership functions of fuzzy numbers A, A² and A³ in Example 10

`PiecewiseLinearApproximation`. For more details, see the following examples and also [8,9].

Example 7 (Approximation of a fuzzy number with two methods NearestEuclidean and Naive)

```
A <- FuzzyNumber(-5, 3, 6, 20,
  left = function(x) pbeta(x,0.4,3),
  right = function(x) 1-x^(1/4),
  lower = function(alpha) qbeta(alpha,0.4,3),
  upper = function(alpha) (1-alpha)^4
)
plot(A, xlim=c(-6,21))
P1 <- piecewiseLinearApproximation(A, method='Naive', knot.n=1)
P1['allknots']
plot(P1, add=TRUE, col=2, lty=2, lwd=2)
print(distance(A, P1), 3)
P2 <- piecewiseLinearApproximation(A, method='NearestEuclidean', knot.n=1)
P2['allknots']
plot(P2, add=TRUE, col=3, lty=3, lwd=2)
print(distance(A, P2), 3)
```

Numerical examples

In this sub-section, two interesting applied examples are given for “FuzzyNumbers” package.

Example 8

```
A <- piecewiseLinearApproximation(PowerFuzzyNumber(1,2,3,4,
  p.left=2, p.right=0.5), knot.n=20)
A+A
plot(A, xlim=c(0,8))
plot(A+A, col=2, add=TRUE)
plot(2*A, col=3, lty=2, lwd=2, add=TRUE) # the same as A+A
```

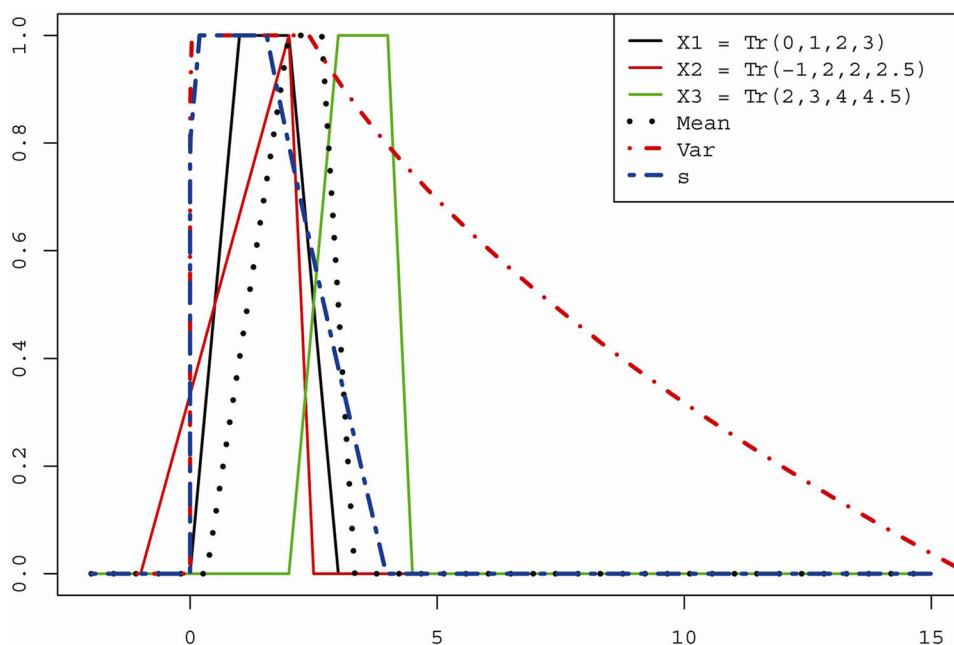
Repeat the running of the comments by `knot.n=1` once again, and compare the observed result with the result of Example 8.

Example 9 Regarding Page 107 from [10], we compute four basic arithmetic operations on two triangular fuzzy numbers $A = T(-1, 1, 3)$ and $B = T(1, 3, 5)$.

Example 10 [9] Consider the triangular fuzzy number $A = Tr(-2, -1, -1, 2)$. Compute and plot the membership functions of fuzzy numbers A^2 and A^3 (Fig. 3).

```
A <- piecewiseLinearApproximation(PowerFuzzyNumber
  (-1, 1, 1, 3, p.left=1, p.right=1), method="Naive", knot.n=150)
B <- piecewiseLinearApproximation( PowerFuzzyNumber
  (1, 3, 3, 5, p.left=1, p.right=1), method="Naive", knot.n=150)
plot(A, xlim=c(-6,15), lwd=2)
plot(B, add=TRUE, col=1, lwd=2)
plot(A+B, add=TRUE, col=2, lwd=2)
plot(A-B, add=TRUE, col=3, lwd=2)
plot(A*B, add=TRUE, col=4, lwd=2)
plot(A/B, add=TRUE, col=5, lwd=2)
```

Fig. 4 Membership functions of fuzzy observation and their mean, variance and standard deviation in Example 11



```
A <- as.PiecewiseLinearFuzzyNumber(
  TrapezoidalFuzzyNumber(-2,-1,-1,2), knot.n=10)
plot(A, xlim=c(-8,8))
plot(A^2, add=TRUE, col=2, lty=2)
plot(A^3, add=TRUE, col=4, lty=4)
```

Example 11 Suppose that we are going to plot the membership functions of mean, standard deviation and variance of three fuzzy numbers $X_1 = Tr(0, 1, 2, 3)$, $X_2 = T(-1, 2, 2.5)$ and $X_3 = Tr(2, 3, 4, 4.5)$ (see Fig. 4).

```
x1 = as.PiecewiseLinearFuzzyNumber(
  TrapezoidalFuzzyNumber(0,1,2,3), knot.n=20)
x2 = as.PiecewiseLinearFuzzyNumber(
  TrapezoidalFuzzyNumber(-1,2,2,2.5), knot.n=20)
```

```
x3 = as.PiecewiseLinearFuzzyNumber(
  TrapezoidalFuzzyNumber(2,3,4,4.5), knot.n=20)
plot(x1, col=1, xlim=c(-2,15))
plot(x2, col=2, add=TRUE)
plot(x3, col=3, add=TRUE)
mean = (x1+x2+x3)/3
plot(mean, col=1, lwd=4, lty=3, add=TRUE)
s2 = (1/3) * ((x1-mean)^2+(x2-mean)^2+(x3-mean)^2)
plot(s2, col=2, lwd=3, lty=4, add=TRUE)
s = fapply(s2, function(x) sqrt(x)) #See "fapply" from "FuzzyNumbers"
plot(s, col=4, lwd=3, lty=6, add=TRUE)
legend("topright", c("X1 = Tr(0,1,2,3)", "X2 = Tr(-1,2,2,2.5)",
  "X3 = Tr(2,3,4,4.5)", "Mean", "Var", "s"), col=c(1,2,3,1,2,4),
  text.col=1, lwd=c(2,2,2,4,3,3), lty=c(1,1,1,3,4,6)
)
```

See [8,9] to orient with other abilities of “FuzzyNumbers” package, such as *applying a fuzzy number on a monotone function and computing the distance of two fuzzy numbers*.

“Calculator.LR.FNs” package

The title of *Calculator.LR.FNs* package is “Calculator for *LR* Fuzzy Numbers” and the version of 1.3 published on CRAN by Parchami in 2018 [13]. In this section, some basic comments and functions of package “Calculator.LR.FNs” has been presented with several numerical examples. The main goal of this package is computing four basic arithmetic operations for *LR* fuzzy numbers. Moreover, this package has the ability of calculating the membership function of the scalar multiplication of a real number and an *LR* fuzzy number.

Introducing LR fuzzy number

It must be noted that the definition of fuzzy number in the “Calculator.LR.FNs” package differs from that in the “FuzzyNumbers” package.

Definition 2 [2] (*LR fuzzy number in “Calculator.LR.FNs” package*) Let fuzzy number N be a fuzzy subset from R with the following membership function:

$$\mu_N(x) = \begin{cases} \text{left.fun}\left(\frac{n-x}{\alpha}\right) & \text{if } x \leq n, \\ \text{right.fun}\left(\frac{x-n}{\beta}\right) & \text{if } x > n, \end{cases} \quad (2)$$

in which $\text{left.fun} : R^+ \cup 0 \rightarrow [0, 1]$ and $\text{right.fun} : R^+ \cup 0 \rightarrow [0, 1]$ are two non-decreasing functions, $\text{right.fun}(0) = \text{left.fun}(0) = 1$ and $\alpha, \beta > 0$. N is called an “LR fuzzy number” and shown by $N = LR(n, \alpha, \beta)$ notation, in which $n, \alpha, \beta, \text{left.fun}$ and right.fun are called core, left spread, right spread, left shape function and right shape function, respectively.

Remark 2 One can consider several different kinds for the left shape and the right shape functions: e.g. $\text{left.fun}(x) = \exp(-x^p)$ and $\text{right.fun}(x) = \max\{0, 1 - |x|^q\}$, where $p, q \geq 0$.

Definition 3 LR fuzzy number $LR(n, \alpha, \beta)$ with $\text{right.fun}(x) = \text{left.fun}(x)$ is named as “L fuzzy number” and shown by notation $L(n, \alpha, \beta)$.

Remark 3 Triangular and normal fuzzy numbers are two common/popular and special kinds of L fuzzy numbers for which the left and right shape functions are, respectively, equal to

$$\text{left.fun}(x) = \begin{cases} 1-x, & 0 \leq x \leq 1 \\ 0, & 1 < x \end{cases}$$

and

$$\text{left.fun}(x) = e^{-x^2}, \quad x \geq 0.$$

Note that the input and output fuzzy numbers in package “Calculator.LR.FNs” must be in the class of all LR fuzzy numbers and this package has the ability to calculate only on the set of all LR fuzzy numbers. So, it is more convenient to introduce the shape functions and then define the related fuzzy numbers. For instance, see the next example.

Example 12 By the following comments, one can easily introduce and plot the LR fuzzy number $A = LR(20, 12, 10)$ in package “Calculator.LR.FNs”, with the left and right shape functions $\text{left.fun}(x) = 1 - x^2, x \geq 0$ and $\text{right.fun}(x) = e^{-x}, x \geq 0$, respectively (Fig. 5).

```
Left.fun = function(x) { (1-x^2)*(x>=0) }
Right.fun = function(x) { (exp(-x))*(x>=0) }
A = LR(20, 12, 10)
LRFN.plot(A, xlim=c(0,60), col=1)
```

Example 13 (Three kinds of LR fuzzy numbers)

Introducing an LR fuzzy number:

```
Left.fun = function(x) { (1-x^2)*(x>=0) }
Right.fun = function(x) { (exp(-x))*(x>=0) }
LRFN.plot( LR(17,5,3), xlim=c(5,40), lwd=2, lty=2, col=2)
```

Introducing an RL fuzzy number:

```
Left.fun = function(x) { (1/(1+x^2))*(x>=0) }
Right.fun = function(x) { (1/(1+(2*abs(x))))*(x>=0) }
LRFN.plot( RL(20,2,3), lwd=2, add=TRUE)
```

Introducing a L fuzzy number (see Fig. 6):

```
Left.fun = function(x) { (1-x)*(x>=0) }
LRFN.plot( L(30,15,5), lwd=2, lty=3, col=4, add=TRUE)
legend("topright", c("LR(17,5,3)", "RL(20,2,3)", "L(30, 15, 5)"),
      col = c(2, 1, 4), text.col = 1, lwd = c(2,2,2), lty = c(2, 1, 3) )
```

Fig. 5 The membership function of LR fuzzy number A in Example 12

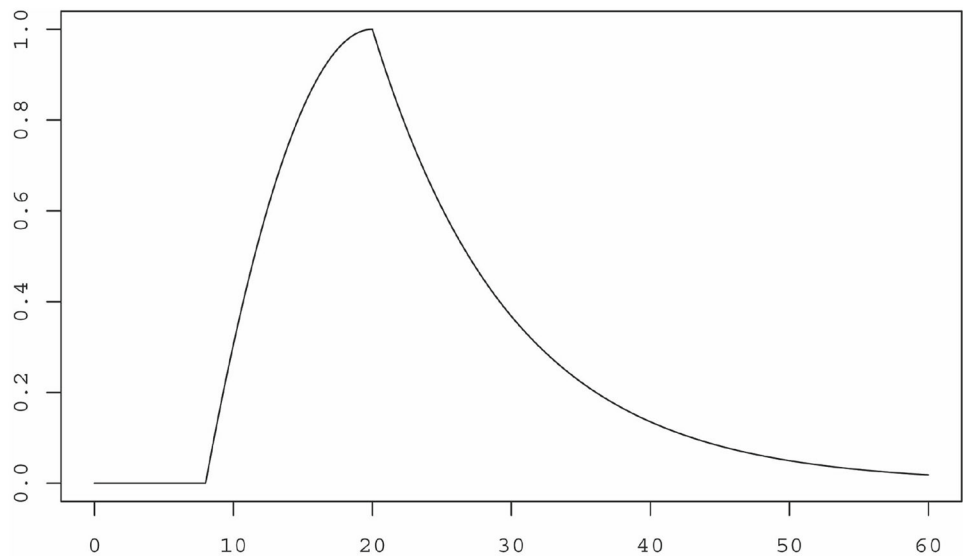
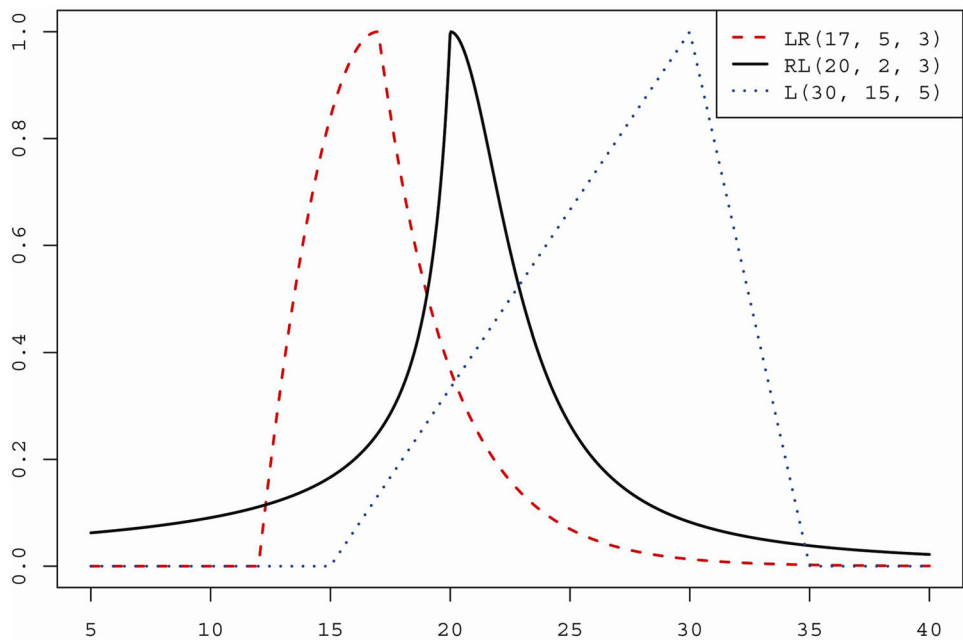


Fig. 6 The membership function of three fuzzy numbers, LR , RL and L in Example 13



Remark 4 Plotting the membership functions of LR fuzzy numbers in “Calculator.LR.FNs” package is possible, using the function `LRFN.plot`. It is more appropriate to define the limits of x ’s axis via the argument `xlim`, in plotting the membership function of the first fuzzy number. Moreover, for drawing several plot on a figure (like Example 13), setting the argument `add=TRUE` is emergent for the second, third, ... fuzzy numbers.

Arithmetic operations on LR fuzzy numbers

Two-dimensional operators’ addition, subtraction, multiplication and division are introduced in package “Calculator.LR.FNs” by functions `a`, `s`, `m` and `d` for two LR fuzzy numbers, respectively. Also, the function `s.m` is consid-

ered for the scalar multiplication of a real number and an LR fuzzy number. Although the base of these operators is Zadeh’s extension principle, in theory the class of LR fuzzy numbers is not close under the operations $*$ and $/$. To close the output of multiplication and division on the class of LR fuzzy numbers, the calculations of multiplication and division in package “Calculator.LR.FNs” is on the basis of the proposed approximation in [2] by LR fuzzy numbers. See [13] for more details and the exact operations formulas.

To clarify further, after introducing the left and right shape functions, suppose that the set of all fuzzy numbers LR , RL and L is named “the class of LR fuzzy numbers”. Now, if the result of arithmetic calculations on “the class of LR fuzzy numbers” belong into this class, then “Calculator.LR.FNs” package gives the result to the user as an LR , RL or L

fuzzy number in the Console window; otherwise, “Calculator.LR.FNs” is not able to calculate and the package “FuzzyNumbers” is proposed for such cases.

Example 14 The following program generates and plots n random LR fuzzy numbers by “Calculator.LR.FNs” package and then calculates and draws the membership function of the mean for the generated fuzzy numbers.

```
n = 10
Left.fun = function(x) { (1-x)*(x>=0)}
Right.fun = function(x) { (exp(-x))*(x>=0)}
xlim = c(2, 18)
sum_x = c(0,0,0,0)
for (i in 1:n) {
  x = rnorm(1,10,3)
  x_l = runif(1,0,3)
  x_r = runif(1,0,2)
  X = c()
  X = LR(x, x_l, x_r)
  LRFN.plot( X, xlim=xlim, ylim=c(0,1), lwd=1,
            lty=1, col=1, add = (i != 1) )
  sum_x = a( sum_x , X )
}
sum_x
X_bar = s.m( 1/n , sum_x )
LRFN.plot(X_bar, lwd=2, lty=2, col=2, add=T)
legend( "topright", c("LR FNs", "mean of LR FNs"),
       col = c(1,2),text.col = 1,lwd = c(1,2),lty = c(1,2))
```

After running above comments in “Calculator.LR.FNs” package, the results of the calculations are presented by the following LR fuzzy numbers in the Console window (see Fig. 7).

Example 15 Consider LR fuzzy numbers $A = LR(5, 0.5, 1)$, $B = LR(2, 0.3, 0.6)$, $C = RL(1, 0.7, 1.5)$ and $D = LR(0.5, 0.5, 1)$ with the shape functions $left.fun(x) = 1 - x^3$, $x \geq 0$ and $right.fun(x) = 1 - x$, $x \geq 0$. After plotting these four fuzzy numbers on a figure, we compute and plot the membership functions of $(A + B)/2$, $[(A + B)/2] - C$ and $\{[(A + B)/2] - C\} \times D$.

```
Left.fun = function(x) { (1-x^3)*(x>=0)}
Right.fun = function(x) { (1-x)*(x>=0)}
A = LR(5, 0.5, 1)
B = LR(2, 0.3, 0.6)
C = RL(1, 0.7, 1.5)
D = LR(0.5, 0.5, 1)
```

```
E = s.m(a(A,B), 1/2) # The mean of A and B
F = s(s.m(a(A,B), 1/2), C)
G = m(F, D)
LRFN.plot(A, xlim=c(-1,6), ylim=c(0,1.6), lwd=3, lty=1, col=1)
LRFN.plot(B, lwd=3, lty=1, col=2, add=TRUE)
LRFN.plot(C, lwd=3, lty=1, col=3, add=TRUE)
LRFN.plot(D, lwd=3, lty=1, col=4, add=TRUE)
LRFN.plot(E, lwd=3, lty=2, col=5, add=TRUE)
LRFN.plot(F, lwd=3, lty=3, col=6, add=TRUE)
LRFN.plot(G, lwd=3, lty=4, col=7, add=TRUE)
```

$$(A + B)/2 = LR(3.5, 0.4, 0.8)$$

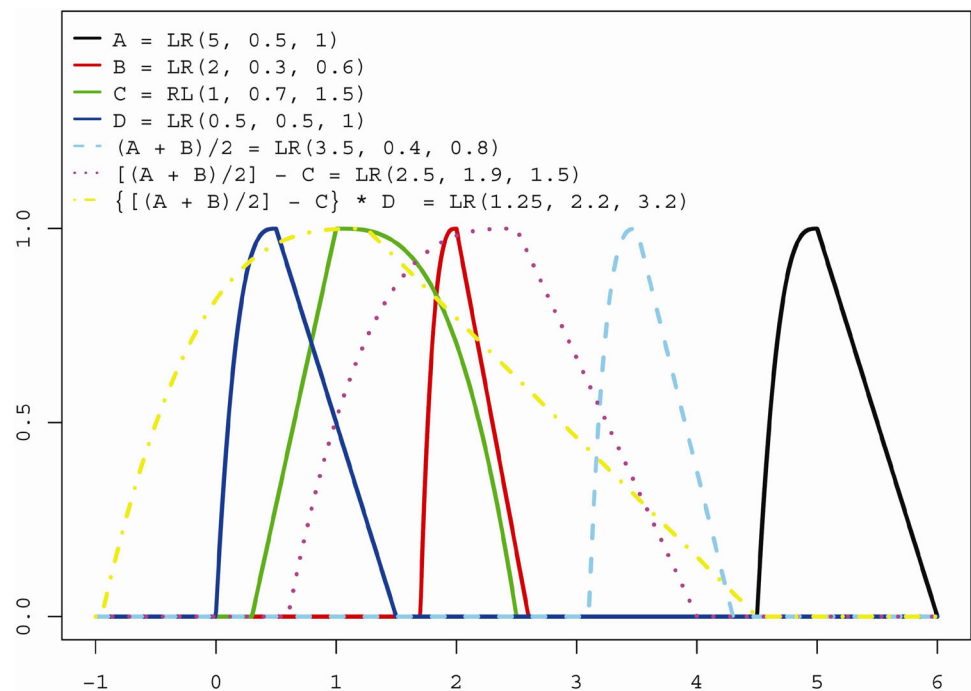
$$[(A + B)/2] - C = LR(2.5, 1.9, 1.5)$$

$$\{[(A + B)/2] - C\} \times D = LR(1.25, 2.2, 3.2)$$

Conclusions: advantages and weak points

Although the classical arithmetic operations can be extended by the extension principle approach, the complexity of this principle causes some computational difficulties/challenges. To resolve these computational challenges, two R packages

Fig. 7 The membership functions of the considered input and output LR fuzzy numbers in Example 15



“FuzzyNumbers” and “Calculator.LR.FNs” have been published on CRAN in 2015 and 2016, respectively. This paper has tried to introduce and explain these R packages briefly, by presenting several numerical examples from [8,9,13]. Package “Calculator.LR.FNs” has the ability of computing the membership function of the result (as a LR fuzzy number) in Console window, which is an advantage with respect to the package “FuzzyNumbers”. Although the package “FuzzyNumbers” can compute the core, support, α cuts and plot the membership function of the result, it cannot give the membership function of the result. From another point of view, the final result of the calculation may not belong to “the class of LR fuzzy numbers” and “Calculator.LR.FNs” is not capable in such cases, and so the package “FuzzyNumbers” is proposed to plot the membership function of the result. This is one of the weak points of “Calculator.LR.FNs” with respect to “FuzzyNumbers”.

Finally, some suggestions are provided to improve the available packages on CRAN in future scientific works:

1. Define/consider the abbreviated version for the name of functions, e.g., introducing function PL instead of function PiecewiseLinearFuzzyNumber in package “FuzzyNumbers”.
2. Notions $+$, $-$, $*$ and $/$ are more appropriate than notions a , s , m and d , which are respectively defined for operators’ addition, subtraction, multiplication and division in package “Calculator.LR.FNs”.
3. Merge and combine some packages (which have a same/similar goal) to achieve a more powerful package.

4. Introduce useful packages to the fuzzy researchers by publishing and comparing related manuscripts.
5. Like the usual calculators for crisp data, the calculator for fuzzy numbers can be developed in future for more complex functions; e.g. for functions e^x , $\log(x)$, $\frac{1}{x}$, x^y and $\sin(x)$.
6. Package “FuzzyNumbers” is able to extend one-variable monotone function $f(\cdot)$ based on extension principle by the `fapply` function for a fuzzy number. Package “FuzzyNumbers.Ext.2” is able to extend two-variable monotone function $f(\cdot, \cdot)$ based on extension principle by the `f2apply` function [14]. Study of the extension of any k -variable monotone function, s.t. $k \in \{1, 2, 3, \dots\}$, is a potential subject for further research.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Blanco-Fernández A, Casals MR, Colubi A, Corral N, García-Bárcana M, Gil MA, González-Rodríguez G, López MT, Lubiano MA, Montenegro M, Ramos-Guajardo AB, de la Rosa de Súa S, Sinova B (2013) Random fuzzy sets: a mathematical tool to develop statistical fuzzy data analysis. Iran J Fuzzy Syst 10:1–28

2. Dubois D, Prade H (1980) Fuzzy sets and systems: theory and applications. Academic Press, Cambridge
3. Carlsson C, Fullér R (2001) On possibilistic mean value and variance of fuzzy numbers. *Fuzzy Sets Syst* 122:315–326
4. Chen W, Tan S (2009) On the possibilistic mean value and variance of multiplication of fuzzy numbers. *J Comput Appl Math* 232:327–334
5. Coroianu L, Gagolewski M, Grzegorzewski P, Adabitar Firozja M, Houlari T (2014) Piece-wise linear approximation of fuzzy numbers preserving the support and core. In: Laurent A (ed) Information processing and management of uncertainty in knowledge-based systems, part II (CCIS 443). Springer, Berlin, pp 244–254
6. Feng Y, Hu L, Shu H (2001) The variance and covariance of fuzzy random variables and their applications. *Fuzzy Sets Syst* 120:487–497
7. Fullér R, Majlender P (2003) On weighted possibilistic mean and variance of fuzzy numbers. *Fuzzy Sets Syst* 136:363–374
8. Gagolewski M, Caha J (2019) FuzzyNumbers Package: tools to deal with fuzzy numbers in R. R package version 0.4-6. <https://cran.r-project.org/web/packages/FuzzyNumbers>
9. Gagolewski M, Caha J (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1). <https://cran.r-project.org/web/packages/FuzzyNumbers>
10. Klir GJ, Yuan B (1995) Fuzzy sets and fuzzy logic: theory and applications. Prentice Hall PTR, New Jersey
11. Körner R, Näther W (2002) On the variance of random fuzzy variables. In: Bertoluzza C, Gil MA, Ralescu DA (eds) Statistical modeling. Analysis and management of fuzzy data. Physica, Heidelberg, pp 22–39
12. Körner R (1997) On the variance of fuzzy random variables. *Fuzzy Sets Syst* 92:83–93
13. Parchami A (2018) Calculator.LR.FNs: Calculator for LR fuzzy numbers. R package version 1.3. <https://CRAN.R-project.org/package=Calculator.LR.FNs>
14. Parchami A (2017) FuzzyNumbers.Ext.2: Apply two fuzzy numbers on a monotone function. R package version 2.3, <https://CRAN.R-project.org/package=FuzzyNumbers.Ext.2>
15. Parchami A, Sadeghpour-Gildeh B, Nourbakhsh M, Mashinchi M (2014) A new generation of process capability indices based on fuzzy measurements. *J Appl Stat* 41:1122–1136
16. Sadeghpour Gildeh B, Rahimpour S (2013) Fuzzy bootstrap test for the mean and variance with D_p , q -distance. *Int J Mach Learn Comput* 3:21–23
17. Viertl R (2011) Statistical methods for fuzzy data. Wiley, Chichester
18. Wikipedia, the free encyclopedia, R programming language. [https://en.wikipedia.org/wiki/R_\(programming_language\)](https://en.wikipedia.org/wiki/R_(programming_language))
19. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8:338–359
20. Zadeh LA (1968) Probability measures of fuzzy events. *J Math Anal Appl* 23:421–427
21. Zhang WG, Wang YL (2007) Notes on possibilistic variances of fuzzy numbers. *Appl Math Lett* 20:1167–1173

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.