

Package ‘Sim.PLFN’

January 20, 2025

Type Package

Title Simulation of Piecewise Linear Fuzzy Numbers

Version 1.0

Date 2017-11-17

Author Abbas Parchami (Department of Statistics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran)

Maintainer Abbas Parchami <parchami@uk.ac.ir>

Description The definition of fuzzy random variable and the methods of simulation from fuzzy random variables are two challenging statistical problems in three recent decades. This package is organized based on a special definition of fuzzy random variable and simulate fuzzy random variable by Piecewise Linear Fuzzy Numbers (PLFNs); see Coroianua et al. (2013) <[doi:10.1016/j.fss.2013.02.005](https://doi.org/10.1016/j.fss.2013.02.005)> for details about PLFNs. Some important statistical functions are considered for obtaining the membership function of main statistics, such as mean, variance, summation, standard deviation and coefficient of variance. Some of applied advantages of 'Sim.PLFN' package are: (1) Easily generating / simulation a random sample of PLFN, (2) drawing the membership functions of the simulated PLFNs or the membership function of the statistical result, and (3) Considering the simulated PLFNs for arithmetic operation or importing into some statistical computation. Finally, it must be mentioned that 'Sim.PLFN' package works on the basis of 'FuzzyNumbers' package.

License LGPL (>= 3)

Imports FuzzyNumbers, DISTRIB

NeedsCompilation no

Repository CRAN

Date/Publication 2017-11-28 18:27:30 UTC

Contents

Sim.PLFN-package	2
cuts.to.PLFN	4
CV	5
Mean	6

PLFN	7
PLFN.to.cuts	10
S.PLFN	11
Sd	13
Sum	14
Trunc	15
Var	16

Index	18
--------------	-----------

Sim.PLFN-package	<i>Simulation of Piecewise Linear Fuzzy Numbers</i>
------------------	---

Description

This package is organized based on a special definition of fuzzy random variable and simulate fuzzy random variable by Piecewise Linear Fuzzy Numbers (PLFNs). Some important statistical functions are considered for obtaining the membership function of main statistics, such as mean, variance, summation, standard deviation and coefficient of variance. Some of applied advantages of 'Sim.PLFN' Package are: (1) Easily generating / simulation a random sample of PLFN, (2) drawing the membership functions of the simulated PLFNs or the membership function of the statistical result, and (3) Considering the simulated PLFNs for arithmetic operation or importing into some statistical computation.

Author(s)

Abbas Parchami

References

- Coroianua, L., Gagolewski, M., Grzegorzewski, P. (2013) Nearest piecewiselinearapproximationof-fuzzy numbers. *Fuzzy Sets and Systems* 233, 26-51.
- Gagolewski, M., Caha, J. (2015) FuzzyNumbers Package: Tools to deal with fuzzy numbers in R. R package version 0.4-1, <https://cran.r-project.org/web/packages=FuzzyNumbers>
- Gagolewski, M., Caha, J. (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1) <http://FuzzyNumbers.rexamine.com>

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
library(FuzzyNumbers)

# Example 1: Let  $x \sim (X \sim N(0, .2) ; s_X^1 \sim \text{Exp}(3) ; s_X^r \sim \text{beta}(1,3) )$ 
n=3; knot.n=3
Sample <- S.PLFN( n, knot.n, type="PLFI",
```

```

X.dist="norm", X.dist.par=c(0,.2),
slX.dist="exp", slX.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)
Sample
Sample[, ,3]

# For plotting random fuzzy sample:
xlim = c( min(Sample), max(Sample) )
plot( cuts.to.PLFN(Sample[, ,1]), type="o", xlim=xlim )
plot( cuts.to.PLFN(Sample[, ,2]), type="o", add=TRUE )
plot( cuts.to.PLFN(Sample[, ,3]), type="o", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

# Example 2:
n=9; knot.n=9
Sample <- S.PLFN( n, knot.n,
X.dist="norm", X.dist.par=c(5,2),
slX.dist="chisq", slX.dist.par=1,
srX.dist="unif", srX.dist.par=c(0,3)
)
Sample
Sample[, ,9]

# For plotting random fuzzy sample:
xlim = c( min(Sample), max(Sample) )
plot( cuts.to.PLFN(Sample[, ,1]), type="b", col=1, xlim=xlim )
plot( cuts.to.PLFN(Sample[, ,2]), type="b", col=2, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,3]), type="b", col=3, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,4]), type="b", col=4, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,5]), type="b", col=5, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,6]), type="b", col=6, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,7]), type="b", col=7, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,8]), type="b", col=8, add=TRUE )
plot( cuts.to.PLFN(Sample[, ,9]), type="b", col=9, add=TRUE )

abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

# Example 3: (Four Arithmetic Operations on PLFNs)

if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)

Sample <- S.PLFN( n=2, knot.n=9,
X.dist="unif", X.dist.par=c(1,3),
slX.dist="beta", slX.dist.par=c(4,3),
srX.dist="beta", srX.dist.par=c(1/3,2/3)
)
Sample

```

```

# For plotting random fuzzy sample:
xlim = c( min(Sample), max(Sample) )
plot( cuts.to.PLFN(Sample[,1]), type="b", xlim=xlim )
plot( cuts.to.PLFN(Sample[,2]), type="b", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

X1 = cuts.to.PLFN( Sample[,1] )
X2 = cuts.to.PLFN( Sample[,2] )

### Now working with four arithmetic operations +, -, * and / are simple as follows:

summ = X1 + X2 ; summ
dist = X1 - X2 ; dist
prod = X1 * X2 ; prod
divi = X1 / X2 ; divi

xlim = c(min(summ["a1"],dist["a1"],prod["a1"],divi["a1"]),
          max(summ["a4"],dist["a4"],prod["a4"],divi["a4"]) )

# For plotting random fuzzy sample:
plot( cuts.to.PLFN(Sample[,1]), type="b", xlim=xlim )
plot( cuts.to.PLFN(Sample[,2]), type="b", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

plot(summ, type="b", xlim=c(0, 12), add=TRUE, col=2, lwd=2)
plot(dist, type="b", xlim=c(0, 12), add=TRUE, col=3, lwd=2)
plot(prod, type="b", xlim=c(0, 12), add=TRUE, col=4, lwd=2)
plot(divi, type="b", xlim=c(0, 12), add=TRUE, col=5, lwd=2)

abline(v=c(X1["a2"],X2["a2"],summ["a2"],dist["a2"],prod["a2"],divi["a2"]), col=
        c(1,1,2,3,4,5), lty=3)
legend( "topright", c("X1 & X2", "X1 + X2", "X1 - X2", "X1 * X2", "X1 / X2"), col=1:5,
        text.col = 1, lwd=2 )

```

cuts.to.PLFN

Convert cuts to Piecewise Linear Fuzzy Number

Description

This function can easily convert a cuts matrix into a Piecewise Linear Fuzzy Number.

Usage

```
cuts.to.PLFN(cuts)
```

Arguments

cuts A matrices, with knot.n row and 2 column, which contains all information about a Piecewise Linear Fuzzy Number.

Value

This function returned a Piecewise Linear Fuzzy Number.

References

Gagolewski, M., Caha, J. (2015) FuzzyNumbers Package: Tools to deal with fuzzy numbers in R. R package version 0.4-1, <https://cran.r-project.org/web/packages=FuzzyNumbers>

Gagolewski, M., Caha, J. (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1) <http://FuzzyNumbers.rexamine.com>

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)

knot.n = 2
T <- PLFN( knot.n=knot.n,
X.dist="norm", X.dist.par=c(0,1),
slX.dist="exp", slX.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)

T
plot(T, type="b")
CUTS <- PLFN.to.cuts(T, knot.n)
CUTS

T2 = cuts.to.PLFN(CUTS)
plot(T2, type="b", col=2, add=TRUE, lwd=3, lty=3)
```

CV

Coefficient of variation for n PLFNs

Description

This function is able to calculate the coefficient of variation (CV) of a sample from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
CV(S.PLFN)
```

Arguments

S.PLFN A sample from Piecewise Linear Fuzzy Numbers (PLFNs), with n PLFNs. This sample is an array with $\text{dim}=\text{c}(\text{knot.n}+2, 2, \text{n})$.

Value

This function returned a Piecewise Linear Fuzzy Number as the coefficient of variation of several PLFNs.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
library(FuzzyNumbers)
n=3; knot.n=4
Sample <- S.PLFN( n, knot.n,
X.dist="norm", X.dist.par=c(3,2),
slX.dist="exp", slX.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)
Sample

# For plotting random fuzzy sample:
xlim = c(0, max(Sample[knot.n+2,2,]))
plot( cuts.to.PLFN(Sample[, ,1]), type="b", xlim=xlim )
plot( cuts.to.PLFN(Sample[, ,2]), type="b", add=TRUE )
plot( cuts.to.PLFN(Sample[, ,3]), type="b", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

FuzzyNumbers::plot(Mean(Sample), col=4, lwd=2, add=TRUE, type="b")
plot(Var(Sample), col=3, lwd=2, add=TRUE, type="b")
plot(Sd(Sample), col=6, lwd=2, add=TRUE, type="b")

CV = CV(Sample)
CV
PLFN.to.cuts(CV, knot.n)
plot(CV, col=2, lwd=2, add=TRUE, type="b")
```

Mean

Mean of n Piecewise Linear Fuzzy Numbers

Description

This function is able to calculate the mean (average) of a sample from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
Mean(S.PLFN)
```

Arguments

S.PLFN A sample from Piecewise Linear Fuzzy Numbers (PLFNs), with n PLFNs. This sample is an array with $\text{dim}=\text{c}(\text{knot.n}+2, 2, \text{n})$.

Value

This function returned a Piecewise Linear Fuzzy Number.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
library(FuzzyNumbers)

n=3; knot.n=4
Sample <- S.PLFN( n, knot.n,
  X.dist="pois", X.dist.par=5,
  slX.dist="exp", slX.dist.par=3,
  srX.dist="beta", srX.dist.par=c(1,3)
)
Sample

# For plotting random fuzzy sample:
xlim = c( min(Sample), max(Sample) )
plot( cuts.to.PLFN(Sample[,1]), type="b", xlim=xlim )
plot( cuts.to.PLFN(Sample[,2]), type="b", add=TRUE )
plot( cuts.to.PLFN(Sample[,3]), type="b", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

M = Mean(Sample)
M
plot( M, type="b", add=TRUE, col=2, lwd=3 )
```

PLFN

Simulate a random Piecewise Linear Fuzzy Number

Description

This function is able to produce / simulate a Piecewise Linear Fuzzy Number (PLFN).

Usage

```
PLFN(knot.n, type = "PLFN", X.dist, X.dist.par, slX.dist, slX.dist.par,
  srX.dist, srX.dist.par)
```

Arguments

knot.n	the number of knots; see package FuzzyNumbers for more details.
type	The possible values of this argument is <code>type = c("Tri", "Tra", "PLFN", "PLFI")</code> . In other words, this function returned one of following fuzzy numbers: (1) Triangular Fuzzy Number (when <code>type = "Tri"</code>), (2) Trapezoidal Fuzzy Number (when <code>type = "Tra"</code>), (3) Piecewise Linear Fuzzy Number (when <code>type = "PLFN"</code>), and (4) Piecewise Linear Fuzzy Interval (when <code>type = "PLFI"</code>).
X.dist	The distribution name of the random variable (for simulate the core of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to <code>stats</code> package.
X.dist.par	A vector of distribution parameters (for simulate the core of random fuzzy number) with considered ordering in <code>stats</code> package.
s1X.dist	The distribution name of the random variable (for simulate the left spread value of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to <code>stats</code> package.
s1X.dist.par	A vector of distribution parameters (for simulate the left spread value of random fuzzy number) with considered ordering in <code>stats</code> package.
srX.dist	The distribution name of the random variable (for simulate the right spread value of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to <code>stats</code> package.
srX.dist.par	A vector of distribution parameters (for simulate the right spread value of random fuzzy number) with considered ordering in <code>stats</code> package.

Value

Considering the `type` argument, this function returned/simulate/create one of following fuzzy numbers: (1) Triangular Fuzzy Number, (2) Trapezoidal Fuzzy Number, (3) Piecewise Linear Fuzzy Number, and (4) Piecewise Linear Fuzzy Interval.

References

- Gagolewski, M., Caha, J. (2015) FuzzyNumbers Package: Tools to deal with fuzzy numbers in R. R package version 0.4-1, <https://cran.r-project.org/web/packages=FuzzyNumbers>
- Gagolewski, M., Caha, J. (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1) <http://FuzzyNumbers.rexamine.com>

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
# Example: Let  $x \sim (X \sim N(3, 0.2) ; s_X^1 \sim \text{Exp}(3) ; s_X^r \sim U(0, 2) )$ 

if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)
```



```

knot.n = 2
P <- PLFN( knot.n, type="Tri",
  X.dist="norm", X.dist.par=c(3,.2),
  slX.dist="exp", slX.dist.par=3,
  srX.dist="unif", srX.dist.par=c(0,2)
)
P
plot(P, lwd=3, type="b")
abline( h=round((knot.n+1):(knot.n+1),4), v=alphacut(P, round((knot.n+1):(knot.n+1),4) ),
  lty=3, col=2 )

P <- PLFN( knot.n, type="Tra",
  X.dist="norm", X.dist.par=c(3,1),
  slX.dist="exp", slX.dist.par=3,
  srX.dist="unif", srX.dist.par=c(0,2)
)
plot(P, lwd=3, type="b")
abline( h=round((knot.n+1):(knot.n+1),4), v=alphacut(P, round((knot.n+1):(knot.n+1),4) ),
  lty=3, col=2 )

knot.n = 2
P <- PLFN( knot.n, #Default: type="PLFN"
  X.dist="norm", X.dist.par=c(3,1),
  slX.dist="exp", slX.dist.par=3,
  srX.dist="unif", srX.dist.par=c(0,2)
)
plot(P, lwd=3)
abline( h=round((knot.n+1):(knot.n+1),4), v=alphacut(P, round((knot.n+1):(knot.n+1),4) ),
  lty=3, col=2 )
#Try once again by knot.n=10

knot.n = 2
P <- PLFN( knot.n, type="PLFI",
  X.dist="norm", X.dist.par=c(3,1),
  slX.dist="exp", slX.dist.par=3,
  srX.dist="unif", srX.dist.par=c(0,2)
)
plot(P, lwd=3, type="b")
abline( h=round((knot.n+1):(knot.n+1),4), v=alphacut(P, round((knot.n+1):(knot.n+1),4) ),
  lty=3, col=2 )

plot(P, type="b", col=2, lty=1, lwd=3, add=FALSE)
# Some of possible types are:
# "p" for points,
# "l" for lines,
# "b" for both,
# "c" for the lines part alone of "b",
# "o" for both over plotted,
# "h" for histogram like (or high-density) vertical lines,
# "s" for stair steps,

```

```

# "S" for other steps,
# "n" for no plotting.

P
P["a1"] #First point of support
P["a2"] #First point of core
P["a3"] #End point of core
P["a4"] #End point of support

core(P)
supp(P)
alphacut(P, 0.5)
abline(h=.5, lty=3)

evaluate(P, 3.5)
round( evaluate(P, seq(0,4, by=.5)), 2)

```

PLFN.to.cuts

Convert Piecewise Linear Fuzzy Number to a cuts matrix

Description

This function can easily convert a Piecewise Linear Fuzzy Number (PLFN) into a cuts matrix.

Usage

```
PLFN.to.cuts(P, knot.n)
```

Arguments

P A Piecewise Linear Fuzzy Number (PLFN).
knot.n the number of knots; see package FuzzyNumbers for more details.

Value

This function returned a matrix which contains core, support and knot.n cuts of the considered PLFN.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```

knot.n = 2
T <- PLFN( knot.n=knot.n,
X.dist="norm", X.dist.par=c(0,1),
slX.dist="exp", slX.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)

T
FuzzyNumbers::plot(T, type="b")
cuts = PLFN.to.cuts(T, knot.n)
cuts
cuts[,"L"] #Lower bounds of cuts
cuts[,"U"] #Upper bounds of cuts
cuts[2,] #Or equivalently cuts["0.6667",]

```

S.PLFN

*Simulate a random sample from Piecewise Linear Fuzzy Numbers***Description**

This function is able to produce / simulate a random sample from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
S.PLFN(n, knot.n, type = "PLFN", X.dist, X.dist.par, slX.dist, slX.dist.par,
srX.dist, srX.dist.par )
```

Arguments

n	the size of random sample of PLSNs.
knot.n	the number of knots; see package FuzzyNumbers for more details.
type	The possible values of this argument is <code>type = c("Tri", "Tra", "PLFN", "PLFI")</code> . In other words, this function returned one of following fuzzy numbers: (1) Triangular Fuzzy Number (when <code>type = "Tri"</code>), (2) Trapezoidal Fuzzy Number (when <code>type = "Tra"</code>), (3) Piecewise Linear Fuzzy Number (when <code>type = "PLFN"</code>), and (4) Piecewise Linear Fuzzy Interval (when <code>type = "PLFI"</code>).
X.dist	The distribution name of the random variable (for simulate the core of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to <code>stats</code> package.
X.dist.par	A vector of distribution parameters (for simulate the core of random fuzzy number) with considered ordering in <code>stats</code> package.
slX.dist	The distribution name of the random variable (for simulate the left spread value of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to <code>stats</code> package.

<code>s1X.dist.par</code>	A vector of distribution parameters (for simulate the left spread value of random fuzzy number) with considered ordering in stats package.
<code>srX.dist</code>	The distribution name of the random variable (for simulate the right spread value of random fuzzy number) is determined by characteristic element <code>T.dist</code> . The names of distributions is similar to stats package.
<code>srX.dist.par</code>	A vector of distribution parameters (for simulate the right spread value of random fuzzy number) with considered ordering in stats package.

Value

Considering the type argument, this function returned/simulate/create one of following fuzzy numbers: (1) Triangular Fuzzy Number, (2) Trapezoidal Fuzzy Number, (3) Piecewise Linear Fuzzy Number, and (4) Piecewise Linear Fuzzy Interval.

References

Gagolewski, M., Caha, J. (2015) FuzzyNumbers Package: Tools to deal with fuzzy numbers in R. R package version 0.4-1, <https://cran.r-project.org/web/packages=FuzzyNumbers>

Gagolewski, M., Caha, J. (2015) A guide to the FuzzyNumbers package for R (FuzzyNumbers version 0.4-1) <http://FuzzyNumbers.rexamine.com>

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
library(FuzzyNumbers)

# Let  $x \sim (X \sim N(0,1) ; s_X \sim \text{Exp}(3) ; s_X \sim \text{beta}(1,3) )$ 

n=3; knot.n=3

Sam <- S.PLFN( n=3, knot.n=4, type="Tra",
X.dist="norm", X.dist.par=c(0,1),
s1X.dist="exp", s1X.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)

Sam
Sam[,,"X3"]

# For plotting random fuzzy sample:
xlim = c( min(Sam), max(Sam) )
plot( cuts.to.PLFN(Sam[,1]), type="b", col=1, xlim=xlim )
plot( cuts.to.PLFN(Sam[,2]), type="b", col=2, add=TRUE )
plot( cuts.to.PLFN(Sam[,3]), type="b", col=3, add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")
```

Sd *Standard of Deviation of n Piecewise Linear Fuzzy Numbers*

Description

This function is able to calculate the Standard of Deviation for a sample with size n from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
Sd(S.PLFN)
```

Arguments

S.PLFN A sample from Piecewise Linear Fuzzy Numbers (PLFNs), with n PLFNs. This sample is an array with $\text{dim}=\text{c}(\text{knot.n}+2, 2, \text{n})$.

Value

This function returned a Piecewise Linear Fuzzy Number.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)

n=3; knot.n=4
Sample <- S.PLFN( n, knot.n,
X.dist="norm", X.dist.par=c(3,2),
s1X.dist="exp", s1X.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)
Sample

# For plotting random fuzzy sample:
xlim = c(0, max(Sample[knot.n+2,2,]))
plot( cuts.to.PLFN(Sample[, ,1]), type="b", xlim=xlim )
plot( cuts.to.PLFN(Sample[, ,2]), type="b", add=TRUE )
plot( cuts.to.PLFN(Sample[, ,3]), type="b", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

plot(Mean(Sample), col=4, lwd=2, add=TRUE, type="b")
plot(Var(Sample), col=3, lwd=2, add=TRUE, type="b")
```

```
S = Sd(Sample)
S
PLFN.to.cuts(S, knot.n)
plot(S, col=2, lwd=2, add=TRUE, type="b")
```

Sum

Summation of n Piecewise Linear Fuzzy Numbers

Description

This function is able to calculate the summation of a sample from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
Sum(S.PLFN)
```

Arguments

S.PLFN A sample from Piecewise Linear Fuzzy Numbers (PLFNs), with n PLFNs. This sample is an array with $\text{dim}=\text{c}(\text{knot.n}+2, 2, \text{n})$.

Value

This function returned a Piecewise Linear Fuzzy Number.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)

n=3; knot.n=4
Sample <- S.PLFN( n, knot.n,
X.dist="pois", X.dist.par=5,
s1X.dist="exp", s1X.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)
Sample
S = Sum(Sample)

# For plotting random fuzzy sample:
xlim = c(min(Sample[knot.n+2,1,]),S["a4"])
plot( cuts.to.PLFN(Sample[,1]), type="o", xlim=xlim )
plot( cuts.to.PLFN(Sample[,2]), type="o", add=TRUE )
plot( cuts.to.PLFN(Sample[,3]), type="o", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")
```

```
plot( S, type="b", add=TRUE, col=2, lwd=3 )
```

Trunc

Simulation from a Truncated Distribution

Description

This function can simulation from a truncated distribution which is used for generating the left and right spreads of Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
Trunc(n, T.dist, T.dist.par, L = -Inf, R = Inf)
```

Arguments

n	The sample size.
T.dist	The distribution name of random variable (which one needs its truncated version) is determined by characteristic element T.dist. The names of distributions is similar to stats package.
T.dist.par	A vector of distribution parameters (which one needs its truncated version) with considered ordering in stats package.
L	The left point of truncation of distribution.
R	The right point of truncation of distribution.

Details

The goal of introducing Trunc function in this package is only using in PLFN and S.PLFN functions.

Value

A vector of random data from the considered truncated distribution.

See Also

DISTRIB

Examples

```
# Truncated Normal Distribution:

data1 = Trunc(n=10^4, T.dist="norm", T.dist.par=c(5,2), L=3, R=10)
hist(data1)

data2 = Trunc(n=1000, T.dist="chisq", T.dist.par=4, L=0, R=12)
hist(data2)
```

```
data3 = Trunc(n=10^4, T.dist="norm", T.dist.par=c(5,2), L=3)
hist(data3)
```

Var *Variance of n Piecewise Linear Fuzzy Numbers*

Description

This function is able to calculate the variance of a sample with size n from Piecewise Linear Fuzzy Numbers (PLFNs).

Usage

```
Var(S.PLFN)
```

Arguments

S.PLFN A sample from Piecewise Linear Fuzzy Numbers (PLFNs), with n PLFNs. This sample is an array with $\text{dim}=\text{c}(\text{knot}.n+2, 2, n)$.

Value

This function returned a Piecewise Linear Fuzzy Number as the variance of PLFNs.

See Also

DISTRIB FuzzyNumbers FuzzyNumbers.Ext.2 Calculator.LR.FNs

Examples

```
if(!require(FuzzyNumbers)){install.packages("FuzzyNumbers")}
library(FuzzyNumbers)

n=3; knot.n=4
Sample <- S.PLFN( n, knot.n,
X.dist="norm", X.dist.par=c(3,2),
slX.dist="exp", slX.dist.par=3,
srX.dist="beta", srX.dist.par=c(1,3)
)
Sample

# For plotting random fuzzy sample:
xlim = c( min(Sample), max(Sample) )
plot( cuts.to.PLFN(Sample[,1]), type="o", xlim=xlim )
plot( cuts.to.PLFN(Sample[,2]), type="o", add=TRUE )
plot( cuts.to.PLFN(Sample[,3]), type="o", add=TRUE )
abline( h=round((knot.n+1):0/(knot.n+1),4), lty=3, col="gray70")

plot(Mean(Sample), col=4, lwd=2, add=TRUE, type="b", pch=2)
```



```
S2 = Var(Sample)
S2
PLFN.to.cuts(S2, knot.n)
plot(S2, col=2, lwd=2, add=TRUE, type="b", pch=3)
```

Index

- * **CV**
 - cuts.to.PLFN, 4
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
- * **FuzzyNumber**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
- * **Mean**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
- * **PLFN.to.cuts**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - S.PLFN, 11
 - Sd, 13
- Sim.PLFN-package, 2
- Sum, 14
- Var, 16
- * **PLFN**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Trunc, 15
 - Var, 16
- * **PowerFuzzyNumber**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
- * **S.PLFN**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Trunc, 15
 - Var, 16
- * **Sd**
 - cuts.to.PLFN, 4
 - CV, 5

- Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
 - * **Sum**
 - Mean, 6
 - Var, 16
 - * **TrapezoidalFuzzyNumber**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
 - * **TriangularFuzzyNumber**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - Var, 16
 - * **Var**
 - cuts.to.PLFN, 4
 - CV, 5
 - Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN-package, 2
 - Sum, 14
 - * **cuts.to.PLFN**
 - cuts.to.PLFN, 4
 - CV, 5
-
- cuts.to.PLFN, 4
 - CV, 5
- Mean, 6
 - PLFN, 7
 - PLFN.to.cuts, 10
 - S.PLFN, 11
 - Sd, 13
 - Sim.PLFN (Sim.PLFN-package), 2
 - Sim.PLFN-package, 2
 - Sum, 14
 - Trunc, 15
 - Var, 16