FuzzySimRes: Epistemic Bootstrap – an Efficient Tool for Statistical Inference Based on Imprecise Data

by Maciej Romaniuk, Przemysław Grzegorzewski, Abbas Parchami

Abstract T he classical Efron's bootstrap is widely used in many areas of statistical inference, including imprecise data. In our new package FuzzySimRes, we adapted the bootstrap methodology to epistemic fuzzy data, i.e., fuzzy perceptions of the usual real-valued random variables. The epistemic bootstrap algorithms deliver real-valued samples generated randomly from the initial fuzzy sample. Then, these samples can be utilized directly in various statistical procedures. Moreover, we implemented a practically oriented simulation procedure to generate synthetic fuzzy samples and provided a real-life epistemic dataset ready to use for various techniques of statistical analysis. Some examples of their applications, together with the comparisons of the epistemic bootstrap algorithms and the respective benchmarks, are also discussed.

1 Introduction

Efron's bootstrap (Efron and Tibshirani, 1993) is a simple but very powerful tool. This useful resampling method is successfully applied in statistical inference, including estimation, hypothesis testing, and other data analysis techniques, e.g., Davison and Hinkley (1997); James et al. (2021); Romaniuk (2019).

In our package **FuzzySimRes**, we adapted the classical bootstrap algorithm to a special kind of imprecise data, i.e., the epistemic random fuzzy numbers (see Couso and Dubois (2014)), which might be treated as fuzzy perceptions of the usual real-valued random variables.

This way, a special resampling methodology, known as the epistemic bootstrap, can be introduced (Grzegorzewski and Romaniuk, 2021, 2022a,c).

Following the suggested methods, we can generate random real-valued samples based on the initial fuzzy sample.

Such a "change of viewpoint" from the "fuzzy world" to its "clear" (i.e., real-value) counterpart can be a very useful and important tool.

This allows all commonly used classical statistical methods (developed for real-valued samples), including statistical tests, estimation procedures, etc., to be directly and easily adapted to fuzzy epistemic samples.

Please note that statistical inference for fuzzy data is usually underdeveloped, poses some problems, and leads to discussions about intuitions, solutions, etc. (e.g., concerning the different approaches to the p-value).

We provide some useful functions in our package **FuzzySimRes**.

They are related to a practically oriented simulation of various types of fuzzy numbers (FNs), the epistemic bootstrap itself, and its applications related to the estimation of important statistical measures of the initial sample, and one- and two-sample statistical tests.

Additionally, we provide a real-life dataset of the epistemic FNs, which can be useful in comparing various approaches to fuzzy statistical inference.

Based on the two general epistemic bootstrap functions, users of **FuzzySimRes** can build their own "epistemic bootstrap statistical tools" to fit their purposes (e.g., the necessity of using tests other than the Kolmogorov-Smirnov one).

In the following, we briefly compare the **FuzzySimRes** package with other existing ones and introduce the necessary notation.

Then, the functions implemented in the package are illustrated with the respective examples.

Finally, the outcomes of these functions are compared, taking into account some benchmarks for different statistical problems using both synthetic and real-life data.

1.1 A brief review of related packages

There are some packages related to fuzzy numbers and their statistical analysis.

Firstly, we should mention **FuzzyNumbers** (Gagolewski and Caha, 2021).

This library aims to provide S4 classes and methods for FNs.

They can be used to construct different types of FNs (e.g., triangular or trapezoidal ones), compute arithmetic operators for fuzzy values, calculate their approximations, and find different characteristics of FNs (like the possibility and necessity values, expected interval, ambiguity, membership functions, among many others) for arbitrary FNs or some of their special types, etc.

Notably, our package **FuzzySimRes** uses S4 objects describing FNs derived from this package.

However, there are no special functions devoted to simulations or resampling methods in the **FuzzyNumbers** package.

Therefore, it can be seen as a kind of "foundation" to deal with FNs.

The next package, **FuzzySTs** (Berkachy and Donzé, 2020), is a collection of various statistical tools, like fuzzification methods, numerical estimations of fuzzy statistical measures and bootstrap distributions of the likelihood ratio, testing hypotheses by fuzzy confidence intervals, and estimation of the fuzzy p-values for epistemic fuzzy data.

These approaches are related to fuzzy notions, like fuzzy p-values, resulting in a strictly fuzzy output (Berkachy and Donzé, 2019).

The **SAFD** (Trutschnig et al., 2013) package joins two kinds of functions.

Similar to **FuzzyNumbers**, it provides basic operations on FNs (like the sum, mean, etc.), but it also contains some strictly statistical functions.

They allow us to simulate FNs and perform bootstrap tests for the equality of the means.

As for the simulation function, this is an implementation of the second procedure described by González-Rodríguez et al. (2009), where a respective basis is perturbed stochastically to generate a new polygonal fuzzy number.

There are important theoretical and practical differences (Parchami et al., 2024) between this approach and the one applied in the **FuzzySimRes** package.

The statistical tests in the **SAFD** library are exclusively based on the classical bootstrap as described by Colubi (2009); Montenegro et al. (2004).

On the other hand, **Sim.PLFN** (Parchami, 2017) can be seen as a kind of "ancestor" of our package.

It allows only the simulation of some kinds of FNs, especially so-called piecewise linear FNs (Coroianu et al., 2013), and calculates a few basic operators (like the sum, mean, and variance) for them.

Another package, **FuzzyStatTra** (Lubiano and de la Rosa de Saa, 2017), also provides basic statistical functions for FNs like the calculation of the mean and medians, indexes, and various distance measures.

Some simulation procedures are also included there, but they are intended for special cases of dependent and independent components described by Sinova et al. (2016).

Therefore, they cannot be considered as "multi-purpose" generation functions when the probability distributions are selected by the user.

We should also mention our previous package, **FuzzyResampling** (Romaniuk et al., 2022), that provides various resampling algorithms other than the classical bootstrap (Romaniuk and Grzegorzewski, 2023).

The main aim of these approaches is to overcome the problem with the repetition of a few distinct values (which is commonly seen in the case of Efron's bootstrap) and to create FNs, which are "similar" (in the sense of some characteristics of FNs) but not "the same" as values from the initial sample (Grzegorzewski et al., 2020; Romaniuk and Hryniewicz, 2019; Grzegorzewski and Romaniuk, 2022b).

Additionally, the tests for the means related to the approach presented by Lubiano et al. (2016) but based on these new resampling methods are also provided.

Nevertheless, **FuzzySimRes** has some unique features.

Firstly, it adds very useful simulation functions (as its acronym – Fuzzy Simulations and Resampling – suggests) to complement **FuzzyNumbers** in this field.

These procedures are very intuitive and practically oriented, as noted by Parchami et al. (2024) (contrary to, e.g., **SAFD** that adds some random noise without keeping track of important characteristics of the input FNs).

Secondly, the so-called epistemic bootstrap is implemented there.

Apart from ready-to-use general epistemic bootstrap functions, special procedures are provided for the estimation of parameters of the underlying statistical model, together with an interface that can be used with various classical statistical tests.

The epistemic bootstrap is a relatively new idea, and the respective algorithms were not implemented in other publicly available software packages (including R itself).

It should be noted that this approach is completely different when compared with the ontic-oriented resampling procedures from the **FuzzyResampling** package that can be seen as a "generalization" of the classical bootstrap procedure (Grzegorzewski et al., 2020; Grzegorzewski and Romaniuk, 2022b).

1.2 Epistemic fuzzy numbers

In the following, we recall some basic concepts and notations concerning fuzzy numbers. For a more detailed introduction, we refer the reader to, e.g., Ban et al. (2015).

A **fuzzy number** (abbreviated further as FN) is an imprecise value characterized by a mapping $\tilde{A} : \mathbb{R} \to [0,1]$ (a **membership function**), such that its α-cut defined by

$$\tilde{A}_\alpha = \begin{cases} \{x \in \mathbb{R} : \tilde{A}(x) \geqslant \alpha\} & \text{if} \quad \alpha \in (0,1], \\ cl\{x \in \mathbb{R} : \tilde{A}(x) > 0\} & \text{if} \quad \alpha = 0, \end{cases} \tag{1}$$

is a nonempty compact interval for each $\alpha \in [0,1]$. Operator cl in (1) denotes the closure. Every FN is completely characterized both by its membership function $\tilde{A}(x)$ and a family of α-cuts $\{\tilde{A}_\alpha\}_{\alpha\in[0,1]}$. There are two special α-cuts: the **core** $\tilde{A}_1 = \text{core}(\tilde{A})$, which contains all values fully compatible with the concept described by $\tilde{A}$, and the **support** $\tilde{A}_0 = \text{supp}(\tilde{A})$ on the real line, for which values are compatible to some extent with the concept modeled by $\tilde{A}$. A family of all FNs will be denoted by $\mathbb{F}(\mathbb{R})$.

There are many possible shapes of the membership functions.

A special family of the **LR-fuzzy numbers** is defined by

$$\tilde{A}(x) = \begin{cases} 0 & \text{if} \quad x < a_1, \\ L\left(\frac{x-a_1}{a_2-a_1}\right) & \text{if} \quad a_1 \leqslant x < a_2, \\ 1 & \text{if} \quad a_2 \leqslant x < a_3, \\ R\left(\frac{a_4-x}{a_4-a_3}\right) & \text{if} \quad a_3 \leqslant x < a_4, \\ 0 & \text{if} \quad x \geqslant a_4, \end{cases} \tag{2}$$

where $L, R : [0,1] \to [0,1]$ are continuous and strictly increasing functions such that $L(0) = R(0) = 0$ and $L(1) = R(1) = 1$, and $a_1, a_2, a_3, a_4 \in \mathbb{R}$, where $a_1 \leqslant a_2 \leqslant a_3 \leqslant a_4$. If L and R are linear functions, i.e., $L\left(\frac{x-a_1}{a_2-a_1}\right) = \frac{x-a_1}{a_2-a_1}$ and $R\left(\frac{a_4-x}{a_4-a_3}\right) = \frac{a_4-x}{a_4-a_3}$, we get a **trapezoidal fuzzy number** (denoted further on as TPFN). Moreover, if $a_2 = a_3$ then we have a **triangular fuzzy number** (abbreviated as TRFN).

In these two cases, we can simply write $A = (a_1, a_2, a_3, a_4)$ (for TPFN) or $A = (a_1, a_2, a_4)$ (for TRFN) to fully describe such FNs.

Another type of LR-fuzzy number is known as the k**-knot piecewise linear fuzzy number** (Coroianu et al., 2019) (or polygonal fuzzy number, see Báez-Sánchez et al. (2012), abbreviated further as PLFN), which is suitable especially in the approximation of more complex FNs.

In this case, L and R functions are polygons consisting of $k \in \mathbb{N}$ segments.

Fuzzy numbers are used to model the results of various experiments that cannot be precisely described, qualified, or measured.

In many cases, we have to draw conclusions and make decisions based on data whose uncertainty comes both from randomness (which classical statistics copes with) and lack of precision (for which fuzzy set theory is perfect for modeling). To model such data, one can use **fuzzy random variables**, known also as **random fuzzy numbers** (Parchami et al., 2024).

It should be noted here that we can look at fuzzy random variables from two different perspectives: **ontic** or **epistemic** (e.g. Couso and Dubois (2014)). The first concerns data that appear to be essentially fuzzy in value, while the second refers to situations where, although precise (accurate) data values exist, they are imprecisely observed (e.g., due to imperfections in measuring devices, inaccuracies caused by people performing the measurements, or how results are reported), so their true actual values remain unknown. This second kind of imprecision is widespread in real-life problems met in engineering, science, and other applications, so further on we limit our attention only to epistemic data.

Following the definition by Kwakernaak (1978) and Kruse (1982), a fuzzy random variable $\widetilde{X}$ can be considered as a *fuzzy perception* of the unknown random variable X, called the *original* of $\widetilde{X}$. More precisely, given a probability space $(\Omega, \mathcal{F}, P)$, a mapping $X : \Omega \to \mathbb{F}(\mathbb{R})$ is said to be a fuzzy random variable (f.r.v.) if for each $\alpha \in [0,1]$ $(\inf X_\alpha) : \Omega \to \mathbb{R}$ and $(\sup X_\alpha) : \Omega \to \mathbb{R}$ are real-valued random variables on $(\Omega, \mathcal{F}, P)$. Similarly, a **fuzzy random sample** $\widetilde{X}_1, \ldots, \widetilde{X}_n$ is a fuzzy perception of a random sample $X_1, \ldots, X_n$ of the usual real-valued random variables. For more details, we refer the reader to Kwakernaak (1978); Kruse (1982).

1.3 Epistemic vs classical bootstrap

There are important differences between the classical Efron's bootstrap (Efron and Tibshirani, 1993) and its epistemic counterpart (Grzegorzewski and Romaniuk, 2021, 2022c, 2024).

In the classical bootstrap approach, the initial sample is directly resampled.

Therefore, in the case of fuzzy input, the output also consists of the same FNs as in the primary sample (with possible repetitions or omitting some of them).

This procedure can be very useful in statistical inference (see, e.g., (Gil et al., 2006; Lubiano et al., 2016; Montenegro et al., 2004)), but the respective statistical tests (or other statistical procedures like the estimation) have to be specially developed for fuzzy-valued samples.

Therefore, there is a need to construct "almost completely new" statistical solutions taking into account various distance measures for fuzzy sets existing in the literature, more complex definitions of the expected value, possible problems with difference operators, etc. (Ban et al., 2015; Heilpern, 1992).

Resampling procedures existing in the **FuzzyResampling** package can be seen as a kind of generalization of this classical bootstrap (in the same manner as the smoothed bootstrap in the case of real-valued samples).

They aim to preserve some important characteristics of FNs (like the value, ambiguity, etc.) but with an alternation of FNs from the initial sample into "new" values occurring in the generated samples (Grzegorzewski et al., 2020; Grzegorzewski and Romaniuk, 2022b).

However, we are still obtaining fuzzy-valued outputs for these methods.

On the other hand, in the epistemic bootstrap, a completely real-valued (i.e., "crisp") sample is generated from a fuzzy-valued initial sample.

This allows the use of directly highly developed statistical tools for real-valued data (various statistical tests, point or interval estimators, etc.) without the need for transforming them into a "new fuzzy world".

Consequently, knowing statistical tools with suitable good properties, the areas of possible applications of epistemic fuzzy data may substantially expand.

To explain it better, consider the following goodness-of-fit testing problem. In Lubiano et al. (2016) and Lubiano et al. (2017), the outcomes of the well-known questionnaire TIMSS-PIRLS 2011 performed by Spanish primary school pupils were considered, while in Ramos-Guajardo et al. (2019) experts' perceptions about different characteristics of the Gamonedo blue cheese were discussed. In both cases, researchers dealt with subjective valuations expressed in natural language, which are inherently imprecise and therefore modeled using ontic fuzzy sets. Thus, the problems mentioned above required the construction of appropriate statistical tools that would enable inferences to be made based on this type of data.

Meanwhile, the epistemic variants of the classical Kolmogorov-Smirnov and Cramer-von Mises tests were directly used for fuzzy data concerning the lifetimes of street light equipment (Hesamian and Taheri, 2013) and electronic circuit thickness (Faraz and Shapiro, 2010) in Grzegorzewski and Romaniuk (2024).

The obtained results were consistent with predictions concerning these real-life samples, like the behavior of the probability distributions of their originals (Gibbons and Chakraborti, 2010).

The example related to the electronic circuit thickness is also considered further in this paper.

Some other applications can also be found in (Grzegorzewski and Romaniuk, 2022a,c,b, 2024).

Moreover, using brute computational force, we can easily improve the quality of the outputs.

However, the results are quite satisfactory also for the limited number of α-cuts.

For instance, using even 10 α-cuts leads to the p-values for the epistemic versions of the goodness-of-fit tests (like the Kolmogorov-Smirnov or Cramer-von Mises tests) very close to their respective benchmarks (Grzegorzewski and Romaniuk, 2024).

2 Overview of FuzzySimRes package

Firstly, we briefly discuss the functions implemented in the **FuzzySimRes** package.

They can be roughly divided into four groups:

1. random generation of FNs of various types,
2. general epistemic bootstrap procedures,
3. epistemic estimation of basic population characteristics from fuzzy samples,
4. interface to statistical tests based on the epistemic bootstrap.

Moreover, a set of real-life epistemic fuzzy data is also included in the package.

All examples in R can be reproduced using the supplementary file.

Taking into account the above-mentioned types of functions, there are many possible applications of the **FuzzySimRes** package:

1. Generation of synthetic fuzzy samples according to the specified probability distributions.

   Such samples can then be used to check the validity and quality of new statistical tools for FNs in a strictly controlled "environment", e.g., to plot power curves for a statistical test (Grzegorzewski and Romaniuk, 2022c) or to check the influence of different model parameters on the estimated p-values (Grzegorzewski and Romaniuk, 2024).

2. Estimation is one of the key problems in statistical inference.

   The same applies to fuzzy-valued data, especially in the epistemic case.

   Then, our considerations about the mean or the standard deviation related to the respective originals of the fuzzy random sample can give us the necessary insight into the parameters of the underlying statistical model (Grzegorzewski and Romaniuk, 2021, 2022c).

3. Statistical tests are the next important subject in statistical inference.

   To accept or reject the null hypothesis, the respective statistical test has to be developed.

   As it was previously mentioned, the epistemic bootstrap allows for direct application of the widely known real-valued tests instead of their "fuzzy-oriented" counterparts.

   Therefore, e.g., the classical goodness-of-fit Kolmogorov-Smirnov or Cramer-von Mises tests can be directly used with the interface provided by the **FuzzySimRes** package (Grzegorzewski and Romaniuk, 2022a, 2024).

4. Real-life fuzzy data are also important to develop statistical procedures.

   Synthetic samples are very useful, but some problems are only visible when the data are provided by a "true source".

   In the **FuzzySimRes** package, there is a special set of such data used to construct a fuzzy statistical control chart (Faraz and Shapiro, 2010) and check the quality of statistical tests based on the epistemic bootstrap (Grzegorzewski and Romaniuk, 2024).

The general workflow for some possible applications (black lines) and the internal order of invoking functions (orange lines) from the **FuzzySimRes** package can be found in Fig. 1.

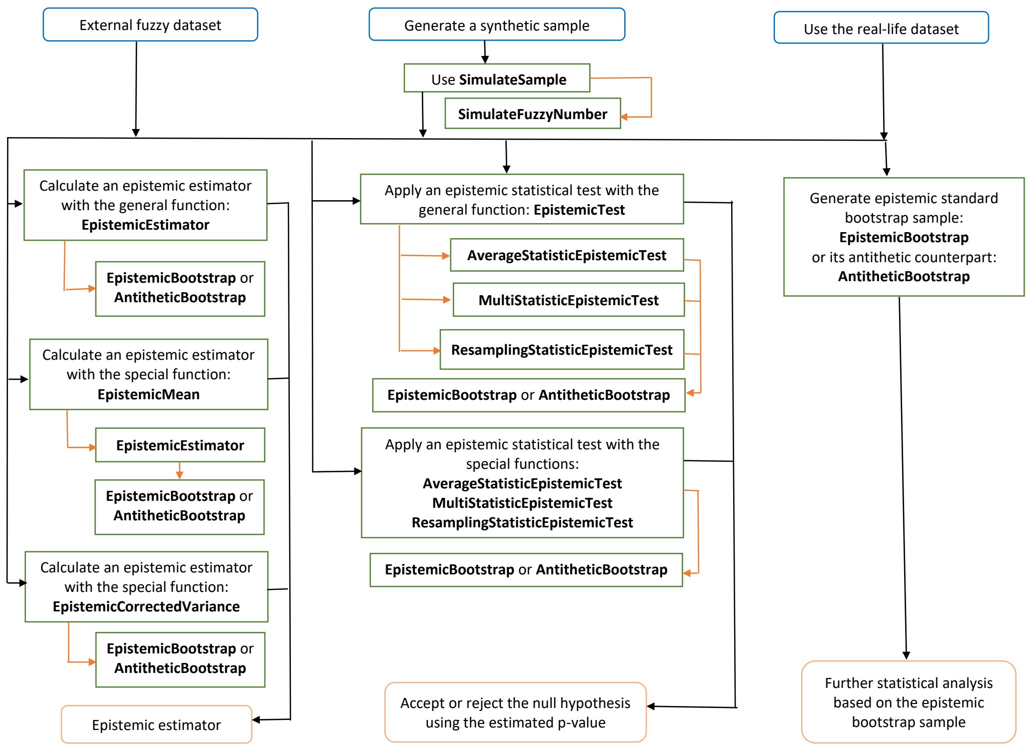

Figure 1: General workflow for possible applications and invoking the functions from the **FuzzySimRes** package.

2.1 Generation of the initial sample

In many cases, synthetic samples of predefined properties are necessary to analyze statistical methods numerically.

Two functions in **FuzzySimRes** allow the generation of random fuzzy variables.

The first one

```
SimulateFuzzyNumber(originalPD,parOriginalPD,incrCorePD,
  parIncrCorePD,suppLeftPD,parSuppLeftPD,
```

```
  suppRightPD,parSuppRightPD,knotNumbers = 0,
  type = "trapezoidal",...)
```

is used to randomly generate a single TPFN (for `type = "trapezoidal"`), TRFN (`type = "triangular"`), or PLFN (`type = "PLFN"`, respectively).

All these types of FNs utilize the respective S4 objects from **FuzzyNumbers**.

To simulate a TPFN $\widetilde{X}$, five independent real-valued random variables are necessary: X for its "true value" (i.e., its *original*), C^l, C^r – the left and right increment of the core, S^l, S^r – the left and right increment of the support, respectively.

To generate these random variables, the functions derived from **stats** (R Core Team, 2023) with the respective parameters are used (see Table 1), e.g., to randomly draw the original X, the function `originalPD` with the parameters `parOriginalPD` should be applied.

Random variable	Function	Parameters
X	`originalPD`	`parOriginalPD`
C^l, C^r	`incrCorePD`	`parIncrCorePD`
S^l	`suppLeftPD`	`parSuppLeftPD`
S^r	`suppRightPD`	`parSuppRightPD`

Table 1: Random variables used to simulate a TPFN.

As a result, we obtain a random TPFN given by $(X - C^l - S^l, X - C^l, X + C^r, X + C^r + S^r)$ (see also Grzegorzewski and Romaniuk (2022a) for a similar procedure).

Obviously, for a TRFN we have $C^l = C^r = 0$ without using the respective parameters in `SimulateFuzzyNumber`.

In the case of a PLFN, the number of knots `knotNumbers` should be greater than zero, and then the specially truncated probability distributions for both arms of the support are applied.

The function `SimulateFuzzyNumber` returns both the generated FN (as `value` in the output list) and its random original X (as `original`).

Let us initialize a random seed and generate a TPFN with the "true origin" described by the normal distribution with the expected value $\mu = 0$ and standard deviation $\sigma = 1$ (denoted by $N(\mu, \sigma)$), the increments of the core given by the uniform distribution on the interval $(0, 0.6)$ (denoted by $U(0, 0.6)$), and the increments of the support from $U(0, 1)$:

```
# seed PRNG
> set.seed(123456)

> SimulateFuzzyNumber(originalPD="rnorm",parOriginalPD=list(mean=0,sd=1),
+   incrCorePD="runif",parIncrCorePD=list(min=0,max=0.6),
+   suppLeftPD="runif",parSuppLeftPD=list(min=0,max=1),
+   suppRightPD="runif",parSuppRightPD=list(min=0,max=1),
+   type="trapezoidal")

$original

[1] 0.6857515

$value

Trapezoidal fuzzy number with:
   support=[-0.316967,1.10087],
   core=[0.480817,0.902528].
```

The second function generates a sample of `n` independent FNs similarly to `SimulateFuzzyNumber`:

```
SimulateSample(n = 1,originalPD,parOriginalPD,incrCorePD,
  parIncrCorePD,suppLeftPD,parSuppLeftPD,
  suppRightPD,parSuppRightPD,knotNumbers = 0,
  type = "trapezoidal")
```

This function returns a list of simulated FNs together with a vector of their respective originals.

Let us generate 10 TPFNs given by the same distributions as in the previous example and print the second simulated value and its "true origin":

```
# seed PRNG
> set.seed(123456)
> sample1 <- SimulateSample(n=10,originalPD="rnorm",
+  parOriginalPD=list(mean=0,sd=1),
+  incrCorePD="runif",parIncrCorePD=list(min=0,max=0.6),
+  suppLeftPD="runif",parSuppLeftPD=list(min=0,max=1),
+  suppRightPD="runif",parSuppRightPD=list(min=0,max=1),
+  type="trapezoidal")

> sample1$original[2]

[1] -1.301602

> sample1$value[2]

$X2

Trapezoidal fuzzy number with:
   support=[-1.937,-0.229014],
   core=[-1.40214,-0.822808].

> plot(sample1$value[[2]])
```

The obtained graph of this exemplary FN can be found in Fig. 2.

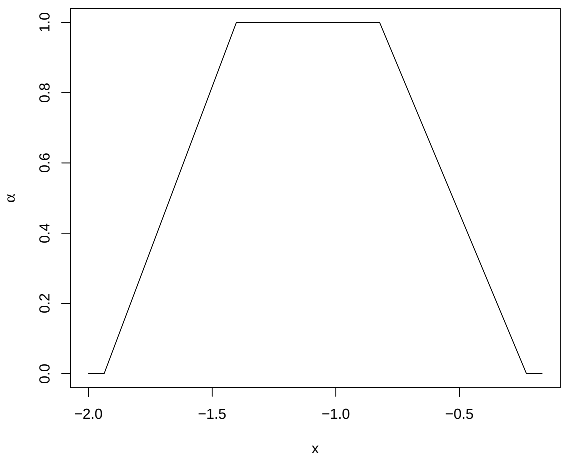

Figure 2: Example of the generated TPFN.

2.2 Epistemic bootstrap

All of the functions described in further sections use two main procedures related to the epistemic bootstrap (Grzegorzewski and Romaniuk, 2021, 2022a,c, 2024).

The first one

```
EpistemicBootstrap(fuzzySample, cutsNumber = 1,...)
```

applies the **standard epistemic bootstrap** (abbreviated as *std*) to a single value or a whole list of FNs given by `fuzzySample`.

This procedure firstly generates uniformly a list of α-cuts (their number is specified by `cutsNumber`). Then, it generates a sample from each of the input FNs, corresponding to the aforementioned list of the α-cuts.

A final output is given as a real-valued matrix, with the number of rows equal to `cutsNumber`, and the number of columns designated by the initial sample size.

This way, we obtain b real-valued bootstrap samples $\mathbb{X}^{*j} = \left(X_1^{*j}, \ldots, X_n^{*j}\right)$, based on the initial fuzzy sample $\tilde{\mathbb{X}} = (\tilde{X}_1, \ldots, \tilde{X}_n)$, where $j = 1, \ldots, b$ and b is equal to `cutsNumber`.

Let us apply the epistemic bootstrap with 3 α-cuts for the previously generated `sample1`, and then show the output rounded to 4 decimal places:

```
> set.seed(123456)

> epistemicOutput <- EpistemicBootstrap(sample1$value, cutsNumber = 3)

> round(epistemicOutput,digits = 4)

           X1      X2     X3     X4      X5     X6     X7      X8      X9    X10
0.7978 0.5323 -1.0784 0.6744 0.8553  0.9501 1.1755 0.6460 -0.5347 -0.0049 1.5937
0.7536 0.5253 -1.4512 1.4851 0.8546  0.9337 1.4773 0.6576 -0.3149 -0.0022 1.8909
0.3913 0.1991 -0.4767 1.1215 0.9443 -0.0089 0.8814 0.4538  0.0633 -0.9345 1.1242
```

The first column shows α-cuts drawn randomly, while the rest of the columns contain values generated from each α-cut.

The second function

```
AntitheticBootstrap(fuzzySample, cutsNumber = 1,...)
```

applies the so-called **antithetic epistemic bootstrap** (denoted further on by *anti*).

Instead of drawing a single value from the given α-cut of each FN, we generate two values: one from this α-cut and the other from $(1-\alpha)$-cut, and then we determine their average.

As indicated in Grzegorzewski and Romaniuk (2022a,c), the antithetic approach improves the quality of some statistical inference methods.

An example of how to use this function can be found in the supplementary file.

The epistemic bootstrap produces a real-valued sample based on the initial fuzzy values.

Therefore, it can be easily applied to estimate various statistical measures of the input values (like the mean) or to conduct many "classical" (i.e., real-valued) statistical tests.

2.3 Estimation of parameters

Estimation of basic population parameters (like the mean) is a fundamental task of most statistical inference problems. Given fuzzy data, we can easily adapt the epistemic bootstrap to estimate the quantities of interest (see (Grzegorzewski and Romaniuk, 2022c)).

A general function

```
EpistemicEstimator(fuzzySample,estimator = "sd",cutsNumber = 1,bootstrapMethod = "std",
  trueValue = NA,...)
```

can be used to determine the desired estimate from the `fuzzySample` of the specified function in `estimator`.

Both the classical epistemic approach (`bootstrapMethod = "std"`) and its antithetic counterpart (`bootstrapMethod = "anti"`) are available.

Since the mean is the most used statistical parameter, it can be obtained using a special function

```
EpistemicMean(fuzzySample,cutsNumber = 1,bootstrapMethod = "std",trueValue = NA,...)
```

instead of the general command.

Besides estimates, the standard error (SE) and the mean squared error (MSE) of the considered estimators are also calculated.

The SE is estimated (for $b > 1$) using the formula

$$\widehat{\mathrm{SE}} = \sqrt{\frac{1}{b-1}\sum_{k=1}^{b}\left(\hat{\theta}\left(\mathbb{X}^{*k}\right) - \bar{\hat{\theta}}\right)^2}, \tag{3}$$

where $\hat{\theta}\left(\mathbb{X}^{*k}\right)$ is the estimator of θ based on the epistemic bootstrap sample for the k-th α-cut and $\bar{\hat{\theta}}$ is the overall mean for $\hat{\theta}\left(\mathbb{X}^{*1}\right),\ldots,\hat{\theta}\left(\mathbb{X}^{*b}\right)$.

If the true (but usually unknown) value of θ is set with `trueValue`, then the MSE is estimated by

$$\widehat{\mathrm{MSE}} = \frac{1}{b}\sum_{k=1}^{b}\left(\hat{\theta}\left(\mathbb{X}^{*k}\right) - \theta\right)^2. \tag{4}$$

Let us estimate the median and its SE for `sample1` using 100 α-cuts and the classical epistemic bootstrap:

```
> set.seed(56789)

> EpistemicEstimator(sample1$value, estimator = "median",cutsNumber = 100)

$value
[1] 0.6287525

$SE
[1] 0.1705336

$MSE
[1] NA
```

To estimate the variance using bootstrap, instead of the classical well-known formula, its more sophisticated and specially corrected variant (Grzegorzewski and Romaniuk, 2022c) can be used with the function

```
EpistemicCorrectedVariance(fuzzySample,cutsNumber = 1,bootstrapMethod = "std",...)
```

As noted in Grzegorzewski and Romaniuk (2022c), this estimator can more closely approximate the desired value, e.g., we have

```
> set.seed(56789)
> EpistemicCorrectedVariance(sample1$value, cutsNumber = 100)

[1] 0.8729738
```

2.4 Statistical tests

The real-valued samples generated by the epistemic bootstrap can also be used for hypothesis testing.

However, given several bootstrap samples, one has to clarify how to merge the obtained test statistics or the p-values (Grzegorzewski and Romaniuk, 2022a).

FuzzySimRes contains a general function

```
EpistemicTest(sample1, sample2, algorithm = "avs", ...)
```

that can be used to activate one of the specially tailored procedures.

By setting `algorithm = "avs"`, the **averaging statistic** (abbreviated as *avs*) is activated, and the function

```
AverageStatisticEpistemicTest(sample1,sample2,bootstrapMethod = "std",
  test = "ks.test",cutsNumber = 1,criticalValueFunction = "KSTestCriticalValue",...)
```

is used.

Similarly, by setting `algorithm = "ms"`, the function

```
MultiStatisticEpistemicTest(sample1,sample2,bootstrapMethod = "std",
  test = "ks.test",cutsNumber = 1,combineMethod = "simes",...)
```

and the **multi-statistic** method (denoted by *ms*) are applied.

Finally, for `algorithm = "res"`, the **resampling** algorithm (abbreviated as *res*) together with the function

```
ResamplingStatisticEpistemicTest(sample1,sample2,bootstrapMethod = "std",
  test = "ks.test",cutsNumber = 1,K = 1,combineMethod = "simes",...)
```

is run.

The above functions can be applied to both one-sample and two-sample statistical tests, where the relevant samples are entered as lists of fuzzy values.

For the one-sample case, `sample2=NULL` should be set.

To use a statistical test, one has to specify the name of the respective function in `test` (e.g., `test="ks.test"` for `ks.test` from **stats** activates the Kolmogorov-Smirnov goodness-of-fit test, abbreviated further on as the KS test).

User-defined functions can also be used if they have at least one or two parameters (`x` for one- or `x,y` for two-sample case, namely) and return a list of at least two values (`statistic` for the output test statistic, and `p.value` for the calculated p-value).

In the case of the *avs* approach, the additional parameter `criticalValueFunction` is required with the name of the function calculating the p-value for a specified critical level of the considered test statistic.

For the KS test, such a procedure is given by `KSTestCriticalValue` available in the **FuzzySimRes**.

To conduct the test, the classical epistemic approach (`bootstrapMethod = "std"`) or its antithetic version (`bootstrapMethod = "anti"`) can be applied.

The p-values (in the case of *ms* or *res* methods) are aggregated with the algorithm specified in `combineMethod`.

Besides `combineMethod="mean"`, i.e., the simple averaging of p-values, all other methods are as in the package **palasso** (Rauschenberger et al., 2020).

Let us generate the second sample with a small shift in location and compare it with the previously generated `sample1` using the two-sample KS test with the *anti* and *ms* approaches for 100 α-cuts:

```
> set.seed(56789)

> sample2 <- SimulateSample(n=10,originalPD="rnorm",
+  parOriginalPD=list(mean=0.5,sd=1),
+  incrCorePD="runif",parIncrCorePD=list(min=0,max=0.6),
+  suppLeftPD="runif",parSuppLeftPD=list(min=0,max=1),
+  suppRightPD="runif", parSuppRightPD=list(min=0,max=1),
+  type="trapezoidal")

> EpistemicTest(sample1$value,sample2$value,algorithm = "ms",
+  bootstrapMethod="anti",cutsNumber=100)

[1] 0.1873127
```

An example of the one-sample KS test can be found in the supplementary file.

2.5 Real-life dataset

FuzzySimRes provides a fuzzy epistemic dataset `controlChartData` concerning electronic circuit thickness, which is one of the most important quality characteristics in the production of electronic boards for vacuum cleaners (see Faraz and Shapiro (2010) for the relevant source).

This dataset is given as a list of 90 TRFNs and contains 30 samples, each of size three.

Every observation has its own label `X.y.z`, where `y` is a sample number, and `z` stands for the element number in a sample, e.g.

```
> controlChartData$X.1.2$

Trapezoidal fuzzy number with:
   support=[70.19,74.15],
   core=[71.4,71.4].
```

is the second value in the first sample.

3 Statistical applications with the package

As it was mentioned, the epistemic bootstrap provides real-valued samples generated from the initial fuzzy sample. It enables us to apply many classical statistical methods instead of using procedures specifically designed for fuzzy data (which are usually underdeveloped in the R environment).

In the following, we present some statistical applications of such approaches for both synthetic and real-life datasets.

In the first case, using `SimulateSample`, the respective samples are generated from the probability distributions described in Table 2.

Available TPFNs are grouped by their types, wherein the normal distribution with the mean μ and standard deviation σ is denoted by $\mathrm{N}(\mu, \sigma)$, the uniform distribution on the interval (a, b) – by $\mathrm{U}(a, b)$, the exponential distribution with the parameter λ – by $\mathrm{Exp}(\lambda)$, the Weibull distribution with the shape k and scale λ parameters – by $\mathrm{Weib}(k, \lambda)$, and the Gamma distribution with the shape α and rate β parameters – by $\Gamma(\alpha, \beta)$, respectively.

In the case of the real-life dataset, the data `controlChartData` embedded in **FuzzySimRes** is applied.

In the following, only some of the results are presented in the tables and graphs to reduce the overall length of the paper.

All of the outputs can be found in the supplementary script file.

Type	X	C^l, C^r	S^l	S^r
$\mathbb{F}_{(\mathrm{N,U,U,U})}$	$\mathrm{N}(0,1)$	$\mathrm{U}(0,0.6)$	$\mathrm{U}(0,1)$	$\mathrm{U}(0,1)$
$\mathbb{F}_{(\mathrm{Weib,Exp,Exp,Exp})}$	$\mathrm{Weib}(2,1)$	$\mathrm{Exp}(5)$	$\mathrm{Exp}(5)$	$\mathrm{Exp}(4)$
$\mathbb{F}_{(\Gamma,\mathrm{U,U,U})}$	$\Gamma(2,2)$	$\mathrm{U}(0,0.6)$	$\mathrm{U}(0,0.8)$	$\mathrm{U}(0,0.8)$

Table 2: Scenarios for simulating fuzzy random variables.

3.1 Comparison of estimators

We start with a comparison of some estimators of the mean, variance, and median for both epistemic approaches, i.e., the *std* and *anti*.

For all types of the TPFNs mentioned in Table 2, the function `EpistemicEstimator` was applied with $b = 100$ α-cuts.

To limit the randomness impact, each numerical experiment was repeated $m = 1000$ times.

Both small ($n = 10$) and moderate ($n = 100$) samples were considered.

Since the function `SimulateSample` produces also the "true values" of the fuzzy samples (i.e., their originals), it gives an opportunity (quite exceptional in real-life applications) to compare the epistemic bootstrap estimators based on fuzzy samples with the results related to these originals.

Then, we can calculate the respective error – **Originals Absolute Error** (abbreviated as OAE) – that measures the absolute difference between the epistemic bootstrap estimator $\hat{\theta}^*_j$ based on the j-th synthetic sample and its counterpart $\hat{\theta}^o_j$ obtained from the originals for this j-th sample, i.e.,

$$\mathrm{OAE} = \frac{1}{m}\sum_{j=1}^{m}\left|\hat{\theta}^*_j - \hat{\theta}^o_j\right|, \tag{5}$$

where m is the number of simulations.

In general, it seems that the *anti* approach gives better results – the resulting estimates are closer to their "true" values, and the respective errors are lower (see Table 3 and the supplementary file).

To facilitate the understanding of Table 3, the best outputs (i.e., the estimators that are the closest to the respective true values of the parameters, and the lowest errors in each case) are given in boldface there.

Of course, the answers may vary for the different error measures (e.g., sometimes the OAE is slightly lower for the *std* approach).

However, the *anti* method clearly provides significant improvement measured with the SE, slightly less important (but still visible) in the case of the MSE.

Taking into account the low additional numerical burden of this approach compared with the *std* method (i.e., generation of two values: from the α-cut and its $(1-\alpha)$ counterpart instead of only a single drawing), the *anti* algorithm should be recommended to users.

The above-mentioned conclusions are similar to the ones discussed in Grzegorzewski and Romaniuk (2021, 2022c).

3.2 Detection of the difference in location

Then, we conducted the power analysis for the two-sample KS test taking into account all the considered epistemic bootstrap approaches.

Two independent samples corresponding to the types of the TPFNs from Table 2 were generated, and a deterministic shift was added to the second sample.

As previously, both small ($n = 10$) and moderate ($n = 100$) samples were considered, and each numerical experiment was repeated $m = 1000$ times.

	Mean		Variance		Median	
	std	anti	std	anti	std	anti
$\mathbb{F}_{(\mathrm{N,U,U,U})}, n = 10$						
Value	-0.0055	**-0.0053**	1.1476	**1.0854**	**-0.0034**	-0.0038
SE	0.1089	**0.0760**	0.2424	**0.1595**	0.1797	**0.1347**
MSE	0.1145	**0.1078**	0.3469	**0.2972**	0.1602	**0.1522**
OAE	0.0407	**0.0405**	0.1555	**0.1105**	0.0874	**0.0819**
$\mathbb{F}_{(\mathrm{N,U,U,U})}, n = 100$						
Value	0.0016	0.0016	1.1472	**1.0850**	**0.0004**	0.0006
SE	0.0345	**0.0242**	0.0999	**0.0498**	0.0705	**0.0573**
MSE	0.0115	**0.0110**	0.0526	**0.0305**	0.0179	**0.0168**
OAE	0.0135	**0.0134**	0.1480	**0.0860**	0.0405	**0.0375**
$\mathbb{F}_{(\mathrm{Weib,Exp,Exp,Exp})}, n = 10$						
Value	0.8917	**0.8912**	0.2884	**0.2671**	0.8536	**0.8517**
SE	0.0636	**0.0440**	0.0728	**0.0478**	0.0941	**0.0706**
MSE	0.0272	**0.0251**	0.0287	**0.0222**	0.0398	**0.0378**
OAE	**0.0409**	0.0412	0.0716	**0.0553**	**0.0609**	0.0621
$\mathbb{F}_{(\mathrm{Weib,Exp,Exp,Exp})}, n = 100$						
Value	**0.8864**	0.8865	0.2828	**0.2614**	0.8417	**0.8387**
SE	0.0203	**0.0141**	0.0295	**0.0152**	0.0359	**0.0290**
MSE	0.0029	**0.0027**	0.0068	**0.0037**	0.0044	**0.0041**
OAE	0.0127	0.0127	0.0676	**0.0462**	0.0252	**0.0247**

Table 3: Numerical comparison of the estimators (for the mean, variance, and median) and their errors (the standard error – SE, the mean squared error – MSE, and the originals absolute error – OAE) based on the epistemic bootstrap (the standard – std, and the antithetic epistemic bootstrap – anti).

Besides the estimation of the null hypothesis rejection percentage for the significance level $\alpha = 0.05$, the p-values for the increasing shift were also obtained and aggregated by simple averaging.

Using `SimulateSample`, which delivers the originals of the simulated fuzzy sample, we can compare the results of the epistemic bootstrap tests with their "crisp" counterpart, so the results of the classical two-sample KS test serve us as a benchmark.

We can see that the estimated p-values (see Fig. 3) and power curves (see Fig. 5) for the moderate sample of TPFNs described by $\mathbb{F}_{(\mathrm{N,U,U,U})}$ are very close to their respective benchmarks, especially for the shift larger than 0.75.

To visualize the results better, the differences in p-values and power curves between the epistemic bootstrap approaches and the classical KS test were also calculated (see Fig. 4 and 6, respectively).

In general, the estimation error for p-values is lower when the *ms* or *res* approaches are used (especially when they are combined with the *anti* method), and the power curves are closer to the respective benchmarks for the *avs* and *ms* algorithms (the *anti* method also has a beneficial effect).

Additional examples can be found in the supplementary file and Grzegorzewski and Romaniuk (2022a).

3.3 Detection of the difference in scale

Next, we conducted the power study of tests to detect the difference in dispersion. This case was modeled by gradually increasing the standard deviation of the second sample when the first one is simulated according to $\mathbb{F}_{(\mathrm{N,U,U,U})}$ type.

As previously, the p-values and power curves (see Fig. 7 and 9) were estimated for the moderate sample and the respective simulation parameters: $m = 1000$, $\alpha = 0.05$, and $b = 100$. A comparison of the epistemic bootstrap approaches and our benchmark (i.e., the two-sample "crisp" KS test) was also done (see Fig. 8 and 10, respectively). It seems that the

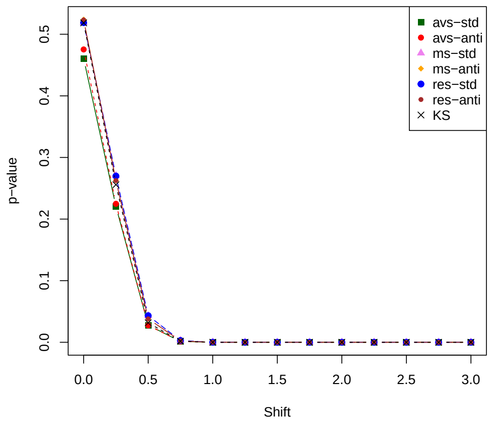

Figure 3: Estimated p-values of the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, and shift in location.

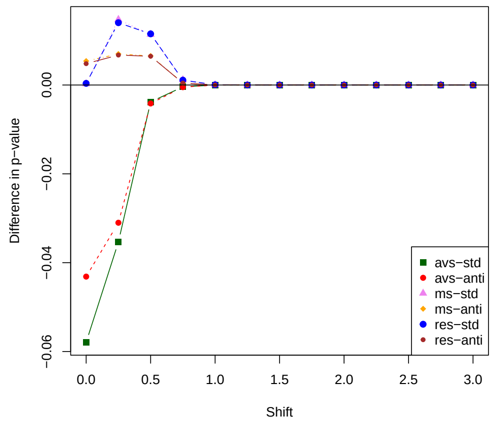

Figure 4: Differences in estimated p-values between the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, and shift in location.

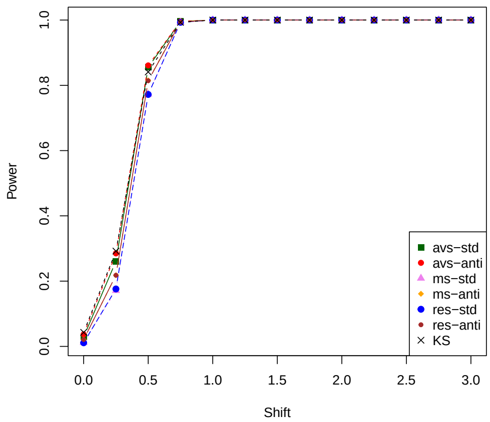

Figure 5: Power curves of the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, and shift in location.

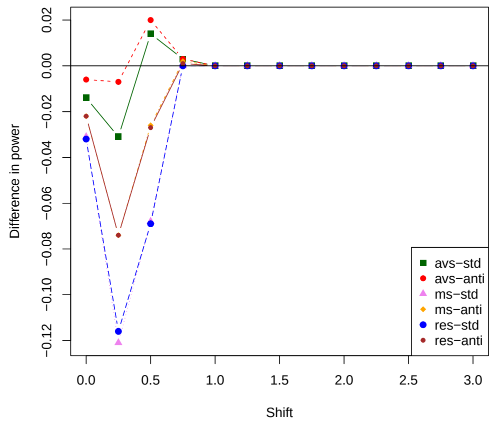

Figure 6: Differences in power curves between the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, and shift in location.

estimation error of p-values is lower for the *ms* or *res* approach and the power curves are closer to the respective results of the "crisp" KS test for the *avs* and *ms* algorithms. Thus, the *anti* method again improves the results.

An additional example for the small sample is provided in the supplementary file.

3.4 Goodness-of-fit test in quality control

Finally, we applied the KS two-sample test for the manufacturing data embedded in **FuzzySimRes**. These fuzzy data can be used to build the respective control charts to check the behavior of the underlying process (Faraz and Shapiro, 2010). But in our experiment, the sample was divided randomly into two parts to check if they came from the same distribution (so they were "not statistically different").

```
> set.seed(5678)
> randomSetsCCD <- sample(length(controlChartData),length(controlChartData)/2)

> EpistemicTest(controlChartData[randomSetsCCD],controlChartData[-randomSetsCCD],
+  algorithm="avs",cutsNumber=1000)

[1] 0.3319477
```

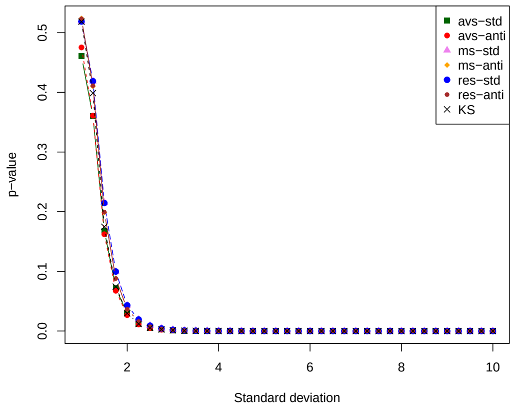

Figure 7: Estimated p-values of the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, difference in scale.

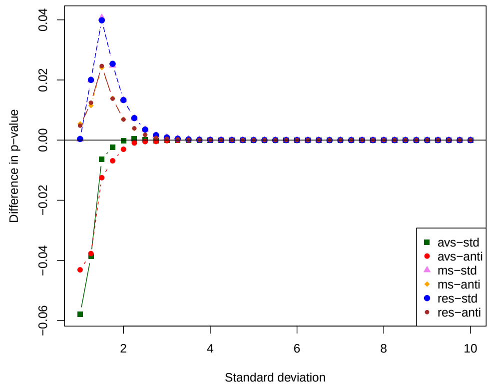

Figure 8: Differences in estimated p-values between the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, difference in scale.

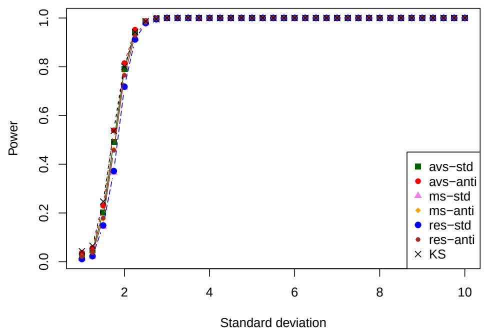

Figure 9: Power curves of the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, difference in scale.

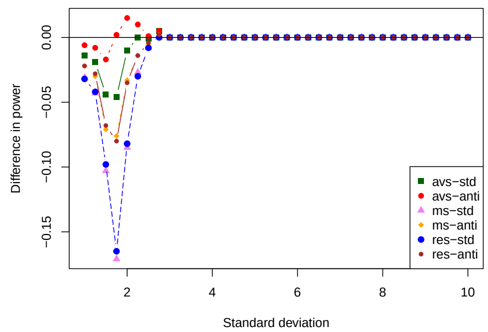

Figure 10: Differences in power curves between the two-sample epistemic and "crisp" KS tests for $\mathbb{F}_{(N,U,U,U)}, n = 100$, difference in scale.

```
> EpistemicTest(controlChartData[randomSetsCCD],controlChartData[-randomSetsCCD],
+  algorithm="ms",combineMethod="mean",cutsNumber=1000)

[1] 0.433548

> EpistemicTest(controlChartData[randomSetsCCD],controlChartData[-randomSetsCCD],
+  algorithm="res",combineMethod="mean",cutsNumber=1000,K=200)

[1] 0.4616578
```

As we can see, all of the considered algorithms do not reject the null hypothesis for the KS test, even for high significance levels.

These results are consistent with the findings in Faraz and Shapiro (2010).

Moreover, as Grzegorzewski and Romaniuk (2024) described, the epistemic KS test clearly indicates the issues caused by the troublesome 21st subsample.

It makes the process out of control and results in the lower p-values in the goodness-of-fit tests.

4 Conclusions

The **FuzzyResampling** package delivers resampling methods developed to overcome some shortcomings of the classical Efron's bootstrap in the fuzzy environment (see also Romaniuk and Grzegorzewski (2023)).

However, this package was intended for the ontic fuzzy data.

Meanwhile, **FuzzySimRes** is a package that has a completely new purpose. The proposed epistemic bootstrap methods allow the generation of real-valued samples from the epistemic fuzzy data, which can then be directly utilized as input values for various classical statistical procedures (like estimators, tests, etc.). It seems that the proposed methods, combined with some well-known statistical techniques, can be competitive with available fuzzy procedures that are not too popular among practitioners. Moreover, as shown in the respective examples, the results of the suggested approaches implemented to imprecise data are comparable with their counterparts – the benchmarks related to the real-valued originals of the fuzzy perceptions.

Of course, further investigations on epistemic bootstrap are still required. They can be aimed both at new resampling epistemic procedures and their applications in statistical inference and machine learning.

References

A. Báez-Sánchez, A. Moretti, and M. Rojas-Medar. On polygonal fuzzy sets and numbers. *Fuzzy Sets and Systems*, 209:54–65, 2012. URL https://doi.org/10.1016/j.fss.2012.04.003. [p3]

A. Ban, L. Coroianu, and P. Grzegorzewski. *Fuzzy Numbers: Approximations, Ranking and Applications*. Polish Academy of Sciences, Warsaw, 2015. [p3, 4]

R. Berkachy and L. Donzé. Testing hypotheses by fuzzy methods: A comparison with the classical approach. In A. Meier, E. Portmann, and L. Terán, editors, *Applying Fuzzy Logic for the Digital Economy and Society*, pages 1–22, Cham, 2019. Springer International Publishing. URL https://doi.org/10.1007/978-3-030-03368-2_1. [p2]

R. Berkachy and L. Donzé. *FuzzySTs: Fuzzy Statistical Tools*, 2020. URL https://CRAN.R-project.org/package=FuzzySTs. R package version 0.2. [p2]

A. Colubi. Statistical inference about the means of fuzzy random variables: Applications to the analysis of fuzzy- and real-valued data. *Fuzzy Sets and Systems*, 160(3):344–356, 2009. URL https://doi.org/10.1016/j.fss.2007.12.019. [p2]

L. Coroianu, M. Gagolewski, and P. Grzegorzewski. Nearest piecewise linear approximation of fuzzy numbers. *Fuzzy Sets and Systems*, 233:26–51, 2013. ISSN 0165-0114. URL https://doi.org/10.1016/j.fss.2013.02.005. [p2]

L. Coroianu, M. Gagolewski, and P. Grzegorzewski. Piecewise linear approximation of fuzzy numbers: algorithms, arithmetic operations and stability of characteristics. *Soft Computing*, 23(19):9491–9505, Oct 2019. URL https://doi.org/10.1007/s00500-019-03800-2. [p3]

I. Couso and D. Dubois. Statistical reasoning with set-valued information: Ontic vs. epistemic views. *International Journal of Approximate Reasoning*, 55:1502–1518, 2014. doi: j.ijar.2013.07.002. URL https://doi.org/10.1016/j.ijar.2013.07.002. [p1, 4]

A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997. URL https://doi.org/10.1017/CBO9780511802843. [p1]

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993. [p1, 4]

A. Faraz and A. F. Shapiro. An application of fuzzy random variables to control charts. *Fuzzy Sets and Systems*, 161(20):2684–2694, 2010. ISSN 0165-0114. URL https://doi.org/10.1016/j.fss.2010.05.004. [p5, 6, 12, 15, 16]

M. Gagolewski and J. Caha. *FuzzyNumbers Package: Tools to Deal with Fuzzy Numbers in R*, 2021. URL https://github.com/gagolews/FuzzyNumbers/. [p1]

J. D. Gibbons and S. Chakraborti. *Nonparametric Statistical Inference*. Chapman and Hall/CRC, 2010. [p5]

M. Gil, M. Montenegro, G. González-Rodríguez, A. Colubi, and M. Casals. Bootstrap approach to the multi-sample test of means with imprecise data. *Computational Statistics and Data Analysis*, 51:148–162, 2006. URL https://doi.org/10.1016/j.csda.2006.04.018. [p4]

G. González-Rodríguez, A. Colubi, and W. Trutschnig. Simulation of fuzzy random variables. *Information Sciences*, 179(5):642–653, 2009. ISSN 0020-0255. URL https://doi.org/10.1016/j.ins.2008.10.018. [p2]

P. Grzegorzewski and M. Romaniuk. Epistemic bootstrap for fuzzy data. In *Joint Proceedings of IFSA-EUSFLAT-AGOP 2021 Conferences*, pages 538–545. Atlantis Press, 2021. URL https://doi.org/10.2991/asum.k.210827.071. [p1, 4, 5, 8, 13]

P. Grzegorzewski and M. Romaniuk. Bootstrapped Kolmogorov-Smirnov test for epistemic fuzzy data. In D. Ciucci, I. Couso, J. Medina, D. Ślęzak, D. Petturiti, B. Bouchon-Meunier, and R. R. Yager, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 494–507, Cham, 2022a. Springer International Publishing. URL https://doi.org/10.1007/978-3-031-08974-9_39. [p1, 5, 6, 7, 8, 9, 11, 14]

P. Grzegorzewski and M. Romaniuk. Bootstrap methods for fuzzy data. In K. T. Atanassov, V. Atanassova, J. Kacprzyk, A. Kałuszko, M. Krawczak, J. W. Owsiński, S. S. Sotirov, E. Sotirova, E. Szmidt, and S. Zadrożny, editors, *Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives*, pages 28–47, Cham, 2022b. Springer International Publishing. URL https://doi.org/10.1007/978-3-030-95929-6_3. [p2, 3, 4, 5]

P. Grzegorzewski and M. Romaniuk. Bootstrap methods for epistemic fuzzy data. *International Journal of Applied Mathematics and Computer Science*, 32(2):285–297, 2022c. URL https://doi.org/10.34768/amcs-2022-0021. [p1, 4, 5, 8, 9, 10, 13]

P. Grzegorzewski and M. Romaniuk. Bootstrapped tests for epistemic fuzzy data. *International Journal of Applied Mathematics and Computer Science*, 34, 2024. (in print). [p4, 5, 6, 8, 16]

P. Grzegorzewski, O. Hryniewicz, and M. Romaniuk. Flexible resampling for fuzzy data. *International Journal of Applied Mathematics and Computer Science*, 30(2):281–297, 2020. URL https://doi.org/10.34768/amcs-2020-0022. [p2, 3, 4]

S. Heilpern. The expected value of a fuzzy number. *Fuzzy Sets and Systems*, 47(1):81–86, 1992. ISSN 0165-0114. URL https://www.sciencedirect.com/science/article/pii/0165011492900629. [p4]

G. Hesamian and S. Taheri. Linear rank tests for two-sample fuzzy data: a p-value approach. *Journal of Uncertain Systems*, 7(2):129–137, 2013. [p5]

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer New York, New York, 2021. URL https://doi.org/10.1007/978-1-0716-1418-1. [p1]

R. Kruse. The strong law of large numbers for fuzzy random variables. *Information Sciences*, 28:233–241, 1982. URL https://doi.org/10.1016/0020-0255(82)90049-4. [p4]

H. Kwakernaak. Fuzzy random variables, part I: Definitions and theorems. *Information Sciences*, 15:1–15, 1978. URL https://doi.org/10.1016/0020-0255(78)90019-1. [p4]

A. Lubiano and S. de la Rosa de Saa. *FuzzyStatTra: Statistical Methods for Trapezoidal Fuzzy Numbers*, 2017. URL https://CRAN.R-project.org/package=FuzzyStatTra. R package version 0.1. [p2]

M. A. Lubiano, M. Montenegro, B. Sinova, S. de la Rosa de Sáa, and M. A. Gil. Hypothesis testing for means in connection with fuzzy rating scale-based data: algorithms and applications. *European Journal of Operational Research*, 251:918–929, 2016. URL https://doi.org/10.1016/j.ejor.2015.11.016. [p2, 4]

M. A. Lubiano, A. Salas, C. Carleos, S. de la Rosa de Sáa, and M. A. Gil. Hypothesis testing-based comparative analysis between rating scales for intrinsically imprecise data. *International Journal of Approximate Reasoning*, 88:128–147, 2017. URL https://doi.org/10.1016/j.ijar.2017.05.007. [p4]

M. Montenegro, A. Colubi, M. Rosa Casals, and M. Ángeles Gil. Asymptotic and bootstrap techniques for testing the expected value of a fuzzy random variable. *Metrika*, 59(1):31–49, 2004. URL https://doi.org/10.1007/s001840300270. [p2, 4]

A. Parchami. *Sim.PLFN: Simulation of Piecewise Linear Fuzzy Numbers*, 2017. URL https://CRAN.R-project.org/package=Sim.PLFN. R package version 1.0. [p2]

A. Parchami, P. Grzegorzewski, and M. Romaniuk. Statistical simulations with LR random fuzzy numbers. *Statistical Papers*, 2024. URL https://doi.org/10.1007/s00362-024-01533-5. (in print). [p2, 3]

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL https://www.R-project.org/. [p7]

A. B. Ramos-Guajardo, Á. Blanco-Fernández, and G. González-Rodríguez. *Applying Statistical Methods with Imprecise Data to Quality Control in Cheese Manufacturing*, pages 127–147. Springer International Publishing, Cham, 2019. URL https://doi.org/10.1007/978-3-030-03201-2_8. [p4]

A. Rauschenberger, I. Ciocanea-Teodorescu, R. X. Menezes, M. A. Jonker, and M. A. van de Wiel. Sparse classification with paired covariates. *Advances in Data Analysis and Classification*, 14:571–588, 2020. URL https://doi.org/10.1007/s11634-019-00375-6. [p11]

M. Romaniuk. On some applications of simulations in estimation of maintenance costs and in statistical tests for fuzzy settings. In A. Steland, E. Rafajłowicz, and O. Okhrin, editors, *Stochastic Models, Statistics and Their Applications*, pages 437–448. Springer International Publishing, 2019. URL https://doi.org/10.1007/978-3-030-28665-1_33. [p1]

M. Romaniuk and P. Grzegorzewski. Resampling fuzzy numbers with statistical applications: FuzzyResampling package. *The R Journal*, 15:271–283, 2023. URL https://doi.org/10.32614/RJ-2023-036. [p2, 17]

M. Romaniuk and O. Hryniewicz. Interval-based, nonparametric approach for resampling of fuzzy numbers. *Soft Computing*, 23:5883–5903, 2019. URL https://doi.org/10.1007/s00500-018-3251-5. [p2]

M. Romaniuk, P. Grzegorzewski, and O. Hryniewicz. *FuzzyResampling: Resampling Methods for Triangular and Trapezoidal Fuzzy Numbers*, 2022. URL https://CRAN.R-project.org/package=FuzzyResampling. R package version 0.6.0. [p2]

B. Sinova, M. A. Gil, and S. Van Aelst. M-estimates of location for the robust central tendency of fuzzy data. *IEEE Transactions on Fuzzy Systems*, 24(4):945–956, 2016. URL https://10.1109/TFUZZ.2015.2489245. [p2]

W. Trutschnig, M. A. Lubiano, and J. Lastra. SAFD — an R package for statistical analysis of fuzzy data. In C. Borgelt, M. Á. Gil, J. M. Sousa, and M. Verleysen, editors, *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, pages 107–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. URL https://doi.org/10.1007/978-3-642-30278-7_10. [p2]

Maciej Romaniuk
Systems Research Institute Polish Academy of Sciences
Newelska 6, 01-447 Warsaw
Poland
WIT Academy
Newelska 6, 01-447 Warsaw
Poland
(0000-0001-9649-396X)
mroman@ibspan.waw.pl

Przemysław Grzegorzewski
Faculty of Mathematics and Information Science, Warsaw University of Technology
Koszykowa 75, 00-662 Warsaw
Poland
Systems Research Institute Polish Academy of Sciences
Newelska 6, 01-447 Warsaw
Poland
(0000-0002-5191-4123)
przemyslaw.grzegorzewski@pw.edu.pl

Abbas Parchami
Department of Statistics, Faculty of Mathematics and Computer
Shahid Bahonar University of Kerman, Kerman
Iran
(0000-0002-0593-7324)
parchami@uk.ac.ir