# Supporting information

**S1 Appendix. Foundations of the fuzzy set theory.**

The difference between classical theory and fuzzy set theory begins in the concept of pertinence of an element to a set. In the classical logic of sets, the concept of pertinence of an element to a set is well defined. The degree of membership is expressed mathematically by the characteristic function, the definition of which is given below.

Definition (1) [Characteristic Function]: given a set *A* in a universe *U*, an element belongs to or does not belong to that set and cannot be partially contained:

However, there are cases in which the degree of membership between elements and sets is not accurate, or rather, it is not known whether an element actually belongs to a set or not. Zadeh (1965) proposed a broader characterization, and generalized the characteristic function so that it could assume an infinite number of values in the interval [0.1].

Definition (2) [Fuzzy Set]: A fuzzy set *A* in a universe *U* is characterized by a membership function *φ*A: *U* → [0,1]. The membership value *φ*A(𝑥) indicates the degree of pertinence of *x* in *A*, where *φ*A(𝑥) = 0 and *φ*A(𝑥) = 1 indicate the non-pertinence and complete pertinence of *x* to the fuzzy set *A*, respectively. The fuzzy set *A* can be represented by a set of ordered pairs:

Definition (3) [Support of a Fuzzy Set]: Given a fuzzy set *A* in a universe *U*, the support of *A*, denoted by supp(*A*), is the (classical) subset of *U* defined by supp(*A*) = {*x* ∈ U : *φ*A(𝑥) > 0}. In general terms, supp(*A*) is the classical set of all elements that have a certain (non-zero) degree of pertinence in *A*.

Definition (4) [α-level]: Let *A* be a fuzzy subset of *U* defined by the relevance function *φ*A : *U* → [0.1] and α ∈ [0.1]. The α-level of *A*, denoted by [*A*]*α*, is the classic subset of *U* defined by:

Definition (5) [Fuzzy Number]: A fuzzy subset *A* is called a fuzzy number when the universe set in which φA is defined is the set of the real numbers R and satisfy the following conditions:

(i) all the α-levels of *A* are non-empty, with 0 ≤ α ≤ 1;

(ii) all the α-levels of *A* are closed intervals of R;

(iii) supp*A* = {*x* ∈ R : φA (*x*) > 0} is limited.

Definition (6) [Zadeh’s Extension Principle]: Let *f* : *X* → *Z* and *A* be a fuzzy subset of *X*. Zadeh’s extension of *f* is the function which, applied to *A*, provides the fuzzy subset of *Z*, whose membership function is defined as:

|  |  |  |
| --- | --- | --- |
|  |  |  |

where 𝑓−1(*𝑧*) = {*x*; *f*(*x*) = *z*}, it is called the pre-image of z.

**S2 Appendix. Fuzzy Set for natural mortality of each species (pseudocode).**

**[** Parameter Definition **]**

**S** : array (mortality space).

var : Real (theoretical maximum length).

var : Real (individual growth coeficiente).

 : array (interval for the i-th IPCC scenario, ).

: Fuzzy Set associated with the range .

 : Fuzzy Set associated with the average annual temperature.

 : Fuzzy Set for average annual temperature under the presence of the i-th scenario.

var **tmptr** : Real (the evaluated temperature).

var **valueM** : Real (numerical value for mortality).

:Object containing the inverse function obtained with Pauly's equation

**temperatureData** : Data Matrix (Database).

**scenarioGroups** : array ("grouping of IPCC scenarios").

**IC** : array ; Confidence interval array (Buckley, 2005).

**CC** : array ; Confidence coefficient associated with the confidence interval.

var **Nt** : Logic; Normality test

**[** End Parameter Definition **]**

***PROGRAM START***

Parameter Reading: **S, , and** ,

**[** Data Reading and Statistical Treatment **]**

# *Accessing the database*

**temperatureData** 🡨 dataReading("Database")

# *Normality Test on the data*

**Nt** 🡨 testNormality(**temperatureData,** **Shapiro-Wilk**)

**[** End Data Reading **]**

**[** Confidence Interval **]**

 % This will be used in the definition of the fuzzy set for average temperature

 **if** Nt **is true**

**CC** 🡨 “select 400 points from the interval [0.01,1]”

 **for k=0 to len(CC)**

 🡨 **CC[k]**

 **IC[k]** 🡨 “Confidence Interval to Confidence Level 1-”

**End for**

 **End if**

**[** End of Confidence Interval **]**

**[** Calculation of the Average Temperature Fuzzy Set **]**

function **tempFuzzySet** (IC,CC)

 **for i = 1 to length(CI)**

 **🡨 CC[i]** % alpha-levels

  **🡨** **CI[i]** % Construction alpha-levels set

 **End for**

**return ()**

**End** function

**[** End Calculation of the Average ... **]**

**[** Membership Function for Average Annual Temperature Values ​​under IPCC scenarios **]**

**for j = 1 to 4, do** # variation of scenarios

 **for i = 1 to length(CI)**

 **🡨 CC[i]** % alpha-levels

% alpha-levels to IPCC scenarios

 **End for**

**End for**

**for i = 1 to 4, do** % Variation of scenarios

 **=** % Sum of fuzzy sets

 **scenarioGroups 🡨** %Storesset informations

**End for**

**[** End Membership Function for Average ... **]**

**[** Membership function degree of the fuzzy subset  **]**

% Returns the membership function degree of the temperature to the set

function **degreeBelongsTemp** (tmptr, )

**if (“**the **tmptr** exceeds the support of the Fuzzy Set **”**)

 **return 0**

 **else**

 **return** (“The degree to which the element **tmptr** belongs to the fuzzy set )

**End** function

**[** End membership function degree **]**

**[** Inverse Function for mortality **]**

**# USED PARAMETERS**

**# valueM** : Real; Numerical value for mortality

**#** :Object containing the inverse function obtained with Pauly's equation

**# Returns the temperature value associated with the mortality value**

function **invMortality** ( valueM, K, )

 % Inverse function obtained with Pauly's equation

 🡨 Inverse FunctionPauly(K,)

**tmptr** =

 **return tmptr**

**End function**

**[** End Inverse Function for Mortality**]**

**[** Calculates Fuzzy Set for Mortality using Zadeh's Extension **]**

**# USED PARAMETERS**

**# list\_M** : array; stores mortality values

**# list\_Degree** : array; stores membership values ​​relative to mortality values

**# value\_Mort** : Real; mortality value

**# value\_Temp** : Real; temperature value

**# degree\_Mort** : Real; mortality membership function degree

**for c = 1 to len(grupoCenarios), do**

 **var list\_M** : : Defines the array that stores mortality values

 **var list\_Degree** : : Defines the array that stores to mortality values

 **for m in range(len(S))** % Space of mortality values

 **value\_Mort** 🡨 **S[m]**

 **value\_Temp** 🡨 invMortality(**val\_Mort**)

 **degree\_Mort** 🡨 degreeBelongsTemp(**val\_Temp**, **scenarioGroups[c]**)

 **list\_M** 🡨 Stores the value of mortality(value\_Mort)

 **list\_Degree** 🡨 Stores the membership values(degree\_Mort)

 **End for**

**End for**

**[** End Calculates Fuzzy Set for Mortality **]**

***END OF PROGRAM***