ALUES

Agricultural Land Use Evaluation System

Contents

1 Installation **1**

2 Dataset **1**
- 2.1 Land Units' Attributes . . . 2
- 2.2 Crop Requirements 2

3 R Functions **3**

4 Methodology **3**
- 4.1 Suitability 3
- 4.2 Overall Suitability 15

5 Demonstration **16**

6 Appendix **21**

21 of August 2014

Arnold R. Salvacion
UP Los Baños
arsalvacion@gmail.com

Al-Ahmadgaid B. Asaad
MSU-IIT
alstated@gmail.com

vignette
Getting Started!

Agricultural Land Use Evaluation System (ALUES) is an R package that evaluates land suitability for different crop production. The package is based on the Food and Agriculture Organization (FAO) and the International Rice Research Institute (IRRI) methodology for land evaluation. Development of ALUES is inspired by similar tool for land evaluation, Land Use Suitability Evaluation Tool (LUSET). The package uses fuzzy logic approach to evaluate land suitability of a particular area based on inputs such as rainfall, temperature, topography, and soil properties. The membership functions used for fuzzy modeling are the following: Triangular, Trapezoidal and Gaussian. The methods for computing the overall suitability of a particular area are also included, and these are the Minimum, Maximum, Product, Sum, Average, Exponential and Gamma. Finally, ALUES utilizes the power of Rcpp library for efficient computation.

1 Installation

The package is not yet on CRAN, and is currently under development on github. To install it, run the following:

```
install.packages('devtools')

library(devtools)
install_github(repo = 'ALUES', username = 'alstat')
```

We want to hear some feedbacks, so if you have any suggestion or issues regarding this package, please do submit it here.

2 Dataset

The package contains several datasets which can be categorized into two:

1. **Land Units' Attributes** - datasets that contain the attributes of the land units of a given location.

2. **Crop Requirements** - datasets that contain the required values of factors of a particular crop for the land units.

2.1 Land Units' Attributes

The package contains sample dataset of land units' attributes from two countries:

1. **Marinduque, Philippines**:

   - `MarinduqueLT` - a dataset consisting the land and terrain characteristics of the land units of Marinduque, Philippines;
   - `MarinduqueTemp` - a dataset consisting the temperature characteristics of the land units of Marinduque, Philippines; and
   - `MarinduqueWater` - a dataset consisting the water characteristics of the land units of Marinduque, Philippines.

2. **Lao Cai, Vietnam**:

   - `LaoCaiLT` - a dataset consisting the land and terrain characteristics of the land units of Lao Cai, Vietnam;
   - `LaoCaiTemp` - a dataset consisting the temperature characteristics of the land units in Lao Cai, Vietnam;
   - `LaoCaiWater` - a dataset consisting the water characteristics of the land units of Lao Cai, Vietnam;

For example, the first six land units in `MarinduqueLT` is shown below

```
head(ALUES::MarinduqueLT)

##       X     Y CECc pHH20 CFragm SoilTe
## 1 121.9 13.52   12    53     11     12
## 2 121.9 13.52   12    52      9     12
## 3 121.9 13.52   12    53     10     12
## 4 121.9 13.52   12    52     10     12
## 5 121.9 13.52   12    54     12     12
## 6 122.0 13.52   13    54     11     12
```

Refer to appendix for complete list of factors.

2.2 Crop Requirements

The crops available in the package are the listed in Table 1. From the table, the codes are suffixed with the land units' characteristics (`TerrainCR`, `SoilCR`, `WaterCR` and `TemperatureCR`) required for the crop.

Crops	Code
Banana	`BANANA-`
Cassava	`CASSAVA-`
Cocoa	`COCOA-`
Coconut	`COCONUT-`
Arabica Coffee	`COFFEEAR-`
Robusta Coffee	`COFFEERO-`
Rainfed Bunded Rice	`RICEBR-`
Irrigated Rice	`RICEIW-`
Rice Cultivation Under Natural Floods	`RICENF-`
Rainfed Upland Rice	`RICEUR-`

Table 1: Crops Dataset Available in ALUES.

For example, below are the required values for the terrain characteristics of the land units on cultivating coconut:

```
ALUES::COCONUTTerrainCR

##       Code S3 S2 S1 S1.1 S2.1 S3.1 Weight.class
## 1   Slope1  6  4  2   NA   NA   NA           NA
## 2   Slope2 16  8  4   NA   NA   NA           NA
## 3   Slope3 30 16  8   NA   NA   NA           NA
## 4    Flood  2  1  1   NA   NA   NA           NA
## 5 Drainage  3  3  2   NA   NA   NA           NA
```

For required characteristics of soil, water and temperature, the codes are `COCONUTSoilCR`, `COCONUTWaterCR` and `COCONUTTemperatureCR`, respectively.

3 R Functions

The package has two functions:

1. `suitability` - computes the suitability scores and classes of the land units base on the requirements of the crop.

2. `overall_suit` - computes the overall suitability of the land units, using the suitability scores obtained from the `suitability` function.

4 Methodology

The details on both functions will be discussed in the following subsections.

4.1 Suitability

In this section, we will get into the details of the `suitability` function.

Usage

```
suitability(x, y, mf = "triangular", sow.month = NULL, min = NULL,
    max = "average", interval = "fixed")
```

`x` a data frame consisting the properties of the land units;

`y` a data frame consisting the crop (e.g. coconut, cassava, etc.) requirements for a given characteristics (terrain, soil, water and temperature);

`mf` membership function, default is set to `"triangular"`. Other fuzzy models are `"Trapezoidal"` and `"Gaussian"`.

`sow.month` sowing month of the crop. Takes integers from 1 to 12 (inclusive), representing the twelve months of a year. So if sets to 1, the function assumes sowing month on January.

`min` factor's minimum value. If `NULL` (default), `min` is set to 0. But if numeric of length one, say 0.5, then minimum is set to 0.5 for all factors. If factors on land units (`x`) have different minimum, then these can be concatenated to vector of `min`s, the length of this vector should be equal to the number of factors in `x`. However, if sets to `"average"`, then `min` is theoretically computed as:

Let X be a factor, then X has the following suitability class: S3, S2 and S1. Assuming the scores of the said suitability class for X are a, b and c, respectively. Then,

$$\min = a - \frac{(b-a)+(c-b)}{2}$$

For factors with suitability class S3, S2, S1, S1, S2 and S3 with scores a, b, c, d, e and f, respectively. `min` is computed as,

$$\min = a - \frac{(b-a)+(c-b)+(d-c)+(e-d)+(f-e)}{5}$$

`max` factor's maximum value. Default is set to `"average"`. If numeric of length one, say 50, then maximum is set to 50 for all factors. If factors on land units (`x`) have different maximum, then these can be concatenated to vector of `max`s, the length of this vector should be equal to the number of factors in `x`. However, if sets to `"average"`, then `max` is computed from the equation below:

$$\max = c + \frac{(b-a)+(c-b)}{2}$$

For factors with suitability class S3, S2, S1, S1, S2 and S3 with scores a, b, c, d, e and f, respectively. Then,

$$\max = f + \frac{(b-a)+(c-b)+(d-c)+(e-d)+(f-e)}{5}$$

`interval` domain for every suitability class (S1, S2, S3, and N). If `"fixed"`, the interval would be 0 to 0.25 for N (Not Suitable), 0.25 to 0.50 for S3 (Marginally Suitable), 0.50 to 0.75 for S2 (Moderately Suitable), and 0.75 to 1 for (Highly Suitable). If `"unbias"`, then the interval is set to 0 to $\frac{a}{\max}$ for N, $\frac{a}{\max}$ to $\frac{b}{\max}$ for S3, $\frac{b}{\max}$ to $\frac{c}{\max}$ for S2, and $\frac{c}{\max}$ to $\frac{\max}{\max}$ for S1.

Output

The function returns the following output:

1. Actual Factors Evaluated;
2. Suitability Score;
3. Suitability Class;
4. Factors' Minimum Values; and,
5. Factors' Maximum Values.

Example: To test the suitability of the land units in Marinduque, Philippines, for terrain requirements of coconut, we have

```
# First 6 of the Marinduque land units
# soil and terrain properties
head(ALUES::MarinduqueLT)

##       X     Y CECc pHH20 CFragm SoilTe
## 1 121.9 13.52   12    53     11     12
## 2 121.9 13.52   12    52      9     12
## 3 121.9 13.52   12    53     10     12
## 4 121.9 13.52   12    52     10     12
## 5 121.9 13.52   12    54     12     12
## 6 122.0 13.52   13    54     11     12

# Soil characteristics of the coconut
# from FAO standards
ALUES::COCONUTSoilCR

##      Code   S3   S2    S1 S1.1 S2.1 S3.1 Weight.class
## 1  CFragm 55.0 35.0  15.0   NA   NA   NA           NA
## 2 SoilDpt 50.0 75.0 100.0   NA   NA   NA           NA
## 3      BS 19.9 19.9  20.0   NA   NA   NA           NA
## 4  SumBCs  1.5  1.5   1.6   NA   NA   NA           NA
## 5      OC  0.7  0.7   0.8   NA   NA   NA           NA
## 6   ECemh 20.0 16.0  12.0   NA   NA   NA           NA
```

Before we run the function, let's check for the possible output. From the land units (`MarinduqueLT`), the only factor available to be evaluated is `CFragm`, for required soil characteristics of the coconut. The first land unit has 11% coarse fragment (CFragm), which falls within the S1 domain of the required soil characteristics, with domain [`min` - 15%), where `min` has default value set to 0. The second to sixth land units also are highly suitable as it falls within the said domain. Let's confirm it using the function,

```
library(ALUES)

## Loading required package:  Rcpp

coco_suit_soil <- suitability(x = MarinduqueLT, y = COCONUTSoilCR)
```

Extract the first 6 of the outputs,

```
lapply(coco_suit_soil, function(x) head(x, n = 6))

## $`Actual Factors Evaluated`
## [1] "CFragm"
##
## $`Suitability Score`
##   CFragm
## 1 0.8533
## 2 0.8800
## 3 0.8667
## 4 0.8667
## 5 0.8400
## 6 0.8533
##
## $`Suitability Class`
##   CFragm
## 1     S1
## 2     S1
## 3     S1
## 4     S1
## 5     S1
## 6     S1
##
## $`Factors' Minimum Values`
## CFragm
##      0
##
## $`Factors' Maximum Values`
## CFragm
##     75
```

Indeed, just what we argued earlier.

Options for `mf` (Membership Function)

The membership function is an option for the type of fuzzy model, the available models are the following:

- Triangular;
- Trapezoidal; and,
- Gaussian.

The suitability scores are computed base on these fuzzy models.

Options for `sow.month` (Sowing Month)

The `sow.month` is the sowing month which takes integers from 1 to 12, representing the twelve months of a year. So if sets to 1, the function assumes sowing month on January. This argument is only use for water and temperature characteristics.

To illustrate this, we will test the land units of Marinduque for the required water and temperature for rainfed bunded rice. Thus, we have

```
library(ALUES)
# Water Characteristics of Marinduque land units
head(MarinduqueWater)

##       X     Y Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1 121.9 13.52 103  65  55  62 136 197 221 191 199 308 277 248
## 2 121.9 13.52 104  65  55  61 135 196 220 190 197 307 276 247
## 3 121.9 13.52 104  66  56  61 134 195 218 187 195 306 275 247
## 4 121.9 13.52 105  65  57  59 131 195 218 189 195 308 273 245
## 5 121.9 13.52 105  65  58  59 131 196 219 190 196 309 272 245
## 6 122.0 13.52 106  66  59  57 128 191 210 179 185 302 268 241

# Temperature Characteristics of Marinduque land units
head(MarinduqueTemp)

##       X     Y  Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct
## 1 121.9 13.52 24.7 25.1 25.9 27.1 27.7 27.4 26.8 26.8 26.7 26.3
## 2 121.9 13.52 24.8 25.0 26.0 27.2 27.8 27.5 26.8 26.8 26.6 26.5
## 3 121.9 13.52 24.9 25.2 26.1 27.3 27.9 27.6 26.9 27.0 26.8 26.6
## 4 121.9 13.52 24.7 25.1 26.0 27.2 27.7 27.5 26.8 26.8 26.6 26.5
## 5 121.9 13.52 24.7 25.1 25.9 27.0 27.7 27.4 26.8 26.8 26.6 26.4
## 6 122.0 13.52 25.0 25.5 26.4 27.5 28.2 27.9 27.4 27.3 27.2 26.9
##    Nov  Dec
## 1 25.9 25.1
## 2 26.0 25.3
## 3 26.2 25.4
## 4 26.0 25.3
## 5 26.0 25.2
## 6 26.4 25.6
```

We will test first the land units for water, and thus we have the following water requirements for rainfed bunded rice,

```
RICEBRWaterCR

##      Code     S3     S2     S1 S1.1 S2.1 S3.1 Weight.class
## 1  WmAv1 100.00 125.00 175.00  500  650  750           NA
## 2  WmAv2 100.00 125.00 175.00  500  650  750           NA
## 3  WmAv3 100.00 125.00 175.00  500  650  750           NA
## 4  WmAv4  29.00  30.00  50.00  300  500  600           NA
## 5 WmhAv2  30.00  40.00  50.00   90   NA   NA           NA
## 6 WmhAv4  29.90  30.00  33.00   80   NA   NA           NA
## 7   WynN   0.44   0.45   0.65   NA   NA   NA           NA
```

The factors to be evaluated here are the following:

- `WmAv1` - Mean precipitation of first month (mm);
- `WmAv2` - Mean precipitation of second month (mm);
- `WmAv3` - Mean precipitation of third month (mm); and
- `WmAv4` - Mean precipitation of fourth month (mm).

If sowing month is set to November, then we have

- `WmAv1` - November;
- `WmAv2` - December;
- `WmAv3` - January; and
- `WmAv4` - February.

So for Novermber, we see the first land unit falls within the domain of S1, that is, 277 mm falls within [175 - 500 mm). And same thing for the first land unit of December, highly suitable. Let's fire up the function to confirm that,

```
rice_suit_water <- suitability(x = MarinduqueWater, y = RICEBRWaterCR)

## Warning:  For water characteristic, make sure to input sowing month
(sow.month), say 1, w/c implies January
## Error:  No factor(s) to be evaluated, since none matches with the crop
requirements.
```

You will have this error if there is no factors to be evaluated. What just happened here is that, the function assumed the data as neither water nor temperature characteristics. Thus, it ignores the `WmAv1`, `WmAv2`, `WmAv3` and `WmAv4` factors. But if we specify the sowing month (`sow.month`) to November (`11`), then we have

```r
rice_suit_water <- suitability(
  x = MarinduqueWater, y = RICEBRWaterCR, sow.month = 11
  )
lapply(rice_suit_water, function(x) head(x, n = 6))
```

```
## $`Actual Factors Evaluated`
## [1] "Nov" "Dec" "Jan" "Feb"
##
## $`Suitability Score`
##      Nov    Dec    Jan    Feb
## 1 0.8207 0.7348 0.3052 0.3714
## 2 0.8178 0.7319 0.3081 0.3714
## 3 0.8148 0.7319 0.3081 0.3771
## 4 0.8089 0.7259 0.3111 0.3714
## 5 0.8059 0.7259 0.3111 0.3714
## 6 0.7941 0.7141 0.3141 0.3771
##
## $`Suitability Class`
##   Nov Dec Jan Feb
## 1  S1  S2  S3  S3
## 2  S1  S2  S3  S3
## 3  S1  S2  S3  S3
## 4  S1  S2  S3  S3
## 5  S1  S2  S3  S3
## 6  S1  S2  S3  S3
##
## $`Factors' Minimum Values`
## Nov Dec Jan Feb
##   0   0   0   0
##
## $`Factors' Maximum Values`
##   Nov   Dec   Jan   Feb
## 880.0 880.0 880.0 714.2
```

The first land unit for November does confirms to be S1, but for December it isn't, and instead S2 is given. This problem will be discussed later on details about the `interval` argument.

Options for `min` (Factors' Minimum Value)

By default, `min = 0` for all factors. This can be assigned to any positive integers, for example, using the cassava soil requirements,

```r
# Default min = 0
cassava_suit_soil <- suitability(x = MarinduqueLT, y = CASSAVASoilCR)
cassava_suit_soil$`Factors' Minimum Values`
```

```
##   CECc SoilTe
##      0      0
```

```
# Change min = 1
cassava_suit_soil <- suitability(
  x = MarinduqueLT, y = CASSAVASoilCR, min = 1
  )
cassava_suit_soil$`Factors' Minimum Values`

##   CECc SoilTe
##      1      1

# Change min = 2
cassava_suit_soil <- suitability(
  x = MarinduqueLT, y = CASSAVASoilCR, min = 2
  )
cassava_suit_soil$`Factors' Minimum Values`

##   CECc SoilTe
##      2      2
```

Now let's try different minimums for factors, we will utilize the following:

	Factors			
	CECc	pHH20	CFragm	SoilTe
`min`	0.4	0.6	0.1	0.3

```
minimum <- c(0.4, 0.6, 0.1, 0.3)
minimum

## [1] 0.4 0.6 0.1 0.3

# Use these minimum
cassava_suit_soil <- suitability(
  x = MarinduqueLT, y = CASSAVASoilCR, min = minimum
  )

## Error:  min length should be equal to the number of factors in x.
```

So we got an error, it is expected, since the length of the vector `min` should be equal to the number of factors in `x`, which is 6. Since we are not interested on the latitude (`X`) and longitude (`Y`) factors of the dataset, then we can ommit the two and rerun the code,

```
cassava_suit_soil <- suitability(
  x = MarinduqueLT[, -(1:2)], y = CASSAVASoilCR, min = minimum
  )

cassava_suit_soil$`Factors' Minimum Values`
```

```
##   CECc SoilTe
##    0.4    0.3
```

Only CECc and SoilTe are returned since these are the factors evaluated.

Options for `max` (Factors' Maximum Value)

By default `max = 'average'`, and just like `min`, `max` can be assigned to any positive integer, example:

```
# Default max = 'average'
cassava_suit_soil <- suitability(x = MarinduqueLT, y = CASSAVASoilCR)
cassava_suit_soil$`Factors' Maximum Values`

##   CECc SoilTe
##  16.05  12.00

# Change max = 110
cassava_suit_soil <- suitability(
  x = MarinduqueLT, y = CASSAVASoilCR, max = 110
  )
cassava_suit_soil$`Factors' Maximum Values`

##   CECc SoilTe
##    110    110
```

For different maximum value on every factor, we will utilize the following and ommit the first two factors in `MarinduqueLT` like what we did in the previous section.

	Factors			
	CECc	pHH20	CFragm	SoilTe
`max`	52.5	8.8	40	14

```
maximum <- c(52.5, 8.8, 40, 14)
banana_suit_soil <- suitability(
  x = MarinduqueLT[, -(1:2)], y = CASSAVASoilCR, max = maximum
  )
banana_suit_soil$`Factors' Maximum Values`

##   CECc SoilTe
##   52.5   14.0
```

Options for `interval` (Domain of Suitability Scores)

The domain of suitability scores are set to default, `'fixed'`, if this option is used, the domain of the suitability scores would be that in Table 2.

Class	N	S3	S2	S1
Domain	[0, 0.25)	[0.25, 0.5)	[0.5, 0.75)	[0.75, 1]

Table 2: Domain for 'fixed'

An example of `interval = 'fixed'` is the one illustrated in Section 4.1. Let us investigate the output of that, here is the crop requirements for water (the crop we are interested in, is the rainfed bunded rice),

```
RICEBRWaterCR

##      Code     S3     S2     S1 S1.1 S2.1 S3.1 Weight.class
## 1  WmAv1 100.00 125.00 175.00  500  650  750           NA
## 2  WmAv2 100.00 125.00 175.00  500  650  750           NA
## 3  WmAv3 100.00 125.00 175.00  500  650  750           NA
## 4  WmAv4  29.00  30.00  50.00  300  500  600           NA
## 5 WmhAv2  30.00  40.00  50.00   90   NA   NA           NA
## 6 WmhAv4  29.90  30.00  33.00   80   NA   NA           NA
## 7   WynN   0.44   0.45   0.65   NA   NA   NA           NA
```

Given that the starting sowing month assigned is November, then the following factors are evaluated:

- `WmAv1` - November;
- `WmAv2` - December;
- `WmAv3` - January; and
- `WmAv4` - February.

So we are going to extract this factors from the dataset, `MarinduqueWater`,

```
head(MarinduqueWater[c(13, 14, 3, 4)])

##   Nov Dec Jan Feb
## 1 277 248 103  65
## 2 276 247 104  65
## 3 275 247 104  66
## 4 273 245 105  65
## 5 272 245 105  65
## 6 268 241 106  66
```

The suitability scores and class of this would be,

```
head(rice_suit_water[[2]])

##      Nov    Dec    Jan    Feb
## 1 0.8207 0.7348 0.3052 0.3714
```

```
## 2 0.8178 0.7319 0.3081 0.3714
## 3 0.8148 0.7319 0.3081 0.3771
## 4 0.8089 0.7259 0.3111 0.3714
## 5 0.8059 0.7259 0.3111 0.3714
## 6 0.7941 0.7141 0.3141 0.3771

head(rice_suit_water[[3]])

##   Nov Dec Jan Feb
## 1  S1  S2  S3  S3
## 2  S1  S2  S3  S3
## 3  S1  S2  S3  S3
## 4  S1  S2  S3  S3
## 5  S1  S2  S3  S3
## 6  S1  S2  S3  S3
```

Focus your attention on suitability scores of `Feb` factor for the first three land units. We have here $0.3714, 0.3714$ and 0.3771. And the domain of this base on Table 2, would be S3, S3 and S3. But, if we refer to the original data, the first three data points in `Feb` factor are all 65. Since `WmAv4` is the corresponding requirements for `Feb` factor, with scores:

Factor	S3	S2	S1	S1	S2	S3	Weight
WmAv4	29	30	50	300	500	600	NA

Table 3: `WmAv4`'s Suitability Requirements.

Then it is easy to pin point what suitability class does the scores of the land units falls into. Which follows that all first three land units falls within class S1. See the problem with `'fixed'` interval? This is the same problem for other factor like `Dec` (December), where instead of S1, we got S2. Users can change the domain though, that is, instead of using the `'fixed'` option, users can assign for example, `interval = c(0, 0.33, 0.56, 0.89, 1)`, which equivalently:

Class	N	S3	S2	S1
Domain	[0, 0.33)	[0.33, 0.56)	[0.56, 0.89)	[0.89, 1]

Table 4: Custom Domains

Assigning new values for parameters of the `interval` won't solve the problem, but this argument has one more option to offer, which does solve the problem, and that is by changing `interval = 'fixed'` to `interval = 'unbias'`. Let's try it,

```
rice_suit_water <- suitability(x = MarinduqueWater, y = RICEBRWaterCR,
    sow.month = 11, interval = "unbias")
lapply(rice_suit_water, function(x) head(x, n = 6))

## $`Actual Factors Evaluated`
```

```
## [1] "Nov" "Dec" "Jan" "Feb"
##
## $`Suitability Score`
##      Nov    Dec    Jan    Feb
## 1 0.8207 0.7348 0.3052 0.3714
## 2 0.8178 0.7319 0.3081 0.3714
## 3 0.8148 0.7319 0.3081 0.3771
## 4 0.8089 0.7259 0.3111 0.3714
## 5 0.8059 0.7259 0.3111 0.3714
## 6 0.7941 0.7141 0.3141 0.3771
##
## $`Suitability Class`
##   Nov Dec Jan Feb
## 1  S1  S1  S3  S1
## 2  S1  S1  S3  S1
## 3  S1  S1  S3  S1
## 4  S1  S1  S3  S1
## 5  S1  S1  S3  S1
## 6  S1  S1  S3  S1
##
## $`Factors' Minimum Values`
## Nov Dec Jan Feb
##   0   0   0   0
##
## $`Factors' Maximum Values`
##   Nov   Dec   Jan   Feb
## 880.0 880.0 880.0 714.2
```

And that supports our argument above.

Weighting

The function, `suitability`, also considers the weights of the factors. An example of crop with no weights is the soil requirement for coconut,

```
ALUES::COCONUTSoilCR

##      Code   S3   S2    S1 S1.1 S2.1 S3.1 Weight.class
## 1  CFragm 55.0 35.0  15.0   NA   NA   NA           NA
## 2 SoilDpt 50.0 75.0 100.0   NA   NA   NA           NA
## 3      BS 19.9 19.9  20.0   NA   NA   NA           NA
## 4  SumBCs  1.5  1.5   1.6   NA   NA   NA           NA
## 5      OC  0.7  0.7   0.8   NA   NA   NA           NA
## 6   ECemh 20.0 16.0  12.0   NA   NA   NA           NA
```

The weights are assigned on the last column, `Weight.class`. And here is the soil requirements for the cassava, with weight on each factor:

```
ALUES::CASSAVASoilCR

##        Code   S3   S2    S1 S1.1 S2.1 S3.1 Weight.class
## 1   CFragm1 35.0 15.0   3.0   NA   NA   NA            2
## 2   CFragm2 55.0 35.0  15.0   NA   NA   NA            3
## 3   SoilDpt 50.0 75.0 100.0   NA   NA   NA            2
## 4     CaCO3 10.0  5.0   1.0   NA   NA   NA            3
## 5      Gyps  2.0  1.0   0.2   NA   NA   NA            3
## 6      CECc 15.9 15.9  16.0   NA   NA   NA            3
## 7        BS 20.1 20.1  20.0   NA   NA   NA            3
## 8    SumBCs  2.1  2.1   2.0   NA   NA   NA            3
## 9     pHH2O  4.5  4.8   5.2    7  7.6  8.2            3
## 10       OC  0.9  0.9   0.8   NA   NA   NA            3
## 11    ECedS  4.0  3.0   2.0   NA   NA   NA            3
## 12   SoilTe  3.0  4.0   9.0   NA   NA   NA            2
```

If a given factor has a weight, then the function will compute the corresponding suitability and then use the weighting score to obtain the appropriate suitability score. The weights of the factors for the default interval (`interval = 'fixed'`) are in Table 5:

Suitability	Factor Weight		
Class	1	2	3
S1	0.833	0.916	1.000
S2	0.583	0.667	0.750
S3	0.333	0.416	0.500
N	0.083	0.167	0.250

Table 5: Weights of the Factors for `'fixed'` Interval.

Thus the function simply divides the interval of the suitability class into three, for three weights.

4.2 Overall Suitability

```
overall_suit(x, method = NULL, interval = NULL, output = NULL)
```

- **`x`** a data frame consisting the suitability scores of a given characteristics (terrain, soil, water and temperature) for a given crop (e.g. coconut, cassava, etc.);
- **`method`** the method for computing the overall suitability, which includes the minimum, maximum, sum, product, average, exponential and gamma. If `NULL`, minimum is used.
- **`interval`** if `NULL`, the interval used are the following: 0-25% (Not suitable, N), 25%-50% (Marginally Suitable, S3), 50%-75% (Moderately Suitable, S2), and 75%-100% (Highly Suitable, S1).

`output` the output to be returned, either the scores or class. If `NULL`, both are returned.

5 Demonstration

Let's assume we are interested on the land units in Lao Cai, Vietnam, for cultivating irrigated rice. So here are the first 6 land units in the said location,

```
library(ALUES)
head(LaoCaiLT) # Land and Terrain Characteristics

##   SlopeD CFragm SoilDpt SoilTe CECc SumBCs pHH2O    BS   OC
## 1      1      0      45      9 18.6   3.30   5.4 50.00 1.40
## 2      1      0      45      9 18.6   3.30   5.4 50.00 1.40
## 3      1      0      45      9 18.6   3.30   5.4 50.00 1.40
## 4      1      0      45      9 18.6   3.30   5.4 50.00 1.40
## 5      1      0      45      9 18.6   3.30   5.4 50.00 1.40
## 6      1      5      65      3 19.8   3.13   5.3 34.72 1.21
##   Flood
## 1     3
## 2     3
## 3     3
## 4     3
## 5     3
## 6     2

head(LaoCaiTemp) # Temperature Characteristics

##     Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct
## 1 11.19 12.55 16.34 19.96 22.68 23.67 23.84 23.28 22.01 19.46
## 2 12.04 13.32 17.10 20.26 22.35 23.29 23.37 23.04 21.91 19.50
## 3 12.27 13.53 17.30 20.47 22.64 23.58 23.64 23.29 22.16 19.77
## 4 15.91 16.11 20.09 23.01 25.95 27.50 27.04 26.64 25.27 23.33
## 5 15.39 15.76 19.70 22.67 25.50 26.93 26.55 26.16 24.83 22.83
## 6 13.13 14.28 18.06 21.13 23.40 24.37 24.32 23.97 22.85 20.58
##     Nov   Dec
## 1 15.85 12.46
## 2 15.99 13.02
## 3 16.23 13.24
## 4 19.36 16.73
## 5 18.92 16.22
## 6 16.96 13.98

head(LaoCaiWater) # Water Characteristics

##     Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep    Oct
## 1 20.49 32.77 47.80 124.2 177.9 268.3 340.6 367.6 237.7 123.71
## 2 35.71 50.45 73.00 155.6 261.9 341.2 409.6 411.9 261.3 136.69
```

```
## 3 33.63 47.38 71.10 151.6 250.3 331.7 402.7 404.6 252.6 128.94
## 4 20.74 27.21 96.28 132.8 184.5 191.7 339.2 353.0 229.9  76.14
## 5 22.17 29.49 91.25 134.4 191.3 212.3 347.4 358.5 229.3  81.75
## 6 30.17 41.50 72.47 147.0 231.1 311.2 389.6 389.5 231.3 108.18
##     Nov   Dec Irrigation
## 1 65.80 20.48          1
## 2 75.82 31.68          1
## 3 73.43 29.21          1
## 4 79.89 12.06          1
## 5 77.71 14.03          1
## 6 69.60 23.80          1
```

And here are the required values for factors of soil, terrain, temperature and water characteristics for irrigated rice,

```
RICEIWTerrainCR # Terrain Characteristics

##       Code S3 S2 S1 S1.1 S2.1 S3.1 Weight.class
## 1   Slope1  4  2  1   NA   NA   NA           NA
## 2    Flood  1  2  3   NA   NA   NA            1
## 3 Drainage  5  4  2   NA   NA   NA            2
## 4   SlopeD  3  2  1   NA   NA   NA            1

RICEIWSoilCR # Soil Characteristics

##       Code   S3   S2   S1 S1.1 S2.1 S3.1 Weight.class
## 1   CFragm 35.0 15.0  3.0   NA   NA   NA            3
## 2  SoilDpt 20.0 50.0 75.0   NA   NA   NA            3
## 3    CaCO3 25.0 15.0  6.0   NA   NA   NA            3
## 4     Gyps 25.0 10.0  3.0   NA   NA   NA            3
## 5     CECc 10.0 12.0 16.0   NA   NA   NA            3
## 6       BS 20.0 35.0 50.0   NA   NA   NA            2
## 7   SumBCs  1.6  2.8  4.0   NA   NA   NA            3
## 8       OC  0.7  0.8  1.5   NA   NA   NA            2
## 9    ECedS  6.0  4.0  2.0   NA   NA   NA            3
## 10     ESP 40.0 30.0 20.0   NA   NA   NA            3
## 11   pHH2O  4.5  5.0  5.5  8.2  8.5  8.8            3
## 12  SoilTe  3.0  4.0  9.0   NA   NA   NA            2

RICEIWTemperatureCR # Temperature Characteristics

##       Code S3 S2 S1 S1.1 S2.1 S3.1 Weight.class
## 1     TgAv 10 18 24   36 36.1 36.1            1
## 2  TmMaxXm 21 26 30   40 45.0 50.0           NA
## 3    TmAv2 10 18 24   36 42.0 45.0            2
## 4 TmMinAv4  7 10 14   25 28.0 30.0           NA

RICEIWWaterCR # Water Characteristics
```

```
##      Code     S3     S2     S1 S1.1 S2.1 S3.1 Weight.class
## 1   WmAv1 100.00 125.00 175.00  500  650  750           NA
## 2   WmAv2 100.00 125.00 175.00  500  650  750           NA
## 3   WmAv3 100.00 125.00 175.00  500  650  750           NA
## 4   WmAv4  29.00  30.00  50.00  300  500  600           NA
## 5 WmhAv2  30.00  40.00  50.00   90   NA   NA           NA
## 6 WmhAv4  29.90  30.00  33.00   80   NA   NA           NA
## 7    WynN   0.44   0.45   0.65   NA   NA   NA           NA
```

Now, we are going to take the suitability scores for every characteristics,

```r
x1 <- LaoCaiLT; y1 <- RICEIWTerrainCR; y2 <- RICEIWSoilCR;
x2 <- LaoCaiTemp; y3 <- RICEIWTemperatureCR;
x3 <- LaoCaiWater; y4 <- RICEIWWaterCR

# Terrain Characteristics
riwTerr_suit <- suitability(x = x1, y = y1, interval = "unbias")

# Soil Characteristics
riwSoil_suit <- suitability(x = x1, y = y2, interval = "unbias")

# Temperature Characteristics
riwTemp_suit <- suitability(
  x = x2, y = y3,
  interval = "unbias",
  sow.month = 10
  )

# Water Characteristics
riwWatr_suit <- suitability(
  x = x3, y = y4,
  interval = "unbias",
  sow.month = 10
  )
```

Next, we will take the overall suitability on all factors in each land unit using the `"average"` method (default is `"minimum"`).

```r
# Terrain Characteristics
riwTerr_ovs <- overall_suit(riwTerr_suit[[2]], method = "average")

# Soil Characteristics
riwSoil_ovs <- overall_suit(riwSoil_suit[[2]], method = "average")

# Temperature Characteristics
riwTemp_ovs <- overall_suit(riwTemp_suit[[2]], method = "average")

# Water Characteristics
```

```
riwWatr_ovs <- overall_suit(riwWatr_suit[[2]], method = "average")

# Combine scores into data frame
rFact_ovs1 <- data.frame(
  "Terrain" = riwTerr_ovs[, 1],
  "Soil" = riwSoil_ovs[, 1],
  "Temp" = riwTemp_ovs[, 1],
  "Water" = riwWatr_ovs[, 1]
  )

# Combine classes into data frame
rFact_ovs2 <- data.frame(
  "Terrain" = riwTerr_ovs[, 2],
  "Soil" = riwSoil_ovs[, 2],
  "Temp" = riwTemp_ovs[, 2],
  "Water" = riwWatr_ovs[, 2]
  )

# Combine the two data frame into a list
rFact_ovs <- list("Scores" = rFact_ovs1, "Class" = rFact_ovs2)

# Return the first 10 of the output
lapply(rFact_ovs, function(x) head(x, n = 10))
```

```
## $Scores
##    Terrain   Soil   Temp   Water
## 1   0.8333 0.4878 0.4529 0.06068
## 2   0.8333 0.4878 0.4569 0.09387
## 3   0.8333 0.4878 0.4637 0.08655
## 4   0.8333 0.4878 0.5531 0.03573
## 5   0.8333 0.4878 0.5406 0.04157
## 6   0.5833 0.0000 0.4846 0.07052
## 7   0.5833 0.0000 0.4889 0.06353
## 8   0.5833 0.0000 0.4886 0.07004
## 9   0.5833 0.0000 0.4797 0.06753
## 10  0.5833 0.0000 0.5020 0.05742
##
## $Class
##    Terrain Soil Temp Water
## 1       S1   S3   S3     N
## 2       S1   S3   S3     N
## 3       S1   S3   S3     N
## 4       S1   S3   S2     N
## 5       S1   S3   S2     N
## 6       S2    N   S3     N
## 7       S2    N   S3     N
## 8       S2    N   S3     N
## 9       S2    N   S3     N
```

```
## 10       S2    N   S2     N
```

Finally, we will take the overall suitability from these characteristics using the `"maximum"` method.

```
riw_ovs <- overall_suit(rFact_ovs[[1]], method = "maximum")
head(riw_ovs, n = 10)

##    Scores Class
## 1  0.8333    S1
## 2  0.8333    S1
## 3  0.8333    S1
## 4  0.8333    S1
## 5  0.8333    S1
## 6  0.5833    S2
## 7  0.5833    S2
## 8  0.5833    S2
## 9  0.5833    S2
## 10 0.5833    S2
```

6 Appendix

List of factors from International Rice Research Institute (IRRI)

Description	**Code**
Base Saturation (%)	BS
CaCO3 (%)	CaCO3
Apparent CEC (cmol (+)kg clay)	CECc
Apparent CEC (cmol (+)kg soil)	CECs
Coarse fragm (vol %)	CFragm
Coarse fragm(vol %) in surface	CFragm1
Coarse fragm (vol %) in depth	CFragm2
ECe (dSm)	ECedS
ECe (mmhoscm)	ECemh
ESP (%)	ESP
Gypsum (%)	Gyps
Organic carbon(%)	OC
Organic carbon (%) (6)	OC6
Organic carbon (%) (7)	OC7
Organic carbon (%) (8)	OC8
pH H2O	pHH2O
Soil depth (cm)	SoilDpt
Surface depth (cm)	SoilDpt1
Texturestruct.	SoilTe
Sum of basic cations (cmol (+)kg of clay)	SumBCc
Sum of basic cations (cmo (+)kg soil)	SumBCs
(Tmax-Tmin)Tmean maturation stage	Tcoef
Mean day temp. at germination (°C)	TdAvg0
Mean day temp. for tillage stage (°C)	TdAvg1
Mean day temp. for vegetative stage (°C)	TdAvg2
Mean dAY temp. of flowering stage (°C)	TdAvg3
Av. Temp. difference between day-night (°C)	TdAvgDiff
Temp diffrence daynight (°C)	TdDiff
Temp. diff day/night flowering stage (°C)	TdDiff3
Mean NIGHT temp. of flowering stage (°C)	TdMinN3
Mean daily min. temp. of coldest month (°C)	TdMinXm
Mean temp. of the growing cycle (°C)	TgAv
Mean max temp. of growing cycle (°C)	TgMaxAv
Meam min. temp. of growing cycle (°C)	TgMinAv
Mean temp at germination (°C)	TmAv0
Mean temp of the initial stage (°C)	TmAv1

Description	**Code**
Mean temp of the vegetative stage(2nd month)(°C)	TmAv2
Mean temp of the flowering stage(3rd month)(°C)	TmAv3
Mean temp of the ripening stage(4-5th month)(°C)	TmAv4
Mean temp. of 4 warmest month (°C)	TmAv4Xm
No of months with mean temp > 38 °C	TmMax38
Average max. temp. warmest month (°C)	TmMaxXm
No of months with mean temp < 13 °C	TmMin13
Mean min. temp. of warmest month (°C)	TmMinAv
Mean min. temp. at germination (°C)	TmMinAv0
Average absol. Min. temp. of the 1st month (°C)	TmMinAv1
Average absol. Min. temp. of the 2nd month (°C)	TmMinAv2
Average absol. Min. temp. of the 3rd month (°C)	TmMinAv3
Average absol. Min. temp. of the 4th month (°C)	TmMinAv4
Average min. temp. coldest month (°C)	TmMinXm
Absolute min temp coldest month (°C)	TmMinXmAb
Mean annual temp. (°C)	TyAv
Mean annual max. temp. (°C)	TyMaxAv
Absol. Min temperature (°C)	TyMinAb
Mean annual min. (°C)	TyMinAv
Relative humidity maturation stage (%)	WmhAv3
Drainage	Drainage
Drainage (4)	Drainage4
Drainage (5)	Drainage5
Flooding (1: No Flood, 2: short time; 3: Long time)	Flood
Slope (%) (1)	Slope1
Slope (%) (2)	Slope2
Slope (%) (3)	Slope3
SlopeD	SlopeD
Length of growing period (days)	CropLen
Sum of basic cations (cmol(+)kg soil)	SumBCs
Sunshine : hoursyear	SunH
Average daylength growing cycle (h)	TgAvDlen
Daylength (h) during yield form. Period	TmAvDlen3
10 days of rainfall (mm)	Wd10
Precipitation of the growing cycle (mm)	WgAv
Relative humidity growing cycle (%)	WghAv
nN growing cycle	WgnN

Description	**Code**
Mean precipitation of 1st month (mm)	WmAv1
Mean precipitation of 2nd month (mm)	WmAv2
Mean precipitation of 3rd month (mm) (flowering)	WmAv3
Mean precipitation of 4th month (mm)	WmAv4
Precipitation of the 5th month (mm)(yield formation)	WmAv5
Precipitation of ripening stage (mm)(6th month)	WmAv6
Monthly precipitation during dry season (mm)	WmAvDry
Length dry season (months : P < 12 PET)	WmDryLen
Months of excessive rain (x)	WmER
Relative humidity of devel. Stage (%)	WmhAv2
Relative humid. of maturation stage (%)	WmhAv3
Relative humidity at harvest (%)	WmhAv4
Relative humidity of coldest month if frost (%)	WmhColdXm
Mean rel. humidity dryest month (%)	WmhDryXm
nN develop stage	WmnN2
nN maturation stage	WmnN4
nN of the 5 dryest months	WmnN5
Monthly rainfall during the sclerification of stone (mm) - August (N hem) February (S hem.)	WmSpecial1
Monthly rainfall during the sclerification of stone (mm) - September (N hem) March (S hem.)	WmSpecial2
Annual precipitation (mm)	WyAv
Annua relative humidity (%)	WyhAv
Mean annual nN	WynN