TATALE: Tools for Assessment with Transformation and Aggregation using simple Logic and Expertise

Manual (Version March 2017)

Authors

Simon Taugourdeau [1,2]

Samir Messad[1]

[1] CIRAD, UMR SELMET, F-34398 Montpellier, France
[2] ISRA LNERV- DP PPZS, Route du front de Terre Dakar, Sénégal

E-mail address :

Simon Taugourdeau: simon.taugourdeau@cirad.fr
Samir Messad: samir.messad@cirad.fr

URL :
http://umr-selmet.cirad.fr/les-produits-et-expertises/produits/tatale

Contents

Overview

The TATALE tool's objective is to sum up a set of variables by creating synthetic scores. TATALE tool has two steps. The first step is the transformation which consists in transforming all the variables into scores that take their values from 0 to 1. The second step is the aggregation. All scores obtained from the transformation are aggregated using simple mathematical methods (mean, minimum, maximum). The adjustment of the parameters of the two steps is defined by users' expertise.

TATALE deals with qualitative and quantitative variables. For qualitative variables, the user chooses values between 0 and 1 to be assigned to each modality. For quantitative variables, users propose transformation based on linear models. He defines a range for the variable where the model would be applied (for example between 5 and 20). Then, user sets a score between 0 and 1 for the two borders of the range (for example 0.2 for 5 and 0.9 for 20). All the values of the variable between 5 and 20 are predicted by the linear model. For one variable, the user can define as many intervals and associated models as he wants.

However, all the transformation thresholds (range and associated scores) required good quality expertise not necessarily available. The function **`prestrans`** can be used to make a transformation based on an idea of the form of the relationships between variables value and scores. It is necessary to have a representative dataset that will be used to adjust the transformation. The user must then choose among the 20 proposed forms (Figure 2) those are best suited to the transformation of the variable. The function **`prestrans`** provides intervals and associated scores that are editable as needed.

In the case where the user would provide a data table and the table of the transformation parameter set, the transformed variables are obtained using the **`trans`** function. A graphical view on the shape of the transformations is possible with the **`graphtrans`** function (see figure 3).

The step of aggregation also relies on expertise. Users choose variables that he wants to aggregate to create new synthetic scores. Synthetic scores can be aggregated as well to create new synthetic scores. Three simple arithmetic aggregation operations are proposed: (i) weighted average of scores (ii) minimum value of scores (iii) maximum value of scores. The number of variables/scores that can be aggregated and the number of successive aggregation is not limited. We can aggregate transformed variables and synthetic scores as well.

`aggr` function of the package is used to aggregate scores or a set of variables which values varying between 0 and 1. Function **`graphaggr`** displays a graphical representation (agregation tree) of the sequence of aggregation.

Finally, the user can perform both transformation and aggregation with **`tatale`** function.

In order to have a synthetic view of the links between functions and steps of the calculation of scores, the figure 1 presents a synoptic view of connections between TATALE functions.

The following part EXAMPLE presents a short example of the construction of synthetic scores. Finally, the part FUNCTIONS describes in detail the parameters and the outputs of the functions of TATALE package.

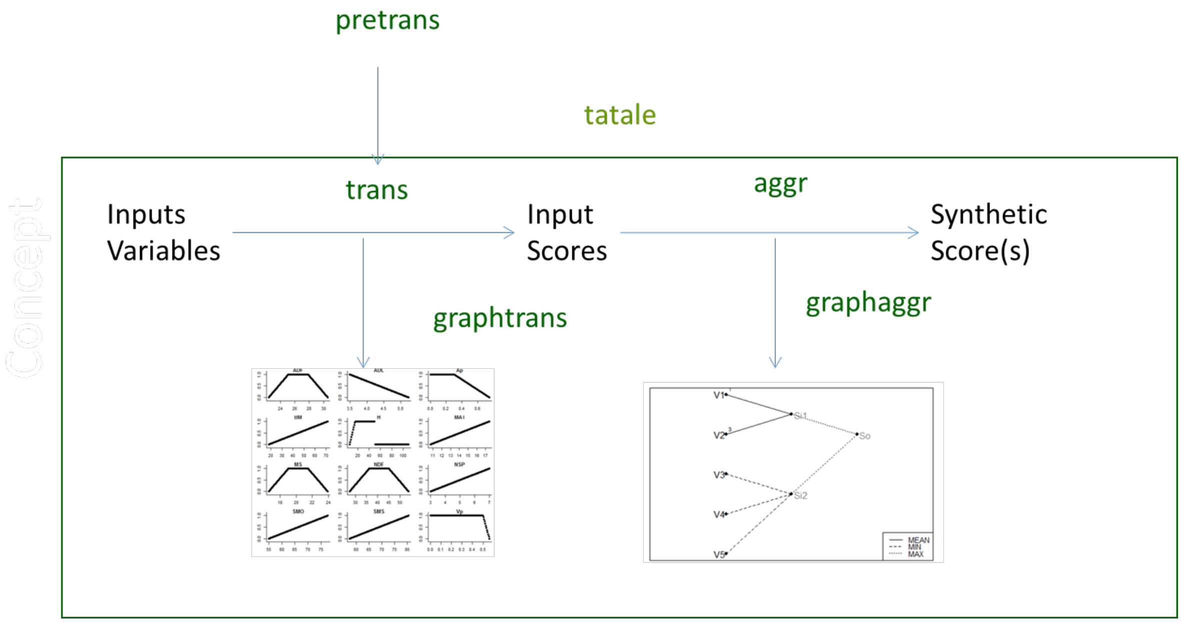

Figure 1. Synoptic view of tatale package functions.

Example

```
set.seed(15)
dataex <- as.data.frame(matrix(ncol = 5,nrow = 100))
colnames(dataex) <- c("V1","V2","V3","V4","V5")
dataex$V1 <- runif(100,min = 100,max = 2000)
dataex$V2 <- runif(100,min = 0.1,max = 0.3)
dataex$V3 <- as.factor(ifelse(runif(100,min=0,max=1)<0.5,"A","B"))
dataex$V4 <- runif(100,min = -5 ,max = 2)
dataex$V5 <- rnorm(100,mean = 12,sd = 5)

### prestans function
fittabex <- data.frame(variable = factor(colnames(dataex)),
                  form  = factor(c("L.I","CLC.D","","CCC.TI","LL.TI")))
tabtransex <- pretrans(dataex, fittabex, graphpretrans = TRUE)

### transformations
graphtrans(tabtransex)
#### transformation
datatex <- trans(data = dataex, tabtrans = tabtransex)
min(datatex)
max(datatex)

#### aggregation
ON <- c("A1","A1","A2","A2","A2","A3","A3")
IN <- c("V1","V2","V3","V4","V5","A1","A2")
method <- c("MEAN","MEAN","MIN","MIN","MIN","MAX","MAX")
weight <- c(2,1,0,0,0,0,0)
tabaggrex <- as.data.frame(cbind(ON,IN,method,weight))
outputa <- aggr(datat=datatex,tabaggr=tabaggrex)

##graphique aggregation
graphaggr(tabaggrex, orientation = "LR",space = 1)

#tatale
TA <- tatale(dataex,tabtransex,tabaggrex,graphtrans = TRUE,graphaggr = FALSE)
TA$aggr
TA$trans
```

Functions

pretrans

Description

From a dataset and a choice of preset forms of transformation, `pretrans` calculate a set of parameters for the transformation of initial variables.

Usage

```
pretrans(data, fittab, graphtrans = FALSE)
```

Arguments

- `data` is the dataset that will be used to calculate transformation parameters. The variables can be quantitative or qualitative.
- `fittab` is a data.frame with two columns encoded as factor. The first column contains the name of the variable that will be transformed. Be careful to use exactly the same name for the variables as in dataset `data`. The second column contains the code of transformation preset forms as presented in the Figure 2 for quantitative variables. For qualitative variables, no information is required.
- `graphtrans` is optional. It creates a graphical representation of transformed variables from the data frame `data`. Each transformed variable is represented in a panel. X-axis is the value of the variable and in the Y-axis the associate scores.

Details

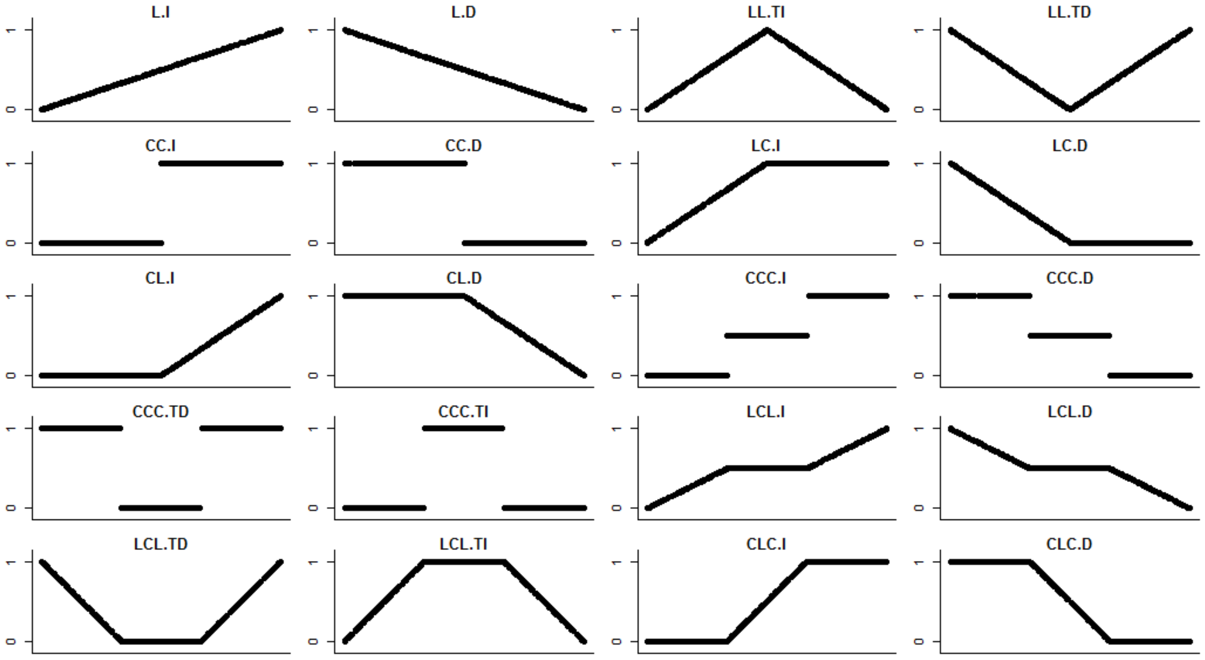

Figure 2 : Graphical representation of the 20 preset forms of transformation of the variables into scores varying between 0 and 1. X-axis represents the variable value and y-axis the associate scores. The code above each preset form is to be used in the `restrains` function.

20 preset forms are proposed by TATALE (Figure 2). For example, in the case of the L.I transformation, the score of 0 and 1 are attributed respectively to the minimum and the maximum value of the variable in the dataset. Some forms are constructed from 2 or 3 linear functions. Depending on the number of

linear functions, the thresholds are defined in the middle, one third or two thirds of the range of the values of the variable to be transformed. For qualitative variables, the function proposes one value score for each modality. The scores are ordered numerical values (0,1, 2, 3, ...) assigned in alphabetical order of the modalities of the variable to be transformed.

Outputs

A data frame with transformation parameters. For the variable with a complex transformation (more than one linear model and qualitative variables), several rows are designed.

- The first column « variable » encoded as a factor is the name of the variables.
- The second column "type" is the type of the variable: "QUAN" for quantitative variable, "QUAL" for qualitative variable.
- The third column "categories" describe categories of qualitative variable.
- The four other columns (v1, s1, v2, s2) are parameters for the transformation. For qualitative variables, only the first column s1 is given. For each modality, the associated score is in the column s1. For quantitative variables, values v1 and v2 are thresholds of the interval of the transformed variable for which linear functions chosen by user are adjusted. Columns s1 and s2 give the associated scores respectively for v1 and v2.

Example

```
set.seed(15)
dataex <- as.data.frame(matrix(ncol = 5,nrow = 100))
colnames(dataex) <- c("V1","V2","V3","V4","V5")
dataex$V1 <- runif(100,min = 100, max = 2000)
dataex$V2 <- runif(100,min = 0.1, max = 0.3)
dataex$V3 <- as.factor(ifelse(runif(100,min = 0,max = 1)<0.5,"A","B"))
dataex$V4 <- runif(100,min = -5, max = 2)
dataex$V5 <- rnorm(100,mean = 12, sd = 5)

###prestans function

fittabex <- data.frame(variable = factor(colnames(dataex)),
                  form  = factor(c("L.I","CLC.D","","CCC.TI","LL.TI")))

tabtransex <- pretrans(dataex, fittabex, graphpretrans = TRUE)
```

trans

Description

Transformation of variables into normalized scores varying between 0 and 1.

Usage

```
trans(data,tabtrans)
```

Arguments

- `data` is the dataset that contains the variables to be transformed.
- `tabtrans`
  A data frame with transformation parameters. For the variable with a complex transformation (more than one linear model and qualitative variables), several rows are designed. This data frame can be calculated by the function `pretrans`.
  - The first column « variable » encoded as a factor is the name of the variables.
  - The second column "type" is the type of the variable: "QUAN" for quantitative variable, "QUAL" for qualitative variable.
  - The third column "categories" describe categories of qualitative variable.
  - The four other columns (v1, s1, v2, s2) are parameters for the transformation. For qualitative variables, only the first column s1 is given. User give for each modality, the associated score in the column s1. For quantitative variables, The 4 columns (v1, s1, v2, s2) should be filled. Values v1 and v2 are thresholds of the interval of the transformed variable for which linear functions chosen by user are adjusted. Columns s1 and s2 give the associated scores respectively for v1 and v2.

Outputs

A data frame that contains the normalized scores varying between 0 and 1 obtained from the transformation of initial variables.

Example

```
set.seed(15)
dataex <- as.data.frame(matrix(ncol = 5,nrow = 100))
colnames(dataex) <- c("V1","V2","V3","V4","V5")
dataex$V1 <- runif(100,min = 100,max = 2000)
dataex$V2 <- runif(100,min = 0.1,max = 0.3)
dataex$V3 <- as.factor(ifelse(runif(100,min = 0,max = 1)<0.5,"A","B"))
dataex$V4 <- runif(100,min = -5,max = 2)
dataex$V5 <- rnorm(100,mean = 12,sd = 5)

###prestans function
fittabex <- data.frame(variable = factor(colnames(dataex)),
                  form  = factor(c("L.I","CLC.D","","CCC.TI","LL.TI")))
tabtransex <- pretrans(dataex,fittabex)

datatrans <- trans(data = dataex,tabtrans = tabtransex)
min(datatrans)
max(datatrans)
```

aggr

Description

This function is used to aggregate variables or scores using three different methods. Successive aggregations is possible.

Usage

```
aggr(datat,tabaggr)
```

Arguments

- `datat` is a table that contains normalized variables varying between 0 and 1. This data frame can be obtained with function `trans` which calculate the transformation.
- `tabaggr` is a data frame provides by the user which describes parameters of the aggregation. A row corresponds to the aggregation of a variable of `datat` into a new variable. These new variables can be aggregated as well to create new synthetic scores. If the user wishes to make successive aggregations, he should specify the aggregation parameters in this order: (i) aggregations of the variables belonging the `datat` table (ii) aggregations between scores created by previous aggregations.
  - The first column should be labeled « ON » and is the name of the new synthetic variable result of the aggregation. It is necessary to create as many rows with the same name in the "ON" column as variables to aggregate. For example, if the new synthetic variable is obtained by the aggregation of 10 variables, the user must specify 10 rows for the new variable. See examples below.
  - The second column should be labeled « IN » and is the name of the variable to be aggregate. Note that theses variables should exist in `datat`.
  - The third column should be labeled « method » and give the method for the aggregation: (i) "MEAN" using the average of scores (ii) "MIN" using the lowest of the values of the variables to be aggregated (iii) "MAX" using the the highest of the values of the variables to be aggregated
  - "weight" is the weight applied if method MEAN is chosen. Choose value: 1 if no weighting.

Outputs

A data frame that contains scores obtained from the aggregation.

Example

```
set.seed(15)
datatex <- matrix(ncol = 5,nrow = 100,runif(500))
colnames(datatex)=c("V1","V2","V3","V4","V5")
ON <- c("A1","A1","A2","A2","A2","A3","A3")
IN <- c("V1","V2","V3","V4","V5","A1","A2")
method <- c("MEAN","MEAN","MIN","MIN","MIN","MAX","MAX")
weight <- c(2,1,0,0,0,0,0)
tabaggrex <- as.data.frame(cbind(ON,IN, method, weight))
output <- aggr(datat=datatex,tabaggr = tabaggrex)
```

graphtrans

Description

graphtrans function produce a graphical representation of the transformation described in the parameter table. The graph is composed of one figure per variable which in the x-axis, the value of the variable and in the y-axis the associate scores. At the opposite of the graphic option of pretrans and TATALE, the graphic do not represent a real data.

Usage

```
graphtrans(tabtrans)
```

Arguments

- tabtrans A data frame with transformation parameters. For more details, see the help of pretrans function, which provides tabtrans as an output or the help of trans function which uses tabtrans as an input parameter.

Outputs

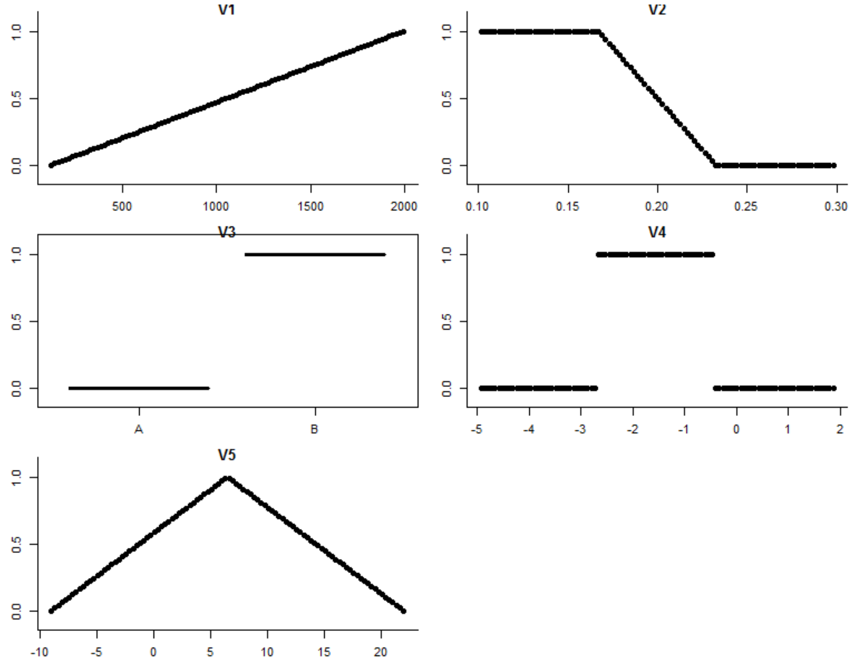

Figure 3 : Example of graphical representation of the transformation of 5 variables with graphtrans function.

Example

```
set.seed(15)
dataex <- as.data.frame(matrix(ncol=5,nrow=100))
colnames(dataex)=c("V1", "V2", "V3", "V4", "V5")
dataex$V1 <- runif(100, min = 100, max = 2000)
dataex$V2 <- runif(100, min = 0.1, max = 0.3)
dataex$V3 <- as.factor(ifelse(runif(100,min = 0,max = 1)<0.5,"A","B"))
dataex$V4 <- runif(100,min = -5,max = 2)
dataex$V5 <- rnorm(100,mean = 12,sd = 5)

###prestans function
fittabex <- data.frame(variable = factor(colnames(dataex)),
                       form   = factor(c("L.I","CLC.D","","CCC.TI","LL.TI")))
tabtransex <- pretrans(dataex, fittabex)

### transformation graphical output
graphtrans(tabtransex)
```

graphaggr

Description

graphaggr produce a graphical representation of the aggregation (aggregation tree) described in the table tabaggr which contains the aggregation for the construction of the scores.

Usage

```
graphaggr(tabaggr, orientation= "LR", space=1)
```

Arguments

- tabaggr is a data frame provides by the user which describes parameters of the aggregation. For more details, see the help of aggr function.
- orientation is the orientation of the aggregation tree. Two options are possible: (i) default option is "LR", the tree is oriented from the left (variables from the transformation) to the right (scores obtained after aggregation). (ii) "TD", the tree is oriented from the top to the bottom.
- space is a graphic parameter to control the distance between labels of variables/scores on the graphic. Default value is 1.

Outputs

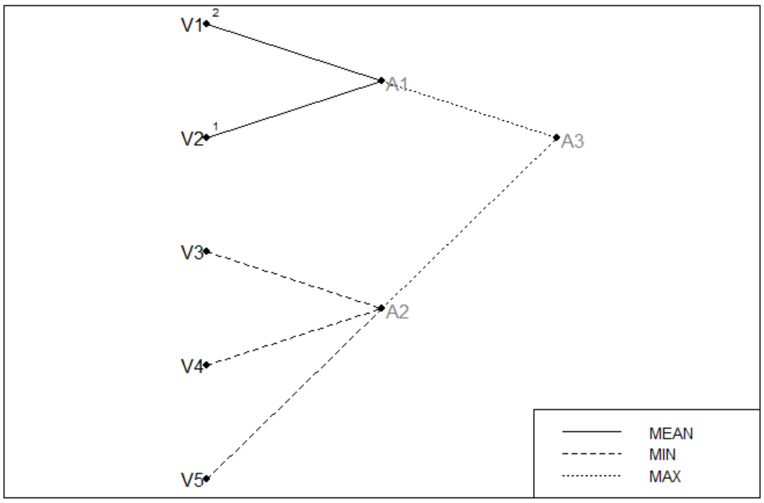

Figure 4 : Example of aggregation obtained with graphaggr function.

Example

```
ON <- c("A1","A1","A2","A2","A2","A3","A3")
IN <- c("V1","V2","V3","V4","V5","A1","A2")
method <- c("MEAN","MEAN","MIN","MIN","MIN","MAX","MAX")
weight <- c(2,1,0,0,0,0,0)
tabaggrex <- as.data.frame(cbind(ON, IN, method, weight))

graphaggr(tabaggrex, orientation="LR", space=1)
```

tatale

Description

tatale performs both transformation and aggregation from a given dataset, set of parameters of transformation and aggregation.

Usage

```
tatale(data,tabtrans, taggr, graphtrans = FALSE, graphaggr = FALSE)
```

Arguments

- data is the dataset that contains the data that need to be transform*ed*
- tabtrans is a data frame with transformation parameters. For more details, see the help of trans function.
- tabaggr is a data frame provides by the user which describes parameters of the aggregation. For more details see the help of aggr function.
- graphtrans is optional. graphtrans is optional. It creates a graphical representation of transformed variables from the data frame data. Each transformed variable is represented in a panel. X-axis is the value of the variable and in the Y-axis the associate scores. See the help of graphtrans function.
- graphaggr is optional. If TRUE, it produces a graphical representation (aggregation tree) of the aggregation of variables and scores described in the table taggr. See the help of graphaggr function.

Outputs

A list containing two data frames.

trans: a data frame that contains the normalized scores varying between 0 and 1 obtained from the transformation of initial variables.

aggr: a data frame that contains scores obtained from the aggregation of transformed variables.

Example

```
set.seed(15)
dataex <- as.data.frame(matrix(ncol=5,nrow=100))
colnames(dataex) <- c("V1","V2","V3","V4","V5")
dataex$V1 <- runif(100,min = 100,max = 2000)
dataex$V2 <- runif(100,min = 0.1,max = 0.3)
dataex$V3 <- as.factor(ifelse(runif(100,min = 0,max = 1)<0.5,"A","B"))
dataex$V4 <- runif(100,min = -5,max = 2)
dataex$V5 <- rnorm(100,mean = 12,sd = 5)

### prestans function

fittabex <- data.frame(variable = factor(colnames(dataex)),
                  form  = factor(c("L.I","CLC.D","","CCC.TI","LL.TI")))

tabtransex <- pretrans(dataex, fittabex, graphpretrans = TRUE)

ON <- c("A1","A1","A2","A2","A2","A3","A3")
IN <- c("V1","V2","V3","V4","V5","A1","A2")
method <- c("MEAN","MEAN","MIN","MIN","MIN","MAX","MAX")
weight <- c(2,1,0,0,0,0,0)
tabaggrex <- as.data.frame(cbind(ON,IN,method,weight))

### tatale
TA <- tatale(dataex,tabtransex,tabaggrex)
TA$aggr
TA$trans
```