R Manual

for the implementation of the Voices of the Hungry
methods to estimate Food Insecurity[i]

VOICES
of the
HUNGRY

[i] The software in R has been developed by Sara Viviani
based on material by Mark Nord and under the supervision of Carlo Cafiero.

R manual

For the implementation of the Voices of the Hungry methods to estimate food insecurity

As the official statistical software of the FAO Voices of the Hungry (VoH) project, R is used for the implementation of methods to estimate food insecurity prevalence. Because of its flexibility in creating new packages, R is the ideal tool to carry out innovative statistical methods. This document presents the main R functions for the psychometric analysis of the Food Insecurity Experience Scale (FIES), a questionnaire used to evaluate the food insecurity severity of a population.

1. Introduction to R and RStudio

R is an open-source statistical environment widely used for data analysis. Thanks to its external packages, it offers a wide variety of statistical methods. To get an overview of its features, visit the task view relevant to your field.

Click here to download R for Mac OS X, Windows or Linux.

Once downloaded, R can be used to program directly in the console, or through a user-friendly compiler, RStudio.

RStudio is an integrated development environment for R. It includes a console, a syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

For online courses you can try the R online simulator
http://tryr.codeschool.com/levels/1/challenges/1
and the some of the lectures by Roger Peng
https://www.youtube.com/watch?v=EiKxy5IecUw

2. Fitting the Rasch model

The R functions used to analyze the FIES data are external. This means that they are not already integrated in the software but need to be uploaded through the corresponding file.

To upload the functions, set the working directory[1] as the one which contains the R scripts, and then type the command:

```
source("Functions.r")
```

in the R console.

---

[1] To set the working directory, use the command `setwd("...")` and include the directory's path between quotes, as in the following example: `setwd("C:\\Users\\Name\\Output")`. Pay attention to the double backslash!

The main function is `RM.w`, that fits the [one parameter logistic (Rasch) model](#)[2] by using the conditional maximum likelihood (CML) approach, with the possibility of including sampling weights and many other features not available in other `R` packages.

The function's syntax is illustrated in detail in the paragraph below.

- ### Estimation of the Rasch model: `RM.w`

  This function computes the parameter estimates of a Rasch model for binary item responses by using weighted [CML estimation](#)[3].

  #### Usage

  ```
  RM.w(.data, .w = NULL, .d=NULL, country=NULL,
       se.control = T, quantile.seq = NULL, write.file = F)
  ```

  #### Arguments

`.data`	Input 0/1 data matrix or data frame; affirmative responses must be coded as 1s. Rows represent individuals, columns represent items. Missing values are inserted as NA.
`.w`	Vector of sampling weights. The length must be the same as the number of rows of `.data`. If left unspecified, all the individuals will be equally weighted (`.w = rep(1, nrow(.data))`).
`.d`	Optional vector for the assumption on the extreme raw score parameters. Default is `0.5` and `(k-0.5)`, `k` being the maximum number of items (columns of .data).
`country`	Name of the country that data refer to.
`se.control`	Are the extreme parameter standard errors fixed to the ones corresponding to raw score `0.5` and `(k-0.5)`? If `FALSE`, the actual standard errors for the extreme parameters are estimated.
`quantile.seq`	Quantiles corresponding to the observed and the expected case fit statistic distributions.
`write.file`	If `TRUE`, a CSV file with the main results will be saved in the working directory.

---

[2] See also page 3 of [Introduction to Item Response Theory applied to Food Security Measurement](#)
[3] See also page 7 of [Introduction to Item Response Theory applied to Food Security Measurement](#)

Details

The weighted CML method is used to estimate the item parameter. Respondent parameters[4] are estimated post-hoc. Cases with missing responses to some items may be included, but will not be used to estimate the Rasch model.

As the parameters for the extreme raw scores (0 and `k)`, are undefined under the CML, some assumptions are needed for population-level prevalence estimates unless the proportions of respondents with those raw scores are so small that they can be considered to be measured as highly secure/highly insecure without error. Vector `.d` gives the possibility to include up to four alternative assumptions on each of the extreme parameters. More in detail, `.d` can be a two, three or four element vector:

- If `length(.d) = 4`, then the first two elements have to refer to the assumptions upon raw score 0, and the second two elements to raw score k. For instance `.d = c(0.1, 0.7, 7.1, 7.6)`, if the maximum raw score is 8.
- If `length(.d) = 3`, then the first two elements can either refer to the assumptions upon raw score 0 or raw score k, and the last one is defined accordingly. For instance `.d = c(0.1, 7.1, 7.6)` or `.d = c(0.1, 0.7, 7.6)`, if the maximum raw score is 8.
- If `length(.d) = 2`, then the first element have to refer to the assumption upon raw score 0, and the second element to raw score k. For instance `.d = c(0.1, 7.6)`, if the maximum raw score is 8.

Examples

```
rr.country1 = RM.w(XX.country1, wt.country1)

### Comment: This is the simplest specification, supplying
only the name of the data file and weight vector and
accepting the defaults for all other specifications. ###

### The output from the analysis will be in the list named
rr.country1 in this example. The following R statements will
display the various parts of the output included in
rr.country1 ###

# Item severity
rr.country1$b

# Item standard error
rr.country1$se.b

# Respondent severity for each raw score
rr.country1$a

# Respondent measurement error for each raw score
rr.country1$se.a
```

---

[4] See also page 8 of Introduction to Item Response Theory applied to Food Security Measurement

```
# Item infit⁵
rr.country1$infit

# Item outfit⁶
rr.country1$outfit

# Rasch reliability based on observed distribution of cases
  across raw scores
rr.country1$reliab

# Rasch reliability⁷ based on equal proportion of cases in
  each non-extreme raw score (more comparable across
  datasets)
rr.country1$reliab.fl

# Respondent infit distribution: observed and expected
quantile.seq = c(0,.01,.02,.05,.10,.25,.50,.75,.90,.95,
                 .98,.99,1)
q.infit = rr.country1$q.infit
q.infit.theor  = rr.country1$q.infit.theor
plot(quantile.seq, q.infit, type = "b", xlab = "Quantiles",
     ylab = "Observed infit", ylim = c(0, 6))
lines(quantile.seq, q.infit.theor, type = "b", col = 2)

# Conditional independence⁸
rr.country1$res.cor

# Rerun analysis to save outputs to csv file with country
  name
rr.country1 = RM.w(XX.country1, wt.country1, country =
                   "country1", write.file = T)
```

3. Data from Gallup World Poll® 2013-2014

VoH receives data collected via the Gallup World Poll ® (GWP), which includes the FIES, from 150 countries. Specific functions for this kind of data have been developed as follows:

- `load.Gallup` loads the complete GWP dataset and prepare it for the psychometric analysis;
- `tab.weight` pursues the weighted and unweighted descriptive analysis for the GWP sample.

- **Loading data: `load.Gallup`**

  This function loads the GWP data from an SPSS format to R and gives the data for the Rasch model application as an output.

---

[5] See also page 7 of Introduction to Item Response Theory applied to Food Security Measurement
[6] See also page 8 of Introduction to Item Response Theory applied to Food Security Measurement
[7] See also page 12 of Introduction to Item Response Theory applied to Food Security Measurement
[8] See also page 15 of Introduction to Item Response Theory applied to Food Security Measurement

Usage

```
load.Gallup(country, wave = 8)
```

Arguments

country	Name of the country the data refer to.
wave	If wave = 8, then data from GWP 2013 are loaded. Otherwise, wave = 9 refers to 2014.

Details

It is necessary that the working directory is set as the one that contains the data.

Examples

```
# Load data
data.country1 = load.Gallup("country1")

# Extract data for the Rasch analysis
XX.country1 = data.country1$data.obj
wt.country1 = data.country1$wt.obj

# Extract demographics
Data.country1 = data.country1$Data
```

- ## Descriptives: `tab.weight`

  This function computes the main descriptive tables, weighted and unweighted, for Rasch scale items and respondents. Optionally, it also computes tables for a standard set of demographic and geographic variables in the Gallup data. A further option computes tables for user-specified variables.

  ### Usage

  ```
  tab.weight(XX, wt, extra.var = F, Data = NULL,
              var.extern = NULL)
  ```

  ### Arguments

XX	Input 0/1 data matrix or data frame; affirmative responses must be coded as 1s. Rows represent individuals, columns represent items. Missing values are inserted as NA.

`wt`	Vector of sampling weights. The length must be the same as the number of rows of `.data`. If left unspecified, all the individuals will be weighted equally (`.w = rep(1, nrow(.data))`).
`extra.var`	If `TRUE`, default demographic and geographic descriptive tables are produced. The considered variables are age, gender, education and area.
`Data`	Complete GWP dataset as an `R` data frame. Needed only if `extra.var = TRUE`.
`var.extern`	If `TRUE`, a user-specified variable is considered for weighted tabulation. The only constrain is that `length(var.extern) = nrow(XX)`.

Details

The only necessary input is the 0/1 matrix of the responses to the questionnaire. In this case, the descriptive analysis of the responses and the distribution of the raw scores are considered.

Examples

```
# Fitting the function
tabs = tab.weight(XX.country1, wt.country1, extra.var =
                  T, Data = Data.country1)

# Simple tables - Unweighted
tabs$tab.extra

# Simple tables - Weighted
tabs$tab.extra.w

# Cross tables - Unweighted
tabs$cross.tab

# Cross tables - Weighted
tabs$cross.tab.w
```