



Organisation des Nations Unies
pour l'alimentation et l'agriculture

Manuel pour la mise en œuvre des méthodes *Voices of the Hungry* de la FAO pour mesurer l'insécurité alimentaire :

Package RM.weights dans R

Rédigé par Sara Viviani

Division des statistiques de la FAO, projet *Voices of the Hungry*

Version 2

Septembre, 2016

Avant-propos

Ce manuel accompagne un package dans R (<https://www.r-project.org/>) qui a été développé pour mesurer la prévalence de l'insécurité alimentaire à partir de l'échelle de mesure de l'insécurité alimentaire fondée sur les expériences (FIES en anglais) en appliquant la méthodologie utilisée par le projet *Voices of the Hungry* (VoH). Le manuel, ainsi que le package R qui l'accompagne, ont été préparés pour aider les analystes de données à effectuer une validation statistique de FIES et de mesurer les taux de prévalence de l'insécurité alimentaire au niveau national et sous-national. Une fonction additionnelle a été la capacité de produire des estimations comparables du taux de prévalence de l'insécurité alimentaire à travers différentes cultures et populations, tant au sein des pays qu'entre eux. Le package peut être utilisé par toute personne désireuse d'apprendre à utiliser le logiciel R et qui a la responsabilité d'analyser les données FIES pour le suivi de la sécurité alimentaire des populations. Les utilisateurs seront les offices nationaux de statistique exécutant leurs fonctions d'évaluation et de surveillance grâce à des enquêtes nationales et d'autres organismes impliqués dans le domaine de l'évaluation de la sécurité alimentaire parmi les différents groupes de population.

Le manuel fournit des instructions et des captures d'écran pour chacune des étapes qui peuvent être effectuées par le logiciel. Il est divisé en 5 parties :

1. **Introduction à R et RStudio**. Cette section introduit la logique générale de R.
2. **Validation statistique**. Cette section donne des directives préliminaires sur l'installation du package « RM.weights », le chargement et codage de données FIES dans R. Il guide ensuite l'utilisateur à travers l'estimation par la fonction « RM.w » des paramètres du modèle de Rasch pondéré et leur interprétation.
3. **Données contenues dans le package provenant du Gallup World Poll® 2013-2014**. Cette section décrit l'ensemble de données de l'échantillon contenues dans le *package* et aide l'utilisateur à appliquer une validation statistique avec des données réelles.
4. **Fonctions supplémentaires du package RM.weights**. Dans cette section, les autres fonctions contenues dans le *package* « RM.weights » sont brièvement abordées..
5. **Un exemple d'affectation discrète et probabiliste et quelques principes d'égalisation**
Cette section aidera l'utilisateur dans l'exécution de l'ensemble du processus d'affectation probabiliste et discrète aux classes d'insécurité alimentaire en utilisant des données réelles. Quelques applications de la procédure d'égalisation pour le calcul des taux de prévalence comparables sont également illustrées.

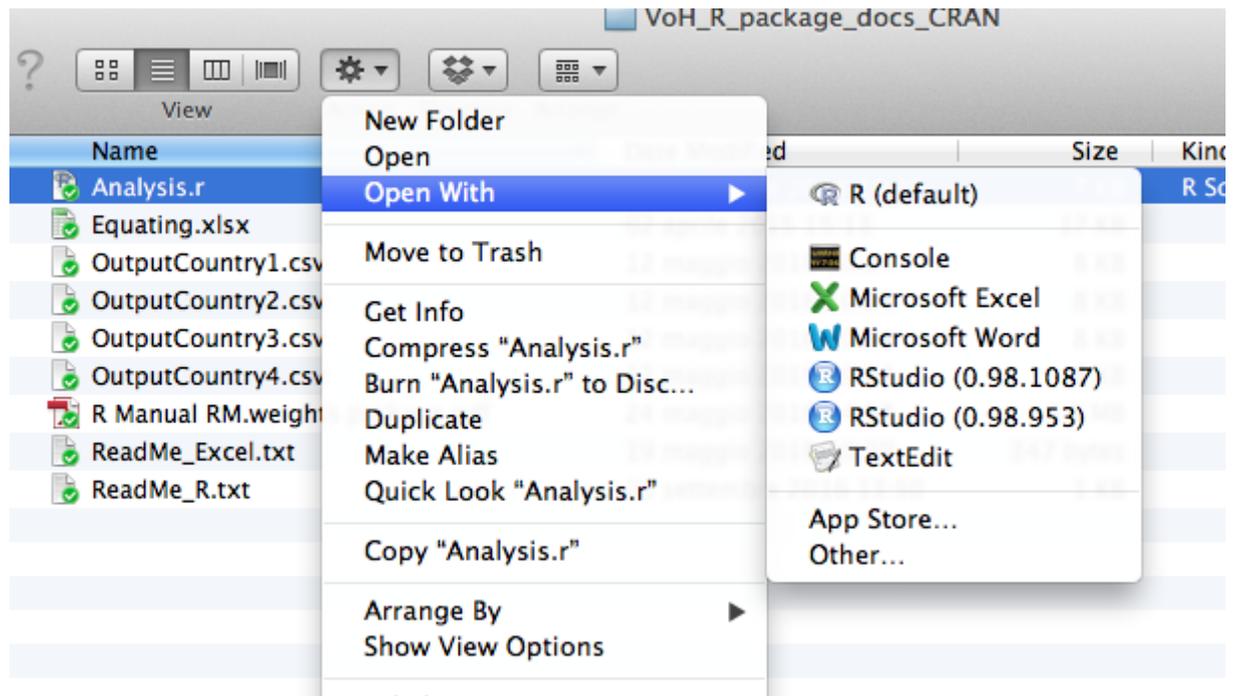
Référence suggérée

Sara Viviani (2016). Manuel pour la mise en œuvre des méthodes *Voices of the Hungry* de la FAO pour mesurer l'insécurité alimentaire : RM.weights package in R. FAO, Rome.

Le logiciel R et le manuel ont été mis au point par Sara Viviani avec une contribution précieuse de Carlo Cafiero, Mark Nord, Chiamaka Nwosu, Filippo Gheri et Gabriela Dos Santos. La mise en œuvre du projet Voices of the Hungry a été rendue possible par le soutien financier direct du Ministère du Développement international du Royaume-Uni et du Royaume de Belgique à travers le mécanisme d'appui aux programmes multidonateurs de la FAO.

AVANT DE COMMENCER : L'utilisateur doit ouvrir le fichier zip nommé « VoH_R_package_docs_CRAN », qui comprend quelques documents détaillés pour l'analyse de données, tels que le fichier de syntaxe nommé « Analysis.r ». Ce fichier peut être ouvert avec Rstudio et est un exemple d'analyse de données qui facilitera l'apprentissage du logiciel et de l'analyse de données.

*Vous pouvez télécharger ce fichier en cliquant sur le lien ci-dessous,
http://www.fao.org/fileadmin/templates/ess/voh/VoH_R_package_docs_CRAN.zip



En tant que logiciel statistique employé par le projet *Voices of the Hungry* (VoH), R est utilisé pour la mise en œuvre de méthodes pour mesurer la prévalence de l'insécurité alimentaire. En raison de sa flexibilité dans la création de nouveaux *packages*, R est l'outil idéal pour effectuer des méthodes statistiques novatrices. Ce document présente les principales fonctions de R pour la validation statistique de l'échelle de mesure de l'insécurité alimentaire fondée sur les expériences, un questionnaire utilisé pour évaluer la gravité de l'insécurité alimentaire d'une population, en utilisant le modèle de Rasch basé sur la théorie des réponses aux items.

Il est recommandé aux utilisateurs de consulter les articles sur la méthodologie de *VoH*, Rapport Technique (2016), disponible à <http://www.fao.org/3/a---i4830e.pdf> et le document de travail sur le modèle de Rasch basé sur la théorie des réponses aux items, disponible à <http://www.fao.org/3/a-i3946e.pdf>.

1. Introduction à R et RStudio

[R](#) est un environnement statistique Open-Source largement utilisé pour l'analyse des données. Grâce à ses *packages* externes, il offre une grande variété de méthodes statistiques. Pour avoir un aperçu de ses fonctionnalités, visitez le [task view](#) relevant de votre domaine.

[Cliquez ici](#) pour télécharger R pour Mac OS X, Windows ou Linux.

Une fois téléchargé, R peut être utilisé pour programmer directement dans la console, ou par l'intermédiaire d'un compilateur facile à utiliser, RStudio.

[RStudio](#) est un environnement de développement intégré pour R. Il comporte une console, une syntaxe - un éditeur de surlignage qui supporte l'exécution directe de code, ainsi que des outils pour le traçage, l'historique, le débogage et la gestion de l'espace de travail.

Pour les cours en ligne vous pouvez essayer le simulateur de R en ligne

<http://tryr.codeschool.com/levels/1/challenges/1>

et quelques cours donnés par Roger Peng

<https://www.youtube.com/watch?v=EiKxy5IecUw>

R est divisé en *packages*, c.-à-d. des modules de programme qui exécutent des techniques statistiques spécifiques. La création de nouveaux *packages* R permet de mettre à jour le logiciel à travers la mise en œuvre de nouvelles techniques. Le dépositaire des *packages* R est appelé CRAN (Comprehensive R Archive Network). Une fois qu'un *package* est chargé sur CRAN, il devient disponible pour chaque utilisateur qui a installé R.

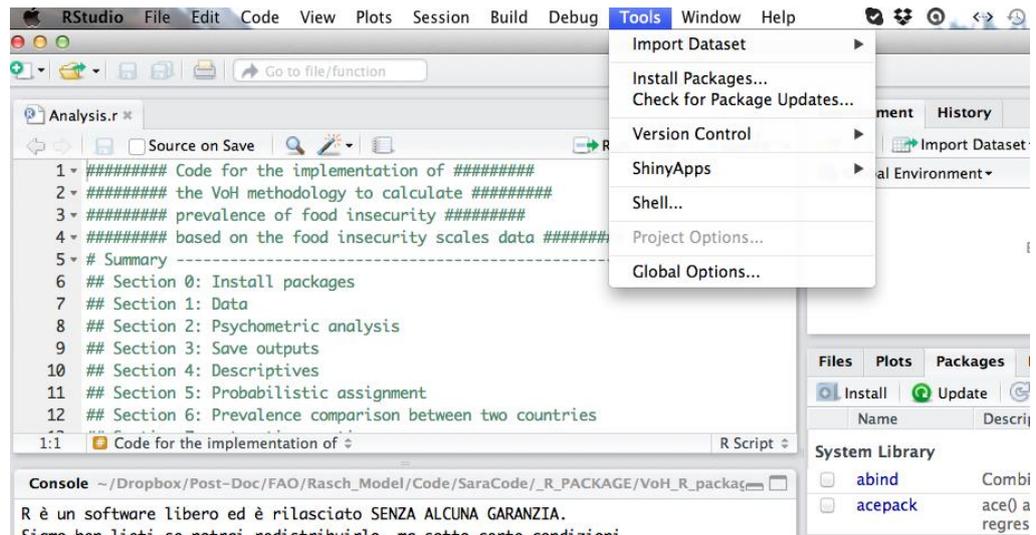
2. Validation statistique

Le projet VoH a mis en œuvre un *package* R pour analyser le FIES appelé "RM.weights", disponible sur CRAN.

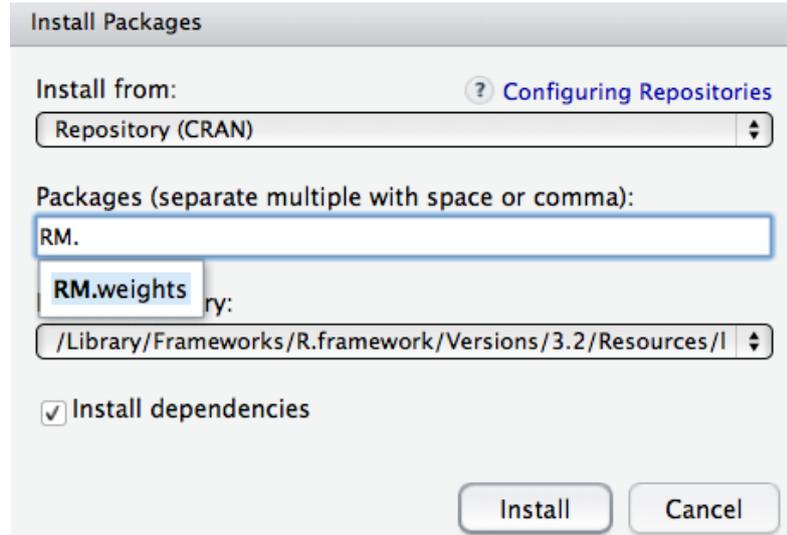
2.1. Installation du *package* « RM.weights » (section 0 dans le script « Analysis.r »)

Les étapes pour installer le *package* « RM.weights » dans RStudio sont les suivantes :

- Ouvrir RStudio (>=3.2)
- Installer le *package* « RM.weights » à partir de la fenêtre du menu de RStudio ("Tools → → Install Packages...")



- Dans la fenêtre « Install Packages », réglez sur « Repository (CRAN) ». Dans « Package archive », tapez « RM.weights » et sélectionnez le *package* « RM.weights ». Dans « Install to library », laissez l'option par défaut. Cocher la case « Install dependency ». Enfin, cliquez sur « Install ».



- Dans la fenêtre de commande RStudio, vous devriez lire les lignes suivantes :

```

15 # Section 0: Install packages -----
16 # The R package for the implementation of the VoH methodology
17 # is called "RM.weights" and it is available on CRAN (R packages' repository)
18 # starting from R version 3.2.
19
20 install.packages("RM.weights")
21 library(RM.weights)
22
23 # Section 1: Data -----
24 # Inside the package, four sample datasets are saved, named
25 # data_FA0_country1, data_FA0_country2, data_FA0_country3
21:1 Section 0: Install packages
R Script

```

```

> install.packages("RM.weights")
provo con l'URL 'http://cran.rstudio.com/bin/macosx/contrib/3.2/RM.weights_1.0.tgz'
Content type 'application/x-gzip' length 111029 bytes (108 KB)

-----

downloaded 108 KB

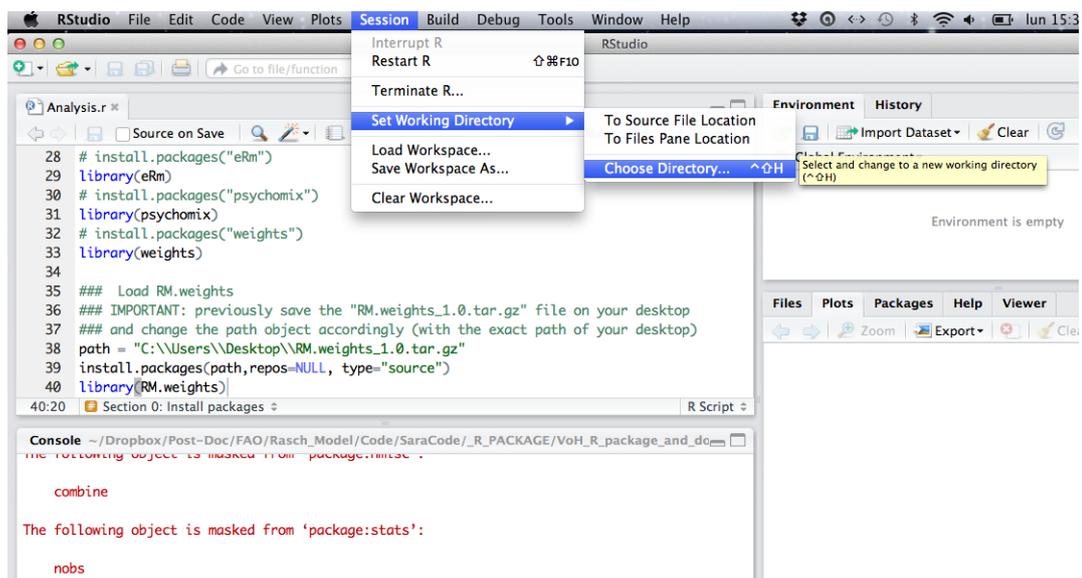
The downloaded binary packages are in
  /var/folders/16/2b0cljns2_d_w3qvcf474cpw0000gn/T//RtmpMAL0Lh/downloaded_packages
>

```

- Pour finalement charger le *package*, veuillez entrer le code suivant dans la fenêtre de commande RStudio
`library(RM.weights)`
- Le *package* est maintenant téléchargé !

2.2. Chargez et codifiez les données FIES dans R

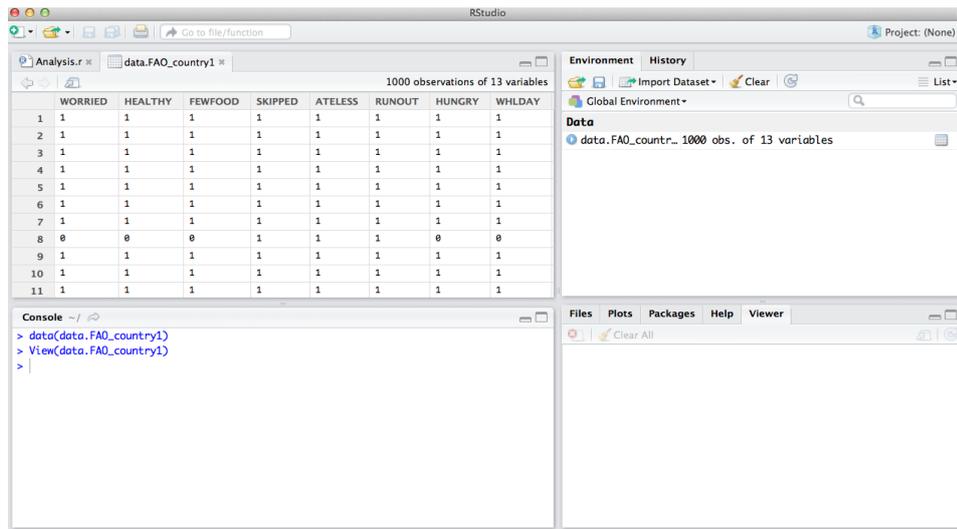
Pour charger votre ensemble de données dans R, la première étape est de définir comme répertoire de travail celui où les données sont enregistrées comme suit :



Par exemple:

```
data = read.spss("datasetname.sav", to.data.frame = T)
```

- Si vos données sont enregistrées en format STATA, vous devez encore utiliser le *package* « foreign », et utiliser la fonction « read.dta ». Si vous utilisez une version de STATA 13.0 ou plus, vous devez d'abord enregistrer les données sous le format « Stata 11/12 Data » pour des raisons de compatibilité.
- Les données seront visibles en cliquant sur l'objet de données situé tout en haut de la fenêtre dans RStudio :



Une fois que les données sont chargées, vous devez extraire les variables FIES de l'ensemble de données. Un moyen facile de le faire est de sélectionner les colonnes correspondant aux données FIES. Par exemple, si les données correspondant au FIES sont enregistrées dans les colonnes 4 à 11, vous pouvez utiliser le code :

```
XX = data[, 4:11]
```

La même chose vaut pour les poids d'échantillonnage, si présent. Par exemple, si ceux-ci sont enregistrés dans la colonne 12 de l'ensemble de données, ils peuvent être extraits avec le code

```
wt = data[, 12]
```

Notez que "data", "XX" et "wt" sont seulement des appellations inventées qui peuvent être modifiées si l'utilisateur préfère d'autres étiquettes.

Remarque : les données FIES (ici appelée XX) doivent être en format zéro/un (les réponses 0 pour non, 1 pour oui). Si les données sont codées différemment (par exemple 1 pour oui et 2 pour non), vous pouvez les recoder comme suit :

```
XX[XX==2] = 0
```

ou les recoder avant d'importer les données.

2.3. Utilisation du *package* « RM.weights »

"RM.weights" est un programme qui regroupe plusieurs fonctions liées au modèle de Rasch.

La fonction principale dans le *package* est appelée “*RM.w*”, et s’adapte au [modèle logistique \(Rasch\) à 1 paramètre¹](#) en utilisant la méthode du maximum de vraisemblance conditionnelle (CML: conditional maximum likelihood), avec la possibilité d’inclure des poids d’échantillonnage et de nombreuses autres fonctionnalités non disponibles dans d’autres *packages* R.

La syntaxe de la fonction est illustrée en détail dans le paragraphe ci-dessous.

2.4. Estimation du modèle de Rasch : la fonction *RM.w*

Cette fonction calcule les estimations des paramètres d’un modèle de Rasch pour les réponses aux Items binaires en utilisant [l’estimation pondérée du CML²](#).

Pour voir l’aide de la fonction, tapez le code “*?RM.w*” sur la console R.

Utilisation

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

Arguments

<code>.data</code>	Entrez 0/1 la matrice de données ou le cadre de données ; les réponses affirmatives doivent être codées 1. Les lignes représentent des individus, les colonnes représentent des Items. Les valeurs manquantes sont insérées comme NA.
<code>.w</code>	Vecteur des poids d’échantillonnage. L’étendue doit être la même comme le nombre de lignes de <code>.data</code> . Si ce n’est pas précisé, tous les Individus seront pondérés de manière égale (<code>.w = rep(1, nrow(.data))</code>).
<code>.d</code>	Vecteur optionnel pour l’hypothèse sur les paramètres de score brut extrêmes. La valeur par défaut est <code>0.5</code> et <code>(k-0.5)</code> , <code>k</code> étant le nombre maximum des Items (colonnes de <code>.data</code>).
<code>country</code>	Nom du pays auquel les données se rapportent.
<code>se.control</code>	Est-ce que les erreurs type de paramètres extrêmes fixés à celles correspondant au score brut <code>0.5</code> et <code>(k-0.5)</code> ? Si <code>FALSE</code> , les erreurs type effectives pour les paramètres extrêmes sont estimées.
<code>quantile.seq</code>	Les quantiles correspondant au cas observé et attendu s’adapte aux distributions.
<code>write.file</code>	Si <code>TRUE</code> , un fichier CSV avec les résultats principaux seront enregistrés dans le répertoire de travail.

¹ Voir aussi la page 3 de l’[Introduction à la théorie des réponses aux items appliquée à la mesure de sécurité alimentaire](#)

² Voir aussi la page 7 de l’[Introduction à la théorie des réponses aux items appliquée à la mesure de sécurité alimentaire](#)

Details

La méthode CML pondérée est utilisée pour estimer le paramètre de l'Item. [Les paramètres de répondant](#)³ sont estimés post-hoc. Les cas pour lesquels il manque des réponses à certains items peuvent être inclus, mais ne seront pas utilisés pour estimer le modèle de Rasch.

Comme les paramètres pour les scores bruts extrêmes (0 et k), ne sont pas définis sous le CML, des hypothèses sont nécessaires pour les estimations de la prévalence au niveau de la population à moins que les proportions de répondants avec ces scores bruts soient si petits qu'ils peuvent être considérés comme étant mesurés comme très sécurisés/très insécurisés sans erreur. Le vecteur `.d` donne la possibilité d'inclure jusqu'à quatre hypothèses alternatives sur chacun des paramètres extrêmes.

Remarque : les hypothèses par défaut sur le paramètre du score brut extrême sont `.d=c(0.5, k-0.5)`. Cela signifie qu'au lieu d'être estimé pour les scores bruts 0 et k (qui conduiraient à des paramètres non définis), les paramètres de score brut extrême sont estimés pour 0,5 et k-0.5. Ces hypothèses sont valables pour les ensembles de données standards où la proportion en score brut 0 ou k n'est pas extrêmement élevé (<40%). Sinon, une flexibilité est introduite dans le *package* pour estimer les paramètres alternatifs de score brut extrêmes comme suit.

`.d` peut être un vecteur à deux, trois ou quatre éléments:

- Si `length(.d) = 4`, alors les deux premiers éléments doivent se référer aux hypothèses sur le score brut 0, et les deux éléments suivants au score brut k. Par exemple `.d = c(0.1, 0.7, 7.1, 7.6)`, si le score brut maximal est 8.
- Si `length(.d) = 3`, alors les deux premiers éléments peuvent renvoyer soit aux hypothèses sur le score brut 0 ou au score brut k, et le dernier est défini en conséquence. Par exemple `.d = c(0.1, 7.1, 7.6)` ou `.d = c(0.1, 0.7, 7.6)`, si le score brut maximal est 8.
- Si `length(.d) = 2`, alors le premier élément doit se référer à l'hypothèse sur le score brut 0, et le deuxième élément au score brut k. Par exemple `.d = c(0.1, 7.6)`, si le score brut maximal est 8.

Exemples

Remarque : L'utilisation la plus simple de la fonction `RM.w` est d'indiquer uniquement le nom des données et les poids d'échantillonnage (toutes les autres spécifications sont déjà définies par défaut). Le résultat est enregistré dans la liste nommée « `rr.country1` » dans cet exemple.

```
data(data.FAO_country1)
```

Questionnaire sur les données et les pondérations :

```
XX.country1 = data.FAO_country1[,1:8]  
wt.country1 = data.FAO_country1$wt
```

³ Voir aussi la page 8 de l'[Introduction à la théorie des réponses aux items appliquée au mesure de sécurité alimentaire](#)

Adaptez le Rasch pondéré :

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

Adaptez le Rasch non pondéré (les pondérations seront toutes configurées à 1)

```
rr.country1.nw = RM.w(XX.country1)
```

Afficher la sévérité de l'item, les erreurs type, [infits](#)⁴ et [outfits](#)⁵...

```
cbind("Item sev."=rr.country1$b, "St.err."=rr.country1$se.b,  
"Infit"=rr.country1$infit, "Outfit"=rr.country1$outfit)
```

```
> cbind("Item sev."=rr.country1$b, "St.err."=rr.country1$se.b,  
+ "Infit"=rr.country1$infit, "Outfit"=rr.country1$outfit)  
      Item sev.  St.err.   Infit   Outfit  
WORRIED -0.4924021 0.1306812 1.1099625 1.1683698  
HEALTHY -0.3646588 0.1285128 1.1642569 1.3562234  
FEWFOOD -0.9972146 0.1411709 0.8373479 0.6208227  
SKIPPED  0.2227441 0.1208220 0.9661706 0.8833550  
ATELESS -0.7017997 0.1346454 0.9428929 0.9720298  
RUNOUT   0.1240431 0.1218599 0.9001897 0.8083734  
HUNGRY   0.5365124 0.1182047 1.0300541 1.0503002  
WHLDAY   1.6727126 0.1196330 1.0409061 1.1386965  
> |
```

Afficher les sévérités de répondant et la mesure des erreurs :

```
cbind("Person par."=rr.country1$a, "Error"=rr.country1$se.a)
```

```
> cbind("Person par."=rr.country1$a, "Error"=rr.country1$se.a)  
      Person par.   Error  
[1,] -2.93364909 1.4767500  
[2,] -2.14729853 1.0919157  
[3,] -1.24833721 0.8489670  
[4,] -0.60359833 0.7703593  
[5,] -0.02875708 0.7539025  
[6,]  0.55596614 0.7832545  
[7,]  1.23189326 0.8739319  
[8,]  2.18697698 1.1233237  
[9,]  3.01139312 1.4767500
```

Afficher la fiabilité de Rasch sur la base de distribution observée des cas à travers les scores bruts

```
rr.country1$reliab
```

⁴ Voir aussi la page 7 de l'[Introduction à la théorie des réponses aux items appliquée au mesure de sécurité alimentaire](#)

⁵ Voir aussi la page 8 de l'[Introduction à la théorie des réponses aux items appliquée au mesure de sécurité alimentaire](#)

Afficher la **fiabilité de Rasch**⁶ fondée sur l'égalité de proportion des cas dans chaque score brut non-extrême (plus comparables entre les ensembles de données)

```
rr.country1$reliab.fl
```

```
> rr.country1$reliab
[1] 0.6853269
> rr.country1$reliab.fl
[1] 0.6969928
```

Calculer la distribution de l'infit des répondants observées et attendues :

```
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)
```

Afficher la matrice d'**indépendance conditionnelle**⁷

```
rr.country1$res.cor
```

```
> rr.country1$res.cor
      WORRIED  HEALTHY  FEWFOOD  SKIPPED  ATELESS  RUNOUT  HUNGRY  WHLDAY
WORRIED  1.00000000 -0.01802200  0.14096227 -0.13366747  0.04254704 -0.08336590 -0.15799628 -0.08449277
HEALTHY -0.01802200  1.00000000  0.11699451 -0.12044348 -0.07755343 -0.11193959 -0.14602194 -0.15128380
FEWFOOD  0.14096227  0.11699451  1.00000000  0.07920136  0.23380594  0.04776520 -0.01864901 -0.01769752
SKIPPED -0.13366747 -0.12044348  0.07920136  1.00000000  0.06420526  0.13198748 -0.03681056  0.12818899
ATELESS  0.04254704 -0.07755343  0.23380594  0.06420526  1.00000000 -0.04515401  0.03764592 -0.09122342
RUNOUT  -0.08336590 -0.11193959  0.04776520  0.13198748 -0.04515401  1.00000000  0.19317732  0.12973828
HUNGRY  -0.15799628 -0.14602194 -0.01864901 -0.03681056  0.03764592  0.19317732  1.00000000 -0.01008275
WHLDAY  -0.08449277 -0.15128380 -0.01769752  0.12818899 -0.09122342  0.12973828 -0.01008275  1.00000000
```

Relancez l'analyse pour enregistrer les résultats en fichier csv avec nom de pays

```
rr.country1 = RM.w(XX.country1, wt.country1, country =
                  "country1", write.file = T)
```

⁶ Voir aussi la page 12 de l'[Introduction à la théorie des réponses aux items appliquée au mesure de sécurité alimentaire](#)

⁷ Voir aussi la page 15 de l'[Introduction à la théorie des réponses aux items appliquée au mesure de sécurité alimentaire](#)

Le résultat sera enregistré dans le répertoire de travail et nommé “Outputcountry1.csv”:

Item	Severity	SE	Infit	SE infit	Outfit	N Yes	Perc Yes	WN Yes	Wperc Yes	N missing	W missing
WORRIED	-0.492402	0.1306812	1.1099625	0.0705261	1.1683698	302	71.394799	713.54683	71.354683	1	0.8516582
HEALTHY	-0.364659	0.1285128	1.1642569	0.0685013	1.3562234	292	69.030733	705.5814	70.55814	0	0
FEWFOOD	-0.997215	0.1411709	0.8373479	0.0794705	0.6208227	335	79.196217	744.76984	74.476984	0	0
SKIPPED	0.2227441	0.120822	0.9661706	0.06055	0.883355	248	58.628842	660.36042	66.036042	2	0.9936879
ATELESS	-0.7018	0.1346454	0.9428929	0.074073	0.9720298	316	74.704492	727.01023	72.701023	2	0.666932
RUNOUT	0.1240431	0.1218599	0.9001897	0.0617431	0.8083734	254	60.047281	669.64327	66.964327	1	0.3700116
HUNGRY	0.5365124	0.1182047	1.0300541	0.0570838	1.0503002	229	54.137116	636.24491	63.624491	2	0.7034776
WHLDAY	1.6727126	0.119633	1.0409061	0.0497944	1.1386965	119	28.132388	532.04294	53.204294	4	4.1036036

3. Données contenues dans le package provenant du Gallup World Poll® 2013-2014.

Le VoH collecte les données recueillies à travers le Gallup World Poll ® (GWP), y compris le FIES, pour 150 pays chaque année. Dans le *package* « RM.weights », quatre ensembles de données de l'échantillon du GWP sont téléchargés. Ces ensembles de données sont appelés `data.FAO_country1`, `data.FAO_country2`, `data.FAO_country3`, `data.FAO_country4`. Les ensembles de données comprennent les données FIES, les poids d'échantillonnage, et d'autres variables démographiques. Pour plus d'informations sur les données, vous pouvez utiliser l'aide comme suit :

```
?data.FAO_country1
```

Pour joindre les données et extraire le FIES et les poids d'échantillonnage, utilisez le code suivant :

```
data(data.FAO_country1)
XX.country1 = data.FAO_country1[,1:8]
wt.country1 = data.FAO_country1$wt
```

Les données peuvent être explorées avec la fonction `tab.weight`.

3.1. Descriptifs : `tab.weight`

Cette fonction calcule les principaux tableaux descriptifs, pondérée et non pondérée, pour les items de l'échelle FIES et des répondants. Il peut également être utilisé pour le calcul des tableaux croisés et simples pour les variables démographiques et géographiques.

Utilisation

```
tab.weight(variab, wt, XX = NULL)
```

Arguments

<code>variab</code>	La variable spécifiée par l'utilisateur considérée pour le calcul pondéré. Elle peut être une variable unique, ou une liste à deux variables, de type de <code>factor</code> , et <code>length(var.extern)</code> doit être égale à <code>nrow(XX)</code> ..
<code>wt</code>	Vecteur des poids d'échantillonnage. L'étendue doit être la même que le nombre de rangs <code>.data</code> . Si laissé non spécifié, tous les individus seront pondérés de manière égale (<code>.w = rep(1, nrow(.data))</code>).
<code>XX</code>	Entrez la matrice de données ou le cadre de données 0/1; les réponses affirmatives doivent être codées 1. Les lignes représentent les personnes, les colonnes représentent les items. Les valeurs manquantes sont insérées comme NA.

Exemples

Définissez les données

```
data(data.FAO_country1)
XX.country1 = data.FAO_country1[,1:8]
wt.country1 = data.FAO_country1$wt
gender = data.FAO_country1$gender
urbanrural = data.FAO_country1$urbanrural
```

Tableau pondéré à une variable par sexe

```
tab.weight(gender, wt.country1)$tab.ext.w
```

```
> tab.weight(gender, wt.country1)$tab.ext.w
Female  Male
 513.2  486.8
>
```

Tableau pondéré à deux variables par sexe et par urbain / rural

```
tab.weight(list(gender,urbanrural),wt.country1)$tab.ext.w
```

```
> tab.weight(list(gender,urbanrural),
+ wt.country1)$tab.ext.w
      Female  Male
Rural 269.63 186.68
Urban 243.57 300.12
> |
```

Calculer les descriptifs de Rasch

```
fit.descr = tab.weight(wt = wt.country1, XX = XX.country1)
```

Afficher la distribution pondérée selon les scores bruts (absolue et relative) :

```
cbind("Abs.RS distrib." = fit.descr$RS.abs.w,
"Rel.RS distrib." = fit.descr$RS.rel.w)
```

```
> cbind("Abs.RS distrib." = fit.descr$RS.abs.w,
+ "Rel.RS distrib." = fit.descr$RS.rel.w)
      Abs.RS distrib. Rel.RS distrib.
[1,]          167.08           0.17
[2,]           37.80           0.04
[3,]           27.07           0.03
[4,]           40.56           0.04
[5,]           42.57           0.04
[6,]           56.01           0.06
[7,]           80.72           0.08
[8,]          128.61           0.13
[9,]          413.64           0.41
>
> |
```

Afficher le pourcentage pondéré et non pondéré de Yes par item :

```
cbind("Weighted perc. of Yes" = fit.descr$Perc.Yes.w,
"Unweighted perc. of Yes" = fit.descr$Perc.Yes)
```

```
> cbind("Weighted perc. of Yes" = fit.descr$Perc.Yes.w, "Unweighted perc. of Yes" = fit.descr$Perc.Yes)
      Weighted perc. of Yes Unweighted perc. of Yes
WORRIED                0.71                0.70
HEALTHY                 0.71                0.69
FEWFOOD                 0.74                0.74
SKIPPED                 0.66                0.65
ATELESS                 0.73                0.72
RUNOUT                  0.67                0.66
HUNGRY                  0.64                0.63
WHLDAY                  0.53                0.52
>
```

4. Fonctions supplémentaires du *package* RM.weights

Le *package* comprend de nombreuses autres fonctions qui peuvent être affichées à l'aide de la commande suivante :

```
help(package="RM.weights")
```

The screenshot shows the RStudio interface. The console window displays the following output:

```
3: package 'Hmisc' was built under R version 3.2.5
4: package 'ggplot2' was built under R version 3.2.4
> help(package="RM.weights")
>
>
>
>
>
>
>
>
>
>
```

The help page for the *RM.weights* package is displayed in the right-hand pane. It includes a "DESCRIPTION file" link and a "Help Pages" section with the following entries:

- [RM.weights-package](#): Rasch model and extensions for survey data, using Conditional Maximum likelihood (CML).
- [data.FAO_country1](#): Food insecurity data for a GWP country (Country1).
- [data.FAO_country2](#): Food insecurity data for a GWP pilot country (Country2).
- [data.FAO_country3](#): Food insecurity data for a GWP pilot country (Country3).
- [data.FAO_country4](#): Food insecurity data for a GWP pilot country (Country4).
- [equating.fun](#): Equating scales between different application contexts.
- [EWaldtest](#): Wald test to check sampling independence under CML.
- [ICC.fun](#): Item characteristic curves computation and plotting under CML.
- [PC.w](#): Estimation of the weighted partial credit model using CML.
- [prob.assign](#): Probabilistic assignment of cases to classes of severity along the latent trait.
- [RM.w](#): Estimation of the weighted Rasch model using CML.
- [RT.thres](#): Rasch-Thurstone thresholds for trichotomous scales under CML.
- [tab.weight](#): Rasch descriptives and weighted tables.

La fonction `prob.assign` est particulièrement intéressante, elle peut être utilisée pour calculer la prévalence de l'insécurité alimentaire en utilisant la même méthodologie du projet VoH. Le code de cette procédure peut être trouvé dans le fichier « Analysis.r », Section 5.

La section 5, dans le fichier « Analysis.r » est préparatoire de la section 6, qui montre comment calculer les mesures de prévalence comparables entre pays ou sous-groupes (par exemple les langues) au sein d'un pays. La première phase de ce processus d'égalisation est aussi signalée dans le fichier « Equating.xlsx ». La section suivante décrit un exemple d'égalisation et d'affectation probabiliste et discret.

5. Un exemple d'affectation discrète et probabiliste et quelques principes d'égalisation

Remarque : Certains principes d'égalisation sont également inclus dans le fichier « Equating.xlsx »

Dans cette section, nous allons illustrer comment attribuer des cas aux classes d'insécurité alimentaire de manière probabiliste et déterministe et nous allons brièvement illustrer quelques applications de la procédure d'égalisation pour le calcul des taux de prévalence comparables.

La classification des cas dans les classes d'insécurité alimentaire peut être menée de deux façons.

La classification déterministe peut être effectuée en fixant des seuils en termes de score brut. Le score brut est la somme des réponses affirmatives aux 8 FIES items par chaque sujet interviewé et peut être calculé dans la R comme suit :

```
rs.country1 = rowSums(XX.country1)
```

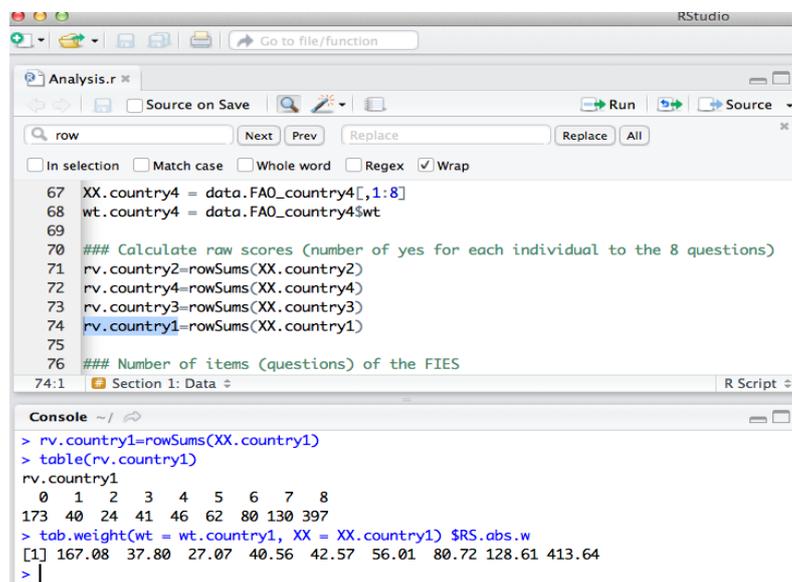
(où XX.country1 est la matrice qui inclut les réponses 0/1 aux FIES)

Pour afficher la distribution (non pondérée) de personnes déclarant un score brut donné, vous pouvez saisir

```
table(rs.country1)
```

Alors que pour la distribution pondérée vous pouvez utiliser la fonction « tab.weight » (voir la section 3 de ce document)

```
tab.weight(wt = wt.country1, XX = XX.country1)$RS.abs.w
```



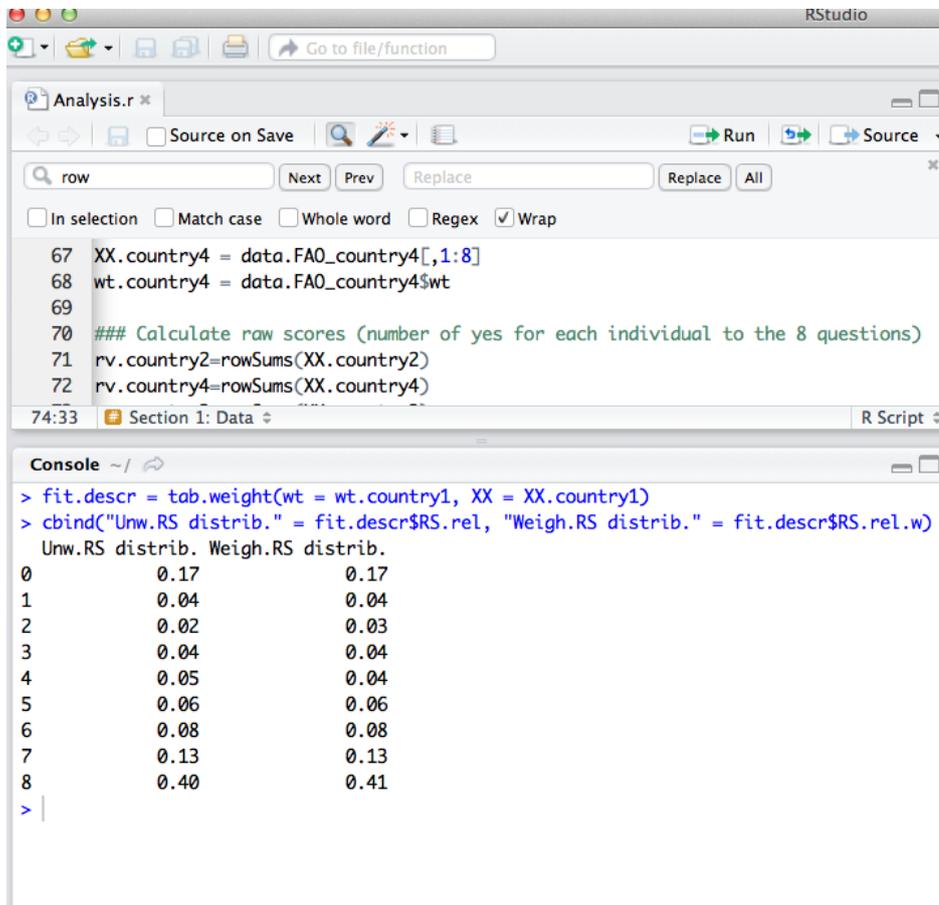
```
67 XX.country4 = data.FAO_country4[,1:8]
68 wt.country4 = data.FAO_country4$wt
69
70 ### Calculate raw scores (number of yes for each individual to the 8 questions)
71 rv.country2=rowSums(XX.country2)
72 rv.country4=rowSums(XX.country4)
73 rv.country3=rowSums(XX.country3)
74 rv.country1=rowSums(XX.country1)
75
76 ### Number of items (questions) of the FIES
74:1 Section 1: Data R Script

Console ~/
> rv.country1=rowSums(XX.country1)
> table(rv.country1)
rv.country1
 0  1  2  3  4  5  6  7  8
173 40 24 41 46 62 80 130 397
> tab.weight(wt = wt.country1, XX = XX.country1) $RS.abs.w
[1] 167.08 37.80 27.07 40.56 42.57 56.01 80.72 128.61 413.64
> |
```

La distribution relative selon les scores bruts peut aussi être calculée avec la fonction « tab.weight » :

```
fit.descr = tab.weight(wt = wt.country1, XX = XX.country1)
```

```
cbind("Unw.RS distrib." = fit.descr$RS.rel,  
      "Weigh.RS distrib." = fit.descr$RS.rel.w)
```



The screenshot shows the RStudio interface. The script editor contains the following code:

```
67 XX.country4 = data.FAO_country4[,1:8]  
68 wt.country4 = data.FAO_country4$wt  
69  
70 ### Calculate raw scores (number of yes for each individual to the 8 questions)  
71 rv.country2=rowSums(XX.country2)  
72 rv.country4=rowSums(XX.country4)
```

The console shows the execution of the code and the resulting output:

```
> fit.descr = tab.weight(wt = wt.country1, XX = XX.country1)  
> cbind("Unw.RS distrib." = fit.descr$RS.rel, "Weigh.RS distrib." = fit.descr$RS.rel.w)  
Unw.RS distrib. Weigh.RS distrib.  
0      0.17      0.17  
1      0.04      0.04  
2      0.02      0.03  
3      0.04      0.04  
4      0.05      0.04  
5      0.06      0.06  
6      0.08      0.08  
7      0.13      0.13  
8      0.40      0.41  
>
```

Cette distribution relative peut être utilisée pour calculer les taux de prévalence de l'insécurité alimentaire à différents niveaux.

Tout d'abord, un moins la distribution cumulative pour tous les niveaux de score brut est calculée :

```
XX.country1 = data.FAO_country1[,1:8]
```

```
wt.country1 = data.FAO_country1$wt
```

```
rv.country1=rowSums(XX.country1)
```

```
cbind("RS"=1:9,"Prev"=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1,  
XX.country1)$RS.rel.w))[-9,]
```

```

132 tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w*100
133 # Example: calculation of prev setting thresholds in terms of raw score
134 XX.country1 = data.FAO_country1[,1:8]
135 wt.country1 = data.FAO_country1$wt
136 rv.country1=rowSums(XX.country1)
137 cbind("RS"=1:9,"Prev"=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w)[-9])
138 ## Distribution by gender
139 Section 4: Descriptives

```

```

> XX.country1 = data.FAO_country1[,1:8]
> wt.country1 = data.FAO_country1$wt
> rv.country1=rowSums(XX.country1)
> cbind("RS"=1:9,"Prev"=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w)[-9])
  RS Prev
[1,] 1 0.83
[2,] 2 0.79
[3,] 3 0.76
[4,] 4 0.72
[5,] 5 0.68
[6,] 6 0.62
[7,] 7 0.54
[8,] 8 0.41
>

```

Ensuite, fixer un seuil, par exemple, de 4 pour classier l'insécurité alimentaire « modérée ou sévère », et de 7 pour classier l'insécurité alimentaire « sévère » les sujets (ou les ménages), conduira aux taux de prévalence suivants :

```
cbind("Threshold"=c(4,7),"Levels"=c("Mod. or severe", "Severe"),
```

```
Prev=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1,
XX.country1)$RS.rel.w)[c(4,7)])
```

```

134 XX.country1 = data.FAO_country1[,1:8]
135 wt.country1 = data.FAO_country1$wt
136 rv.country1=rowSums(XX.country1)
137 cbind("RS"=1:9,"Prev"=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w)[-9])
138 cbind("Threshold"=c(4,7),"Levels"=c("Mod. and severe", "Severe"),
139 Prev=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w)[c(4,7)])
140 ## Distribution by gender
141 Section 4: Descriptives

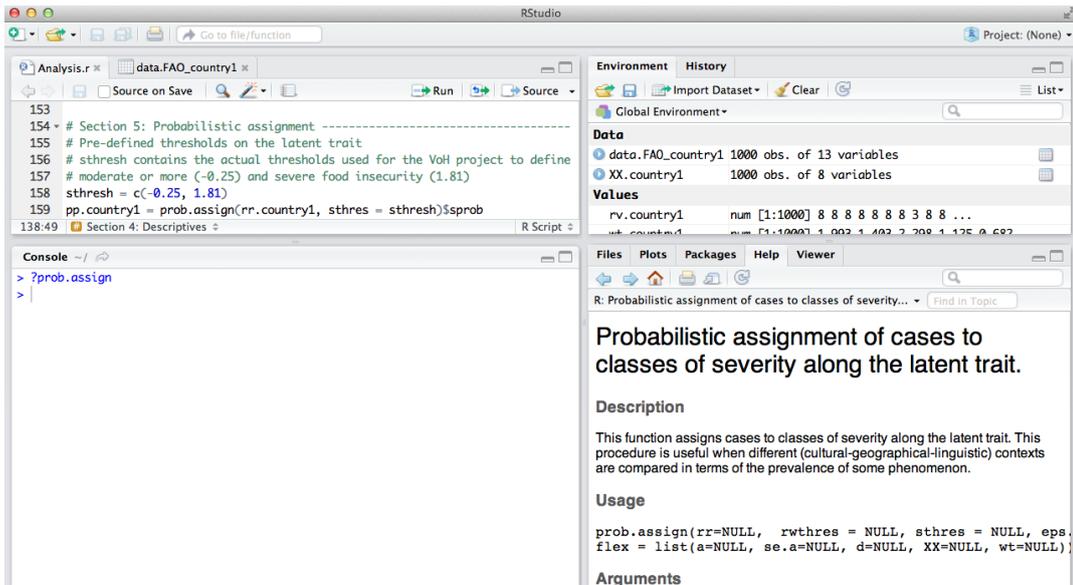
```

```

> XX.country1 = data.FAO_country1[,1:8]
> wt.country1 = data.FAO_country1$wt
> rv.country1=rowSums(XX.country1)
> cbind("Threshold"=c(4,7),"Levels"=c("Mod. and severe", "Severe"),
+ Prev=1-cumsum(tab.weight(as.factor(rv.country1), wt.country1, XX.country1)$RS.rel.w)[c(4,7)])
  Threshold Levels      Prev
[1,]      4 "Mod. and severe" "0.72"
[2,]      7 "Severe"         "0.54"
>

```

La classification probabiliste (ou **affectation probabiliste**), d'autre part, peut être effectuée en fixant des seuils en termes de caractère latent. L'avantage de l'utilisation de cette approche est que, après une procédure d'égalisation, elle peut être utilisée pour calculer les taux de prévalence comparables entre les pays. Dans le *package* « RM.weights », la fonction permettant d'exécuter l'affectation probabiliste est appelée "prob.assign" (tapez ?prob.assign dans R pour voir l'aide):



Par exemple, pour calculer la probabilité, pour un pays donné, d'être au-delà des seuils prédéterminés sur le caractère latent on peut écrire le code suivant :

```
sthresh = c(-0.25, 1.81)
```

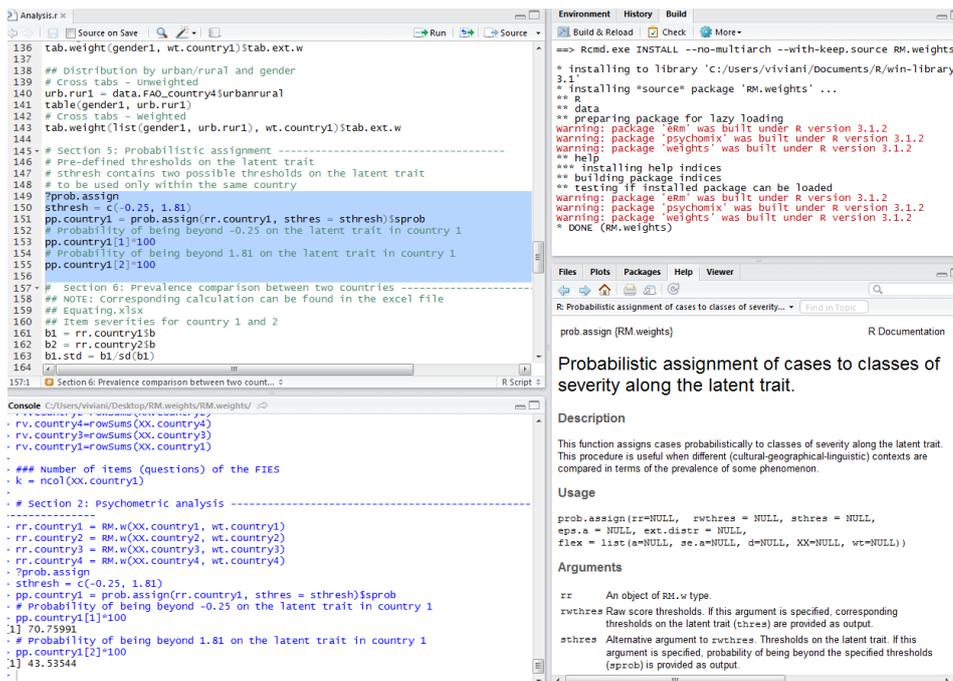
```
pp.country1 = prob.assign(rr.country1, sthres = sthresh)$sprob
```

La probabilité d'être au-delà de -0,25 sur le caractère latent dans le pays 1 est

```
pp.country1[1]*100
```

pendant que la probabilité d'être au-delà de 1,81 sur le caractère latent dans le pays 1 est

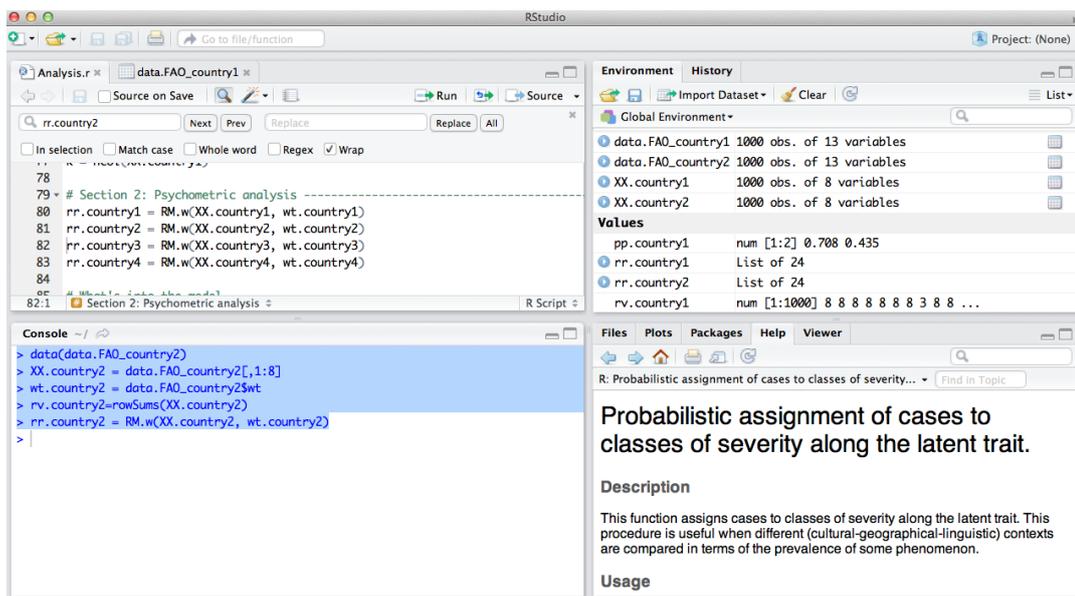
```
pp.country1[2]*100
```



Supposons maintenant que nous voulons calculer un taux de prévalence comparable de l'insécurité alimentaire entre les pays 1 et pays 2 (country 1 et country 2).

Nous chargeons les données pour country 2 et adaptons le modèle de Rasch :

```
data(data.FAO_country2)
XX.country2 = data.FAO_country2[,1:8]
wt.country2 = data.FAO_country2$wt
rv.country2=rowSums(XX.country2)
rr.country2 = RM.w(XX.country2, wt.country2)
```



Nous déterminons ensuite une mesure commune entre *country 1* et *country 2* fondée sur les sévérités de l'item (b1 et b2 dans le code suivant) :

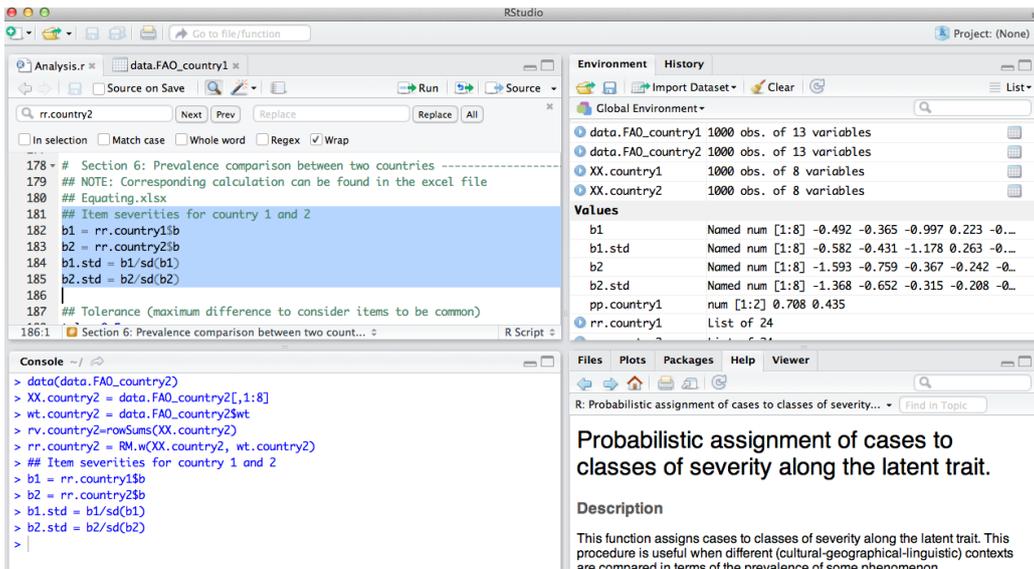
```
b1 = rr.country1$b
```

```
b2 = rr.country2$b
```

La première étape consiste à normaliser les sévérités de l'item à un écart type de 1 :

```
b1.std = b1/sd(b1)
```

```
b2.std = b2/sd(b2)
```

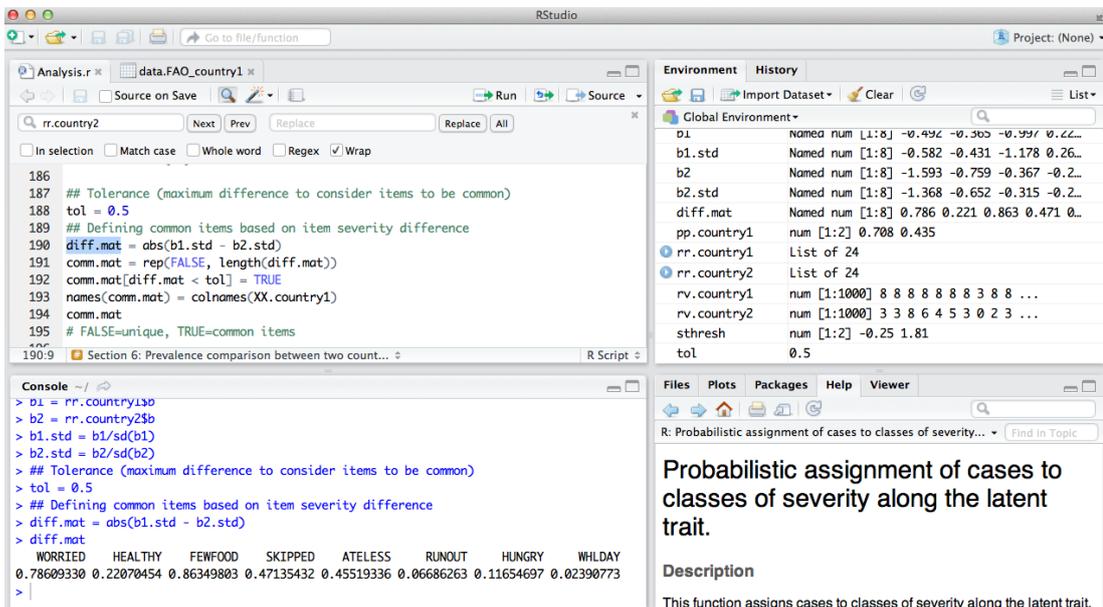


Nous calculons la différence entre les sévérités de l'item normalisé et nous les comparerons à une valeur seuil fixe :

$tol = 0.5$

$diff.mat = abs(b1.std - b2.std)$

$diff.mat$



Le vecteur “comm.mat” sera VRAI si la différence entre les niveaux de sévérités standardisés est inférieure à la tolérance fixée, et faux dans le cas contraire :

$comm.mat = rep(FALSE, length(diff.mat))$

$comm.mat[diff.mat < tol] = TRUE$

$names(comm.mat) = colnames(XX.country1)$

$comm.mat$

```

186
187 ## Tolerance (maximum difference to consider items to be common)
188 tol = 0.5
189 ## Defining common items based on item severity difference
190 diff.mat = abs(b1.std - b2.std)
191 comm.mat = rep(FALSE, length(diff.mat))
192 comm.mat[diff.mat < tol] = TRUE
193 names(comm.mat) = colnames(XX.country1)
194 comm.mat
195 # FALSE=unique, TRUE=common items
196
194:9 Section 6: Prevalence comparison between two count...
R Script

```

```

> ## Defining common items based on item severity difference
> diff.mat = abs(b1.std - b2.std)
> diff.mat
      WORRIED  HEALTHY  FEWFOOD  SKIPPED  ATELESS  RUNOUT  HUNGRY  WHLDAY
0.78609330 0.22070454 0.86349803 0.47135432 0.45519336 0.06686263 0.11654697 0.02390773
> comm.mat = rep(FALSE, length(diff.mat))
> comm.mat[diff.mat < tol] = TRUE
> names(comm.mat) = colnames(XX.country1)
> comm.mat
      WORRIED  HEALTHY  FEWFOOD  SKIPPED  ATELESS  RUNOUT  HUNGRY  WHLDAY
FALSE      TRUE  FALSE      TRUE      TRUE      TRUE      TRUE      TRUE

```

Dans cet exemple, les seuls (c.-à-d pas commune) items sont “WORRIED” (INQUIET) et “FEWFOOD”(PEUD’ALIMENT).

En utilisant le vecteur « comm.mat », nous définissons une métrique basée sur la moyenne et l’écart-type des items communs dans les deux pays :

$mean.comm = c(mean(b1.std[comm.mat]), mean(b2.std[comm.mat]))$

$sd.comm = c(sd(b1.std[comm.mat]), sd(b2.std[comm.mat]))$

mean.comm

sd.comm

```

196
197 ## Defining a metric based on mean and standard deviation of common items
198 ## in both countries
199 mean.comm = c(mean(b1.std[comm.mat]), mean(b2.std[comm.mat]))
200 sd.comm = c(sd(b1.std[comm.mat]), sd(b2.std[comm.mat]))
201 # Cells F14 and G14 in Excel
202 mean.comm
203 # Cells F15 and G15 in Excel
204 sd.comm
205
206 ## New standardized item variables
199:1 Section 6: Prevalence comparison between two count...
R Script

```

```

> comm.mat
      WORRIED  HEALTHY  FEWFOOD  SKIPPED  ATELESS  RUNOUT  HUNGRY  WHLDAY
FALSE      TRUE  FALSE      TRUE      TRUE      TRUE      TRUE      TRUE
> mean.comm = c(mean(b1.std[comm.mat]), mean(b2.std[comm.mat]))
> sd.comm = c(sd(b1.std[comm.mat]), sd(b2.std[comm.mat]))
> # Cells F14 and G14 in Excel
> mean.comm
[1] 0.2933567 0.2804627
> # Cells F15 and G15 in Excel
> sd.comm
[1] 0.9753521 0.9547392

```

Les nouvelles sévérités d’item normalisé sont également fondées sur cette mesure commune :

$b.1.std.new = (b1.std * sd.comm[1]) + mean.comm[1]$

$b.2.std.new = (b2.std * sd.comm[2]) + mean.comm[2]$

$cbind(b.1.std.new, b.2.std.new)$

The screenshot shows RStudio with the following R code in the editor:

```

rr.country2
sd.comm
## New standardized item severities
b.1.std.new = (b1.std * sd.comm[1]) + mean.comm[1]
b.2.std.new = (b2.std * sd.comm[2]) + mean.comm[2]
# Cells M3:M10 and N3:N10 in Excel
cbind(b.1.std.new, b.2.std.new)
# Graph
plot(b.1.std.new, b.2.std.new, pch = 5, col = "blue", xlab = "Country1",
      ylab = "Country2", xlim = c(-3,3), ylim = c(-3,3))

```

The Environment pane shows the following variables:

Variable	Type	Dimensions	Values
b.1.std.new	Named num	[1:8]	-0.274 -0.127 -0.856 0.55...
b.2.std.new	Named num	[1:8]	-1.0256 -0.3417 -0.0202 0...
b1	Named num	[1:8]	-0.492 -0.365 -0.997 0.22...
b1.std	Named num	[1:8]	-0.582 -0.431 -1.178 0.26...
b2	Named num	[1:8]	-1.593 -0.759 -0.367 -0.2...
b2.std	Named num	[1:8]	-1.368 -0.652 -0.315 -0.2...
comm.mat			
diff.mat	Named num	[1:8]	0.786 0.221 0.863 0.471 0...
mean.comm	num	[1:2]	0.293 0.28
pp.country1	num	[1:2]	0.708 0.435
rr.country1	List of 24		

The Console shows the execution of the code:

```

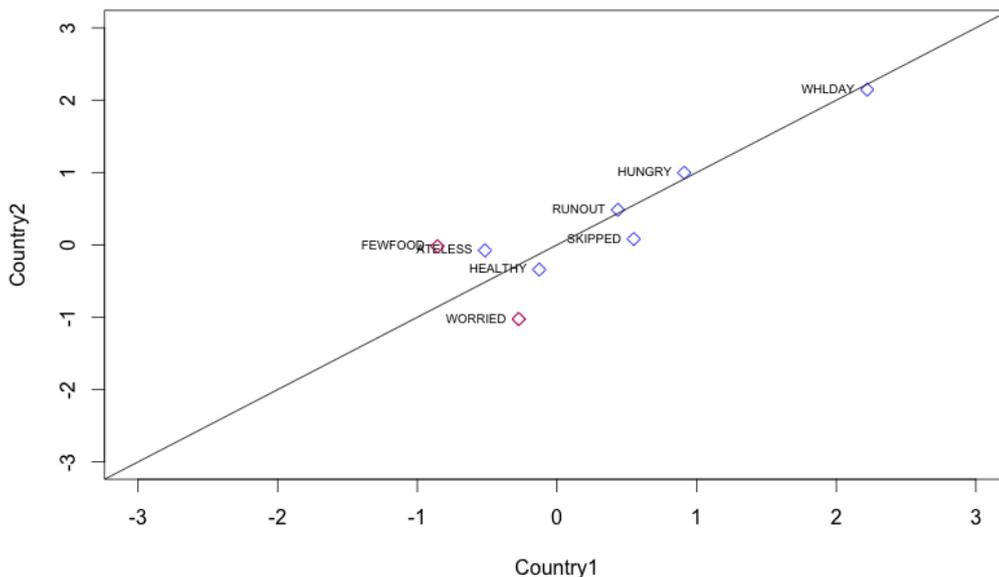
> # Cells M3:M10 and N3:N10 in Excel
> cbind(b.1.std.new, b.2.std.new)
      b.1.std.new b.2.std.new
WORRIED -0.2741519 -1.02556622
HEALTHY -0.1269238 -0.34165085
FEWFOOD -0.859638 -0.02015275
SKIPPED  0.5500760  0.08173618
ATELESS -0.5154890 -0.07669800
RUNOUT  0.4363201  0.48424120
HUNGRY  0.9117037  0.99701367
WHLDAY  2.2212092  2.14474666

```

The Environment pane also shows a description for the function:

Probabilistic assignment of cases to classes of severity along the latent trait.
 Description
 This function assigns cases to classes of severity along the latent trait.

Ces nouvelles sévérités normalisées sont tracées ci-dessous (points bleus pour courants, les points rouges pour des items uniques et la ligne de 45 degrés indique la sévérité égale) :



en utilisant le code suivant :

```
plot(b.1.std.new, b.2.std.new, pch = 5, col = "blue", xlab = "Country1", ylab = "Country2",
      xlim = c(-3,3), ylim = c(-3,3))
```

```
abline(c(0,1))
```

```
text(b.1.std.new, b.2.std.new, colnames(XX.country1), cex = 0.6, pos=2)
```

```
points(b.1.std.new[!comm.mat], b.2.std.new[!comm.mat], col = 2, pch = 5)
```

Les taux de prévalence comparables doivent être calculés selon les seuils ajustés à la même métrique commune.

Les seuils de déclaration sur la métrique des items communs :

```
int1=mean.comm[1]
slop1=sd.comm[1]/sd(b1)
int2=mean.comm[2]
slop2=sd.comm[2]/sd(b2)
sthresh = c(-0.25, 1.81)
sthresh.new1 = (sthresh - int1)/slop1
sthresh.new2 = (sthresh - int2)/slop2
```

Calculer de la prévalence en utilisant les seuils d'égalisation

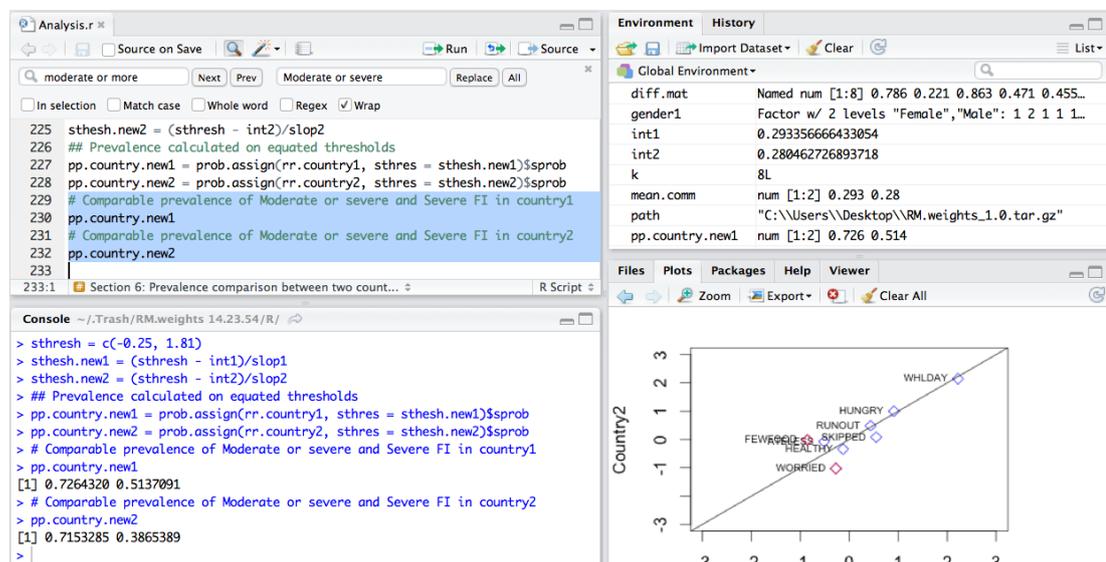
```
pp.country.new1 = prob.assign(rr.country1, sthres = sthresh.new1)$prob
pp.country.new2 = prob.assign(rr.country2, sthres = sthresh.new2)$prob
```

La prévalence comparable de Modéré, Sévère ou Sévère FI en *country1* est

```
pp.country.new1
```

alors que la prévalence comparable de Modéré, Sévère ou Sévère FI en *country2* est

```
pp.country.new2
```



Nous pouvons alors conclure que le taux de prévalence FI dans *country 1* (environ 51 %) est plus élevé que dans *country 2* (environ 39 %) parce que les deux quantités sont calculées sur une métrique commune.

5.1. La fonction “equating.fun”

Le processus d'égalisation comme décrit dans la section précédente est une étape particulièrement délicate de l'analyse, et il est préférable de ne pas l'exécuter automatiquement. Cependant, à titre de comparaison, la fonction « equating.fun » dans le *package* « RM.weights » peut être utilisée parallèlement avec le processus manuel pour effectuer l'égalisation et calculer les taux de prévalence comparables. Le processus de calibrage exécuté par cette fonction est le suivant : lors de la première étape, l'item le plus discordant (s'il dépasse la tolérance) est retiré de l'ensemble des items courants. Si tous les autres items s'harmonisent bien avec la norme mondiale (ou le pays de référence), la fonction s'arrête, sinon, le deuxième item discordant est écarté de l'ensemble commun, et ainsi de suite jusqu'à un nombre maximum d'items uniques.

Utilisation

```
equating.fun(rr1, st=NULL, tol = .35, spec.com1 = 1:8, spec.com2=1:8, thres = c(-0.25, 1.83),  
maxuniq=3, plot=F,iterative=T, excl.prior1, excl.prior2)
```

Arguments

rr1	Le modèle de Rasch adapté à la fonction RM.w pour le pays concerné.
st	Les mesures des paramètres de sévérité d'item pour le pays de référence ou de la norme. Le nombre d'items de la norme peut être différent du nombre d'items du pays concerné.
tol	La tolérance exprimée dans la métrique de la norme, au-dessus desquels items sont considérés comme unique (c.-à-d ne contribuant pas à la métrique commune entre les deux pays). La valeur par défaut est 0,35.
spec.com1	A priori un ensemble de numéros d'items comparables pour le pays concerné à la norme. Les items doivent être spécifiés dans le même ordre qu'ils sont utilisés pour la comparaison.
spec.com2	A priori un ensemble d'items comparables pour la norme. L'étendue de spec.com1 et spec.com2 doit être la même.
thres	Seuils (le long du trait latent) Correspondant auquel le taux de prévalence comparable du phénomène concerné est calculé.
maxuniq	A priori un nombre maximal d'items uniques autorisé.
plot	Argument logique. Si TRUE (vrai), un fichier pdf avec un tracé de la sévérité d'item équivalent est produit dans le répertoire de travail. La valeur par défaut est FALSE (faux).
iterative	Argument logique. Si TRUE, un processus itératif pour identifier quel item est unique est effectué. Autrement, les items uniques sont spécifiés dans les arguments excl.prior1 et excl.prior2 et sont pris comme fixé. La valeur par défaut est TRUE.
excl.prior1	Ensemble de numéros uniques fixés d'items pour les pays concernés. Elle doit être indiquée si seulement iterative = FALSE.
excl.prior2	Ensemble de numéros uniques fixé d'items pour la norme. Elle doit être indiquée si seulement iterative = FALSE.

Valeur

scale	Le paramètre scale (d'échelle) à appliquer à la métrique du pays concerné pour être considéré comme l'équivalent de la norme.
shift	Le paramètre shift à appliquer à la métrique du pays concerné pour être considéré comme équivalent au pays de référence. Les paramètres des items normalisés pour le pays concerné seront : $b.country.st = shift + scale * b.country$ où $b.country$ correspond à la sévérité des items mesurés pour le pays concerné.
common	Le vecteur de logique, TRUE lorsque l'item dans le pays concerné est commun.
prevs	Les taux de prévalence pour le pays concerné, calculés sur la métrique de la norme.
prevs.rs	Les taux de prévalence pour le pays concerné, calculés sur la métrique de la norme, calculé à chaque seuil de score brut.
cor.comm.items	Corrélation entre les items communs.

Un exemple d'application peut être trouvé ci-dessous (Section 7 de « Analysis.r »). L'exemple entraîne le calcul des taux de prévalence comparables pour les *country 1* utilisant comme norme la norme mondiale 2014-2015 calculé par VoH.

```
# VoH 2014-2015 global standard
b.tot=c(-1.2590036, -0.8991436, -1.0876362, 0.4163556, -0.2506451, 0.4466926,
0.8065710, 1.8268093)
# Equating of country 1 to the global standard
ee=equating.fun(rr.country1, st=b.tot, tol=0.5)
# Equated prevalence rates
ee$prevs*100
# Correlation between common items
ee$cor.comm.items
# Producing a plot of the item severities
ee=equating.fun(rr.country1, st=b.tot, tol=0.5, plot=T)
# The plot will be saved as a pdf called "Equating_plot.pdf" file in the working
directory
```