Journal of Statistical Software

October 2016, Volume 74, Issue 5. doi: 10.18637/jss.v074.i05

PerFit: An R Package for Person-Fit Analysis in IRT

Jorge N. Tendeiro
University of Groningen

Rob R. Meijer
University of Groningen

A. Susan M. Niessen
University of Groningen

Abstract

Checking the validity of test scores is important in both educational and psychological measurement. Person-fit analysis provides several statistics that help practitioners assessing whether individual item score vectors conform to a prespecified item response theory model or, alternatively, to a group of test takers. Software enabling easy access to most person-fit statistics was lacking up to now. The **PerFit** R package was written in order to fill in this void. A theoretical overview of relatively simple person-fit statistics is provided. A practical guide showing how the main functions of **PerFit** can be used is also given. Both numerical and graphical tools are described and illustrated using examples. The goal is to show how person-fit statistics can be easily applied to testing of questionnaire data.

Keywords: item response theory, aberrant response behavior, person fit, R.

1. Introduction

It is well known that total scores or estimated trait values do not always reflect the trait or proficiency level that a questionnaire or test intends to measure. The answers provided by test takers or respondents to questionnaires may be biased due to factors unrelated to the trait of interest. For example, cheating or item-preknowledge may inflate test scores whereas inattention or guessing may deflate test scores. It is important to be able to detect whether test scores are invalid, that is, whether test scores are biased and therefore not indicative of the true latent trait being measured. This fact is acknowledged in several guidelines for the reporting of test scores (e.g., International Test Commission 2014, Olson and Fremer 2013), and it applies to both educational and psychological measurement. Several person-fit statistics (PFSs) have been proposed to detect inconsistent, aberrant, or misfitting patterns of item scores. There are some empirical applications of person-fit analyses in the literature; see for example Meijer, Egberink, Emons, and Sijtsma (2008), Meijer and Tendeiro (2014), and

Conijn (2013) for applications in primary education, high-stakes testing, and clinical testing, respectively. However, some researchers and practitioners find it difficult to implement person-fit analyses in practice. This might be explained by the mathematical complexity of (some of the) PFSs and by the lack of software that helps practitioners conducting this type of analysis. It is our hope that the R (R Core Team 2016) **PerFit** package (Tendeiro 2016) discussed in this paper can be helpful to a wide range of practitioners (also, see Meijer, Niessen, and Tendeiro 2016 and Tendeiro and Meijer 2014 for accessible person-fit overviews).

This paper is divided into three main parts. First, we start with a theoretical overview of person-fit analysis in item response theory (IRT; Embretson and Reise 2000). The exposition will focus on PFSs that are currently implemented in **PerFit**. Second, we illustrate how the main functions of **PerFit** can be used in practice. Both numerical and graphical tools are described using empirical data that are also included in the R package. We explain how inconsistent item score patterns can be detected and we present a statistical tool that may be used in order to help finding an explanation for the cause of the misfit. Finally, we will summarize the main points of the paper and discuss further planned extensions for the package in the future.

2. Person-fit analysis in IRT

In the sequel we refer to persons as *test takers*, although all procedures also apply to questionnaires data. Moreover, we use θ to denote the trait or proficiency level that the questionnaire or test is measuring.

Let X_i be the random variable denoting the score on item i ($i = 1, \ldots, I$). The observed score of test taker n ($n = 1, \ldots, N$) on item i, that is, a realization of random variable X_i, will be denoted by x_{ni}. Items are *dichotomous* when there are $m = 2$ possible score categories or *polytomous* when there are three or more score categories ($m \geq 3$). In IRT it is assumed that θ induces the item responses. It is also assumed that the ordering of the item score categories reflects the hypothesized ordering on θ. The following item scores coding is used: 0, 1 for dichotomous items (e.g., 0 = incorrect, 1 = correct) and 0, 1, ..., $(m - 1)$ for polytomous items (e.g., 0 = disagree, 1 = neutral, 2 = agree). It is important that all items are keyed in the same direction, especially if the total score (or some measure based on the total score) is used.

An item score vector (or pattern) and its associated total score will be denoted by $\mathbf{x}_n = (x_{n1}, x_{n2}, \ldots, x_{nI})$ and $s_n = \sum_{i=1}^{I} x_{ni}$, respectively. The goal of person-fit analysis is to detect whether an individual item score pattern $\mathbf{x}_n$ ($n = 1, \ldots, N$) is unusual or unexpected given the item score patterns in a group of test takers (*group*-based analysis) or, alternatively, given a prespecified IRT model (*model*-based analysis). For comprehensive overview studies for both types of approaches we refer to Meijer and Sijtsma (2001) and Karabatsos (2003). Most of the PFSs currently available in **PerFit** are group-based. The exceptions are the log-likelihood statistics introduced by Drasgow, Levine, and Williams (1985) and improved by Snijders (2001). In what follows we will focus on the PFSs and related functions that are available in **PerFit** (also, see Tendeiro and Meijer 2014).

2.1. IRT models

It is important to clarify what type of IRT model assumptions should be verified before

person-fit analyses can be conducted. Typically, a functional relationship between θ and the probability of answering at or above a specific score category is conceptualized in IRT. This function is known as the item response function (IRF) in the dichotomous case or item step response function (ISRF) in the polytomous case and it is denoted as $\mathsf{P}_{ih}(\theta) = \mathsf{P}(X_i \geq h|\theta)$ for $i = 1, \ldots, I$ and $h = 1, \ldots, m-1$. We will use the acronym 'IRF' to refer to function $\mathsf{P}_{ih}(\theta)$ in either the dichotomous or the polytomous case. For dichotomous items there is only one IRF ($\mathsf{P}_i(\theta) = \mathsf{P}(X_i = 1|\theta)$) whereas for polytomous items there are $(m-1)$ IRFs (observe that $\mathsf{P}_{i0}(\theta) \equiv 1$).

Strictly speaking, group-based PFSs do not require closed-form mathematical IRFs to be estimated. Hence, *nonparametric* IRT models (NIRT; Sijtsma and Molenaar 2002) can be used. In NIRT the following model assumptions are typically imposed: (1) Unidimensionality: All items in the test predominantly measure the same latent trait, θ. (2) Local independence: Answers to different items are statistically independent conditional on θ. (3) Latent monotonicity of the IRFs: Each IRF is monotonically nondecreasing in θ. The NIRT model that satisfies assumptions 1 through 3 is known as the monotone homogeneity model (MHM; Mokken 1971), which is the most general nonparametric model. The double monotonicity model (DMM; Mokken 1971) further imposes the assumption that the IRFs of different items do not intersect. This assumption is known as the invariant item ordering (IIO) assumption. There are several tutorials available to aid practitioners interested in fitting these types of models to empirical data; see for example Sijtsma and Molenaar (2002) and Meijer, Tendeiro, and Wanders (2015). The R **mokken** package (Van der Ark 2007, 2012) can be used to check these assumptions in practice.

Both the MHM and the DMM are parameter-free, both in terms of person and of item parameters. Instead of estimating θs, the MHM (and therefore the DMM) relies on the total score statistic and on the property of stochastic ordering of the latent trait (SOL; Hemker, Sijtsma, Molenaar, and Junker 1997). The SOL property states that test takers are stochastically ordered on θ by means of the total score statistic. This ordering holds for dichotomous items and, for most practical purposes, also for polytomous items (however, see Van der Ark 2005). Also, instead of estimating item difficulty parameters, one uses the so-called item-step *difficulties*. An item-step difficulty π_{ih} is the population proportion of persons who answer at or above score category h for item i. This parameter is usually estimated by means of the corresponding sample proportion in the dataset. The item-step difficulty of a dichotomous item is the proportion of persons with score 1 and is also referred to as the item's *proportion-correct* score (denoted as π_i). Without loss of generality, and unless stated otherwise, in what follows it is assumed that the item steps are ordered in increasing order of difficulty, that is, in decreasing order of item-step difficulty. In particular for dichotomous items, this means that items are labeled in increasing order of difficulty (i.e., item 1 being the easiest and item I the most difficult). This property is especially useful when IIO is met (thus for the DMM) because it implies that the same ordering of item-step difficulties holds at any θ level. Nearly all group-based PFSs that are available in **PerFit** involve computations based on item/total scores and item-step difficulties (proportion-correct scores for dichotomous items).

Parametric IRT models (PIRT) are also often used to describe data (e.g., Embretson and Reise 2000). In this case closed-form mathematical functions for the IRFs are chosen, usually based on the logistic function. These functions are defined in terms of both person parameters (θs) and item parameters (their number and nature depend on the model of choice), which need to be estimated. There are several software packages available that allow fitting PIRT

Test taker A	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1	1	0	1	0	0	0	1	0	0
Test taker B	0	0	0	0	1	1	0	0	1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0
Test taker C	1	1	1	1	1	0	1	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	0
Test taker D	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	1

Table 1: Item score patterns corresponding to the PRFs displayed in Figure 1. Items are in increasing order of difficulty.

models to data. Within R some useful packages are **eRm** (Mair and Hatzinger 2007), **irtoys** (Partchev 2016), **ltm** (Rizopoulos 2006), **mcIRT** (Reif 2014), and **TAM** (Kiefer, Robitzsch, and Wu 2016). The quality of the fit of the model to the data requires analyzing the discrepancy between the observed item scores and the values expected under the PIRT model of interest (e.g., Maydeu-Olivares 2015). Several goodness of fit statistics are implemented in R, such as Pearson's χ^2 and the likelihood-ratio statistics (see packages **eRm**, **irtoys**, and **ltm**).

Practitioners are advised to analyze the fit of the IRT model to real data before attempting to interpret the validity of individual item response patterns.

2.2. Rationale behind person-fit

A typical test taker, when given a set of dichotomous items, is expected to answer very easy items correctly and very difficult items incorrectly. The more difficult the item, the lower the probability of answering the item correctly. This is the general principle behind the so-called *person response function* (PRF), which relates item difficulty with the probability of correctly answering an item. **PerFit** allows computing PRF plots; we will discuss this function at more length in Section 3.2. As an example, Figure 1 shows nonparametric PRFs (Emons, Sijtsma, and Meijer 2004; Sijtsma and Meijer 2001) for four different test takers that were estimated based on the answers of 1000 test takers on 26 dichotomous items (dataset `IntelligenceData` in **PerFit**). The x-axes display the item difficulty in terms of the values $(1 - \pi_i)$, thus values close to zero concern easier items and values close to one concern more difficult items.

The top-left panel shows the PRF of a typical test taker: The probability of answering an item correctly decreases as the item difficulty increases. The plots in the remaining panels of Figure 1 display deviations to the expected behavior. For example, the probability that test taker B answers an item correctly seems to *increase* with item difficulty in the interval $(0, .5)$. The PRF for test taker C increases in the interval $(.6, .8)$. Test taker D seems to be the most atypical of the four shown here: The PRF has mostly an increasing trend.

Table 1 shows the item scores of test takers A, B, C, and D, where the items (columns) are rearranged in increasing order of difficulty. Note that test takers B and D incorrectly answered several easy items. This fact helps explaining the increasing trend in the corresponding PRFs for low item difficulty values. Test taker C, on the other hand, answered 6 out of the 7 most difficult items correctly whereas several easier items were incorrectly answered. The corresponding PRF in Figure 1 does reflect this in terms of a local increasing trend for relatively difficult items.

Person-fit analysis can be used to detect test takers in a sample that display some kind of atypical response behavior as shown in the examples above. It should be observed that there are many psychological processes that may trigger atypical responses to a test or questionnaire. Some examples include cheating, guessing, plodding, alignment errors, extremely

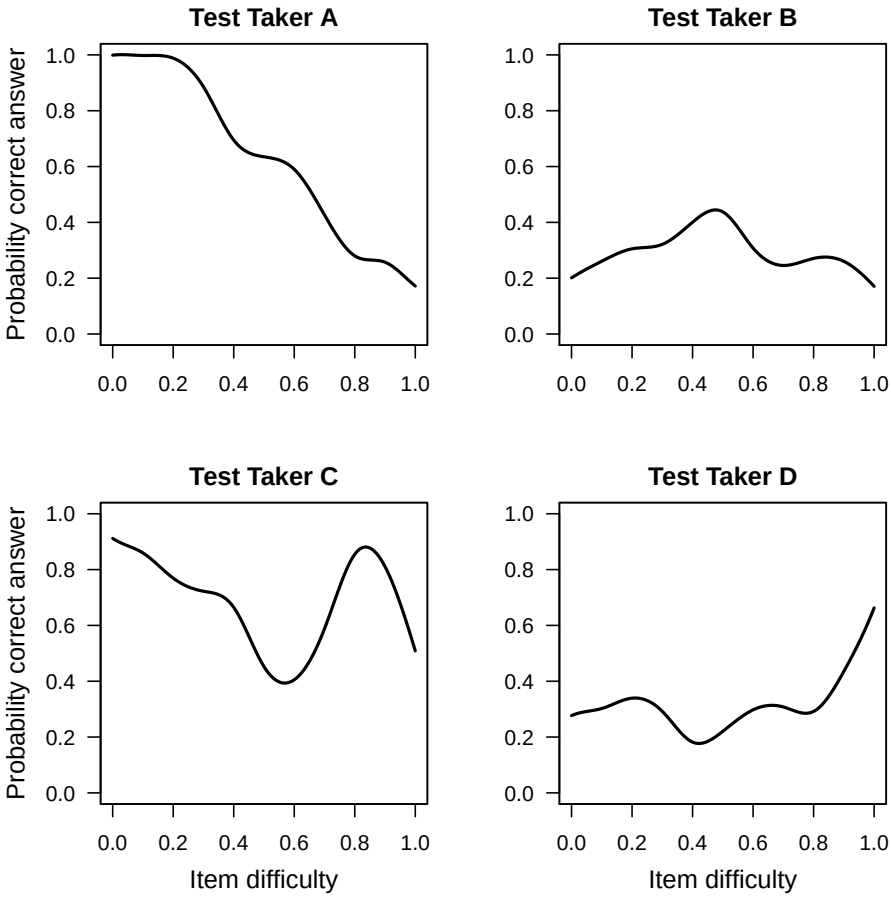

Figure 1: Person response functions of four test takers in dataset `IntelligenceData` from package **PerFit**.

creative interpretation of items, and deficient preparation in specific subparts of the contents being evaluated (Meijer 1996). It is possible that different psychological processes lead to similar atypical item score patterns, thus the *interpretation* of person-fit results is often not easy. As a general rule, person-fit analyses typically do not provide conclusive evidence as to exactly which abnormal phenomenon caused a strange item score pattern. Practitioners should complement and possibly follow up the analysis using other available tools, such as interviews with test takers (see Meijer *et al.* 2008) or relating the values of PFSs with other variables (Conijn 2013).

2.3. Person-fit statistics

Many PFSs (also referred to as *appropriateness indices*; Levine and Drasgow 1983) have been proposed in the literature. Karabatsos (2003) and Meijer and Sijtsma (2001) (see also Tendeiro and Meijer 2014 for an overview focusing on group-based statistics) discuss the most relevant statistics. In this paper we briefly describe each PFS that is implemented in **PerFit**.

Through **PerFit** (version 1.4) 18 PFSs can be calculated. Table 2 displays a summary of what the package offers.

Nonparametric IRT model				
PFS	**PerFit** function	Literature reference	Item type	Dist. tail
r.pbis	`r.pbis`	Donlon and Fischer (1968)	D	Lower
C	`C.Sato`	Sato (1975)	D	Upper
G, G_n	`G`, `Gnormed`	Van der Flier (1977), Meijer (1994)	D	Upper
A, E	`A.KB`, `E.KB`	Kane and Brennan (1980)	D	Lower
D	`D.KB`	Kane and Brennan (1980)	D	Upper
$U3$, $ZU3$	`U3`, `ZU3`	Van der Flier (1982)	D	Upper
C^*	`Cstar`	Harnisch and Linn (1981)	D	Upper
NCI	`NCI`	Tatsuoka and Tatsuoka (1982,1983)	D	Lower
H^t	`Ht`	Sijtsma (1986), Sijtsma and Meijer (1992)	D	Lower
G^p	`Gpoly`	Molenaar (1991), Emons (2008)	P	Upper
G_n^p	`Gnormed.poly`	Molenaar (1991), Emons (2008)	P	Upper
$U3^p$	`U3poly`	Emons (2008)	P	Upper
Parametric IRT model				
PFS	**PerFit** function	Literature reference	Item type	Dist. tail
l_z	`lz`	Drasgow *et al.* (1985)	D	Lower
l_z^p	`lzpoly`	Drasgow *et al.* (1985)	P	Lower
l_z^*	`lzstar`	Snijders (2001)	D	Lower

Table 2: Person-fit statistics included in **PerFit** (version 1.4). D = Dichotomous, P = Polytomous, Dist. tail = Distribution tail.

Likelihood-based PFSs

In the dichotomous case the log-likelihood of item score pattern $\mathbf{x}_n$ is given by

$$\log L(\theta_n) = \sum_{i=1}^{I} x_{ni} \ln \mathsf{P}_i(\theta_n) + (1 - x_{ni}) \ln[1 - \mathsf{P}_i(\theta_n)], \tag{1}$$

where P_i is typically the 1-, 2-, or the 3-parameter logistic model (1PLM, 2PLM, or 3PLM, respectively; Embretson and Reise 2000). Equation 1 is readibly extendable to the polytomous case by using the score category probabilities $\widetilde{\mathsf{P}}_{ih}(\theta) = \mathsf{P}_{ih} - \mathsf{P}_{i(h+1)}$ for $h = 0, \ldots, m-1$ with an associated indicator variable δ_{ih} (1 if the corresponding score category h was chosen, 0 otherwise):

$$\log L(\theta_n) = \sum_{i=1}^{I} \sum_{h=0}^{m-1} \delta_{ih} \ln \widetilde{\mathsf{P}}_{ih}(\theta_n). \tag{2}$$

Levine and Rubin (1979) proposed to use Equation 1 as a PFS denoted by l_0. Low l_0 values indicate misfitting item score patterns. However, the l_0 statistic has some limitations because it is not standardized nor is its distribution under a fitting IRT model known. Drasgow *et al.* (1985) proposed a standardized normal version of l_0 denoted as l_z (l_z^p in the polytomous case). Values of l_z are interpreted similarly as l_0, that is, the smaller the (negative) l_z values the stronger the indication of misfit. Statistic l_z is asymptotically (in terms of the number of items I) standard normally distributed. However, this approximation only holds when true θ values are used (Molenaar and Hoijtink 1990), which is an unrealistic requirement in practice. When estimated θ parameters replace θ in Equation 1 then it has been shown that

the variance of l_z is reduced (Meijer and Tendeiro 2012, Molenaar and Hoijtink 1990, Nering 1995, 1997, Reise 1995). Snijders (2001) proposed a corrected statistic known as l_z^* which takes the sampling variability of the estimated θ parameters into account. Magis, Raîche, and Béland (2012) thoroughly explained the computational formulas of l_z^*.

Group-based PFSs

Group-based or nonparametric PFSs are not constrained to fitting closed-form mathematical models to the data (although the unidimensionality, local independence, and monotonicity assumptions previously mentioned need still be checked; see Sijtsma and Molenaar 2002). More specifically, an item score pattern is flagged as misfitting in case it deviates to a large extent from the most typical response behavior in the sample. Group-based PFSs are usually computed using the item-step difficulties in the polytomous case (which are simply equal to the proportion-correct scores in the dichotomous case).

Donlon and Fischer (1968) considered the personal point-biserial correlations. *r.pbis* is the correlation between the test taker's item score pattern $\mathbf{x}_n$ and the vector of item proportion-correct scores in the sample $\mathbf{p} = (p_1, \ldots, p_I)$. The p_i score, also known as the difficulty or p-value of item i ($i = 1, \ldots, I$), is the proportion of persons who answered item i correctly. Low values of *r.pbis* may be indicative of misfit of the response vector.

Sato (1975) proposed the PFS C, which is a covariance ratio measuring the extent to which an item score pattern deviates from the perfect pattern (Guttman pattern). Assume that the items are ordered in increasing order of difficulty, that is, $p_1 \geq p_2 \geq \cdots \geq p_I$, without loss of generality. The formula for C is

$$C = 1 - \frac{\mathsf{COV}(\mathbf{x}_n, \mathbf{p})}{\mathsf{COV}(\mathbf{x}_n^*, \mathbf{p})}, \tag{3}$$

where $\mathbf{x}_n^*$ is the so-called Guttman vector containing correct answers for the s_n easiest items (i.e., with the largest p-values) only. C is zero for Guttman vectors and its value increases (without theoretical bound) for item score patterns that deviate more from the perfect Guttman pattern, thus higher values indicate more deviant response behavior. Harnisch and Linn (1981) further adapted C by norming it between 0 and 1:

$$C^* = \frac{\mathsf{COV}(\mathbf{x}_n^*, \mathbf{p}) - \mathsf{COV}(\mathbf{x}_n, \mathbf{p})}{\mathsf{COV}(\mathbf{x}_n^*, \mathbf{p}) - \mathsf{COV}(\mathbf{x}_n', \mathbf{p})}, \tag{4}$$

where $\mathbf{x}_n'$ is the reversed Guttman vector containing correct answers for the s_n hardest items (i.e., with the smallest p-values) only. C^* is sensitive to the so-called Guttman errors. A Guttman error is a pair of scores (0, 1), where the 0-score pertains to the easiest item and the 1-score pertains to the hardest item. C^* ranges between 0 (perfect Guttman vector) and 1 (reversed Guttman vector).

The $U3$ statistic (Van der Flier 1982) is similar to C^* except that it is based on the log-odds of the item proportion-correct scores. It is given by

$$U3 = \frac{f(\mathbf{x}_n^*) - f(\mathbf{x}_n)}{f(\mathbf{x}_n^*) - f(\mathbf{x}_n')}, \tag{5}$$

where $f(\mathbf{x}_n)$ denotes the summation $\sum_{i=1}^{I} x_{ni} \log[p_i/(1-p_i)]$. A standardized normal version (statistic $ZU3$) was also developed (see Van der Flier 1982). However, the adequacy of this

asymptotic approximation has been questioned by some researchers (e.g., Emons, Meijer, and Sijtsma 2002).

Kane and Brennan (1980) introduced the agreement (A), disagreement (D), and dependability (E) statistics, which assess the similarity (A and E) and dissimilarity (D) between the vectors $\mathbf{x}_n$ and $\mathbf{p}$. The formulas are

$$A = \sum_{i=1}^{I} x_{ni} p_i, \quad D = \max(A|s_n) - A, \quad \text{and} \quad E = \frac{A}{\max(A|s_n)}, \tag{6}$$

where $\max(A|s_n)$ equals the sum of the s_n largest p_i scores. Small values of A and E (i.e., in the left tail of the sampling distribution) are potentially indicative of aberrant response behavior, whereas large values of D (i.e., in the right tail of the sampling distribution) are potentially indicative of aberrant response behavior.

Sijtsma (1986; see also Sijtsma and Meijer 1992) proposed the H^t statistic, which is based on an idea presented in Mokken (1971). Sijtsma observed that Mokken had already introduced an index H_i that allowed assessing the scalability of an item to the Guttman model (Guttman 1944, 1950). Sijtsma (1986) used the same index applied to the *transposed* data in order to detect test takers that would not comply with the Guttman model. Assume, without loss of generality, that the rows of the data matrix are ordered to increasing order of total score s_n ($n = 1, \ldots, N$). The H^t statistic is given by

$$H^t = \frac{\sum_{n \neq m}(t_{nm} - t_n t_m)}{\sum_{n>m}(t_m - t_n t_m) + \sum_{n<m}(t_n - t_n t_m)}, \tag{7}$$

with $t_n = s_n/I$, $t_m = s_m/I$, and t_{nm} is the proportion of items answered correctly by both test takers n and m (Sijtsma and Molenaar 2002, p. 57). H^t takes its maximum value 1 when $t_{nm} = t_n$ ($n < m$) and $t_{nm} = t_m$ ($n > m$). This means that no test taker with a total score smaller/larger than t_n can answer an item correctly/incorrectly that test taker n has answered incorrectly/correctly, respectively. H^t equals zero when the average covariance of the response pattern of test taker n with the other response patterns equals zero.

Van der Flier (1980; see also Meijer 1994; Tatsuoka and Tatsuoka 1982) proposed the number of Guttman errors as a PFS, here denoted G. Thus, G is the number of (0, 1) pairs of item scores in the item score vector with the incorrect answer given to the easiest item and the correct answer given to the most difficult item. G is an integer and its upper bound is a function of the test length, which is inconvenient. As a result, this statistic has been normalized, leading to statistic G_n. The normalization is done against the maximum number of possible Guttman errors given the test taker's total score s_n (which is $s_n(I - s_n)$ in the dichotomous case). The norm conformity index (NCI) defined by Tatsuoka and Tatsuoka (1983) is perfectly linearly related to G ($NCI = 1 - 2G_n$).

A version of the $U3$ statistic adapted to polytomous items was proposed by Emons (2008). This statistic relies on the concept of *item steps* (Molenaar 1982). Assume that all items have the same number m of *ordered* score categories. A test taker may take step h ($h = 1, \ldots, m-1$), from score category $(h-1)$ to h, only if all previous steps were taken. The goal is to associate the test taker's latent ability with the probability of taking each item step. Define π_{ih} as the difficulty of item step h on item i, consisting of the population proportion of test takers with a score on item i at least equal to h. The item step difficulties of all items can be estimated in a sample and ranked in decreasing order (i.e., in increasing order of item-step

difficulty), so that $\widehat{\pi}_1 \geq \widehat{\pi}_2 \geq \cdots \geq \widehat{\pi}_{I(m-1)}$. Item-step scores of a test taker (1 if a step is taken, 0 otherwise) are collected in an $I(m-1)$ vector $\mathbf{y} = (y_1, y_2, \ldots, y_{I(m-1)})$. The $U3^p$ statistic (Emons 2008) is defined by

$$U3^p = \frac{\sum_{k=1}^{X_+} \text{logit}(\widehat{\pi}_k) - W(\mathbf{y})}{\sum_{k=1}^{X_+} \text{logit}(\widehat{\pi}_k) - \min(W|X_+)}, \tag{8}$$

with $W(\mathbf{y}) = \sum_{k=1}^{I(m-1)} y_k \,\text{logit}(\widehat{\pi}_k)$ and $X_+ = \sum_{k=1}^{I(m-1)} y_k$. The value $\min(W|X_+)$ cannot be written in closed form; its value can be found by means of a recursive algorithm (Emons 2008).

Extensions of the Guttman-based dichotomous statistics to polytomous items are also available in the package. Using the same notation introduced in the previous, statistic G^p is defined by

$$G^p = \sum_{l<k}^{I(m-1)} y_k(1 - y_l) \tag{9}$$

and the normed polytomous version is

$$G_n^p = \frac{G^p}{\max(G^p|X_+)}. \tag{10}$$

In this case also the value $\max(G^p|X_+)$ needs to be computed by means of a recursive algorithm (Emons 2008).

2.4. Comparative performance of PFSs

Given the wealth of PFSs available to practitioners, it is important to know which statistics outperform others, that is, which statistics have the highest power to detect aberrant response behavior while keeping the false positive rates under control. Assessing the performance of competing statistical strategies is traditionally done by means of simulation studies, and in rare cases using empirical data. Results from several studies are indeed available (Birenbaum 1985, 1986; Drasgow, Levine, and McLaughlin 1987; Harnisch and Linn 1981; Harnisch and Tatsuoka 1983; Karabatsos 2003; Kogut 1987; Li and Olejnik 1997; Meijer 1994; Meijer, Muijtjens, and van der Vleuten 1996; Nering and Meijer 1998; Noonan, Boss, and Gessaroli 1992; Rogers and Hattie 1987; Rudner 1983; Tendeiro and Meijer 2014). Findings may vary according to the design of the simulation study (see Rupp 2013) and which PFSs were used in the study. However, a tentative general conclusion is that simple group-based PFSs like H^t and $U3$ do not perform worse, and in some cases perform better, than most model-based PFSs across different types of datasets (Karabatsos 2003; Tendeiro and Meijer 2014). This is good news for practitioners in the sense that (1) group-based PFSs are typically easier to compute, and (2) nonparametric measurement models are less restrictive to empirical data than parametric logistic IRT models (Sijtsma and Molenaar 2002).

2.5. Current software available

From the overview given above it is clear that there are many PFSs available that can be used to assess the validity of a test taker's test score. One of the most serious limitations of PFSs, however, is their implementation in practice. The relatively advanced statistical

knowledge required to apply person-fit techniques in educational and psychological assessment and the lack of accessible software withholds substantive researchers and practitioners from using person-fit analyses with their own data. The few existing options available (**WPERFIT**, Ferrando and Lorenzo 2000; **PERSONz**, Choi 2010) do not offer group-based PFSs which were shown to perform very well in several simulation studies. Some PFSs are already available in R, such as l_z (see packages **irtoys**; **ltm**; **mirt**, Chalmers 2012; and **sirt**, Robitzsch 2016) and the Rasch model-based outfit and infit statistics (Wright and Masters 1990; see packages **eRm**, **mirt**, and **sirt**). Only the **sirt** package currently offers some group-based PFSs (C^*, D, $r.pbis$, and $U3$). **PerFit** offers a broader set of PFSs, both model- and group-based. In particular, **PerFit** includes H^t which has been shown to outperform most PFSs in several simulation studies (Karabatsos 2003; Tendeiro and Meijer 2014), l_z^* as an updated version of the commonly used l_z statistic, and several PFSs suited to polytomously scored data.

Moreover, the ability to estimate PRFs that reflect *observed* deviations of individual response vector patterns with respect to the test takers group trend is of great value, as discussed in Section 2.2. Package **irtProb** (Raîche 2014) already allows plotting person characteristic curves, which are *expected* curves under parametric IRT models. We propose a graphical tool (function `PRFplot()`) that relies on observed item scores to better interpret person misfit, based on a nonparametric approach.

In combination with the other R packages already mentioned, **PerFit** is deemed as a major contribution to users relying on only one computing platform in order to conduct their data analyses.

3. Using the PerFit package

3.1. Computing PFSs

Package **PerFit** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=PerFit` and can be installed from there using:

```
R> install.packages("PerFit")
```

The PFSs available in **PerFit** are listed in Table 2. Depending on the PFS, $N \times I$ matrices with scores 0, 1 (in the dichotomous case) or scores $0, 1, \ldots, m-1$ (in the polytomous case) need to be loaded in R. The number of score options is required to be the same for all items. Missing values are allowed. The package includes three datasets by default. The `InadequacyData` ($N = 806$, $I = 28$) and the `IntelligenceData` ($N = 1000$, $I = 26$) datasets consist of dichotomously scored items whereas `PhysFuncData` ($N = 714$, $I = 10$) consists of polytomously scored items (each with three item categories).

Typically, to use a PFS function one may only provide the data matrix (argument `matrix`). For example, to compute the H^t values for dataset `IntelligenceData`, run the following code:

```
R> library("PerFit")
R> data("IntelligenceData", package = "PerFit")
R> Ht("IntelligenceData")
```

Function `Ht` has more arguments but all have predefined defaults. In general, the arguments for dichotomous PFS functions are

```
(matrix,
   NA.method = "Pairwise", Save.MatImp = FALSE,
   IP = NULL, IRT.PModel = "2PL", Ability = NULL, Ability.PModel = "ML",
   mu = 0, sigma = 1)
```

and the arguments for polytomous PFS functions are

```
(matrix, Ncat,
   NA.method = "Pairwise", Save.MatImp = FALSE,
   IP = NULL, IRT.PModel = "GRM", Ability = NULL, Ability.PModel = "EAP")
```

The default approach to dealing with missing values is pairwise deletion (i.e., `NA.method = "Pairwise"`). This procedure is adequate under the missing completely at random (MCAR) mechanism, and Zhang and Walker (2008) showed that the pairwise deletion method performed well under MCAR. Alternatively, three imputation methods are also available (see Zhang and Walker 2008 for an overview of these and related methods):

- `NA.method = "Hotdeck"`: Hotdeck imputation replaces missing responses of a test taker ('recipient') by item scores from the test taker which is closest to the recipient ('donor'), based on the recipient's nonmissing item scores. The similarity between nonmissing item scores of recipients and donors is based on the sum of absolute differences between the corresponding item scores. The donor's response pattern is deemed to be the most similar to the recipient's response pattern in the group, so item scores of the former are used to replace the corresponding missing values of the latter. When multiple donors are equidistant to a recipient, one donor is randomly drawn from the set of all donors.

- `NA.method = "NPModel"` (default): The nonparametric model imputation method is similar to the hotdeck imputation, but item scores are generated from Bernoulli distributions (for dichotomous items) or multinomial distributions (for polytomous items) with probabilities defined by donors with similar total score than the recipient (based on all items except the NAs).

- `NA.method = "PModel"`: The parametric model imputation method is similar to the hotdeck imputation, but item scores are generated from Bernoulli distributions (for dichotomous items) or multinomial distributions (for polytomous items) with probabilities estimated by means of parametric IRT models (see below).

The user may choose to save the imputed data matrix to a file (`Save.MatImp = TRUE`).

A word of caution concerning missing values is in order here. When the proportion of missing values is small and follows the MCAR mechanism, the missing values procedures available within **PerFit** are probably well suited for most practical purposes. But in case more elaborate schemes to tackle missing values are required (e.g., by means of multiple imputation or maximum likelihood estimation) then users are advised to resort to specialized R packages (e.g., **Amelia**, Honaker, King, and Blackwell 2011; **mi**, Su, Gelman, Hill, and Yajima 2011; **MissingDataGUI**, Cheng, Cook, and Hofmann 2016). However, when the proportion of missing

values is large, both strategies (pairwise deletion and imputation) are not to be recommended because they may lead to biased person-fit results. For large proportions of missing values it is probably better not to calculate a person-fit statistic. Therefore, we strongly advice the user to *first* inspect the proportion of missing values for each test taker and then to decide, on the basis of these results, what the best strategy may be. One may even conceive considering both options and then to compare results.

The estimation of parametric item and ability parameters for group-based PFSs is only required if the parametric model imputation method is chosen. For all likelihood-based PFSs in **PerFit**, the model parameters need to be either provided (arguments `IP` and `Ability`) or internally estimated by the function. As an example, the following code shows how item parameters estimated using package **mirt** can be provided to compute the l_z statistic:

```
R> library("mirt")
R> mod <- mirt(IntelligenceData, 1)
R> ip.mirt <- coef(mod, IRTpars = TRUE, simplify = TRUE, digits = Inf)$
+    items[, c("a", "b", "g")]
R> lz.out  <- lz(IntelligenceData, IP = ip.mirt)
```

The default for dichotomous data is based on fitting the 2PLM to the data using functions available from the **irtoys** package (`IRT.PModel = "2PL"`). Ability parameters are by default estimated via maximum likelihood (`Ability.PModel = "ML"`). Alternatively, other dichotomous IRT models that can be used to describe the data are the 1PLM (`IRT.PModel = "1PL"`) and the 3PLM (`IRT.PModel = "3PL"`), whereas the ability parameters may also be estimated by means of Bayes modal estimation (`Ability.PModel = "BM"`) or weighted likelihood estimation (`method = "WL"`). For polytomous data the defaults are the graded response model (`IRT.PModel = "GRM"`) with expected a posteriori estimates of ability parameters (`Ability.PModel = "EAP"`). Other model options for polytomous data include the partial credit model (`IRT.PModel = "PCM"`) and the generalized partial credit model (`IRT.PModel = "GPCM"`), whereas estimation of ability parameters may be performed by means of empirical Bayes estimation (`Ability.PModel = "EB"`) or multiple imputation estimation (`Ability.PModel = "MI"`).

The output from a PFS function is an object of class '`PerFit`', which consists of a list with 12 elements:

1. `PFscores`: A list of length N with the values of the PFS.

2. `PFStatistic`: The PFS used.

3. `PerfVects`: A message indicating whether perfect response vectors (all-0s or all-1s) were removed from the analysis. Perfect response vectors need to be removed for some dichotomous-based PFSs.

4. `ID.all0s`: Row indices of all-0s response vectors removed from the analysis (if applicable).

5. `ID.all1s`: Row indices of all-1s response vectors removed from the analysis (if applicable).

6. `matrix`: The data matrix after missing values imputation (if applicable; otherwise it is the original data matrix).

7. `Ncat`: The number of response categories.

8. `IRT.PModel`: The parametric IRT model used in case `NA.method = "PModel"`, otherwise `NULL`.

9. `IP`: The matrix of item parameters in case `NA.method = "PModel"`, otherwise `NULL`.

10. `Ability.PModel`: The method used to estimate abilities in case `NA.method = "PModel"`, otherwise `NULL`.

11. `Ability`: The vector of N ability parameters in case `NA.method = "PModel"`, otherwise `NULL`.

12. `NAs.method`: The method used to deal with missing values.

For polytomous PFSs (`Gpoly`, `Gnormed.poly`, `lzpoly`, and `U3poly`) the number of score categories is set using the argument `Ncat`. For example, using the `PhysFuncData` dataset:

```
R> data("PhysFuncData", package = "PerFit")
R> U3poly.out <- U3poly(PhysFuncData, Ncat = 3)
R> U3poly.out$PFscores

         PFscores
Resp.1     0.2633
Resp.2     0.0776
Resp.3     0.3437
...
Resp.713   0.0000
Resp.714   0.0000
```

Based on the same data, next is an example using the `lzpoly` statistic where the item parameters for the graded response model were estimated previously and used later in the function:

```
R> library("ltm")
R> IP.est <- coef(grm(PhysFuncData, constrained = FALSE, IRT.param = TRUE))
R> lzpoly.out <- lzpoly(PhysFuncData, Ncat = 3, IP = IP.est)
R> lzpoly.out$PFscores

         PFscores
Resp.1    -1.1397
Resp.2    -0.5307
Resp.3    -0.8348
...
Resp.713   0.3991
Resp.714   0.5972
```

3.2. Extra functions

Aside from the PFSs currently included in the package there are more functions available, namely: `PerFit.PFS`, `PerFit.SE`, `cutoff`, `flagged.resp`, the `print`, `summary`, and `plot` methods in **PerFit**, and `PRFplot`.

The `PerFit.PFS` function

```
PerFit.PFS(matrix, method = NULL, simplified = TRUE, NA.method = "Pairwise",
  Save.MatImp = FALSE, IP = NULL, IRT.PModel = NULL, Ability = NULL,
  Ability.PModel = NULL, mu = 0, sigma = 1)
```

Function `PerFit.PFS` is a wrapper allowing to compute more than one person-fit statistic simultaneously. A vector of several PFSs may be provided to argument `method`. If `simplified = TRUE`, an $N \times k$ data frame is returned, where N is the number of test takers and k is the number of PFS methods. If `simplified = FALSE` a list of k '`PerFit`' objects is returned.

```
R> PFS.out <- PerFit.PFS(IntelligenceData, method = c("U3", "lzstar", "Ht"),
+    simplified = TRUE)
R> PFS.out

      U3  lzstar     Ht
1 0.2406 -1.1838 0.2794
2 0.1131  0.4650 0.4356
3 0.2076 -0.2973 0.4352
4 0.1381  0.2051 0.3969
5 0.1739 -0.0106 0.4451
6 0.1230  0.1905 0.4463
...
```

The `PerFit.SE` function

```
PerFit.SE(x)
```

Function `PerFit.SE` computes jackknife standard errors for the scores of the person-fit statistic in object x. The process consists of computing the PFS for each dataset omitting the ith item ($i = 1, \ldots, I$), followed by the computation of the jackknife standard deviation of the I values of the PFS, for each test taker in the dataset.

The output is a matrix with two columns: `PFscores` shows the values of the person-fit statistic and `PFscores.SE` shows the estimated standard errors. For example, the standard errors of the H^t values computed from the intelligence data can be found as follows:

```
R> Ht.out <- Ht(IntelligenceData)
R> Ht.SE <- PerFit.SE(Ht.out)
R> Ht.SE
```

```
     PFscores PFscores.SE
[1,]   0.2794      0.1436
[2,]   0.4356      0.1081
[3,]   0.4352      0.2023
[4,]   0.3969      0.1144
[5,]   0.4451      0.1579
[6,]   0.4463      0.1179
...
```

The `cutoff` *function*

```
cutoff(x, ModelFit = "NonParametric", Nreps = 1000, IP = x$IP, IRT.PModel =
  x$IRT.PModel, Ability = x$Ability, Ability.PModel = x$Ability.PModel,
  mu = 0, sigma = 1, Blvl = 0.05, Breps = 1000, CIlvl = 0.95, UDlvl = NA)
```

This function allows estimating a cutoff value at a prespecified level. PFS values in the sample at or more extreme than the cutoff value lead to flagging the corresponding item score vectors as misfitting. The direction according to which a value may be considered extreme (i.e., larger than/smaller than the cutoff) varies. The last column in Table 2 shows which tail of the distribution indicates misfit. The `cutoff` function routinely reports which tail of the PFS distribution is used (`Tail = "lower"` or `Tail = "upper"`).

The procedure consists of generating `Nreps` model-fitting item score vectors based on parametric models (`ModelFit = "Parametric"`) or on sample proportions of test takers per answer category (`ModelFit = "NonParametric"`). The item parameters (either parametric or nonparametric) are estimated from the original data matrix. The `Nreps` ability parameters (under the parametric approach) or total scores (under the nonparametric approach) are randomly sampled with replacement from the set of N ability parameters or total scores in the sample, respectively. The probability of each simulee endorsing each answer option is then estimated under each modeling framework. Next, item scores are randomly generated by means of the multinomial distribution. Finally, `Nreps` values of the PFS corresponding to model-fitting item response patterns are computed.

A bootstrap procedure is then used to approximate the sampling distribution of the quantile of level `Blvl` (resp., `1 - Blvl`) for "lower" (resp. "upper") types of PFS, based on `Breps` resamples. The cutoff (and its standard error) is given by the median (standard deviation) of this bootstrap distribution. A bootstrap percentile confidence interval of level `CIlvl` is also reported.

The cutoff may alternatively be manually entered by the user (e.g., when it is available from prior data calibration) by means of `UDlvl`. The goal is then to compute the proportion of test takers in the sample that are flagged by means of this cutoff value. In this case the bootstrap standard error and confidence intervals are not reported.

The `cutoff` function should be used on objects of class '`PerFit`'. For example, the cutoff of the H^t values computed from the intelligence data based on a nominal 10% error rate is found as follows:

```
R> set.seed(123)
R> cutoff(Ht.out, Blvl = 0.10)
```

```
$Cutoff
[1] 0.3054

$Cutoff.SE
[1] 0.0063

$Prop.flagged
[1] 0.141

$Tail
[1] "lower"

$Cutoff.CI
  2.5%  97.5%
0.2956 0.3216

attr(,"class")
[1] "PerFit.cutoff"
```

The seed was fixed in the command above for replicability, since the estimation of the cutoff value may vary between runs due to the resampling procedure.

The output is an object of class '`PerFit.cutoff`', consisting of a list with five elements: The cutoff value (`$Cutoff`), the associated standard error (`$Cutoff.SE`), the proportion of item score vectors in the sample that will be flagged by means of this cutoff (`$Prop.flagged`), which tail of the distribution is relevant (`$Tail`), and the `CIlvl`% confidence interval (`$Cutoff.CI`).

The ***`flagged.resp`*** *function*

```
flagged.resp(x, cutoff.obj = NULL, scores = TRUE, ord = TRUE, ModelFit =
  "NonParametric", Nreps = 1000, IP = x$IP, IRT.PModel = x$IRT.PModel,
  Ability = x$Ability, Ability.PModel = x$Ability.PModel, mu = 0, sigma = 1,
  Blvl = 0.05, Breps = 1000, CIlvl = 0.95, UDlvl = NA)
```

This function finds which test takers in the sample are flagged by a PFS, based on a cutoff value. The PFS is specified by means of the '`PerFit`' object `x`. The cutoff (i.e., an object of class '`PerFit.cutoff`') may be provided using the argument `cutoff.obj`. If no cutoff is provided then it will be internally computed, for which the function parameters `ModelFit` through `UDlvl` are required (see function `cutoff` above). If `scores = TRUE` then the test takers' item scores will be shown in the output, either in the original item order (`ord = FALSE`) or in increasing difficulty order determined by decreasing proportion-correct values (`ord = TRUE`). The latter option is useful for interpretation as we explained in Section 2.2.

Table 1 (page 4) shows four item score vectors from the intelligence data (test takers A, B, C, and D correspond to rows 6, 584, 832, and 772, respectively), with items ordered in increasing order of difficulty (`ord = TRUE`). The last three item score vectors were detected by $U3$ using a cutoff at a 1% nominal level, as shown below.

```
R> U3.out <- U3(IntelligenceData)
R> set.seed(72)
R> U3.cutoff <- cutoff(U3.out, Blvl = .01)
R> flagged.resp(U3.out, cutoff.obj = U3.cutoff, ord = TRUE)

$Scores
      FlaggedID It8 It10 It7 It14 It4 It9 It11 It16 It6 It1 It20 It23 It17
 [1,]       278   1    1   0    1   1   0    1    0   0   1    1    1    1
 [2,]       478   1    1   1    1   1   1    1    1   1   1    1    1    0
 [3,]       479   0    1   0    1   1   1    1    1   1   0    1    1    1
 [4,]       486   1    1   1    1   1   1    1    1   0   1    1    1    1
 [5,]       558   1    1   1    1   0   1    1    1   1   1    1    1    1
 [6,]       584   0    0   0    0   1   1    0    0   1   0    0    1    1
 [7,]       588   0    1   1    0   1   1    0    1   0   1    1    1    1
 [8,]       606   0    1   1    1   1   1    1    1   1   1    1    1    1
 [9,]       731   1    1   1    1   0   0    1    1   1   1    1    1    1
[10,]       772   0    0   1    0   0   1    0    1   0   0    0    0    1
[11,]       832   1    1   1    1   1   0    1    0   1   1    1    0    0
[12,]       971   0    1   1    0   1   1    0    1   0   0    1    1    1
[13,]       990   1    1   0    0   1   0    1    1   1   1    0    0    1

      It13 It3 It12 It2 It18 It25 It5 It15 It22 It21 It24 It19 It26 PFscores
 [1,]    0   1    0   1    1    1   0    1    1    1    0    0    1   0.5069
 [2,]    1   1    1   1    1    1   0    1    1    1    1    1    1   0.5304
 [3,]    0   0    0   0    0    1   0    0    1    0    1    0    1   0.4155
 [4,]    1   1    1   1    1    1   1    1    0    0    1    1    1   0.4034
 [5,]    1   0    1   1    1    1   0    1    1    1    1    1    1   0.6387
 [6,]    0   1    0   0    0    0   0    1    0    0    1    0    0   0.4608
 [7,]    0   1    0   0    1    0   0    0    1    1    1    1    0   0.4547
 [8,]    1   1    1   1    1    1   1    1    1    1    1    1    0   0.4011
 [9,]    1   0    0   0    1    1   0    0    1    0    1    0    1   0.3929
[10,]    0   0    0   0    1    0   1    0    0    0    1    0    1   0.6032
[11,]    1   1    0   0    0    0   1    1    1    1    1    1    0   0.3798
[12,]    0   1    0   0    0    1   1    0    1    0    0    0    1   0.4396
[13,]    1   0    0   0    0    0   1    0    1    1    0    1    0   0.3909

$MeanItemValue
   I8   I10    I7   I14    I4    I9   I11   I16    I6    I1   I20   I23
0.871 0.868 0.861 0.846 0.835 0.821 0.776 0.718 0.657 0.626 0.602 0.505
  I17   I13    I3   I12    I2   I18   I25    I5   I15   I22   I21   I24
0.499 0.497 0.472 0.452 0.406 0.377 0.344 0.342 0.313 0.255 0.139 0.133
  I19   I26
0.131 0.022

$Cutoff.lst
$Cutoff
[1] 0.3784
```

```
$Cutoff.SE
[1] 0.0201

$Prop.flagged
[1] 0.013

$Tail
[1] "upper"

$Cutoff.CI
  2.5%  97.5%
0.3320 0.4064

attr(,"class")
[1] "PerFit.cutoff"

$PFS
[1] "U3"
```

The output is a list with three elements. The first element is a matrix, `$Scores`, with the row rank score identifying the flagged test takers (first column), the item scores, and the PFS values (last column). The second element is a vector with the proportion-correct values (`$MeanItemValue`). The third element is the '`PerFit.cutoff`' object. When the option `scores = FALSE` is used then the output omits the item scores and the items' proportion-correct values.

The `print` *and* `summary` *methods in* **PerFit**

A summary for objects of class '`PerFit`' is also available. To make the summary more practically useful to the user we combined the information provided by the object with the outcome from both the `cutoff` and the `flagged.resp` functions. Continuing the example above:

```
R> summary(U3.out, cutoff.obj = U3.cutoff)

PFS = U3
Cutoff = 0.3784 (SE = 0.0201).
Tail = upper.
Proportion of flagged respondents = 0.013.
(N.B.: The cutoff varies each time cutoff() is run due to bootstrapping.)

Identified respondents - 13 in total:
    278 478 479 486 558 584 588 606 731 772 832 971 990
```

The summary lists the flagged respondents as well as the details of the cutoff criterion on which this decision was based. We find this type of information more useful than merely reporting summaries of the scores of the PFSs.

Printing a '`PerFit`' object display the PFS scores on the screen.

The ***`PRFplot`*** *function*

```
PRFplot(matrix, respID, h = .09, N.FPts = 101, VarBands = FALSE,
  VarBands.area = FALSE, alpha = .05, Xlabel = NA, Xcex = 1.5, Ylabel = NA,
  Ycex = 1.5, title = NA, Tcex = 1.5, NA.method = "Pairwise", Save.MatImp =
  FALSE, IP = NULL, IRT.PModel = "2PL", Ability = NULL, Ability.PModel =
  "ML", mu = 0, sigma = 1, message = TRUE)
```

Function `PRFplot` plots nonparametric PRFs with optional variability bands (see argument `VarBands`) for all test takers identified in vector `respID`. It applies to dichotomous data only. The PRF relates item difficulty (i.e., the values $(1 - \pi_i)$) on the x-axis with the associated probability of correct response on the y-axis. The PRF is typically nonincreasing; misfitting item score vectors will display deviations from this expected pattern. Missing values are addressed using one of the single imputation methods previously explained (for which arguments `NA.method` through `sigma` apply).

The current implementation of the PRF is based on nonparametric kernel smoothing (Emons *et al.* 2004). The value of the PRF at each focal point (representing a nonparametric difficulty parameter between 0 and 1) is estimated as a weighted sum score, where scores pertaining to items with difficulty close to the focal point are given the largest weights. The total number of focal points is defined by `N.Fpts` (by default `N.Fpts = 101`). The weights are functions of the Gaussian kernel function. It is necessary to specify a bandwidth value (argument `h`) in order to compute the weights. The h value controls the trade-off between bias and sampling variation (Emons *et al.* 2004). Small h values reduce bias but increase variance, leading to PRFs that capture too much measurement error. Large h values, on the other hand, increase bias which renders PRFs that are often too flat, thus missing potentially relevant misfitting response behavior. Therefore, it is important to carefully specify the h value. Emons *et al.* (2004, pp. 10–13), after a simulation study, advised that "h values between 0.07 and 0.11 are reasonable choices". The function's default is `h = 0.09`.

Variability bands of level alpha (`alpha = 0.05` by default) can also be added to the plot. These bands are computed following the jackknife procedure explained in Emons *et al.* (2004).

The PRFs and variability bands for each test taker are approximated by means of functional data objects (e.g., Ramsay, Hooker, and Graves 2009), with the help of the **fda** package (Ramsay, Wickham, Graves, and Hooker 2014). This procedure follows two steps:

- Compute a B-splines basis system. This basis consists of a set of piecewise polynomials, all of degree three/order four (i.e., cubic polynomial segments), with one knot per break point. Any two consecutive splines, sp_1 and sp_2, with common break point BP, verify $sp_1(BP) = sp_2(BP)$, $sp_1'(BP) = sp_2'(BP)$, and $sp_1''(BP) = sp_2''(BP)$. At 0 and 1 (extremes of the x-range) four knots are used.

- Specify coefficients for the B-splines basis system computed above and then create functional data objects. The procedure is based on smoothing using regression analysis (Ramsay *et al.* 2009, Section 4.3).

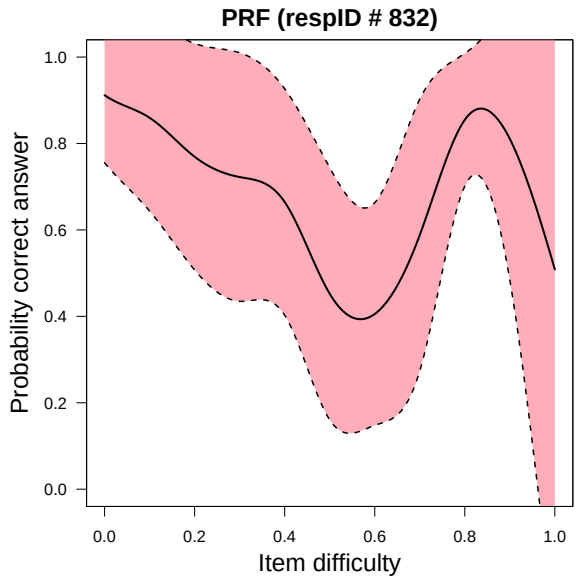

Figure 2: `PRFplot(IntelligenceData, 832, VarBands = TRUE, VarBands.area = TRUE, message = FALSE)`.

Function `PRFplot` outputs a list with three functional data objects of class '`fd`' (see package **fda**). `PRF.FDO` is the functional data object of the PRFs for all test takers; `VarBandsLow.FDO` (resp. `VarBandsHigh.FDO`) is the functional data object of the lower-bound (resp. upper-bound) of the variability bands, for the entire sample.

Figure 1 (page 5) shows the PRFs for test takers 6, 584, 832, and 772 from the intelligence data. Figure 2 reproduces the third plot with variability bands included, and includes the R code in the caption that produces this plot disregarding the axes labels and the title formatting.

The `plot` method for **PerFit**

```
plot(x, cutoff.obj = NULL, ModelFit = "NonParametric", Nreps = 1000, IP =
  x$IP, IRT.PModel = x$IRT.PModel, Ability = x$Ability, Ability.PModel =
  x$Ability.PModel, mu = 0, sigma = 1, Blvl = 0.05, Breps = 1000, CIlvl =
  0.95, UDlvl = NA, Type = "Density", Both.scale = TRUE, Cutoff = TRUE,
  Cutoff.int = TRUE, Flagged.ticks = TRUE, Xlabel = NA, Xcex = 1.5, title =
  NA, Tcex = 1.5, col.area = "lightpink", col.hist = "lightblue", col.int =
  "darkgreen", col.ticks = "red", ...)
```

Finally, the R `plot` method was also extended for objects of class '`PerFit`'. This plot displays the empirical distribution of the scores of the PFS specified by the '`PerFit`' class object `x`. A histogram, density, or a combination of both displays is possible (argument `Type` with options `"Histogram"`, `"Density"` (default), and `"Both"`).

The cutoff score can be added to the plot (`Cutoff = TRUE`). Either the user provides a previously computed '`PerFit.cutoff`' object or the function internally computes it (for which arguments `ModelFit` through `UDlvl` are required). The adequate "flagging region" is colored, thus indicating the range of values for which the PFS flags test takers as potentially displaying

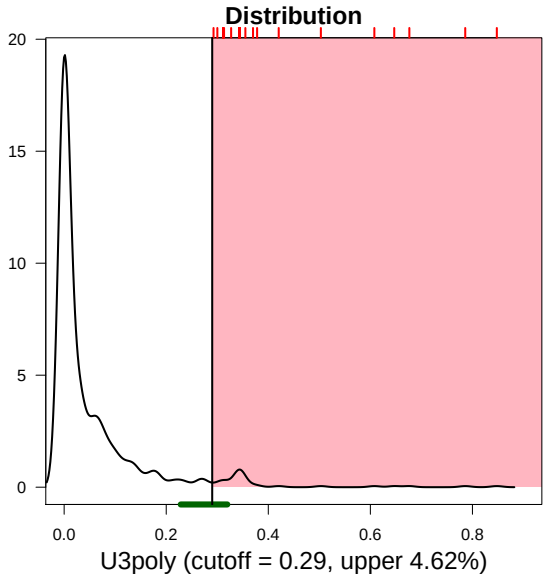

Figure 3: `plot(U3poly(PhysFuncData, Ncat = 3), Blvl = 0.01)`.

aberrant behavior. It is possible to display the `CIvlv`% confidence interval of the cutoff value on the x-axis (`Cutoff.int = TRUE`). The width of this interval may give the user a better idea of the error associated to the estimation of the cutoff. It is also possible to add ticks to the upper-side of the plot marking the positions of the flagged item score vectors. This information may be useful to visually distinguish between flagged test takers which are very close to the cutoff versus those that are further away in the rejection region direction. Finally, the option `both.scale` was included to better tune the scale of the y-axis.

As an example, Figure 3 shows the sample distribution of the $U3^p$ scores from the physical functioning data with a bootstrapped 1% cutoff value. Item score vectors with $U3^p$ values larger than 0.29 are flagged as potentially displaying aberrant answering behavior.

4. Discussion

PerFit is a new R package which offers the most widely used PFSs currently available. Developments in person-fit are ongoing and our goal is to keep extending **PerFit**. Some of the methods that are likely to be included in future versions of the package include a wider variety of model-based PFSs (see Karabatsos 2003 and Meijer and Sijtsma 2001), methods suited to computerized adaptive tests data (the CUSUMs approach; see for example Van Krimpen-Stoop and Meijer 2001), multitest person-fit approaches, and related methods based on the concept of person reliability (e.g., Ferrando 2015).

References

Birenbaum M (1985). "Comparing the Effectiveness of Several IRT Based Appropriateness Measures in Detecting Unusual Response Patterns." *Educational and Psychological Measurement*, **45**(3), 523–534.

Birenbaum M (1986). "Effect of Dissimulation Motivation and Anxiety on Response Pattern Appropriateness Measures." *Applied Psychological Measurement*, **10**(2), 167–174. doi:10.1177/014662168601000208.

Chalmers RP (2012). "**mirt**: A Multidimensional Item Response Theory Package for the R Environment." *Journal of Statistical Software*, **48**(6), 1–29. doi:10.18637/jss.v048.i06.

Cheng X, Cook D, Hofmann H (2016). ***MissingDataGUI**: A GUI for Missing Data Exploration*. R package version 0.2-5, URL https://CRAN.R-project.org/package=MissingDataGUI.

Choi SW (2010). "**PERSONz**: Person Misfit Detection Using the Lz Statistic and Monte Carlo Simulations." *Applied Psychological Measurement*, **34**(6), 457–458. doi:10.1177/0146621609360359.

Conijn JM (2013). *Detecting and Explaining Person Misfit in Non-Cognitive Measurement*. Ph.D. thesis, Tilburg University, Ridderkerk.

Donlon TF, Fischer FE (1968). "An Index of an Individual's Agreement with Group-Defined Item Difficulties." *Educational and Psychological Measurement*, **28**(1), 105–113. doi:10.1177/001316446802800110.

Drasgow F, Levine MV, McLaughlin ME (1987). "Detecting Inappropriate Test Scores with Optimal and Practical Appropriateness Indices." *Applied Psychological Measurement*, **11**(1), 59–79. doi:10.1177/014662168701100105.

Drasgow F, Levine MV, Williams EA (1985). "Appropriateness Measurement with Polychotomous Item Response Models and Standardized Indices." *British Journal of Mathematical and Statistical Psychology*, **38**(1), 67–86. doi:10.1111/j.2044-8317.1985.tb00817.x.

Embretson SE, Reise SP (2000). *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates Publishers, Mahwah.

Emons WHM (2008). "Nonparametric Person-Fit Analysis of Polytomous Item Scores." *Applied Psychological Measurement*, **32**(3), 224–247. doi:10.1177/0146621607302479.

Emons WHM, Meijer RR, Sijtsma K (2002). "Comparing Simulated and Theoretical Sampling Distributions of the U3 Person-Fit Statistic." *Applied Psychological Measurement*, **26**(1), 88–108. doi:10.1177/0146621602026001006.

Emons WHM, Sijtsma K, Meijer RR (2004). "Testing Hypotheses About the Person-Response Function in Person-Fit Analysis." *Multivariate Behavioral Research*, **39**(1), 1–35. doi:10.1207/s15327906mbr3901_1.

Ferrando P (2015). "Assessing Person Fit in Typical-Response Measures." In S Reise, D Revicki (eds.), *Handbook of Item Response Theory Model: Applications to Typical Performance Assessment*, pp. 128–155. Routledge, New York.

Ferrando P, Lorenzo U (2000). "**WPerfit**: A Program for Computing Parametric Person-Fit Statistics and Plotting Person Response Curves." *Educational and Psychological Measurement*, **60**(3), 479–487. doi:10.1177/00131640021970547.

Guttman L (1944). "A Basis for Scaling Qualitative Data." *American Sociological Review*, **9**(2), 139–150. doi:10.2307/2086306.

Guttman L (1950). "The Basis for Scalogram Analysis." In S Stouffer, L Guttman, E Suchman, P Lazarsfeld, S Star, J Claussen (eds.), *Measurement and Precision*, pp. 60–90. Princeton University Press, Princeton NJ.

Harnisch DL, Linn RL (1981). "Analysis of Item Response Patterns: Questionable Test Data and Dissimilar Curriculum Practices." *Journal of Educational Measurement*, **18**(3), 133–146. doi:10.1111/j.1745-3984.1981.tb00848.x.

Harnisch DL, Tatsuoka KK (1983). "A Comparison of Appropriateness Indices Based on Item Response Theory." In R Hambleton (ed.), *Applications of Item Response Theory*. ERIBC, Vancouver.

Hemker BT, Sijtsma K, Molenaar IW, Junker BW (1997). "Stochastic Ordering Using the Latent Trait and the Sum Score in Polytomous IRT Models." *Psychometrika*, **62**(3), 331–347. doi:10.1007/bf02294555.

Honaker J, King G, Blackwell M (2011). "**Amelia II**: A Program for Missing Data." *Journal of Statistical Software*, **45**(7), 1–47. doi:10.18637/jss.v045.i07.

International Test Commission (2014). *ITC Guidelines for Quality Control in Scoring, Test Analysis, and Reporting of Test Scores*. URL http://intestcom.org/.

Kane MT, Brennan RL (1980). "Agreement Coefficients as Indices of Dependability for Domain-Referenced Tests." *Applied Psychological Measurement*, **4**(1), 105–126. doi:10.1177/014662168000400111.

Karabatsos G (2003). "Comparing the Aberrant Response Detection Performance of Thirty-Six Person-Fit Statistics." *Applied Measurement in Education*, **16**(4), 277–298. doi:10.1207/s15324818ame1604_2.

Kiefer T, Robitzsch A, Wu M (2016). ***TAM**: Test Analysis Modules*. R package version 1.995-0, URL https://CRAN.R-project.org/package=TAM.

Kogut J (1987). "Detecting Aberrant Item Response Patterns in the Rasch Model." *Technical report*, University of Twente. Research Report 87-3.

Levine MV, Drasgow F (1983). "Appropriateness Measurement: Validating Studies and Variable Ability Models." In D Weiss (ed.), *New Horizons in Testing: Latent Trait Test Theory and Computerized Adaptive Testing*, pp. 109–131. Academic Press, New York.

Levine MV, Rubin DB (1979). "Measuring the Appropriateness of Multiple-Choice Test Scores." *Journal of Educational Statistics*, **4**(4), 269–290. doi:10.2307/1164595.

Li M, Olejnik S (1997). "The Power of Rasch Person-Fit Statistics in Detecting Unusual Response Patterns." *Applied Psychological Measurement*, **21**(3), 215–231. doi:10.1177/01466216970213002.

Magis D, Raîche G, Béland S (2012). "A Didactic Presentation of Snijders's l_z^* Index of Person Fit with Emphasis on Response Model Selection and Ability Estimation." *Journal of Educational and Behavioral Statistics*, **37**(1), 57–81. doi:10.3102/1076998610396894.

Mair P, Hatzinger R (2007). "Extended Rasch Modeling: The **eRm** Package for the Application of IRT Models in R." *Journal of Statistical Software*, **20**(9), 1–20. doi:10.18637/jss.v020.i09.

Maydeu-Olivares A (2015). "Evaluating the Fit of IRT Models." In S Reise, D Revicki (eds.), *Handbook of Item Response Theory Model: Applications to Typical Performance Assessment*, pp. 111–127. Routledge, New York.

Meijer RR (1994). "The Number of Guttman Errors as a Simple and Powerful Person-Fit Statistic." *Applied Psychological Measurement*, **18**(4), 311–314. doi:10.1177/014662169401800402.

Meijer RR (1996). "Person-Fit Research: An Introduction." *Applied Measurement in Education*, **9**(1), 3–8. doi:10.1207/s15324818ame0901_2.

Meijer RR, Egberink IL, Emons WHM, Sijtsma K (2008). "Detection and Validation of Unscalable Item Score Patterns Using Item Response Theory: An Illustration with Harter's Self-Perception Profile for Children." *Journal of Personality Assessment*, **90**(3), 227–238. doi:10.1080/00223890701884921.

Meijer RR, Muijtjens AM, van der Vleuten CM (1996). "Nonparametric Person-Fit Research: Some Theoretical Issues and an Empirical Example." *Applied Measurement in Education*, **9**(1), 77–89. doi:10.1207/s15324818ame0901_7.

Meijer RR, Niessen ASM, Tendeiro JN (2016). "A Practical Guide to Check the Consistency of Item Response Patterns through Person-Fit Statistics: Examples and a Computer Program." *Assessment*, **23**(1), 52–62. doi:10.1177/1073191115577800.

Meijer RR, Sijtsma K (2001). "Methodology Review: Evaluating Person Fit." *Applied Psychological Measurement*, **25**(2), 107–135. doi:10.1177/01466210122031957.

Meijer RR, Tendeiro JN (2012). "The Use of the l_z and l_z^* Person-Fit Statistics and Problems Derived from Model Misspecification." *Journal of Educational and Behavioral Statistics*, **37**(6), 758–766. doi:10.3102/1076998612466144.

Meijer RR, Tendeiro JN (2014). "The Use of Person-Fit Scores in High Stakes Educational Testing: How to Use Them and What They Tell Us." *Technical Report 14-03*, Law School Admission Council.

Meijer RR, Tendeiro JN, Wanders RBK (2015). "The Use of Nonparametric IRT to Explore Data Quality." In S Reise, D Revicki (eds.), *Handbook of Item Response Theory Model: Applications to Typical Performance Assessment*, pp. 85–110. Routledge, New York.

Mokken RJ (1971). *A Theory and Procedure of Scale Analysis*. De Gruyter, Berlin, Germany.

Molenaar IW (1982). "Mokken Scaling Revisited." *Kwantitatieve Methoden*, **3**, 145–164.

Molenaar IW (1991). "A Weighted Loevinger H-Coefficient Extending Mokken Scaling to Multicategory Items." *Kwantitatieve Methoden*, **12**, 97–117.

Molenaar IW, Hoijtink H (1990). "The Many Null Distributions of Person Fit Indices." *Psychometrika*, **55**(1), 75–106. doi:10.1007/bf02294745.

Nering ML (1995). "The Distribution of Person Fit Using True and Estimated Person Parameters." *Applied Psychological Measurement*, **19**(2), 121–129. doi:10.1177/014662169501900201.

Nering ML (1997). "The Distribution of Indexes of Person-Fit Within the Computerized Adaptive Testing Environment." *Applied Psychological Measurement*, **21**(2), 115–127. doi:10.1177/01466216970212002.

Nering ML, Meijer RR (1998). "A Comparison of the Person Response Function to the lz Person-Fit Statistic." *Applied Psychological Measurement*, **22**(1), 53–69. doi:10.1177/01466216980221004.

Noonan BW, Boss MW, Gessaroli ME (1992). "The Effect of Test Length and IRT Model on the Distribution and Stability of Three Appropriateness Indexes." *Applied Psychological Measurement*, **16**(4), 345–352. doi:10.1177/014662169201600405.

Olson J, Fremer J (2013). *TILSA Test Security Guidebook: Preventing, Detecting, and Investigating Test Security Irregularities*. Council of Chief State School Officers, Washington, DC.

Partchev I (2016). ***irtoys**: A Collection of Functions Related to Item Response Theory (IRT)*. R package version 0.2.0, URL https://CRAN.R-project.org/package=irtoys.

Raîche G (2014). ***irtProb**: Utilities and Probability Distributions Related to Multidimensional Person Item Response Models*. R package version 1.2, URL https://CRAN.R-project.org/package=irtProb.

Ramsay JO, Hooker G, Graves S (2009). *Functional Data Analysis with R and MATLAB*. Springer-Verlag, New York. doi:10.1007/978-0-387-98185-7.

Ramsay JO, Wickham H, Graves S, Hooker G (2014). ***fda**: Functional Data Analysis*. R package version 2.4.4, URL https://CRAN.R-project.org/package=fda.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reif M (2014). ***mcIRT**: IRT Models for Multiple Choice Items*. R package version 0.41, URL https://CRAN.R-project.org/package=mcIRT.

Reise SP (1995). "Scoring Method and the Detection of Person Misfit in a Personality Assessment Context." *Applied Psychological Measurement*, **19**(3), 213–229. doi:10.1177/014662169501900301.

Rizopoulos D (2006). "**ltm**: An R Package for Latent Variable Modelling and Item Response Theory Analyses." *Journal of Statistical Software*, **17**(5), 1–25. doi:10.18637/jss.v017.i05.

Robitzsch A (2016). ***sirt**: Supplementary Item Response Theory Models*. R package version 1.12-2, URL https://CRAN.R-project.org/package=sirt.

Rogers H, Hattie JA (1987). "A Monte Carlo Investigation of Several Person and Item Fit Statistics for Item Response Models." *Applied Psychological Measurement*, **11**(1), 47–57. doi:10.1177/014662168701100103.

Rudner LM (1983). "Individual Assessment Accuracy." *Journal of Educational Measurement*, **20**(3), 207–219. doi:10.1111/j.1745-3984.1983.tb00200.x.

Rupp AA (2013). "A Systematic Review of the Methodology for Person Fit Research in Item Response Theory: Lessons about Generalizability of Inferences from the Design of Simulation Studies." *Psychological Test and Assessment Modeling*, **55**(1), 3–38.

Sato T (1975). *The Construction and Interpretation of S-P Tables*. Meiji Tosho, Tokyo.

Sijtsma K (1986). "A Coefficient of Deviance of Response Patterns." *Kwantitatieve Methoden*, **7**(22), 131–145.

Sijtsma K, Meijer RR (1992). "A Method for Investigating the Intersection of Item Response Functions in Mokken's Nonparametric IRT Model." *Applied Psychological Measurement*, **16**(2), 149–157. doi:10.1177/014662169201600204.

Sijtsma K, Meijer RR (2001). "The Person Response Function as a Tool in Person-Fit Research." *Psychometrika*, **66**(2), 191–207. doi:10.1007/bf02294835.

Sijtsma K, Molenaar IW (2002). *Introduction to Nonparametric Item Response Theory*. Sage Publications, Thousand Oaks.

Snijders TAB (2001). "Asymptotic Null Distribution of Person Fit Statistics with Estimated Person Parameter." *Psychometrika*, **66**(3), 331–342. doi:10.1007/bf02294437.

Su YS, Gelman A, Hill J, Yajima M (2011). "Multiple Imputation with Diagnostics (**mi**) in R: Opening Windows into the Black Box." *Journal of Statistical Software*, **45**(2), 1–31. doi:10.18637/jss.v045.i02.

Tatsuoka KK, Tatsuoka MM (1982). "Detection of Aberrant Response Patterns and Their Effect on Dimensionality." *Journal of Educational Statistics*, **7**(3), 215–231. doi:10.3102/10769986007003215.

Tatsuoka KK, Tatsuoka MM (1983). "Spotting Erroneous Rules of Operation by the Individual Consistency Index." *Journal of Educational Measurement*, **20**(3), 221–230. doi:10.1111/j.1745-3984.1983.tb00201.x.

Tendeiro JN (2016). **PerFit**: *Person Fit*. R package version 1.4.1, URL https://CRAN.R-project.org/package=PerFit.

Tendeiro JN, Meijer RR (2014). "Detection of Invalid Test Scores: The Usefulness of Simple Nonparametric Statistics." *Journal of Educational Measurement*, **51**(3), 239–259. doi:10.1111/jedm.12046.

Van der Ark LA (2005). "Stochastic Ordering Of the Latent Trait by the Sum Score under Various Polytomous IRT Models." *Psychometrika*, **70**(2), 283–304. doi:10.1007/s11336-000-0862-3.

Van der Ark LA (2007). "Mokken Scale Analysis in R." *Journal of Statistical Software*, **20**(11), 1–19. doi:10.18637/jss.v020.i11.

Van der Ark LA (2012). "New Developments in Mokken Scale Analysis in R." *Journal of Statistical Software*, **48**(5), 1–27. doi:10.18637/jss.v048.i05.

Van der Flier H (1977). "Environmental Factors and Deviant Response Patterns." In Y Poortinga (ed.), *Basic Problems in Cross-Cultural Psychology*. Swets & Zeitlinger Publishers, Lisse.

Van der Flier H (1980). *Vergelijkbaarheid van Individuele Testprestaties [Comparability of Individual Test Performance]*. Ph.D. thesis, University of Groningen.

Van der Flier H (1982). "Deviant Response Patterns and Comparability of Test Scores." *Journal of Cross-Cultural Psychology*, **13**(3), 267–298. doi:10.1177/0022002182013003001.

Van Krimpen-Stoop EMLA, Meijer RR (2001). "CUSUM-Based Person-Fit Statistics for Adaptive Testing." *Journal of Educational and Behavioral Statistics*, **26**(2), 199–218. doi:10.3102/10769986026002199.

Wright BD, Masters GN (1990). "Computation of OUTFIT and INFIT Statistics." *Rasch Measurement Transactions*, **3**(4), 84–85.

Zhang B, Walker CM (2008). "Impact of Missing Data on Person-Model Fit and Person Trait Estimation." *Applied Psychological Measurement*, **32**(6), 466–479. doi:10.1177/0146621607307692.

Affiliation:

Jorge N. Tendeiro
Department of Psychometrics and Statistics
Faculty of Behavioral and Social Sciences
University of Groningen
Grote Kruisstraat 2/1,
9712 TS, Groningen, The Netherlands
Telephone: +31/50/363-6953
Fax: +31/50/363-6304
E-mail: j.n.tendeiro@rug.nl

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
October 2016, Volume 74, Issue 5 *Submitted:* 2014-08-29
doi:10.18637/jss.v074.i05 *Accepted:* 2015-09-04