Psychometrics With R: A Review Of CRAN Packages For Item Response Theory

Thomas Rusch, Patrick Mair, Reinhold Hatzinger

Discussion Paper Series
Paper 2013/2, November 2013

Center for Empirical Research Methods
http://wu.ac.at/methods

WIRTSCHAFTS UNIVERSITÄT WIEN VIENNA UNIVERSITY OF ECONOMICS AND BUSINESS

Discussion Paper Series of the
Center for Empirical Research Methods
WU Vienna
Welthandelsplatz 1, D4
1020 Vienna
Austria

Psychometrics with R: A review of CRAN packages for Item Response Theory

Thomas Rusch
WU (Wirtschafts-universität Wien)

Patrick Mair
Harvard University

Reinhold Hatzinger
WU (Wirtschafts-universität Wien)

Abstract

In this paper we review the current state of R packages for Item Response Theory (IRT). We group the available packages based on their purpose and provide an overview of each package's main functionality. Each of the packages we describe has a peer-reviewed publication associated with it. We also provide a tutorial analysis of data from the 1990 Workplace Industrial Relation Survey to show how the breadth and flexibility of IRT packages in R can be leveraged to conduct even challenging item analyses with versatility and ease. These items relate to the type of consultations that are carried out in a firm when major changes are implemented. We first use unidimensional IRT models just to discover that they fit do not fit well. We then use nonparametric IRT to explore the possible causes for the scaling problem. Based on the results from the exploration, we finally use a two-dimensional model on a subset of the original items to achieve a good fit with a sensible interpretation, namely that there are two types of consultations a firm may engage in: consultations with workers/representatives from the firm and with official union representatives. The different items relate mostly to one of these dimensions and firms can be scaled well along these two dimensions.

Keywords: Item Response Theory, review, free software, R, CRAN, Workplace Industrial Relation Survey, Rasch Model, multidimensional IRT, Mokken scale analysis, kernel smoothing.

1. Introduction

Since its publication by Ihaka and Gentleman (1996), R (R Development Core Team 2012) has become a successful language for statistical computing and graphics. Today, it is often regarded as the *lingua franca* of computational statistics. It is both a statistical software package and a programming language well-suited for statistical problems and much of its popularity is probably due to it being free, libre, open source software. This means it can be used, distributed, copied and shared without restrictions. It enables and encourages learning, collaboration, adaptation and inspection of the code base and allows reproducible statistical analyses. Another part of R's success is its extensibility and the effort of many leading scientists to provide good and reliable software. R is increasingly used in many academic and applied settings, from researchers in various subjects for answering substantive questions to statisticians and practitioners for providing state-of-the-art implementations of newly developed techniques. This holds true for psychometrics and specifically item response theory

(IRT) as well.

In recent years, an ever growing number of R packages has been developed to conduct psychometric analyses by various authors. See for example the "Psychometrics Task View" (Mair and Hatzinger 2007b) for a description of which packages there are and what they can be used for[1]. In this chapter we will provide a glance at the IRT functionality in R. Due to the dynamic and open development model, the ecosystem of IRT packages on the Comprehensive R Archive Network (CRAN) can be seen as one of the most comprehensive software collections for IRT currently available[2]. The packages allow to conduct among others Rasch, classic IRT, multidimensional or nonparametric IRT analyses (e.g., with packages **eRm** (Mair, Hatzinger, and Maier 2013), **ltm** (Rizopoulos 2013), **lme4** (Bates, Maechler, and Bolker 2011), **plRasch** (Li and Hong 2007), **mirt** (Chalmers 2013), **mokken** (Van Der Ark 2013), **KernSmoothIRT** (Mazza, Punzo, and McGuire 2013), **MCMCpack** (Martin, Quinn, and Park 2012), **pscl** (Jackman 2011), **DPpackage** (Jara, Hanson, Quintana, Mueller, and Rosner 2012)), allows to fit mixtures of IRT models (e.g., with packages **mRm** (Preinerstorfer 2012), **psychomix** (Frick, Strobl, Leisch, and Zeileis 2012b), **mixRasch** (Willse 2009), **psychotree** (Zeileis, Strobl, Wickelmaier, and Kopf 2011)), investigate differential item functioning (e.g., with packages **difR** (Magis, Beland, and Raiche 2012), **psychotree**, **lordif** (Choi, Gibbons, and Crane 2012)), calibrate items or equate tests based on IRT (e.g., **ltm**, **plink** (Weeks 2011), **EstCRM** (Zopluoglu 2012)), **kequate** (Andersson, Branberg, and Wiberg 2013)), conduct computerized adaptive testing (**catR** (Magis and Gilles 2012) or concerto (The Psychometrics Center of the University of Cambridge 2012)) and various other possibilities.

Most of the functionality in IRT packages for R has been developed by psychometricians and hence aims at providing user-friendly software for practitioners. The typical user may therefore be any person working in an academic or applied area where state-of-the-art IRT analysis are employed. This includes academics and practitioners who develop scales or psychological tests as well as people looking for answers to substantive questions using IRT.

2. Functionality of IRT packages in R

The IRT functionality offered in R can be roughly divided into the following categories: uni- and multidimensional Rasch modeling, uni-and multidimensional IRT modeling, nonparametric IRT modeling, differential item functioning (DIF) and modeling of additional heterogeneity (i.e., mixture models) as well as item calibration, test equating and computerized adaptive testing. In what follows we list packages associated with the different categories for which peer-reviewed articles exist and describe the main functions and what they are meant for (as of May 2013). Thus we want this review to serve as an extended version of the short description in the "CRAN task views" and as an encyclopediac first glance on the broadness of IRT in R. For the underlying theory we also include references to the according chapters of the second edition of the Handbook of Item Response Theory (van der Linden & Hambleton,

[1] A critical discussion of the "Psychometrics Task View" can be found in Ünlü and Yanagida (2011).

[2] This variety is only possible by the continued valuable effort of many people. They should also be properly attributed for their work. Unfortunately, due to the development dynamics and the diversity of contributed code, a snapshot like this review will always be incomplete and must choose what to focus on. Hence, we decided to only discuss packages in more detail that have an accompanying peer-reviewed publication (as of August 2013) associated with it. This also helps to at least add an extra layer of reliability to the software presented.

2014).

2.1. Rasch modeling

Frequentist Approaches A comprehensive package for Rasch modeling based on conditional maximum likelihood estimation is the **eRm** package (Mair *et al.* 2013). It fits the ordinary Rasch model for dichotomous data (Verhelst, Vol. 1, Chap. 3) with function `RM`, the linear logistic test model (`LLTM`; see Janssen, Vol. 1, Chap. 3), the rating scale model (`RSM`; Andrich, Vol. 1, Chap. 5) and its linear extension (`LRSM`), the partial credit model (`PCM`; Masters, Vol. 1, Chap. 7) and its linear extension (`LPCM`), see also Mair and Hatzinger (2007a). Furthermore it allows to measure change (Fischer, Vol. 3, Chap. 20) by conveniently estimating LLRA and the polytomous equivalents (`LLRA`; Hatzinger and Rusch (2009); Rusch, Maier, and Hatzinger (2013)). It offers various plotting possibilities, e.g., for item characteristic curves (`plotICC`), goodness-of-fit (`plotGOF`), differential item functioning (`plotDIF`) or person/item maps (`plotPImap`, `plotPWmap`), possibilities to test goodness of fit such as Andersen's LR Test (`LRtest`), quasi-exact nonparametric tests (`NPtest`) or tests adapted from logistic regression (`gofIRT`), compute item or scale information (`item_info`, `test_info`) and extract person parameters (`person.parameter`) or person and item fit statistics (`person.fit`, `item.fit`). It also features a stepwise item selection procedure (`stepwiseIt`) and functions to simulate Rasch data or various violations.

Multilevel and generalized Rasch models for dichotomous items (DeBoeck & Wilson, Vol. 1, Chap. 34; Fox & Glas, Vol. 1, Chap. 24) can be estimated as generalized linear mixed models for the binomial family using the package **lme4**. Here the `glmer` (or `lmer`) function estimates mixed-effect models with crossed or partially crossed random effects by ML with adaptive Gauss-Hermite Quadrature (defaulting to Laplace Approximation). How this can be used for IRT is described in detail in Doran, Bates, Bliese, and Dowling (2007); De Boeck, Bakker, Zwitser, Nivard, Hofman, Tuerlinckx, and Partchev (2010). This way it is possible to fit item covariate models such as the 1PL, the LLTM, a multidimensional 1PL, person covariate models such as the JML version of the 1PL or the latent regression 1PL and models with person-by-item covariates such as DIF models, local dependency models and the dynamic 1PL. There are many methods such as `summary` to be used with the fitted objects.

The package **plRasch** allows to fit models of the Rasch family in a log-linear-by-linear association model (LLLA; see Anderson, Li, and Vermunt (2007)). Its main focus is to provide estimation for multidimensional Rasch models. To achieve this there are the functions `llla` or `RaschPLE` to compute maximum likelihood estimates or pseudo-likelihood estimates of parameters of the LLLA which allows Rasch models for polytomous (or dichotomous) items and multiple (or single) latent traits (the traits can even be correlated). Additionally, robust standard errors for the pseudo-likelihood estimates can be computed.

Bayesian Approaches Dedicated functions for Bayesian Rasch models are not available. However, a number of general purpose packages allow to conduct generic Bayesian inference with Gibbs sampling (Sinharay & Johnson, Vol. 2, Chap. 13; Junker, Patz & Vanhoudnos, Vol. 2, Chap. 15), most prominently interfacing JAGS, e.g., **rjags** (Plummer 2011). These packages can also be used for IRT models (including the Rasch model). To summarize, diagnose and process MCMC output, the package **coda** (Plummer, Best, Cowles, and Vines

2006) can be used.

2.2. Parametric IRT modeling

Frequentist Approaches For IRT models with one or more item parameters, the package **ltm** offers ample functionality. Using marginal maximum likelihood estimation, it allows fitting of the 1PL model (`rasch`) and, additionally, functions for estimating Birnbaum's 2PL (`ltm`) and 3PL models (`tpm`) to dichotomous items (Hambleton & van der Linden, Vol. 1, Chap. 2). For these items the `ltm` function additionally allows to fit 2PL latent trait models up to two trait dimensions, i.e., a linear multidimensional logistic model (Reckase, Vol. 1, Chap. 12). For polytomous items, the user has the choice of the graded response model (`grm`; Samejima, Vol. 1, Chap. 6) as well as the generalized partial credit model (`gpcm`; Muraki, Vol. 1, Chap. 9). All the functions allow to introduce restrictions on the parameters, and hence allow to fit difficulty-plus-guessing models as well. Goodness of fit can be assessed by bootstrap χ^2 tests (`GoF`), fit to two- or three-way margins (`margins`), likelihood ratio tests (`GoF`), test of unidimensionality (`unidimtest`) and information criteria like AIC and BIC (`AIC`) as well as item and person fit indices (`item.fit`, `person.fit`). Person parameter estimates can be obtained via `factor.scores`. Item information can be extracted with `information`. The package also features a number of standard methods for generics such as `plot` for the graphical display of the results like item characteristic curves, item information, item person maps, standard error of measurement, `anova` for model comparison, `coef` for parameter extraction, `residuals` for residuals and `vcov` for the variance-covariance matrix. See Rizopoulos (2006) for more information. The package also contains a data simulation module, `rmvlogis`.

Models for nominal polytomous responses (Thissen, Vol. 1, Chap. 4) can be fitted with functions from the **plink** package. It allows fitting of the nominal response model (`nrm`) and the multiple-choice model (`mcm`). Moreover, it contains the functions `drm` to fit 1PL, 2PL and 3PL models, `grm` for the graded response model and `gpcm` for partial credit or generalized partial credit model. See Weeks (2010). The package's main focus however is on item calibration, which is why it gets treated more thoroughly in Section 2.5.

Multidimensional IRT models (Reckase, Vol. 1, Chap. 12) can be estimated via full maximum likelihood utilizing a fixed quadrature EM method (Aitkin, Vol. 2, Chap. 12) or a Metropolis-Hastings Robbins-Monro method with the **mirt** package (see Chalmers (2012) for details). Specifically, it allows to fit uni- and multivariate Rasch or 1-4PL models as well as confirmatory and exploratory item response models for polytomous nominal items, partial credit items and rating scales (all with `mirt`), item bifactor analysis (`bfactor`), and allows partially-compensatory item response modeling in conjunction with other IRT models. Factor scores and person parameters can be extracted with `fscores`. Various visualisations of the fitted models can be obtained with `itemplot`. Additionally, there are methods for many generics like `coef`, `anova`, `fitted`, `residuals`, `plot` and `summary` for the object classes returned from the model fitting functions. A simulation module is also included (`simdata`). It is also worth noting that the package strives for easy integration with the functions in **plink**.

Bayesian Approaches Apart from the general purpose packages mentioned in Section 2.1, there are dedicated functions in different packages that allow to estimate Bayesian IRT models more conveniently, mostly by setting up the Bayesian model automatically. The package

MCMCpack (which is built around Scythe (Pemstein, Quinn, and Martin 2011)) provides a number of IRT models (see Martin, Quinn, and Park (2011)) and uses pre-initialized Gibbs-Sampling to estimate item and person parameters. The user can fit unidimensional IRT models with item difficulty and discrimination parameters and normal priors for all parameters (`MCMCirt1d`) as well as a hierarchical version (`MCMCirtHier1d`) and a dynamic version (`MCMCdynamicIRT1d`) of the unidimensional IRT model. Additionally, there are utilities for fitting k-dimensional versions of a model with difficulty and discrimination parameter, i.e., for k latent traits with `MCMCirtKd` and for fitting the same model for heteroscedastic subject errors with `MCMCirtKdHet`. The homoscedastic model is also available in a robust version (`MCMCirtKdRob`). Note that the functions are meant for ideal-point analysis and less for item scaling, hence the argument `store.items` should be used with these functions if the latter is of interest. Objects returned from functions in **MCMCpack** can be used automatically with **coda**.

2.3. Non- and Semiparametric IRT modeling

Frequentist Approaches Nonparametric IRT analysis along the lines of Mokken scale analysis (Sijtsma & Molenaar, Vol 1., Chap. 18) can be done with the **mokken** package. It includes automated item selection algorithms (`aisp`) and various checks of model assumptions for two item response models: the montone homogeneity model and the double monotonicity model. Among those are scalability coeffficients (`coefH`), the according test statistics (`coefZ`) and model property checks (`check.reliability`, `check.monotonicity`, `check.iio`, `check.errors`, `check.pmatrix`, `check.restscore`). For many classes of objects there are also methods for generics like `plot`, e.g., `plot(check.iio(data))` or `summary`. A detailed description of the package and its functionality is provided in Van der Ark (2007, 2012).

The **KernSmoothIRT** package fits nonparametric item and option characteristic curves using kernel smoothing (see Ramsay, Vol. 1, Chap. 20) for dichotomous or polytomous multiple choice, rating scale, partial credit and nominal items. The main function is `ksIRT`. It allows for using a default, user supplied smoothing bandwidth or optimal selection of the smoothing bandwidth using cross-validation. The `plot` method for class `ksIRT` features a variety of exploratory plots for item characteristic curves, option characteristic curves, density plots, expected value plots, principal components, item and test information. More details can be found in Mazza, Punzo, and McGuire (in press).

Bayesian Approaches The **DPpackage** package allows to various non-and semiparametric models, see Jara, Hanson, Quintana, Müller, and Rosner (2011). Among them are semiparametric IRT models. It offers functions to fit a semiparametric 1PL model or Rasch-Poisson models (Jansen, Vol. 1, Chap. 15) with different prior specifications for the random effects. For Dirichlet process priors these are the functions `DPrasch` and `DPraschpoisson`, for Dirichlet process mixtures these are `DPMrasch` and `DPMraschpoisson`. For linear dependent Dirichlet process priors or a mixture of those, the functions are `LDDPrasch` and `LDDPraschpoisson`. Furthermore there are functions for finite Polya tree priors or mixtures of finite Polya tree priors with `FPTrasch` and `FPTraschpoisson`.

2.4. Differential Item Functioning and Additional Heterogeneity

Differential Item Functioning (DIF) The **difR** package contains several methods to detect DIF (Gamerman, Goncalves & Soares, Vol. 3, Chap. 4) in dichotomously scored items, some of those based on IRT, as explained in Magis, Beland, Tuerlincx, and De Boeck (2010). The functions `difLord` (via Lord's χ^2 method), `difLRT` (based on Likelihood Ratio Tests) and `difRaju` (using Raju's area method) aim to detect both uniform and nonuniform DIF effects in models up to the 3PL model for two groups. The `difGenLord` method (a generalization of Lord's χ^2) can deal with uniform and nonuniform DIF with more than two groups. The convenience functions `selectDif` and `selectGenDif` can be used for employing all DIF methods in the package. The functions from **ltm** are underlying the DIF calculations.

The package **lordif** provides a logistic regression and IRT framework for detecting various types of differential item functioning in dichotomous and polytomous items. The core function is called `lordiff` and for objects returned from it, there are methods for generics like `summary`, `plot` and `print`. It also features an optional Monte Carlo simulation procedure (`montecarlo`). A detailed explanation of the package functionality and the underlying idea of combining logistic regression with IRT for detecting DIF can be found in Choi, Gibbons, and Crane (2011). IRT based DIF is assessed by using the graded response model implementation `grm` in **ltm**.

Detection of DIF by means of recursive partitioning based on psychometric models is implemented in **psychotree**. Currently, DIF for the Rasch model (`raschtree`) can be checked Strobl, Kopf, and Zeileis (in press). The trees can be visualised with `plot` and the parameter stability tests (which are used to assess DIF) can be extracted via `sctest`. There also are methods for `summary`, `print` and various other generics.

Nonparametric DIF analyses can be conducted with **KernSmoothIRT**. Here, versions of smoothed item characteristic curves, option characteristic curves, density plots, expected value plots can be obtained for different groups specified by the user.

Mixture Models Traditional unrestricted mixtures of Rasch models (binary mixed Rasch models; von Davier & Rost, Vol. 1, Chap. 23) are implemented in the **mRm** package and used in Preinerstorfer and Formann (2012). It employs conditional maximum likelihood estimation and model selection based on AIC and BIC in binary mixed Rasch models with the function `mrm`. Standard methods for `print` and `plot` are implemented and it also features a simulation module.

Functionality to fit binary mixed Rasch models is also offered by the **psychomix** package. Currently, the core function `raschmix` allows to fit mixed Rasch models based on conditional maximum likelihood estimation via the EM algorithm. Different parametrizations for the score distribution can be used. As a distinguishing feature, the function also allows estimation with concomitant variables. The mixture model can be plotted with `plot` and summarized with `summary` and effects of concomitant variables can be visualised via `effectsplot`. There also is a simulation module included (`simRaschmix`). More details can be found in Frick, Strobl, Leisch, and Zeileis (2012a).

For polytomous items (as well as dichotomous items) the **mixRasch** (Willse 2009) package can be used. The function `mixRasch` estimates various mixture Rasch models, including

the dichotomous Rasch model, the rating scale model, and the partial credit model by joint maximum likelihood estimation.

2.5. Item Calibration, Test Equating and Adaptive Testing

Item Calibration Item and ability parameter calibration (see Berger, Vol. 3, Chap. 1; van der Linden & Barrett, Vol. 3, Chap. 2) can be done with the package **plink**. The package features functions to compute linking constants and conduct chain linking based on item response theory of unidimensional or multidimensional psychological tests for multiple groups that have a common item design. Preprocessing is done with the functions (`irt.pars`, `sep.pars`) to create objects to be used for calibration. For separate calibration, the `plink` method is used. In the unidimensional case it provides Mean/Mean, Mean/Sigma, Haebara, and Stocking-Lord methods for dichotomous and/or polytomous items. For multidimensional constructs there are functions for a least squares method (Reckase-Martineau method) and extensions of the Haebara and the Stocking-Lord method. For this, single or multiple dilation parameters for multidimensional extensions of all the unidimensional dichotomous and polytomous item response models are used. Results can be conveniently displayed with `summary`. The package includes `read` methods for importing item and/or ability parameters from other IRT software (e.g. BILOG, MULTILOG, BMIRT) or from **eRm** and **ltm**. The `plot` methods enable plotting of item response curves or surfaces, vector plots, and comparison plots for examining parameter drift. See Weeks (2010) for a detailed description and illustration of the package.

Test Equating Two packages already described offer test equating functionality: **plink** and **ltm**. The former contains `equate` methods which conduct IRT true score and observed score equating for unidimensional single- or mixed-format item parameters for two or more groups. **ltm** contains a function `testEquatingData` which conducts test equating by common items.

Furthermore, there is a dedicated package **kequate** (Andersson, Bränberg, and Wiberg in press) that contains many test equating functionalities based on the kernel method (von Davier, Holland, and Thayer 2004). The main function is `kequate` which takes different equating designs as arguments. The package includes methods for standard generics (`summary`, `plot`), as well as a number of extractor functions to conveniently access results of interest.

Adaptive Testing The **catR** package allows for computerized adaptive testing using IRT methods (van der Linden, Vol. 3, Chap. 11). Calculations are based on a 4PL model. The function `createItemBank` creates an item bank from a matrix of item parameters. With the `randomCAT` function, the user can generate response patterns within a computerized adaptive testing (CAT) framework. There is a choice of several starting rules (see also `startItems`), next item selection routines (see also `nextItem`), stopping rules and ability estimators (see also `thetaEst`, `eapEst`). With `nextItem` the item exposure and content balancing can be controlled. There are some functions that may also be of interest beyond a CAT framework, mostly for extracting different types of item information. More details are provided in Magis and Raîche (2012).

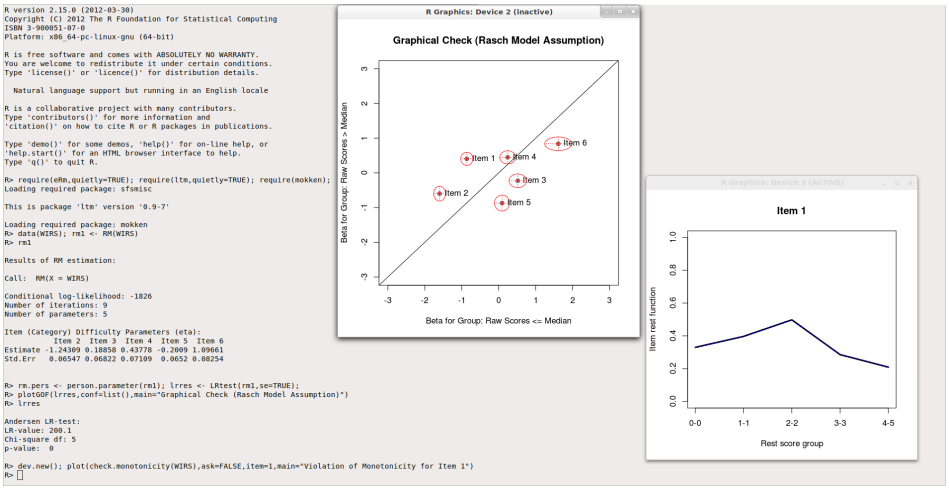

Figure 1: A screenshot of an example R session fitting and assessing a Rasch model for the WIRS data. Note the command line interface. The two plots show a graphical model check from **eRm** of the Rasch model for the WIRS data, with 95%-confidence for the parameter estimates (left plot) and the estimated item step function of Item 1 as provided by **mokken** to check the monotonicity assumption (right plot; note that Item 1 has a non-montonic item step function).

3. Using IRT packages in R

The user interface for R is command line based[3]. Hence using IRT packages in R allows to interact directly with R via input of text strings on the R prompt (abbreviated with `R>` in the following). A screenshot of an R session can be found in Figure 3.

For the user, these text strings typically are function calls which generally look like this: `foo(arguments)` to call function `foo` with arguments `arguments`. The functions are provided by the authors of a package. There is no authorative way how the arguments of the function call must look like, hence the decision which arguments are passed to a function and how it is done lies solely with the function's author. For details on the function and its usage, one can use `?foo` or `help.search('foo')` to access the documentation. Typically the function will be used in one of the following ways:

`foo(data=dataset, option1=o1, option2=o2, ...)` Here the data are supplied to the function as an R object (e.g., data frame or matrix) that the function knows how to use. The argument is called `data` (sometimes also called `X`) and the object passed to the function is called `dataset`. Additional options can be supplied as `option1`, `option2`, etc. Please note that here `data, option1, option2` identify which argument is supplied and `dataset, o1, o2` are the values of the arguments. For example, **eRm** uses this type of function call. The desired model is fitted by calling the corresponding function on a data matrix that is already in the format as expected by the function. To fit a linear partial credit model with given weight matrix `W1` to a data set we called `dat1`, this would be `LPCM(X=dat1, W=W1, sumO=TRUE)`. Note that the first argument `X` identifies

[3]There are various efforts to provide graphical user interfaces, e.g., **Rcmdr** (Fox 2005). We are currently working on a plug-in to **Rcmdr** to make use of IRT functionality.

what will be the data set to be used, the second argument `W` specifies the weight matrix of the linear decomposition and the option `sum0` tells the function to fix the scale such that the sum of item parameters is zero. Of course this way, the data `dat1` must have the correct structure for a LPCM (see `?LPCM`). This type of function call is the most common in IRT packages for R.

`foo(formula, data=dataset, option1=o1, ...)` Sometimes the function call expects a formula object to be passed and (optionally) to specify where the variables in the formula can be found (the `data` argument). To illustrate, a formula object might look like `y~x1+x2`, which means that the variable `y` is explained by an additive model of `x1` and `x2` (see Venables and Smith (2002) for an explanation on formula objects). The variables `y, x1, x2` can be found in the object supplied for `data`. While this is usually the most popular and recommended way for using R functions, IRT packages rarely use it. One exception is **ltm** which allows to fit a custom latent trait model with a formula interface. For instance, for a model that includes an interaction term of two latent traits `z1, z2` to explain the data matrix `dat1` this would be `ltm(dat1~z1 * z2)`.

`foo(object, option1, ...)` Functions that are not meant for model fitting but rather carry out additional calculation, plotting, diagnostics, etc. on a fitted model are usually called in this way. Here `object` is the object returned from fitting the model with one of the functions from above. As an example, assume we stored the output of an `LPCM` call in an object `model1` by writing `model1<-LPCM(X=dat1,W=W1,sum0=TRUE)`. We can then obtain information on the model fit with `summary(model1)`.

Usually function calls return objects which can and should be stored in a variable. This is accomplished with the assignment operator `<-` ("gets"), e.g., `model1<-PCM(X)`. This way, the output of a function is accessible throughout the whole session and can be used for further analyses (e.g., `person.parameter(model1)` or `plotPImap(model1)`) or be saved.

4. A Tutorial Analysis

The IRT functionality in R is not uniform in usage due to it being the effort of different independent contributors as well as due to its command line interface. It is therefore more difficult to describe the various interfaces compared to software with a GUI. We decided that it is possibly best to provide an exemplary, fully reproducible session for a subset of available packages.[4]

To illustrate, we will provide an example analysis of the WIRS data from package **ltm**. The data (Bartholomew 1998) are six dichotomous items taken from a section of the 1990 Workplace Industrial Relation Survey (WIRS) dealing with management/worker consultation in firms as reported by a senior official. A response of "1" means the type of consultation has been held. The items were (more details can be found by typing `help(WIRS)`):

Item 1: Informal discussion with individual workers

[4]However, this is by no means comprehensive. We do believe that it is probably best to learn using R by providing a tutorial which can be used as a starting point for interested readers, who then can take it from there.

Item 2: Meeting with groups of workers

Item 3: Discussions in established joint consultative committee

Item 4: Discussions in specially constituted committee to consider the change

Item 5: Discussions with the union representatives at the establishment

Item 6: Discussions with paid union officials from outside

A feature of these items is that some of them are difficult to scale with classic IRT. We will first start and attempt to Rasch model the items (using package **eRm**). We then use higher parametrized IRT models (2- and 3PL from **ltm**) and continue with nonparametric IRT (**mokken**, **KernSmoothIRT**). Eventually, we will go back to parametric IRT but this time using multidimensional models (**mirt**).

First, we must make the data accessible. For this example, the data are already available if the **ltm** package is installed and can be accessed with the command `data` after the package has been loaded with `library` (we also load the other packages).

```
R> library(ltm)
R> library(eRm)
R> library(mokken)
R> library(KernSmoothIRT)
R> library(mirt)

R> data(WIRS)
```

Often data have to be made available in R from an outside source. This can be a CSV or text file which can be read in with `read.table` and stored in an object that exists during the R session, e.g. `data<-read.table(file='pathToFile')`. The package **foreign** allows to read in system files from other statistical software such as SPSS, SAS or Stata (see R Core Team (2013)).

We begin with a Rasch analysis of the items by utilizing the `RM` function from **eRm**

```
R> rm1 <- RM(WIRS)
R> rm1

Results of RM estimation:

Call:  RM(X = WIRS)

Conditional log-likelihood: -1826.263
Number of iterations: 9
Number of parameters: 5

Item (Category) Difficulty Parameters (eta):
              Item 2     Item 3     Item 4      Item 5     Item 6
Estimate -1.24309438 0.18857870 0.43778161 -0.20091943 1.09661438
Std.Err   0.06546551 0.06822303 0.07108721  0.06519662 0.08254387
```

We see the difficulty, b_i, and their standard errors for all items $i \in \{2, 3, 4, 5, 6\}$ listed as `item (category) difficulty parameters (eta)` (b_1 is fixed at zero). Additional fit information, item easiness and reparametrized values can be obtained by using the `summary` method.

```
R> summary(rm1)

Results of RM estimation:

Call:  RM(X = WIRS)

Conditional log-likelihood: -1826.263
Number of iterations: 9
Number of parameters: 5

Item (Category) Difficulty Parameters (eta): with 0.95 CI:
       Estimate Std. Error lower CI upper CI
Item 2   -1.243      0.065   -1.371   -1.115
Item 3    0.189      0.068    0.055    0.322
Item 4    0.438      0.071    0.298    0.577
Item 5   -0.201      0.065   -0.329   -0.073
Item 6    1.097      0.083    0.935    1.258

Item Easiness Parameters (beta) with 0.95 CI:
            Estimate Std. Error lower CI upper CI
beta Item 1    0.279      0.065    0.152    0.406
beta Item 2    1.243      0.065    1.115    1.371
beta Item 3   -0.189      0.068   -0.322   -0.055
beta Item 4   -0.438      0.071   -0.577   -0.298
beta Item 5    0.201      0.065    0.073    0.329
beta Item 6   -1.097      0.083   -1.258   -0.935
```

Estimated item characteristic curves (ICC) for the Rasch model can be found in the left panel of Figure 4.

Under the column `Estimate` in the part labeled `Item Easiness Parameters (beta)`, we see that the "easiest" item is Item 2 followed by Item 1 ("easy" here means most readily answered with "1" by the firm's official). The item with the lowest probability to be answered with "1" is Item 6, followed by Item 1 and Item 5 which are similarly appealing in this analysis. So, according to the officials, meetings with groups of workers, individual workers and local union representatives are more likely to be done. We can also look at the ICC curves (left panel in Figure 4).

```
R> plotjointICC(rm1, main="Item Characteristics (1PL)",ylab="Probability")
```

How well does the Rasch model fit these items? We will use Ponocny's nonparametric T10 for global model fit by means of subgroup invariance and a test to assess constant discrimination of item 1 against a scale comprising of all other items respectively (we sample `n=1000` matrices).

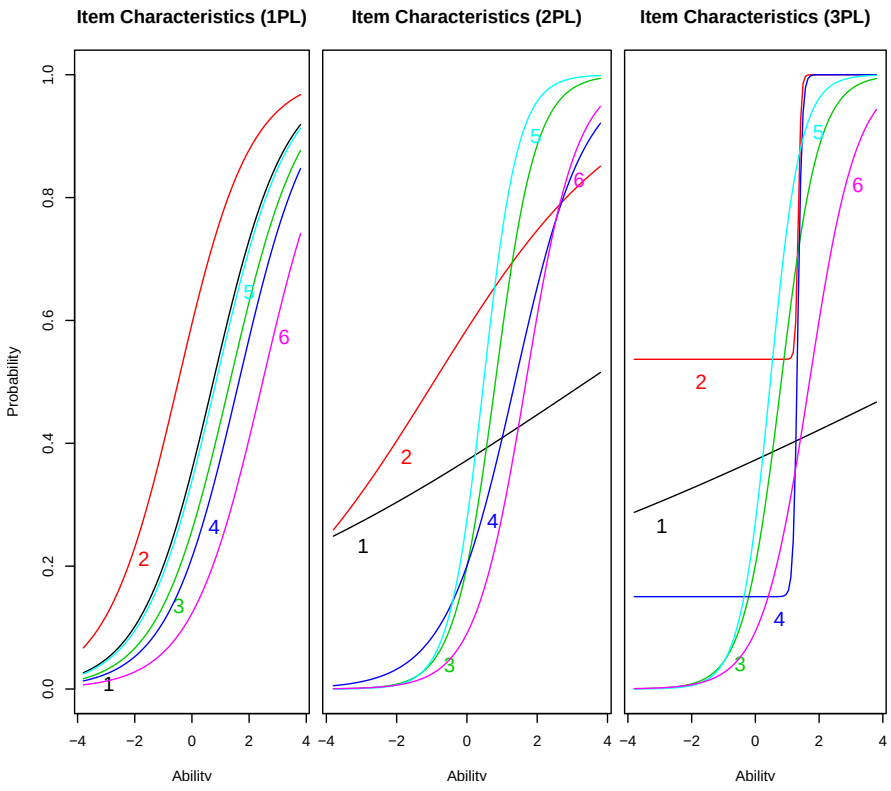

Figure 2: Item characteristic curves for the Rasch Model (1PL; left panel), the 2PL (middle panel) and 3PL (right panel) models fitted to WIRS data.

```
R> set.seed(210485)
R> t10 <- NPtest(as.matrix(WIRS),n=1000,method="T10")
R> t10

Nonparametric RM model test: T10 (global test - subgroup-invariance)
Number of sampled matrices: 1000
Split: median
Group 1: n =  333   Group 2: n = 672
one-sided p-value: 0

R> Tpbis1 <- NPtest(as.matrix(WIRS),n=1000,method="Tpbis",idxt=1,idxs=2:6)
R> Tpbis1

Nonparametric RM model test: Tpbis (discrimination)
    (pointbiserial correlation of test item vs. subscale)
Number of sampled matrices: 1000
Test Item: 1
Subscale  - Items: 2 3 4 5 6
one-sided p-value (rpbis too low): 0
```

As can be seen both null hypothesis—T10's of equal item difficulties in both subgroups and Tpbis of equal item discrimination for item 1 vs. the other items—have to be rejected. The Rasch model does not fit well to the data [5]. The lack of fit can also be seen by graphically assessing subgroup invariance with the median score as split criterion or by checking item infit statistics (see Figure 4) using the commands

```
R> lrres <- LRtest(rm1,se=TRUE)
R> plotGOF(lrres,conf=list())
R> plotPWmap(rm1)
```

As there is reason to be believe that there is different item discrimination, we proceed with using higher parametrized models. With functions in the **ltm** package, we can fit the 2PL (`ltm(WIRS~z1)`) and the 3PL (`tpm(WIRS)`) to the WIRS data. First the 2PL:

```
R> twoPL <- ltm(WIRS~z1,IRT.param=TRUE)
R> summary(twoPL)

Call:
ltm(formula = WIRS ~ z1, IRT.param = TRUE)

Model Summary:
   log.Lik      AIC      BIC
 -3420.066 6864.131 6923.084
```

[5] If the scale is iteratively changed to conform to the Rasch model eventually (function `stepwiseIt(rm1)`), Items 1 and 2 are removed and Items 3 to 6 are selected.

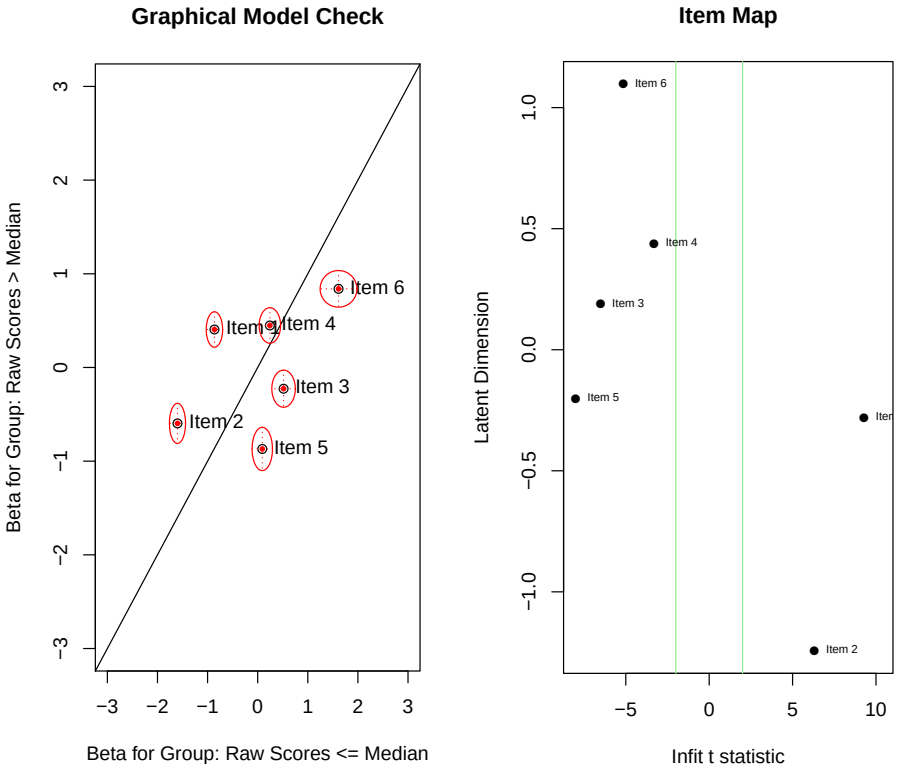

Figure 3: Graphical model check of the Rasch model for the WIRS data, with 95%-confidence ellipses for the parameter estimates (left panel) and a Bond-and-Fox pathway map (right panel) which displays the location of each item against its infit t-statistic.

```
Coefficients:
                value std.err  z.vals
Dffclt.Item 1  3.4011  2.1036  1.6168
Dffclt.Item 2 -0.9421  0.2981 -3.1603
Dffclt.Item 3  0.8094  0.0813  9.9602
Dffclt.Item 4  1.3689  0.1618  8.4601
Dffclt.Item 5  0.4763  0.0616  7.7289
Dffclt.Item 6  1.6805  0.1614 10.4124
Dscrmn.Item 1  0.1534  0.0940  1.6326
Dscrmn.Item 2  0.3677  0.0990  3.7158
Dscrmn.Item 3  1.7180  0.2399  7.1617
Dscrmn.Item 4  1.0101  0.1415  7.1379
Dscrmn.Item 5  2.0324  0.3230  6.2918
Dscrmn.Item 6  1.3746  0.1902  7.2268

Integration:
method: Gauss-Hermite
quadrature points: 21

Optimization:
Convergence: 0
max(|grad|): 0.0057
quasi-Newton: BFGS
```

which leads to interesting results compared to the Rasch analysis. The middle panel of Figure 4 shows the estimated item characteristic curves for the 2PL model solution.

```
R> plot(twoPL, main="Item Characteristics (2PL)")
```

Notably, the Items 1 and 2 have very flat discrimination. For firms with low values on the latent trait, Item 1 and Item 2 are those that are most easily answered with a "1", whereas the opposite is true for firms high on the latent trait. Note, however, that the 2PL cannot be considered to fit the data well as can be seen from the item or person fit (the latter not shown):

```
R> item.fit(twoPL)

Item-Fit Statistics and P-values

Call:
ltm(formula = WIRS ~ z1, IRT.param = TRUE)

Alternative: Items do not fit the model
Ability Categories: 10

           X^2 Pr(>X^2)
Item 1  72.4122  <0.0001
```

```
Item 2 110.7422  <0.0001
Item 3 176.9384  <0.0001
Item 4 301.1355  <0.0001
Item 5 241.8647  <0.0001
Item 6 117.4444  <0.0001
```

```
R> person.fit(twoPL)
```

The behavior of the items in the 2PL model (especially Items 1 and 2) suggests that there either is the possibility that officials of firms with low latent trait values answer in a socially desirable manner leading to some items appearing more difficult for those higher up the latent trait and/or multidimensionality of the six item scale.

To check the first possibility, we fit a 3PL model (with guessing parameter):

```
R> threePL <- tpm(WIRS)
R> threePL

Call:
tpm(data = WIRS)

Coefficients:
        Gussng  Dffclt  Dscrmn
Item 1   0.002   5.128   0.102
Item 2   0.537   1.348  23.147
Item 3   0.000   0.810   1.699
Item 4   0.150   1.329  15.141
Item 5   0.000   0.470   2.102
Item 6   0.000   1.697   1.338

Log.Lik: -3406.941
```

The output lists the c_i, b_i and a_i in the first, second and third column respectively. The item characteristics curves under the 3PL can be found in the right panel of Figure 4.

```
R> plot(threePL, main="Item Characteristics (3PL)")
```

We see that the 3PL suggests substantial "guessing" (here better interpreted as a general tendency towards this type of consultation) for Items 2 ($c_2 = 0.537$) and 4 ($c_4 = 0.15$) to be present. They behave interestingly in terms of discrimination as well: up to a certain point on the latent trait they have a near constant probability to score "1" and after the more or less same threshold on the latent trait is crossed (around 1.5), the probability increases to one (all firms located higher than this position consulted with commitees and worker groups, a case of separation). Judging from the parameters estimated ($a_2 = 23.15$ and $a_4 = 15.14$) and their standard errors, there likely is an estimation artifact due to separation or a unidimensional 3PL is generally not apt to model all these items simultaneously. Item 1 displays the same behavior as found in the 2PL model, a very low discrimination between different values of the latent trait. Items 3, 5 and 6 appear to be well scalable with the 3PL.

The 3-PL fits significantly better than the Rasch model or the 2PL, as can be checked with `anova(twoPL,threePL)`,

```
R> anova(rm1,twoPL)

 Likelihood Ratio Table
          AIC     BIC  log.Lik    LRT df p.value
rm1   7032.56 7066.95 -3509.28
twoPL 6864.13 6923.08 -3420.07 178.43  5  <0.001

R> anova(twoPL,threePL)

 Likelihood Ratio Table
            AIC     BIC  log.Lik   LRT df p.value
twoPL   6864.13 6923.08 -3420.07
threePL 6849.88 6938.31 -3406.94 26.25  6  <0.001
```

but judging by the item fit statistics and the other results, it still does not fit the data very well, notwithstanding the insight gained from applying the 3PL:

```
R> item.fit(threePL)

Item-Fit Statistics and P-values

Call:
tpm(data = WIRS)

Alternative: Items do not fit the model
Ability Categories: 10

            X^2 Pr(>X^2)
Item 1 153.3031  <0.0001
Item 2  17.6565   0.0136
Item 3 375.7460  <0.0001
Item 4  68.2599  <0.0001
Item 5 289.2301  <0.0001
Item 6 504.2906  <0.0001
```

Therefore the results suggest further analyses. We will first use nonparametric approaches for exploration of the encountered problems, followed by multidimensional modelling.

We use the functionality from the **mokken** and the **KernSmootIRT** packages to investigate whether a monotone homogeneity model can be assumed to hold for the scale. Note that this model is less restrictive than the parametric IRT models, assuming scale properties that are, however, also found in parametric IRT (such as local independence, unidimensionality, etc.). Hence, this model can be used to assess necessary conditions for parametric IRT to hold.

We can judge the scale by a number of criteria and see if the monotone homogeneity model holds. Judging by scalability `coefH(WIRS)$H`, the scale is even less than "weak".

```
R> coefH(WIRS)

$Hij
       Item 1   se      Item 2   se      Item 3  se      Item 4   se
Item 1                   -0.490  (0.052)  0.090  (0.039)  -0.139  (0.042)
Item 2  -0.490  (0.052)                    0.122  (0.059)   0.393  (0.061)
Item 3   0.090  (0.039)   0.122  (0.059)                    0.260  (0.039)
Item 4  -0.139  (0.042)   0.393  (0.061)  0.260  (0.039)
Item 5   0.089  (0.033)   0.178  (0.049)  0.414  (0.039)   0.290  (0.044)
Item 6   0.159  (0.060)   0.157  (0.087)  0.360  (0.053)   0.218  (0.049)
       Item 5  se      Item 6  se
Item 1  0.089  (0.033)  0.159  (0.060)
Item 2  0.178  (0.049)  0.157  (0.087)
Item 3  0.414  (0.039)  0.360  (0.053)
Item 4  0.290  (0.044)  0.218  (0.049)
Item 5                  0.474  (0.057)
Item 6  0.474  (0.057)

$Hi
       Item H  se
Item 1  -0.058 (0.023)
Item 2   0.023 (0.033)
Item 3   0.249 (0.020)
Item 4   0.192 (0.023)
Item 5   0.263 (0.018)
Item 6   0.283 (0.028)

$H
 Scale H se
   0.154 (0.015)
```

Additionally, Items 1 and 2 and 4 have negative item-pair scalabilities with each other respectively and no item has an item scalability of 0.3 or above. The assumptions on unidimensionality and/or local independence are therefore likely not fullfilled for this scale.

Similar results can be reported for monotonicity, either by employing `check.montonicity(WIRS)`

```
R> mono <- check.monotonicity(WIRS)
R> summary(mono)

       ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac zmax #zsig crit
Item 1 -0.06  10   7     0.7  0.29 1.04  0.1041 5.06     5  331
Item 2  0.02   6   0     0.0  0.00 0.00  0.0000 0.00     0    0
Item 3  0.25   6   0     0.0  0.00 0.00  0.0000 0.00     0    0
Item 4  0.19   6   0     0.0  0.00 0.00  0.0000 0.00     0    0
Item 5  0.26   6   0     0.0  0.00 0.00  0.0000 0.00     0    0
Item 6  0.28  10   0     0.0  0.00 0.00  0.0000 0.00     0    0
```

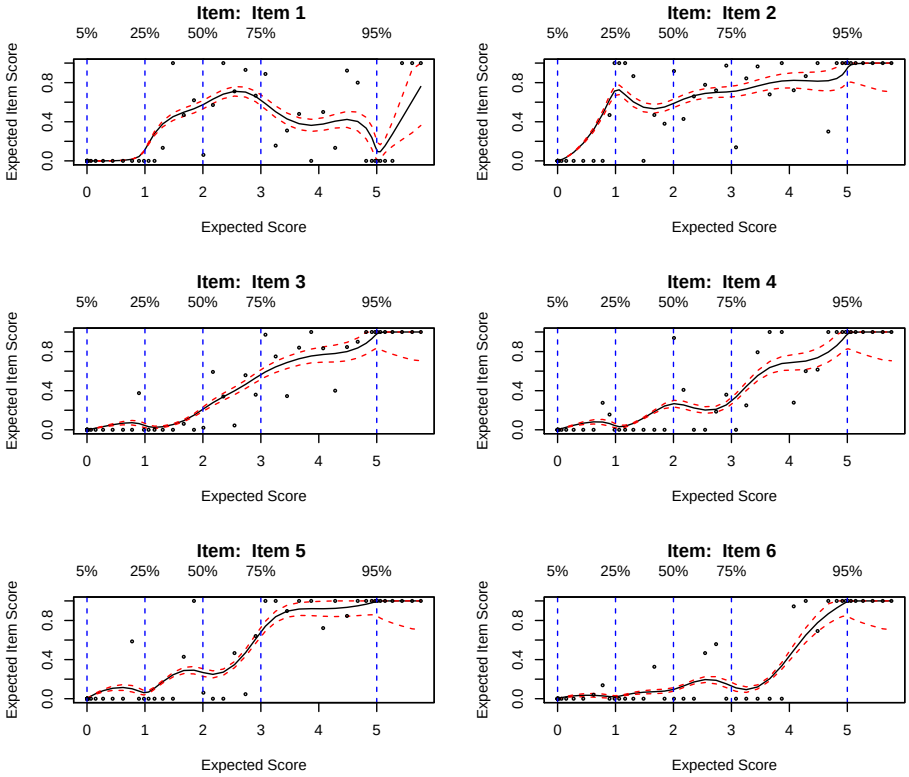

Figure 4: Smoothed item characteristic curves for the "1" response of the WIRS data with approximate pointwise 95% confidence intervals. The dots show the observed frequency of the "1" response for the latent trait value regions over which was smoothed.

or by estimating the item characteristic of a "1" response with kernel smoothing directly from the response pattern, see Figure 4.

```
R> ks1 <- ksIRT(responses=WIRS, key=rep(1,ncol(WIRS)),format=2)

R> plot(ks1, plottype="EIS",items=1:6)
```

We see that Item 1 clearly violates the monotonicity assumption. Item 2 appears to be a borderline case. This also explains the odd results in the 2PL and 3PL analyses. All the other items seem to behave monotonically.

We can also check non-intersection of the item step response function as the third assumption of the monotone homogeneity model by using either `check.pmatrix(WIRS)` or `check.restscore(WIRS)` and invariant item ordering `check.iio(WIRS)`. Both results suggest many violations of this assumptions (and always for Item 1).

```
R> restscore.list <- check.restscore(WIRS)
R> summary(restscore.list)

       ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac zmax #zsig crit
Item 1 -0.06  15    5    0.33  0.59 1.64  0.1091 8.61     5  332
```

```
Item 2  0.02  13   1    0.08  0.06 0.06  0.0043 1.44     0   42
Item 3  0.25  13   2    0.15  0.44 0.49  0.0376 7.47     1  163
Item 4  0.19  15   2    0.13  0.34 0.39  0.0257 5.93     1  132
Item 5  0.26  13   3    0.23  0.59 0.77  0.0591 8.61     2  225
Item 6  0.28  15   1    0.07  0.14 0.14  0.0092 2.99     1   67
```

```
R> iio.list <- check.iio(WIRS)
R> summary(iio.list)
```

```
$method
[1] "MIIO"

$item.summary
       ItemH #ac #vi #vi/#ac maxvi  sum sum/#ac  zmax #zsig crit
Item 2  0.02  13   1    0.08  0.06 0.06  0.0043  1.39     0   41
Item 1 -0.06  15   5    0.33  0.59 1.64  0.1091 11.72     5  348
Item 5  0.26  13   3    0.23  0.59 0.77  0.0591 11.72     2  241
Item 3  0.25  13   2    0.15  0.44 0.49  0.0376  8.75     1  169
Item 4  0.19  15   2    0.13  0.34 0.39  0.0257  6.80     1  136
Item 6  0.28  15   1    0.07  0.14 0.14  0.0092  3.03     1   67

$backward.selection
       step 1 step 2
Item 2      0      0
Item 1      4     NA
Item 5      1      0
Item 3      1      0
Item 4      1      0
Item 6      1      0

$HT
[1] 0.2791073
```

Based on these diagnostics we can conclude that the whole set of items does not conform to a Mokken scale, which carries over to any of the basic parametric IRT models. Either there are violations of local independence, unidimensionality, monotone item characteristics or any combination of these. Even the most general of these models, the monotone homogeneity model (nonparametric graded response model) does not hold. This is particularly striking for Item 1, "Informal discussion with individual workers", which defies any attempt of classic IRT modelling.

To help us out of this situation, the **mokken** package allows to partition the items into scales for which the assumptions do hold, by using the `aisp` function. Two algorithms are included, the classic algorithm (the default `search='normal'`) and a genetic algorithm (`search='ga'`). We will use the genetic algorithm for illustration (in this case both algorithms yield the same partition).

```
R> aisp(WIRS,search='ga')
```

```
       Scale
Item 1     0
Item 2     2
Item 3     1
Item 4     2
Item 5     1
Item 6     1
```

The algorithm suggest the following: Items 3, 5 and 6 should form a scale, Items 2 and 4 a second. Item 1 is not Mokken scalable (see the monotonicity issues above). This corroborates our suspicion from the preceding parametric analyses, namely that Item 3, 5 and 6 form a regular scale that can likely be modeled with a 1- or 2PL, that Items 2 and 4 are qualitatively different from the others and that Item 1 cannot be scaled with any of the usual IRT models (here a nonmonotone ICC might be necessary).

We could now go back to fit IRT models to the two scales that meet the Mokken scale properties. However, we will use a 2-dimensional IRT model to model both subscales simultaneously. We thus remove the first item and rearrange the other items for convenience.

```
R> WIRS2Scales <- WIRS[,c(3,5,6,2,4)]
```

The new scale `WIRS2Scales` now consists of the two Mokken subscales comprising Items 3, 5, 6 and Items 2 and 4. Thus, as a last step, we set up a two-dimensional 3PL model estimated with the **mirt** package.

```
R> mmod1 <- mirt(WIRS2Scales,2,SE=TRUE,,itemtype="3PL")
R> mmod1

Call:
mirt(data = WIRS2Scales, model = 2, itemtype = "3PL", SE = TRUE)

Full-information item factor analysis with 2 factors
Converged in 179 iterations with 21 quadrature.
Log-likelihood = -2738.19
AIC = 5514.379; AICc = 5577.713
BIC = 5607.722; SABIC = 5547.376
G2 (12) = 21.62, p = 0.0421
X2 (12) = 18.77, p = 0.0943
RMSEA (G2) = 0.028; RMSEA (X2) = 0.024
CFI (G2) = 0.971; CFI (X2) = 0.99
TLI (G2) = 0.938; TLI (X2) = 0.979
```

We see that the fit is quite good (judging by the χ^2, RMSEA, CFI and TLI) . The item characteristic surfaces (ICS) for all items can be obtained by

```
R> itemplot(mmod1,item=1,sub="WIRS Item 3")
R> itemplot(mmod1,item=2,sub="WIRS Item 5")
R> itemplot(mmod1,item=3,sub="WIRS Item 6")
R> itemplot(mmod1,item=4,sub="WIRS Item 2")
R> itemplot(mmod1,item=5,sub="WIRS Item 4")
```

and are found in Figures 5 and 6.

The corresponding uni- and multidimensional item parameter estimates are (note the estimation problems for the standard errors):

```
R> coef(mmod1)

Rotation:  oblimin

$`Item 3`
        a1     a2      d     g  u
pars 2.072 -0.299 -1.600 0.015  1
SE     NaN  0.517  0.118 0.013 NA

$`Item 5`
        a1     a2      d     g  u
pars 2.040 -0.105 -1.025 0.013  1
SE   0.561  0.263  0.366 0.042 NA

$`Item 6`
        a1     a2      d    g  u
pars 1.424 -0.207 -2.303 0.00  1
SE   2.061  0.691  2.771 0.12 NA

$`Item 2`
         a1    a2     d     g  u
pars -0.081 4.148 0.307 0.118  1
SE    0.376 2.516 1.257 0.213 NA

$`Item 4`
        a1    a2      d     g  u
pars 2.976 1.806 -4.848 0.123  1
SE   1.626   NaN    NaN 0.024 NA

$GroupPars
     MEAN_1 MEAN_2 COV_11 COV_21 COV_22
pars      0      0      1  0.349      1
SE       NA     NA     NA     NA     NA
```

We see that the items that form the first Mokken scale (Items 3, 5 and 6) have substantial discrimination (here called "factor slope") on mainly one dimension (the first latent trait θ_1; the column is labeled `a_1` for the $a_{i1}, i \in \{3, 5, 6, 2, 4\}$, e.g., for Item 5 on trait 1 it is $a_{51} = 2.0$) with only small discriminatory contributions (smaller than the standard errors in fact) to the other dimension (see Figure 5). Item 4 also has substantial discrimination on this trait, but less than the other three. Marginally, for this latent trait the "easiest" type of consultations a firm engages in is Item 5 ("discussions with the union representatives at the establishment", `d=-1.0`, the b_i are labeled as `d` and can be interpreted as easiness parameters), followed by Item 3 ("discussions in established joint consultative committee", `d=-1.6`) and

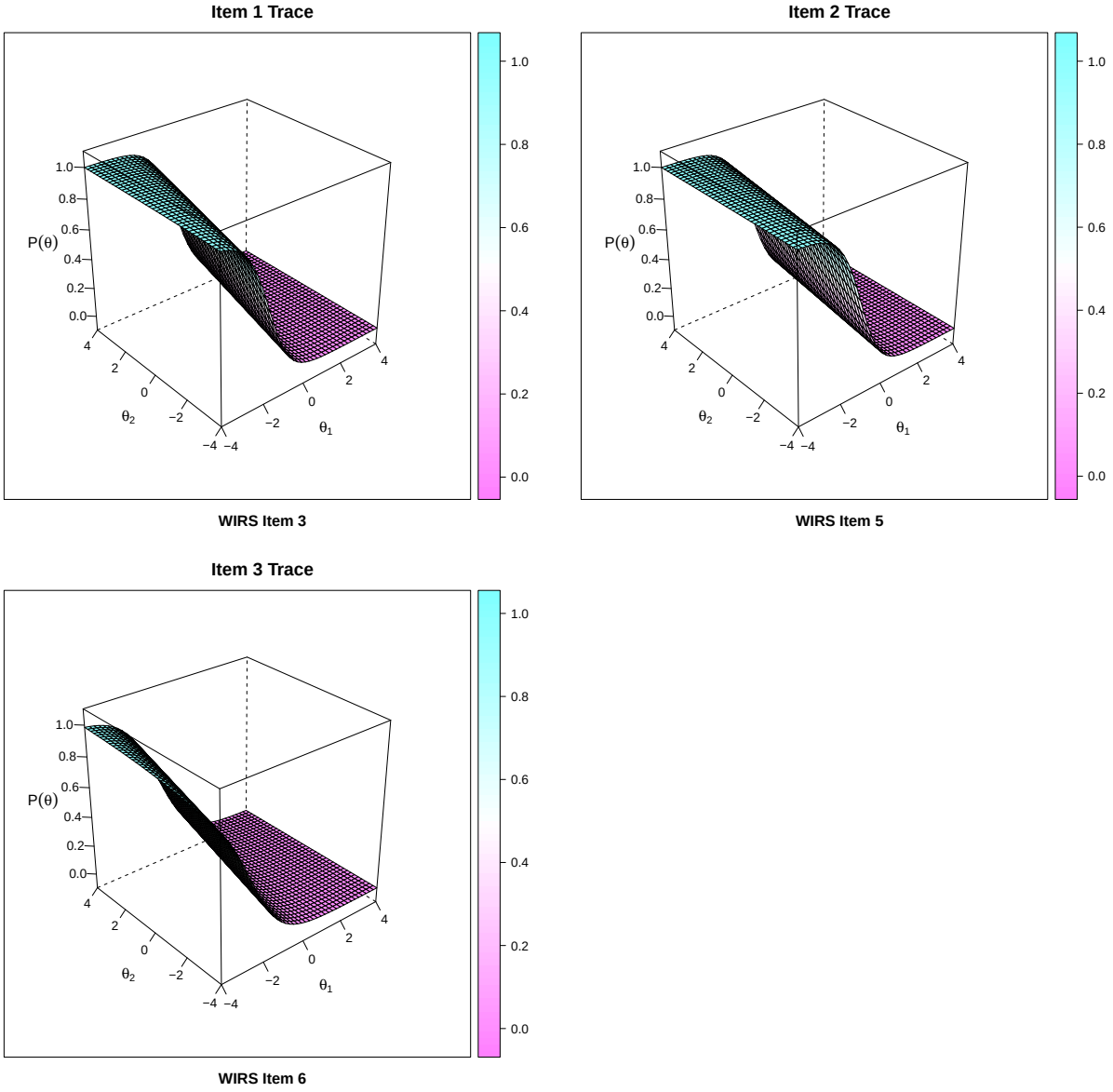

Figure 5: Item characteristic surfaces for the two-dimensional 3PL model for WIRS Items 3, 5, 6.

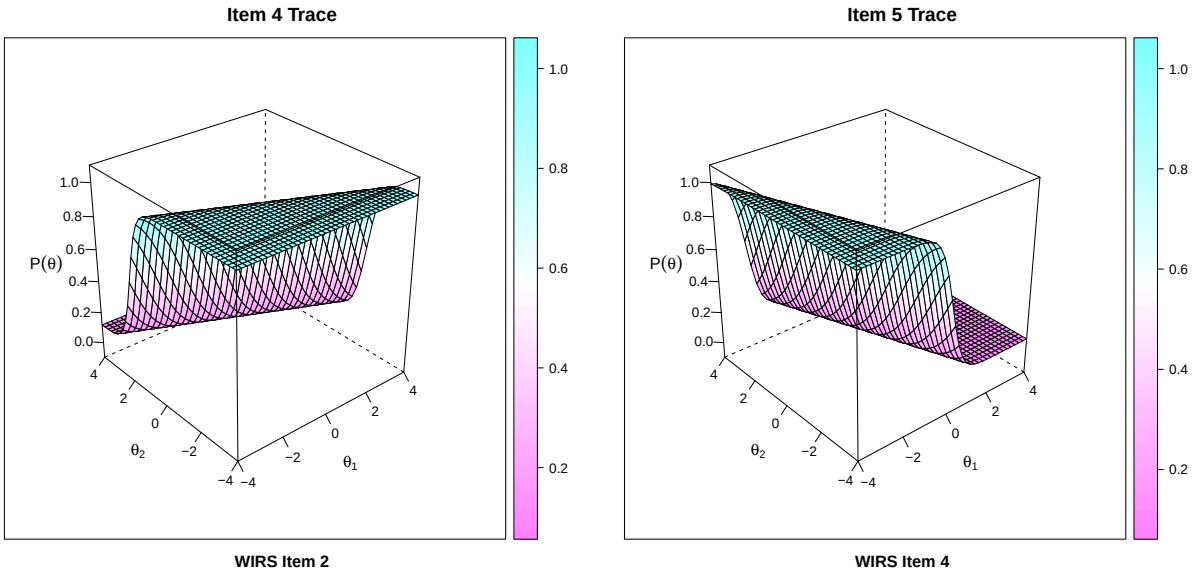

Figure 6: Item characteristic surfaces for the two-dimensional 3PL model for WIRS Items 2 and 4.

Item 6 ("discussions with paid union officials from outside", `d=-2.3`). Item 4's ("discussions in specially constituted committee to consider the change") easiness for this trait depends largely on whether one score high or low on the second. For the second latent trait, Item 2 ("meeting with groups of workers") exhibits the behavior of being nearly exclusively related to this trait (see the column labeled `a_2` which contains the a_{i2}). Item 4 is discriminatory on this trait too, but less so than for the first trait. Here the position on the second trait determines how likely Item 4 is affirmed conditional on the location on the first latent trait, with firms that rank high on the second trait being less likely to answer in the affirmative on Item 4. This can be also be seen in Figure 6 where the latter two items ICS are displayed, as well as in the column of the output labeled `COV_21`, which gives the estimated correlation of both latent traits (which is not that high here as apart from Item 4, the items do not load particularly high on more than one trait).

To summarize, we illustrated an item analysis that uses a number of packages provided in the R environment. Starting from modeling a six item scale with the 1PL, we soon found the need to use 2- and 3PL. After these attempts proved fruitless, a nonparametric analysis detected a number of violations of parametric IRT assumptions when treating the items like they form a single scale and suggested to divide the scale into two parts. Following this suggestion the resulting subscales were subsequently successfully modeled with a multidimensional 3PL. Apparently, a firm's policy on management/worker consultation as measured by the WIRS items is (at least) a two-dimensional construct. Item 2 and also Item 4 relate to consultation with representative of workers and special commitees, whereas Items 3, 5, 6 and to a lesser degree Item 4 relate to the consultations with the union or joint commitees. The two dimensions might therefore represent consultations with local representatives and focus groups or grass roots structures from the firm, whereas the other dimension represents consultations with official representatives that usually do not come from the firm directly. Informal consultations with individual workers seems to be a class on its own. We believe this analysis clearly shows

how the typical user can utilize the diversity and flexibility of IRT packages in R.

5. Support for IRT packages in R

For support, the package maintainer of each package should be contacted. Alternatively, there is the R-help mailing list https://stat.ethz.ch/mailman/listinfo/r-help. Some of the packages described here are developed openly on R-Forge (r-forge.r-project.org), which also offers a discussion and help forum for each project.

6. Availability of R and IRT packages

The full version of R can be obtained free (as in beer and speech) from r-project.org and the presented packages can be downloaded via the Comprehensive R Archive Network (CRAN) for Windows, Mac OS and Linux. For other plattforms, the source code can be downloaded and compiled. Add-on packages like the presented packages for IRT can be simply installed from within R. For example, once R has been installed, an IRT package (say **packagename**) can be installed by typing

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

in the command line, and after installation loaded for each R session as

```
R> require(packagename)
```

See Venables and Smith (2002) for a free introduction to basic R usage. A number of free manuals are also available via http://cran.r-project.org/manuals.html.

References

Anderson C, Li Z, Vermunt J (2007). "Estimation of Models in a Rasch Family for Polytomous Items and Multiple Latent Variables." *Journal of Statistical Software*, **20**(6), 1–36.

Andersson B, Branberg K, Wiberg M (2013). *kequate: The kernel method of test equating*. R package version 1.3.1, URL http://CRAN.R-project.org/package=kequate.

Andersson B, Bränberg K, Wiberg M (in press). "Performing the Kernel Method of Test Equating with the Package kequate." *Journal of Statistical Software*.

Bartholomew D (1998). "Scaling Unobservable Constructs in Social Science." *Applied Statistics*, **47**(1), 1–13.

Bates D, Maechler M, Bolker B (2011). *lme4: Linear Mixed-effects Models using S4 Classes*. R package version 0.999375-42, URL http://CRAN.R-project.org/package=lme4.

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

Chalmers RP (2013). *mirt: Multidimensional Item Response Theory*. R package version 0.9.0, URL http://CRAN.R-project.org/package=mirt.

Choi S, Gibbons L, Crane P (2011). "lordif: An R Package for Detecting Differential Item Functioning Using Iterative Hybrid Ordinal Logistic Regression/Item Response Theory and Monte Carlo Simulations." *Journal of Statistical Software*, **39**(8), 1–30.

Choi S, Gibbons L, Crane P (2012). *lordif: Logistic Regression Differential Item Functioning using IRT*. R package version 0.2-2, URL http://CRAN.R-project.org/package=lordif.

De Boeck P, Bakker M, Zwitser R, Nivard M, Hofman A, Tuerlinckx F, Partchev I (2010). "The Estimation of Item Response Models with the lmer Function from the lme4 Package in R." *Journal of Statistical Software*, **39**(12), 1–28.

Doran H, Bates D, Bliese P, Dowling M (2007). "Estimating the Multilevel Rasch Model: With the lme4 Package." *Journal of Statistical Software*, **20**(2), 1–18.

Fox J (2005). "The R Commander: A Basic Statistics Graphical User Interface to R." *Journal of Statistical Software*, **14**(9), 1–42.

Frick H, Strobl C, Leisch F, Zeileis A (2012a). "Flexible Rasch Mixture Models with Package psychomix." *Journal of Statistical Software*, **48**(7), 1–25.

Frick H, Strobl C, Leisch F, Zeileis A (2012b). *psychomix: Psychometric Mixture Models*. R package version 1.0-0, URL http://CRAN.R-project.org/package=psychomix.

Hatzinger R, Rusch T (2009). "IRT models with relaxed assumptions in eRm: A manual-like instruction." *Psychology Science Quarterly*, **51**(1), 87.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.

Jackman S (2011). *pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University*. Department of Political Science, Stanford University, Stanford, California. R package version 1.04.1, URL http://pscl.stanford.edu/.

Jara A, Hanson T, Quintana F, Müller P, Rosner G (2011). "DPpackage: Bayesian Semi- and Nonparametric Modeling in R." *Journal of Statistical Software*, **40**(5), 1–30.

Jara A, Hanson T, Quintana FA, Mueller P, Rosner G (2012). *DPpackage: Bayesian nonparametric modeling in R*. R package version 1.1-4, URL http://CRAN.R-project.org/package=DPpackage.

Li Z, Hong F (2007). *plRasch: Log Linear by Linear Asscociation Models*. R package version 0.1, URL http://CRAN.R-project.org/package=plRasch.

Magis D, Beland S, Raiche G (2012). *difR: Collection of methods to detect dichotomous differential item functioning (DIF) in psychometrics*. R package version 4.2.

Magis D, Beland S, Tuerlincx F, De Boeck P (2010). "A general framework and an R package for the detection of dichotomous differential item functioning." *Behavior Research Methods*, **42**(3), 847–862.

Magis D, Gilles R (2012). *catR: Procedures to generate IRT adaptive tests (CAT)*. R package version 2.3, URL http://CRAN.R-project.org/package=catR.

Magis D, Raîche G (2012). "Random Generation of Response Patterns under Computerized Adaptive Testing with the R Package catR." *Journal of Statistical Software*, **48**(8), 1–31.

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

Mair P, Hatzinger R (2007b). "Psychometrics Task View." *R News*, **7**(3), 38–40.

Mair P, Hatzinger R, Maier M (2013). *eRm: Extended Rasch Modeling*. R package version 0.15-2, URL http://CRAN.R-project.org/package=eRm.

Martin A, Quinn K, Park JH (2011). "MCMCpack: Markov Chain Monte Carlo in R." *Journal of Statistical Software*, **42**(9), 22.

Martin A, Quinn K, Park JH (2012). *MCMCpack: Markov chain Monte Carlo (MCMC) Package*. R package version 1.2-3, URL http://CRAN.R-project.org/package=MCMCpack.

Mazza A, Punzo A, McGuire B (2013). *KernSmoothIRT: Nonparametric Item Reponse Theory*. R package version 5.0, URL http://CRAN.R-project.org/package=KernSmoothIRT.

Mazza A, Punzo A, McGuire B (in press). "KernSmoothIRT: An R Package allowing for Kernel Smoothing in Item Response Theory." *Journal of Statistical Software*.

Pemstein D, Quinn K, Martin A (2011). "The Scythe Statistical Library: An Open Source C++ Library for Statistical Computation." *Journal of Statistical Software*, **42**(12), 1–26.

Plummer M (2011). *rjags: Bayesian graphical models using MCMC*. R package version 3-3.

Plummer M, Best N, Cowles K, Vines K (2006). "CODA: Convergence Diagnosis and Output Analysis for MCMC." *R News*, **6**(1), 7–11.

Preinerstorfer D (2012). *mRm: An R package for conditional maximum likelihood estimation in mixed Rasch models*. R package version 1.1.2, URL http://CRAN.R-project.org/package=mRm.

Preinerstorfer D, Formann A (2012). "Parameter recovery and model selection in mixed Rasch models." *British Journal of Mathematical and Statistical Psychology*, **65**(2), 251–262.

R Core Team (2013). *R Data Import/Export*. URL http://cran.r-project.org/doc/manuals/R-data.html.

R Development Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rizopoulos D (2006). "ltm: An R package for Latent Variable Modelling and Item Response Theory Analyses." *Journal of Statistical Software*, **17**(5), 1–25.

Rizopoulos D (2013). *ltm: Latent Trait Models under IRT*. R package version 0.9-9, URL http://CRAN.R-project.org/package=ltm.

Rusch T, Maier M, Hatzinger R (2013). "Linear Logistic Models with Relaxed Assumptions in R." In B Klaussen, D van den Poel, A Ultsch (eds.), *Algorithms from & for Nature and Life*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 337–347. Springer-Verlag, Heidelberg.

Strobl C, Kopf J, Zeileis A (in press). "A New Method for Detecting Differential Item Functioning in the Rasch Model." *Psychometrika*.

The Psychometrics Center of the University of Cambridge (2012). "Concerto: R-Based Online Adaptive Testing Platform." Retrieved 29-05-2012. From http://www.psychometrics.cam.ac.uk/page/300/concerto-testing-platform.htm.

Ünlü A, Yanagida T (2011). "R you ready for R?: The CRAN Psychometrics Task View." *British Journal of Mathematical and Statistical Psychology*, **64**(1), 182–186.

Van der Ark LA (2007). "Mokken Scale Analysis in R." *Journal of Statistical Software*, **20**(11), 1–19.

Van der Ark LA (2012). "New Developments in Mokken Scale Analysis in R." *Journal of Statistical Software*, **48**(5), 1–27.

Van Der Ark LA (2013). *mokken: Mokken Scale Analysis in R*. R package version 2.7.5, URL http://CRAN.R-project.org/package=mokken.

Venables W, Smith D (2002). *An Introduction to R*. Network Theory Ltd., Bristol.

von Davier AA, Holland PW, Thayer DT (2004). *The kernel method of test equating*. Springer.

Weeks J (2010). "plink: An R Package for Linking Mixed-Format Tests Using IRT-Based Methods." *Journal of Statistical Software*, **35**(12), 1–33.

Weeks J (2011). *plink: IRT Separate Calibration Linking Methods*. R package version 1.3-1, URL http://CRAN.R-project.org/package=plink.

Willse J (2009). *mixRasch: Mixture Rasch Models with JMLE*. R package version 0.1, URL http://CRAN.R-project.org/package=mixRasch.

Zeileis A, Strobl C, Wickelmaier F, Kopf J (2011). *psychotree: Recursive Partitioning Based on Psychometric Models*. R package version 0.12-1, URL http://CRAN.R-project.org/package=psychotree.

Zopluoglu C (2012). *EstCRM: Calibrating Parameters for the Samejima's Continuous IRT Model*. R package version 1.2, URL http://CRAN.R-project.org/package=EstCRM.

Affiliation:

Thomas Rusch
Center for Empirical Research Methods
WU (Wirtschaftsuniversität Wien)
Welthandelsplatz 1, D4
1020 Wien, Austria
E-mail: Thomas.Rusch@wu.ac.at

Patrick Mair
Department of Psychology
Harvard University
33 Kirkland Street
Cambridge, MA 02138
United States of America
E-mail: mair@fas.harvard.edu