Lexter Manual

April 21, 2021

Cees Glas

Nikky van Buuren

c.a.w.glas@utwente.nl

support@shinylexter.com

Contents

1. Introduction

Item response theory (IRT) provides a useful and theoretically well-founded framework for educational measurement. It supports such activities as the construction of measurement instruments, linking and equating measurements, and evaluation of test bias and differential item functioning. Further, IRT has provides the underpinnings for item banking, optimal test construction and various flexible test administration designs, such as multiple matrix sampling, flexi-level testing and computerized adaptive testing.

Lexter is a further development of the MIRT package (Glas, 2010). Lexter supports inferences for three IRT models: marginal maximum likelihood (MML) estimation for the generalized partial credit model (GPCM) and the sequential model (SEQM), and Bayesian estimation using Marcov chain Monte Carlo (MCMC) computational techniques for the grade response model (GRM). For dichotomously scored items these three models are equivalent. A guessing parameter can be included in the model, so the package can be used to estimate the parameters for both the 1-, 2- and 3-parameter logistic models. For polytomously scored item, the package support reweighting the response categories if some of the categories are unobserved. Further, the program can provide inferences for a between-items multidimensional version of the GPCM, GRM and SEQM.

Summing up, Lexter supports

- MML and MCMC estimation of the item and population parameters,
- CML estimation of item parameters,
- ML, WML and EAP estimation of person parameters,
- Item fit analysis,
- Analysis of differential item functioning,
- Person fit analysis,
- Data simulation.

Besides in this manual, more information on Lexter can be found on the website https://shinylexter.com/ which includes links to Lexter Installation Manual, Lexter OnePager and example data and runs.

2. The model and estimation methods

2.1 Models for dichotomously scored responses

2.1.1. The Rasch model

For MML estimation for dichotomous response data Lexter always starts with estimating and testing the Rasch model. A response of a student *n* to an item *i* will be coded by a stochastic variable Y_{ni}. In the sequel, upper-case characters will denote stochastic variables. The realizations will be lower case characters. In the present case, there are two possible realizations, defined by

$$y_{ni} = \begin{cases} 1 & \text{if person } n \text{ responded correctly to item } i \\ 0 & \text{if this is not the case.} \end{cases} \quad (1)$$

Lexter supports the case where not all students responded to all items. To indicate whether a response is available, we define a variable

$$d_{ni} = \begin{cases} 1 & \text{if a response of person } n \text{ to item } i \text{ is available} \\ 0 & \text{if this is not the case.} \end{cases} \quad (2)$$

In an incomplete design, the definition of the response variable Y_{ni} is generalized such that it assumes an arbitrary constant if no response is available.
To introduce the Rasch model (Rasch, 1960, also abbreviated as 1PLM) define the logistic function:

$$\Psi(x) = \frac{\exp(x)}{1 + \exp(x)}$$

The 1PLM is then defined as

$$p(Y_{ni} = 1 \mid \theta_n, b_i) \; = \; \Psi(\theta_n - b_i) \quad (3)$$

that is, the probability of a correct response is given by a logistic function with argument $\theta_n - b_i$. Note that the argument has the same linear form as in Formula (1). Using the abbreviation $P_i(\theta_n) = p(Y_i = 1 \mid \theta_n, b_i)$, the two previous formulas can be combined to

$$P_i(\theta_n) = \frac{\exp(\theta_n - b_i)}{1 + \exp(\theta_n - b_i)} \quad . \tag{4}$$

The probability of a correct response as a function of ability, $P_i(\theta)$, is the so-called item response function of item *i*. Two examples of the associated item response curves are given in Figure 2.1. The x-axis is the ability continuum θ. For two items, with distinct values of b_i, the probability of a correct response is plotted for different values of θ. The item response curves increase with the value of θ, so this parameter can be interpreted as an ability parameter. Note that the order of the probabilities of a correct response for the two items is the same for all ability levels.

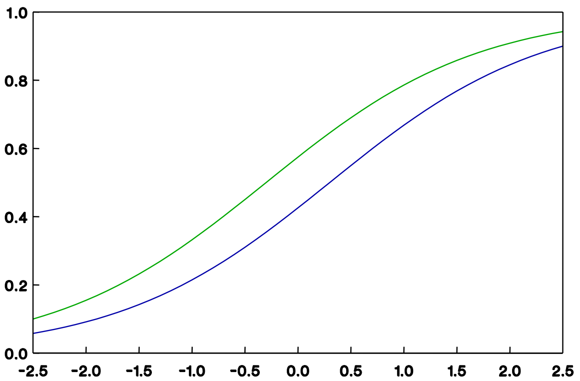


Figure 2.1 Response curves for two items in the Rasch model.

2.1.2 Two- and three-parameter models

In the 3-parameter logistic model (3PLM, Birnbaum, 1968) the probability of a correct response, depends on three item parameters, a_i, b_i and c_i which are called the discrimination, difficulty and guessing parameter, respectively. The model is given by

$$\begin{aligned} P_i(\theta_n) &= c_i + (1 - c_i) + \Psi(a_i(\theta_n - b_i)) \\ &= c_i + (1 - c_i)\frac{\exp(a_i(\theta_n - b_i))}{1 + \exp(a_i(\theta_n - b_i))} \end{aligned} \tag{5}$$

The 2PLM follows by setting the guessing parameter equal to zero, so upon introducing the constraint $c_i = 0$ and the 1PLM follows upon introducing the additional constraint $a_i = 1$.

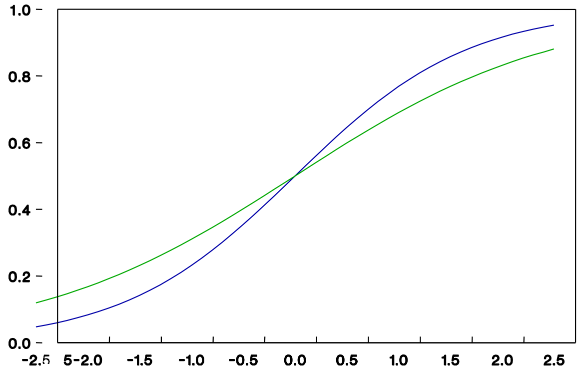


Figure 2.2 Response curves for two items in the 2PLM.

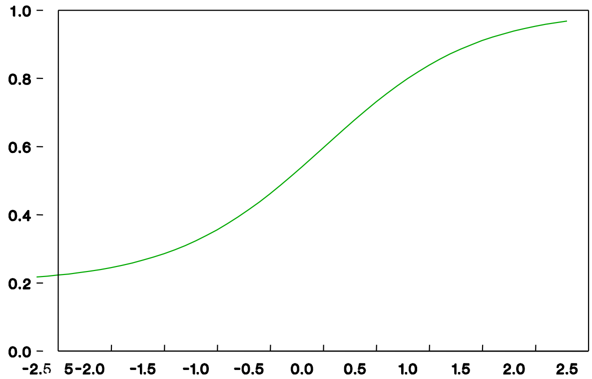


Figure 2.3 Response curve for an item in the 3PLM

Two examples of response curves of the 2PLM are shown in the Figure 2.2. It can be seen that under the 2PLM the response curves can cross. The parameter a_i determines the steepness of the response curve: The higher a_i, the steeper the response curve. The parameter a_i is called the discrimination parameter because it indexes the dependence of the item response on the latent variable θ. An item response curve for the 3PLM is given in Figure 2.3. The value of the guessing parameter was equal to 0.20, that is, $c_i = 0.20$. So the probability of a correct response of students with a very low ability level is still equal to the guessing probability, in this case, to 0.20.

Verhelst and Glas (1995) considered a version of the 2PLM where the parameter a_i is an integer constant, which will be denoted here as w_i. The main reason for considering this version of the 2PLM was that the parameters of this model can be estimated with a CML method that does not rely on assumptions of population distributions. The probability of a correct response under this model, the so-called OPLM, is given by $\Psi(w_i(\theta_n - b_i))$. In case the option of MML estimation in Lexter is chosen, also a model given by $\Psi(w_i a_i(\theta_n - b_i))$ can be estimated.

2.2 Models for polytomously scored responses

In the sequel, the response to an item *i* can be in one of the categories $m=0,\ldots,M_i$. So it will be assumed that every item has a unique number of response categories $1+M_i$ The response of a student *n* to an item *i* will be coded by stochastic variables Y_{nim}. As above, upper-case characters will denote stochastic variables, the analogous lower-case characters the realizations. So

$$y_{nim} = \begin{cases} 1 & \text{if person } i \text{ responded in category } m \text{ on item } i \\ 0 & \text{if this is not the case,} \end{cases} \quad (6)$$

for $m = 0,\ldots,M_i$. A dichotomous item is the special case where $M_i = 1$, and the number of response variables is then equal to two. To model responses to proficiency items (as opposed to, for instance, attitude items), the response curve for the lowest category should be monotone decreasing, the response curve for the highest category should be monotone increasing and the response curves for the intermediate categories should be single peaked. An example is given in Figure 2.5.

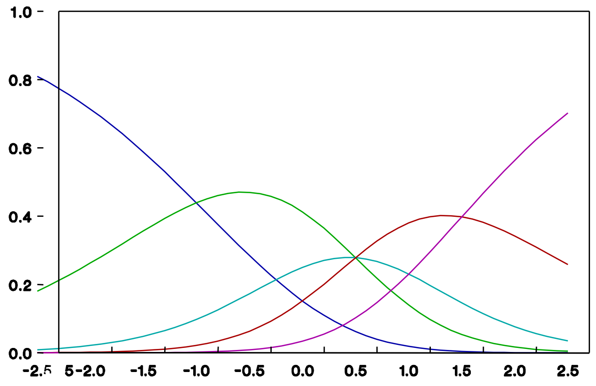


Figure 2.5. Response curves of a polytomously scored item.

Various IRT models give rise to sets of item-category curves as in Figure 2.5 (Mellenbergh, 1995). Examples are the GMPC (Muraki, 1992), the steps model (Tutz, 1990) and the graded response model (Samejima, 1969). Though the rationales underlying the models are very different, the practical implications are often negligible, because their item-category response curves are so close that they can hardly be distinguished in the basis of empirical data (Verhelst, Glas, & the Vries, 1997). Further, for dichotomously scored items, the three models are completely equivalent.

2.2.1 The GPCM

The model estimated by Lexter is given by

$$P_{nim}(\theta_n) \quad = \quad p(Y_{nim}=1 \mid \theta_n, a_i, b_i) \;=\; \frac{\exp(a_i w_{im}\theta_n - \sum_{g=1}^{m} b_{im})}{\sum_{h=0}^{M_i}\exp\left[a_i w_{im}\theta_n - \sum_{g=1}^{h} b_{ig}\right]} \quad , \qquad (7)$$

for $m=0,\ldots,M_i$. The item parameters of the GPCM are a discrimination parameter a_i that models the extent to which the item is related to the latent ability variable θ and item location parameters b_{im} that are the points where adjacent response curves cross. Further, w_{im} are scoring weights. Usually, these weights are just the category indices, that is, 0,1,2,3,…., but they can also be adjusted to accommodate categories that did not get any responses. For instance, if category 2 was unobserved, this category can be removed from the model and the weights become 0,1,3,…….

2.2.2. The SEQM

The definition of the SEQM starts with the assumption that a polytomously scored item with M_i response categories consists of M_i virtual dichotomously scored items, say item steps. If a respondent scored m , it is assumed the respondent has taken m item steps correctly, but stumbles on the next item step. The probability of taking an item step is modeled by the 2PLM. The model then becomes

$$P_{nim}(\theta_n) \quad = \quad P(Y_{nim}=1 \mid \theta_n, b_i) \quad = \quad \left(\prod_{j=1}^{m}\Psi(a_i w_{ij}(\theta_n - b_{ij})\right)\left(1-\Psi(a_i w_{im+1}(\theta_n - b_{im+1})\right) \qquad (9)$$

for $m = 0, \ldots, M_i-1$, and

$$P_{niM_i}(\theta_n) \quad = \quad P(Y_{niM_i} = 1 \mid \theta_n, b_i) \quad = \quad \left(\prod_{j=1}^{M_i} \Psi(a_i w_{ij}(\theta_n - b_{ij}) \right)$$

for $m = M_i$.

2.2.3. The GRM

An alternative, yet older, approach to modelling responses to polytomously scored items can is the model proposed by Samejima (1969). Allowing for scoring weights w_i , the probability of scoring in a response category j is given by

$$P_{nim}(\theta_n) \quad = \quad P(Y_{nim} = 1 \mid \theta_n, b_i) \quad = \quad \Psi(a_i w_{im}(\theta - b_{im})) - \Psi(a_i w_{im}(\theta_n - b_{i(m+1)})) \qquad (8)$$

for $m=1,\ldots M_i - 1$,

$$P_{ni0}(\theta_n) \quad = \quad P(Y_{ni0} = 1 \mid \theta, b_{ik}) \quad = \quad 1 - \Psi(a_i w_{i0}(\theta - b_{i1}))$$

for $m = 0$, and

$$P_{niM_i}(\theta_n) \quad = \quad P(Y_{niM_i} = 1 \mid \theta_n, b_{iM_i}) \quad = \quad \Psi(a_i w_{iM_i}(\theta_n - b_{iM_i})) \, ,$$

for $m = M_i$. To assure that the differences in Formula (8) are positive, the restriction $b_{i1} < b_{i2} <, \ldots, < b_{iMi}$ is imposed.

2.3 Multidimensional IRT and latent regression models

Lexter support the estimation and testing of a so-called between-items multidimensional IRT model. This entails that it is assumed that the complete item pool is made up of Q distinct subsets and each subset is modelled by a unidimensional IRT model. So θ_n is a vector with elements $\theta_{nq}, \;\; q = 1,\ldots,Q$, which are the Q ability parameters associated with the Q subtest. Response probabilities are defined as above, conditionally on relevant ability parameters, that is, as $P_{nim}(\theta_{nq})$. Further, it is assumed that the ability parameters $\theta_{nq}, \;\; q = 1,\ldots,Q$, have a Q-variate normal distribution with a mean-vector $\boldsymbol{\mu}$ with the elements μ_q, $q=1,\ldots,Q$, and a covariance matrix $\boldsymbol{\Sigma}$.

In MML and MCMC estimation, it is assumed that the ability parameters are sampled from one or more normal distributions. One way of viewing this is, is to assume that group membership is a dummy-coded covariate. One step further is to explain ability parameters by a general set of covariates. Usually, involving covariates is done in the framework of a unidimensional IRT model. As a generalization, it is also possible to consider a multivariate latent regression model in the framework of a multidimensional IRT model. Then the model becomes

$$\theta_{nq} = \sum_{k=1}^{K} \beta_{qk} X_k + \varepsilon_{nq} \ ,$$

where X_k, $k = 1,\dots,K$, are covariates and β_{qk} are regression coefficients. The residuals ε_{nq} are independent over respondents and have a Q-variate normal distribution with a mean of zero and a covariance matrix $\mathbf{\Sigma}$.

2.4 Estimation Methods

2.4.1 MML

MML maximizes a log-likelihood that is marginalized with respect to an assumed ability distribution. In the most general case, the multidimensional case with covariates, this log-likelihood is defined as a sum over respondents n of contributions

$$\log P(\boldsymbol{y}_n; b, \boldsymbol{\Sigma}, \beta) \quad = \quad \log \int \dots \int P(\boldsymbol{y}_n \mid \boldsymbol{\theta}_n, b) g(\boldsymbol{\theta}_n \mid \boldsymbol{\Sigma}, \beta) d\theta_{n1}, \dots, d\theta_{nQ}, \qquad (10)$$

where $P(\boldsymbol{y}_n \mid \boldsymbol{\theta}_n; b)$ is the probability of response pattern $\boldsymbol{y}_n$ given the ability and item parameters, that is,

$$P(\boldsymbol{y}_n \mid \boldsymbol{\theta}_n, b) = \prod_{q,i,j} P_{ij}(\theta_{nq})^{y_{nqij}} \ ,$$

and $g(\boldsymbol{\theta}_n \mid \boldsymbol{\Sigma}, \beta)$ is a multivariate normal density with a definition that may include covariates with regression coefficients β.

2.4.2 CML

Consider the partial credit model, that is, the model defined by Formula 7 with the discrimination parameter a_i fixed to one. Then $s_n = \sum_{i,j} w_{im} Y_{nim}$ is a sufficient statistic for the ability parameter. Conditioning on the value of this sufficient statistic results in a probability of a response pattern given by

$$P(\boldsymbol{y}_n \mid s_n; b) \quad = \quad \frac{\exp\left(-\sum_{i,m} w_{im} y_{nim} b_{im}\right)}{\sum_{\{\boldsymbol{z}_n \mid s_n\}} \exp\left(-\sum_{i,m} w_{im} z_{nim} b_{im}\right)} \ ,$$

where $\{ \boldsymbol{z}_n \mid s_n \}$ is the set of all response patterns $\boldsymbol{z}_n$ resulting in sufficient statistic value s_n. Maximizing these probabilities over the item parameters over all respondents results in the so-called CML estimates. Note that these estimates do not depend on assumptions regarding the ability distribution.

2.4.3 Bayesian MCMC

In Bayesian estimation procedures, the parameters in the model are seen as stochastic variables with a distribution, the so-called prior distribution. The likelihood function is multiplied with these priors to produce a so-called posterior distribution which is conditional on the data and proportional to

$$\prod_n P(\boldsymbol{y}_n \mid \boldsymbol{\theta}_n, b) g(\boldsymbol{\theta}_n \mid \boldsymbol{\Sigma}, \beta) p(b), p(\boldsymbol{\Sigma}) p(\beta) \,,$$

where the three right-hand factors are the priors for the item parameters, the covariance matrix of the ability parameters and the regression parameters, respectively. Convenient choices for these prior distributions can be found in Béguin and Glas (2001). The MCMC algorithm is based on iteratively simulating the model parameters. After a number of burn-in iterations, these draws constitute a sample from the posterior distribution. For further information refer to Albert (1992) and Béguin and Glas (2001).

3. Data collection designs

In the introduction of the previous chapter, it was shown that one of the important features of IRT is the possibility of analyzing so-called incomplete designs. In incomplete designs the administration of items to persons is such, that different groups of persons have responded to different sets of items. In the present section, a number of possible data collection designs will be discussed.

A design can be represented in the form of a persons-by-items matrix. As an example, consider the design represented in Figure 3.1. This figure is a graphical representation of a design matrix with as entries the item administration variables d_{ni} ($n = 1,...,N$ and $i = 1,...,$K) defined by Formula (3) in the previous chapter. The item administration variable d_{ni} was equal to 1 if person n responded to item i, and 0 otherwise. At this moment, it is not yet specified what caused the missing data. There may be no response because the item was not presented, or because the item was skipped, or because the item was not reached. In the sequel it will be discussed under which circumstances the design will interfere with the inferences. For the time being assume that the design was fixed by the test administrator and that the design does not depend on an a-priori estimate of the ability level of the respondents.

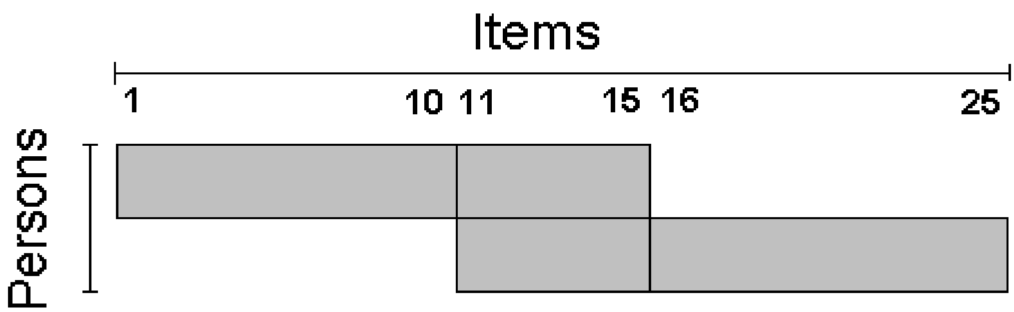


Figure 3.1 Design linked by common items.

In the example, the total number of items is $K = 25$. The design consists of two groups of students, the first group responded to the items 1 to 15, and the second group responded to items 11 to 25. In general, assume that B different subsets of the total of K items have been administered, each to an exclusive subset of the total sample of respondents. These subsets of items will be indicated by the term 'booklets'. Let I be the set of the indices of the items, so $I = \{1,...,K\}$. Then the booklets are formally defined as non-empty subsets of I_b of I, for $b = 1,...,B$. Let K_b denote the number of elements of I_b , that is, K_b is the number of items in booklet b. Next, let V denote the set of the indices of the respondents, so $V = \{1,...,N\}$, where N is the total number of respondents in the sample. The sub-sample of respondents getting booklet b is denoted by V_b and the number of respondents administered booklet b is denoted N_b. The subsets V_b are mutually exclusive, so $N = \Sigma_b N_b$.

To obtain parameters estimates on a common scale, the design has to be linked. For instance, the design of Figure 3.1 is linked because the two booklets are linked by the items 11 to 15, which are common to both booklets. A formal definition of a linked design entails that for any two booklets a and b in the design, there must exist a sequence of booklets with item index sets $I_a, I_{b1}, I_{b2}..., I_b$ such that any two adjacent booklets in the sequence have common items or are administered to samples from the same ability distribution. The sequence may just consist of I_a and I_b . Assumptions with respect to ability distributions do not play a part in CML estimation. So CML estimation is only possible if the design is linked by sequence $I_a, I_{b1}, I_{b2}..., I_b$ where adjacent booklets have common items.

Figure 3.2 Linking by common persons.

This definition may lead to some confusion because it interferes with the more commonly used terms "linking by common items" and "linking by common persons". Figure 3.1 gives an example of a design linked by common items because the two booklets have common items. Figure 3.2 gives an example of a design that is commonly labeled "linked by common persons". The definition of a linked design applies here because the first and second booklet have common items and the second and last booklet have common items. Further, the first and last booklet are linked via the second booklet.

An example of linking via common ability distributions is given in Figure 3.3. Again, common items link the middle two booklets. The respondents of the first two booklets are assumed to be drawn from the first ability distribution and the respondents of the last two booklets are assumed to be drawn from a second ability distribution. It must be emphasized that, in general, designs linked by common items are far preferable to designs that are only linked by common distributions, since the assumptions concerning these distributions add to the danger that the model as a whole does not fit the data. Assumptions on ability distributions should be used to support answering specific research questions, not as a ploy for mending poor data collection strategies.

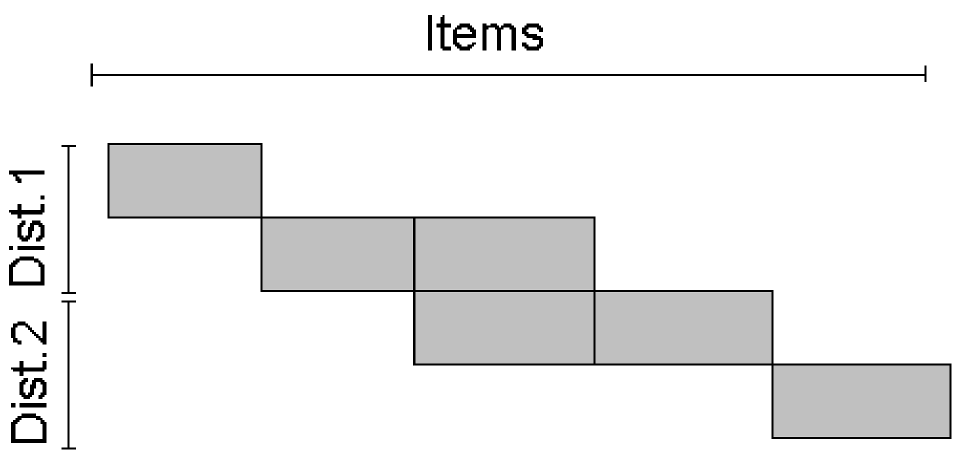


Figure 3.3 Linking by a common distribution.

4. The structure of the data file

The program Lexter is suited for analyzing incomplete designs and handling missing data. The data can be organized in two ways. We will start with the recommended one. As an example consider Figure 4.1. Suppose that there are $K = 10$ items in the design and 2 booklets. The first booklet contains the items 1-5 and 7-8. So the students administered this booklet responded to 7 items. The second booklet contained the items 1-2, 6, 9-10. So the students administered this booklet responded to 5 items. Every record in the data file pertains to a student. Note that there are 10 students. The columns 1-2 contain the booklet number. Note that there are 2 booklets. Further, the columns 3-5 contain an optional student ID. The student IDs must be integer valued. They are echoed on files containing the student's ability scores. The columns 7-13 contain the item responses. In the example, the responses are scored between 0 and 3. A 9 stands for a missing response. Only the responses to items figuring in a booklet are entered into the data file. For the first booklet, the program is informed that the columns 7-11 pertain to the items 1-5, and the columns 12-13 pertain to the items 7-8. In the same manner, for the second booklet, the columns 7-11 pertain to the items 1, 2, 6, 9 and 10, in that order. The order of the booklet numbers, the student IDs and the response vector in the data file is not fixed. That is, the booklet numbers may be in any position in a record, the IDs may also be in any position, and the response vector may start in any position. Further, the data file may also contain the values of covariates, also in any position.

Position:

1	2	3	4	5	6	7	8	9	10	11	12	13
	1			1		1	0	2	1	3	0	9
	1			2		0	0	2	1	0	0	0
	1			3		0	1	0	3	2	1	0
	1			4		0	0	9	1	3	3	3
	1			5		1	0	0	0	2	0	2
	2			6		2	0	0	0	0		
	2			7		1	1	1	9	3		
	2			8		3	3	3	3	3		
	2			9		2	1	2	1	2		
	2		1	0		0	0	0	0	0		

Figure 4.1

Recommended format of a data file

An alternative way of organizing the data file is depicted in Figure 4.2. The person and booklet IDs are in the same positions. However, the program is now told that both booklets consisted of the same 10 items and the design is handled by entering the missing data code 9 for the items not responded to. The reason for recommending the first format is efficiency of computer storage. The computational procedures are generally not affected.

Position:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1			1		1	0	2	1	3	9	0	9	9	9
	1			2		0	0	2	1	0	9	0	0	9	9
	1			3		0	1	0	3	2	9	1	0	9	9
	1			4		0	0	9	1	3	9	3	3	9	9
	1			5		1	0	0	0	2	9	0	2	9	9
	2			6		2	0	9	9	9	0	9	9	0	0
	2			7		1	1	9	9	9	1	9	9	9	3
	2			8		3	3	9	9	9	3	9	9	3	3
	2			9		2	1	9	9	9	2	9	9	1	2
	2		1	0		0	0	9	9	9	0	9	9	0	0

Figure 4.2

Alternative format of a data file

The following points should be kept in mind:

- The booklet number has the same position in all records. It is a positive integer. However, the booklet numbers need not be consecutive. A booklet number is always required, even if there is only one booklet. The records pertaining to a specific booklet need not be consecutively grouped together.
- In every record, the booklet number, the (optional) person ID and the data should start in the same column.
- The item responses are integers. They are unweighted item scores, starting from zero.
- Blanks in the data file are interpreted as zeros.
- The missing data code in the examples above is 9, but all entries which are above the value for the highest category M_i are considered missing.

5. Running the program

5.1. Introduction

Lexter is a new and extended version of MIRT (Glas, 2010). A new shell has been developed in ShinyR, Together with the MIRT files, the program consists of four packages

Program	Program file	Functionality
ShinyLexter	LexterSetup_*version*.exe	Interface for defining design and settings and running analyses with LEXTER applications in a stand-alone installation package
Lexter_MML	LEXTER.EXE	Stand-alone application MML estimation and testing
Lexter_CML	LEXTER_CML.EXE	Stand-alone application CML estimation and testing
Lexter_MCMC	LEXTER_MCMC.EXE	Stand-alone application MCMC estimation and testing

5.2. Installation

The following steps can be taken to download and install ShinyLexter

1. Download the most recent version of ShinyLexter from https://www.shinylexter.com
2. Unzip the .exe file and install the program by running LexterSetup.exe
3. In the installation steps, check the box "Create a desktop shortcut"
4. Continue by pressing "Next" and "Install" and the installation starts,

The following steps can be taken to run the program Lexter after successful installation.

1. Go to your desktop and double click Lexter desktop icon
2. New window should open with ShinyLexter in Google Chrome portable
3. Note: The first time the program opens the start-up process can take some time, because R is checking and loading all packages in the background.

Note: The location of the full Lexter installation can be found with the Windows file explorer command %LocalAppData%\Lexter\ This is also the location where the TTT files can be found.

The standard flow of control for a Lexter run is as follows: the ShinyLexter user interface is used to define the data structure and the computations. When all definitions are entered, clicking the button “Run Lexter” engages the computational modules. After the computations, the results are displayed in the ShinyLexter user interface.

It is also possible to use the three computational packages without the ShinyLexter user interface. Running computations with the stand-alone versions is only recommended to advanced users who might want to create batch-jobs. The method to achieve this is explained in the next section.

5.2.1 Running without the ShinyLexter App and running in Batch

When a run is defined in the ShinyLexter user interface, the definitions are written in a file RUNNAME.TTT, which is located at the installation site %LocalAppData%\Lexter. This file functions as the command file for the computational modules, which are also located at that site. The format of the file RUNNAME.TTT is outlined in Appendix A. The computational modules echo RUNNAME.TTT as RUNNAME.TTT0 at the work directory of the user (for details see next paragraphs).

If a run file RUNNAME.TTT or RUNNAME.TTT0 is available, either issued by the user interface or created from scratch in a text editor, a computations module (say LEXTER.EXE) can also be run directly without the user interface. There are several options.

- Get the path of the installation, say PATH=%LocalAppData%\Lexter. Enter PATH\LEXTER RUNNAME on the command line to run the program. The output of the program is written to the work directory defined in the file RUNNAME.TTT or RUNNAME.TTT0.
- Copy the computational package needed to a new directory, say D:\Lexter. the path of this directory should not contain blanks! (This will also hold for the run files and output files created by the program). So do not put the software on your desktop. Run the program from the new location from the command prompt, say D:\Lexter\LEXTER RUNNAME, or, if the computational module and the file RUNNAME.TTT or RUNNAME.TTT0 are in the same location, by entering LEXTER RUNNAME. Again, the output of the program is written to the work directory defined in the file RUNNAME.TTT or RUNNAME.TTT0.

A batch run can be created by building a batch file, say JOB.BAT, where every line defines a run, so for instance:

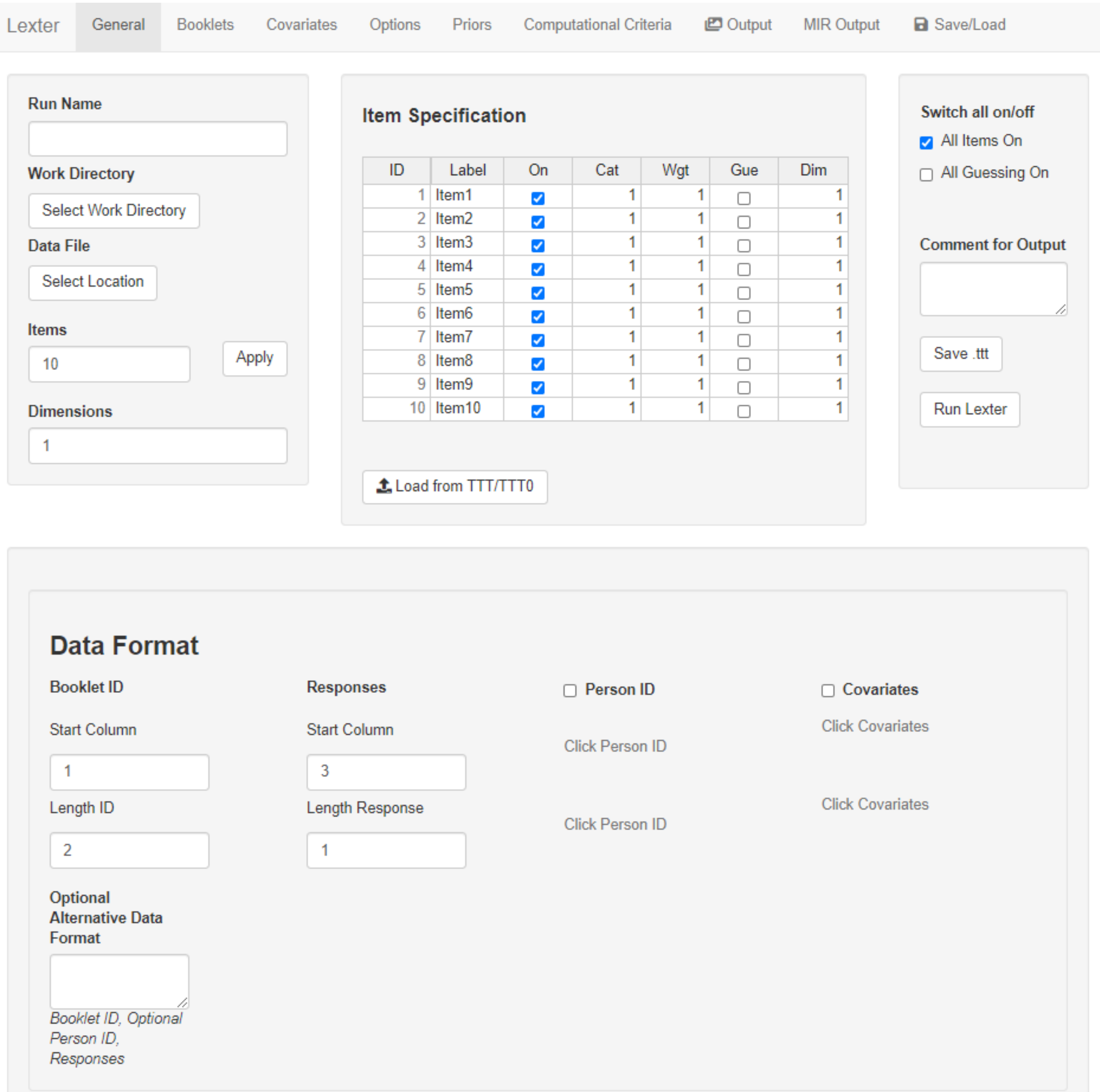
```
LEXTER RUNNAME1
LEXTER RUNNAME2
LEXTER RUNNAME3
```

defines three separate runs which are executed sequentially.

5.3. Starting with the ShinyLexter app

5.3.1 Open ShinyLexter

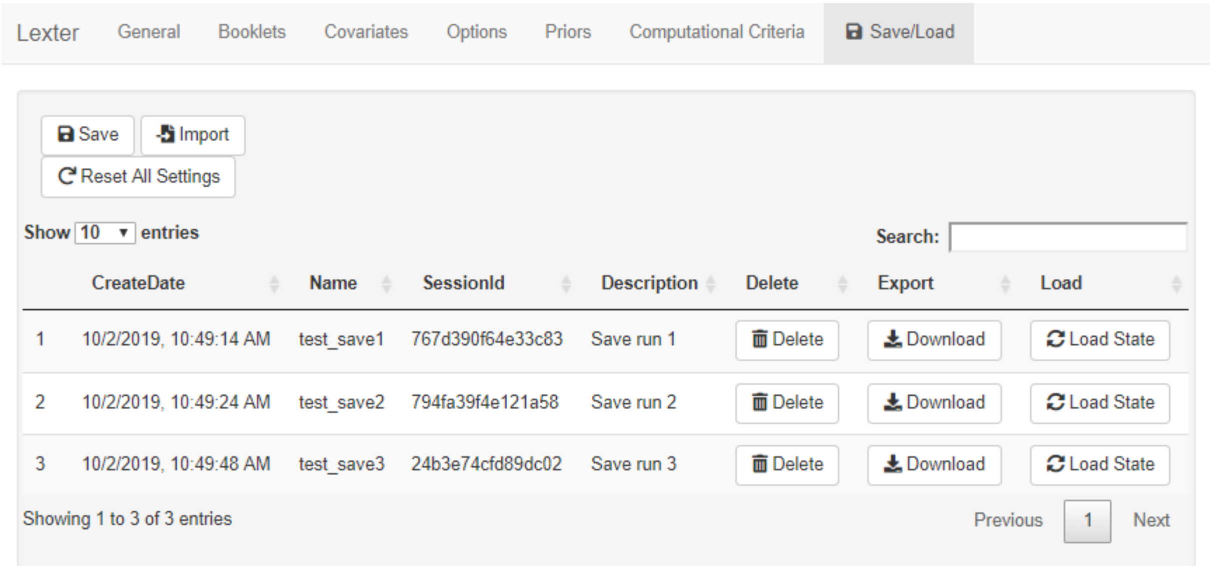
To start the ShinyLexter app, open Google Chrome and start the desktop app. Google Chrome (portable browser) now opens a screen with different tabs. The main tab is named "General" and should immediately open and load the following screen.



After starting off the app, there are four possibilities. One can start with specifying a completely new run. The other three options involve loading previously specified files, such as earlier runs or loading a newly specified booklet design in Excel.

Load saved state from .LEX file

The second option is to load an existing ShinyLexter run and edit it in the browser. The previously saved runs on this pc, can be found directly under the Save/Load tab. This tab also provides the possibility to "Load" a previously saved state, which means a previously saved run opens with the settings as saved in the so-called Shiny state.



Under this tab, there are several buttons:

- *Save*: which can be used to save the current state locally
- *Load State*: which can be used to load one of the previously saved states
- *Download*: once a state is saved, it can be downloaded. The file appears in the directory where the downloads are usually stored. The address is also displayed bottom left on your screen. This is necessary if you want to share the state with other users or other PC's. A .lex file is downloaded once this button is pressed.
- *Import*: after you have received a .lex file from another pc or another user, it can be imported by using the "import" button. Once imported, the state appears in your list and you may load the state with the "Load State" button.

In summary, depending on your goals, you can decide to save or load a state or to download and import a state. The latter is important when sharing files across different folders and PC's. Note that it is also possible to save states during the process of defining a run, for your own reference. Further, it may also be convenient to create a "basis" .lex file to define settings such as the work directory, as a starting file for a number of new run definitions.

The .lex files are summary files containing the state of a previous run. When you work locally and save states, these states are only saved on your PC and cannot be shared with others. This is where the .lex files come in. When you have saved a state, and you want to share this state, you press “Download” and a .lex file is created. When you share this file with others, or you receive a .lex file from someone, you can import these files into your PC and use them in your ShinyLexter.

Load from TTT or TTT0 file

The third option is to load an existing TTT or TTT0 file, with the button “Load from TTT/TTT0” below the item specification table (see screenshot below). The TTT0 file is generated in a previous run and can be used as a blue print for defining the next run. If the file is created from scratch, it is very important to make sure the TTT to be loaded is in the exact format as specified in the appendix. Press “Load from TTT(0)” button and select the .TTT file to be loaded into Shiny.

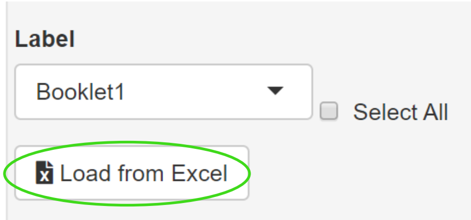
Item Specification

ID	Label	On	Cat	Wgt	Gue	Dim
1	Item1	☑	1	1	☐	1
2	Item2	☑	1	1	☐	1
3	Item3	☑	1	1	☐	1
4	Item4	☑	1	1	☐	1
5	Item5	☑	1	1	☐	1
6	Item6	☑	1	1	☐	1
7	Item7	☑	1	1	☐	1
8	Item8	☑	1	1	☐	1
9	Item9	☑	1	1	☐	1
10	Item10	☑	1	1	☐	1

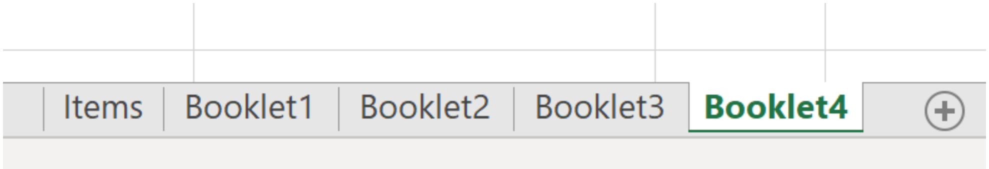
Load from TTT(0)

Load from Excel

Finally, it is possible to load a design from an Excel file with .xlsx extension.



One can create an Excel file with one tab specifying the items, and per Booklet a tab specifying the Item design. It is important to add an ID and label to the items. The first tab should be called “Items” and the next tabs can be called by the booklets names, for instance the following structure in Excel will give us the specification of 4 booklets:



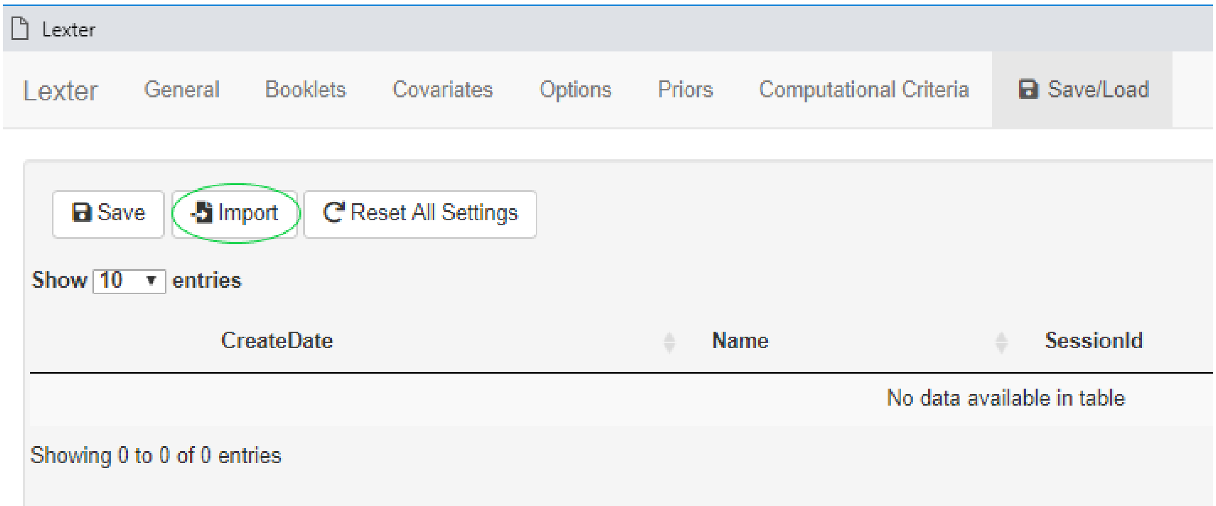
The first sheet named “Items” should contain the columns “ID” and “Label” and will specify the items in the design. The other sheets with the Booklets names will have the columns “ID”, “Label” and “Selected”. To correctly specify the design, the first sheet will contain all the items that are available in the dataset. Then, each sheet for each booklet contains the ID and Label of the items that are available in the dataset for each specific Booklet in the dataset. Finally, the column “Selected” is an indicator with the possible values 0 and 1. The value in this column indicates if the item has to be selected in the design. For an example of the standard XLSX format that reads in correctly, please check ShinyLexter.com for this file.

In summary, the following sheets and columns of the .XLSX file have these effects in the ShinyLexter app:

- Sheet “Items” changes Item Specification in the ShinyLexter app
- Sheets “Name_of_Booklet1”,“Name_of_Booklet2”…”Name_of_BookletB” will change the Booklet tab in the app:
    - Column Label adds these Items to the design of each “Name_of_Booklet”
    - Column Selected causes the item to be selected or not in Booklet specification

5.3.2. Sharing files: Importing in ShinyLexter

When you have received a file with a.lex extension, this file can be "imported" in ShinyLexter. See in the green circle below where to find the Import button in ShinyLexter, then follow these steps:



1. Open the ShinyLexter program and go to the Save/Load tab
2. If you are currently working in ShinyLexter, do not forget to save a state before loading a new state
3. Press "Import" Button
4. Find the .lex file you want to import
5. Select the file and wait patiently, the imported state is now imported and should appear in the table below
6. Once imported, press the button "Load state" in the right column of the table
7. Check the other tabs to find the new settings after loading the state.
8. Do not forget to check and if needed change all the data file and work directory locations to an existing location on your own PC

5.3.3. Sharing files: Downloading in ShinyLexter

When you want to share one of your settings in the ShinyLexter app with others, you can download a copy of your saved state (.lex file) to share with others. Follow these steps to download and share your settings:

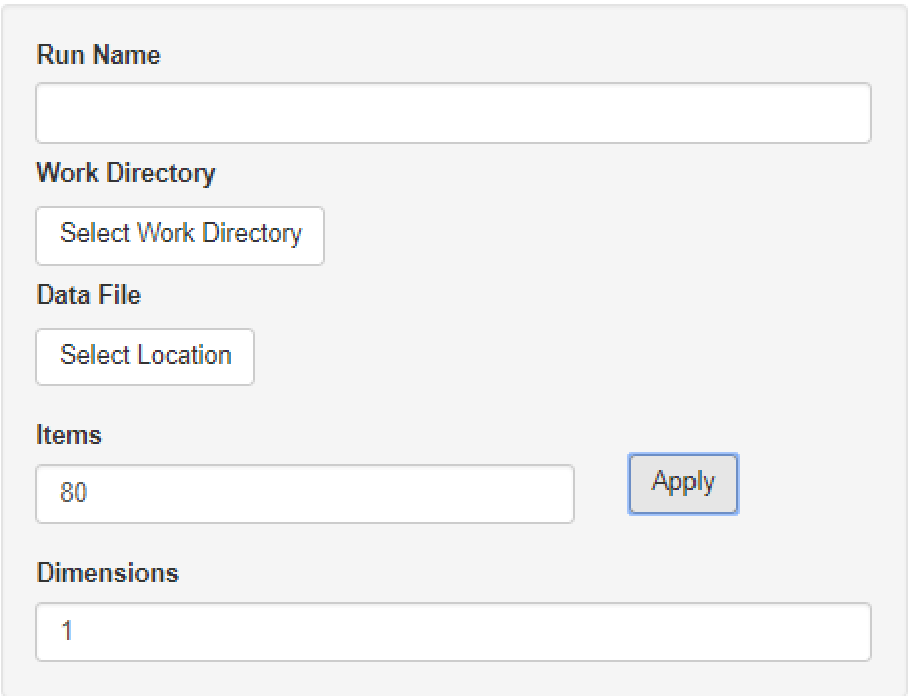
1. Go to the Save/Load tab and make sure to save the current state, give it a name and description
2. This state appears in the table below

3. Press the download button next to the row of the saved state you want to download:
4. The .lex file is saved into your normal Downloads folder
5. You can now share this file with others and import the .lex file on other PC's or within other sessions.

5.4. Defining a run - General tab

The general tab can be used to define a run or to adjust an existing run. After selecting the run, the various options are filled as follows.

5.4.1. General: Main specifications



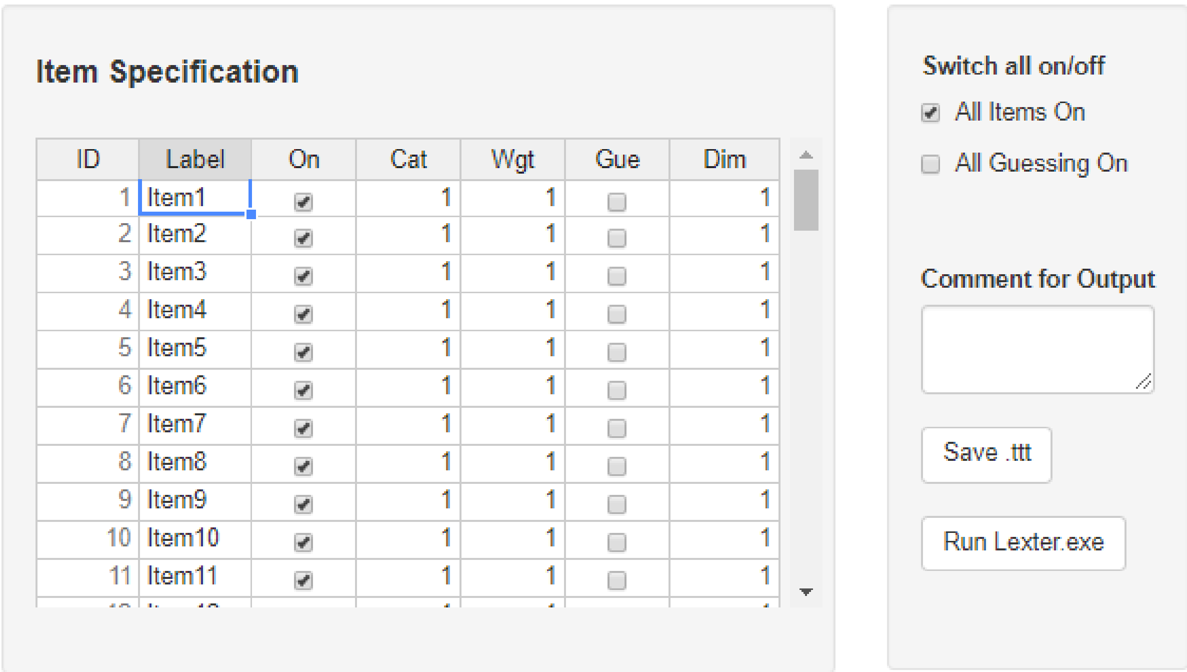
In the left panel, the main specifications for the run can be added. In this example 80 items are added, by pressing the “Apply” button. Note that the “Apply” button must always be pressed before continuing with entering further definitions or importing item labels.

The fields in <**General: Main specifications**> have the following meaning.

Field	*Description and Input*
Run Name	Name for this run, with a maximum of 20 characters. Spaces in the run name will automatically become underscores.
Work Directory	Select the directory where run and output files should be saved
Data File	Name of the data file. Name and path cannot contain blanks. Data file can be selected by clicking the button.

#Items	Total number of items present in the design, labeled K.
#Dim	Number of dimensions. For unidimensional analyses, set equal to 1. For between-items multidimensional analysis choose a number between 2 and 4.

5.4.2. General: Item Specification



After having pressed Apply, the number of items in the table under “Item Specifications” updates accordingly by adding rows for the specified amount of items. For each of the items, settings such as their Label can be modified. The shell has the capability to handle copying and pasting. This is also an efficient manner to change the contents of this table. Its functionalities are comparable to Microsoft Excel.

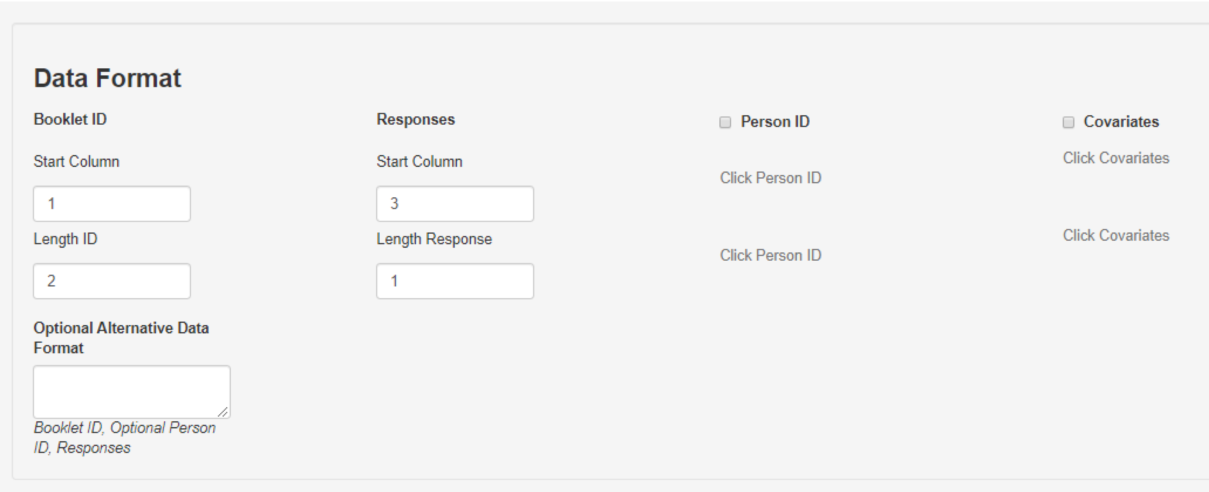
The table in <**General: Item specifications**> contains these adjustable columns, and mean the following:

Field	*Description and Input*
ID	Item number. The items are labeled $i = 1,\ldots,K$. Items with a higher ItId are accessed by scrolling down.
Label	Item label. The default labels are displayed. The labels can be edited. The maximum number of characters is 20.

On	By using the checkbox <**On** >, items can be selected or ignored in an analysis. Deselecting an item overrides selections within booklets defined in the **Booklets** tab. So by setting an item to “off”, the item may be present in the data of certain booklets, but the item will be ignored in the analysis. With the checkbox “All items on”, all of the items can be switched on/off at the same time.
Cat	Number of response categories minus the zero category. The item responses are coded $m=0,\ldots,M_i$. Note that *M.* is the number to be entered here. So for dichotomous items, $M_i=1$.
Wgt	Item scoring weight. An integer between 1 and 8. This weight is defined in the general model as displayed in (20) and (21) as w_k.
Gue	A toggle <**0/1**> to add a guessing parameter to an item to impose the 3PLM for that item. With the checkbox “All guessing on”, all of the guessing parameters can be switched on/off.
Dim	Dimension on which the item loads. This can also be ‘0’, meaning all dimensions.
Comment for Output	Any comments on this run can be added to this textbox.
Export .ttt	This button will save a Run file called RUNNAME.TTT, which can be used to call Lexter.EXE
Run Lexter.exe	This button will perform a call to Lexter.exe; the desired model will be estimated with the settings as saved in the above .ttt file.

An example of using the property of copying and pasting in the “Item specification” table, is setting all Guessing parameters to 1. Set one of the guessing parameters to 1 and copy this field, then select all other fields, and paste. Labels can easily be copied and pasted from another program (i.e. Excel) into this Interface as well.

5.4.3. General: Data Format



The data format of the selected ‘Data File’ has to be provided to the program in the fields of <**General: Data Format**>, which makes the program capable of distinguishing between the various columns in the file.

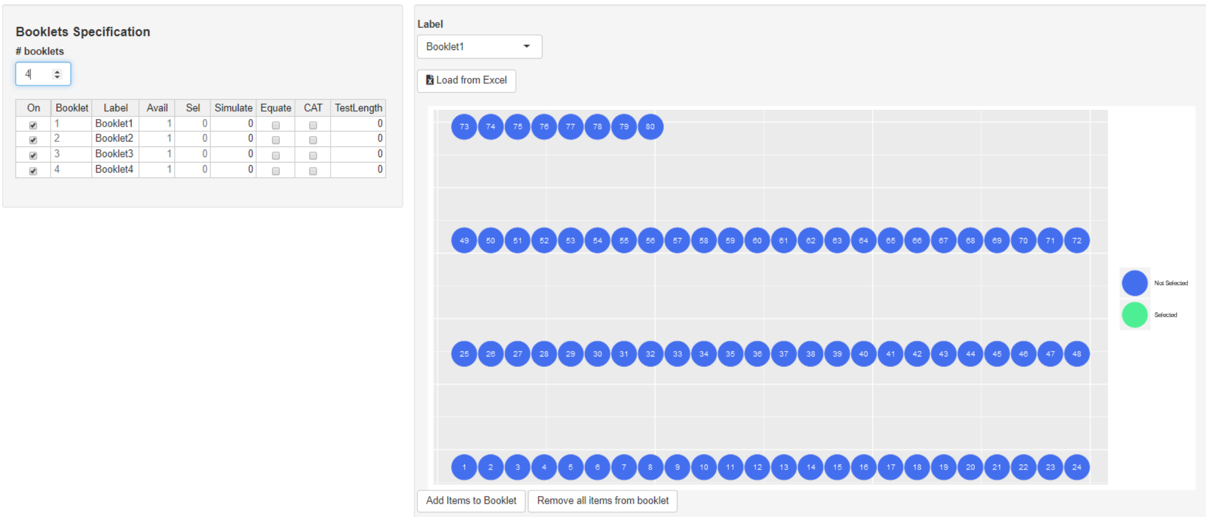
The following options are available:

Field	*Description and Input*
BookletID	Position and length of the Booklet ID in a record.
Data	Position of the first item response and the number of positions for each item response
Person ID	If checkbox is ticked, the position and length of the person ID. This person ID is echoed in files with person parameter estimates. The person ID is a character string. The maximum number of characters is 20.
Covariates	If checkbox is ticked, the position of the first covariate in the data. On the tab “Covariates” it is possible to specify length of each covariate.
Optional Alternative Data Format	Here the advanced user can write a Fortran format. If there is any text in this box, the program will use this text as the data format. The definitions in the d as described above will be overwritten, except for the flag that indicates the presence of a person ID. The motivation for the options is that the definitions in the standard Data Format box cannot handle all possibilities for a format. For instance, a format (I2,T4,A4,T12,10I2,T50,2I1,T22,2I4) would indicate a format where the first 2 positions are the booklet

	number, positions 4 to 7 the person ID, positions 10 to 38 the first 10 items in the Item Specification filed, positions 50 and 51 the next two items in the Item Specification field, and the positions 22 to 29 are for the last two items in the Item Specification field. All numbers must be right-aligned.

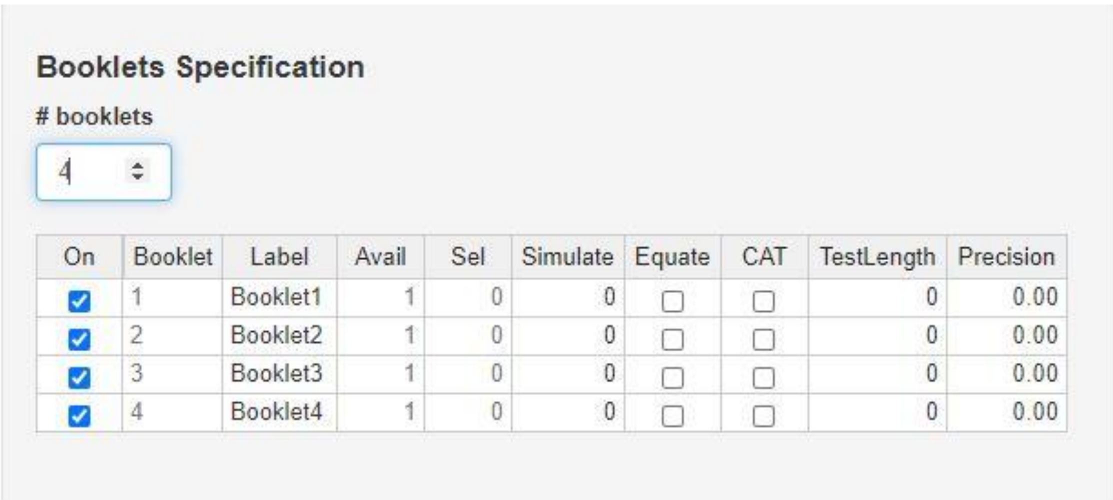
5.5. The Booklets tab

The next tab contains options to specify the booklets characteristics and add items to these booklets. Basically, the design of the tests will be defined here.



5.5.1. Booklets Specification

The names Tests and Booklets have the same meaning in this manual. In the left-hand panel, the example contains 4 booklets.

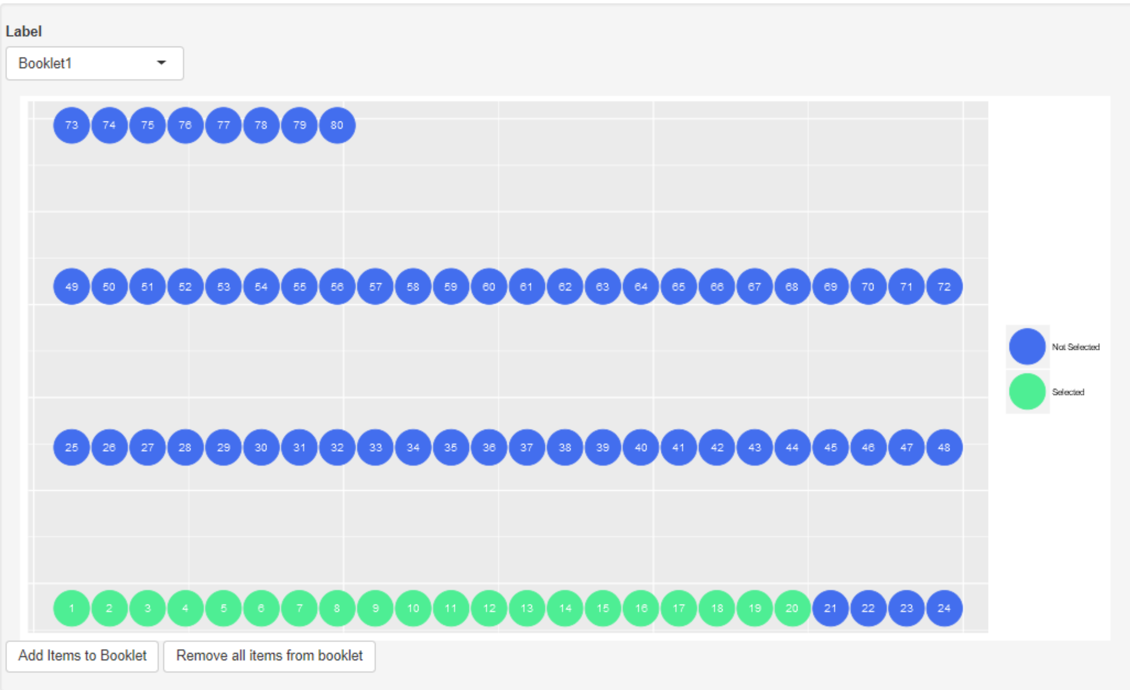


On	Booklet	Label	Avail	Sel	Simulate	Equate	CAT	TestLength	Precision
☑	1	Booklet1	1	0	0	☐	☐	0	0.00
☑	2	Booklet2	1	0	0	☐	☐	0	0.00
☑	3	Booklet3	1	0	0	☐	☐	0	0.00
☑	4	Booklet4	1	0	0	☐	☐	0	0.00

The booklets labels can be edited directly in the table, in the column <**Label**>. The maximum number of characters for the label is 20. Also, booklets can be switched on or off with **<On>**. In addition, one can add simulations for the booklets under <**Simulate**> and indicate which for which two booklets the linking error must be computed under <**Equate**>. The procedure for computing this error can be found in Glas (2019). The book containing the relevant chapter is open source and can be downloaded via https://link.springer.com/book/10.1007%2F978-3-030-18480-3. The columns <**CAT**> and <**TestLength**> pertain to simulation data simulation and will further be explained in section 5.5.4 below. The other two columns <**Avail**> and <**Sel**> are automatically changed through adding items in the panel on the right.

5.5.2. Adding items to booklets

By selecting the ID numbers of the items, items can be added to the booklets. In the present example, the number of items for Booklet 1 equals 20. This number cannot exceed the total number of items, defined on the **General** tab as, in this case, 80.



The area with the blue dots, see screenshot above, gives the opportunity to define which items are present in each booklet. This is done by dragging and dropping, so holding the mouse and drag over the area one wants to select. Also, adding items in a certain order, must correspond to the order in which they appear in the booklet.

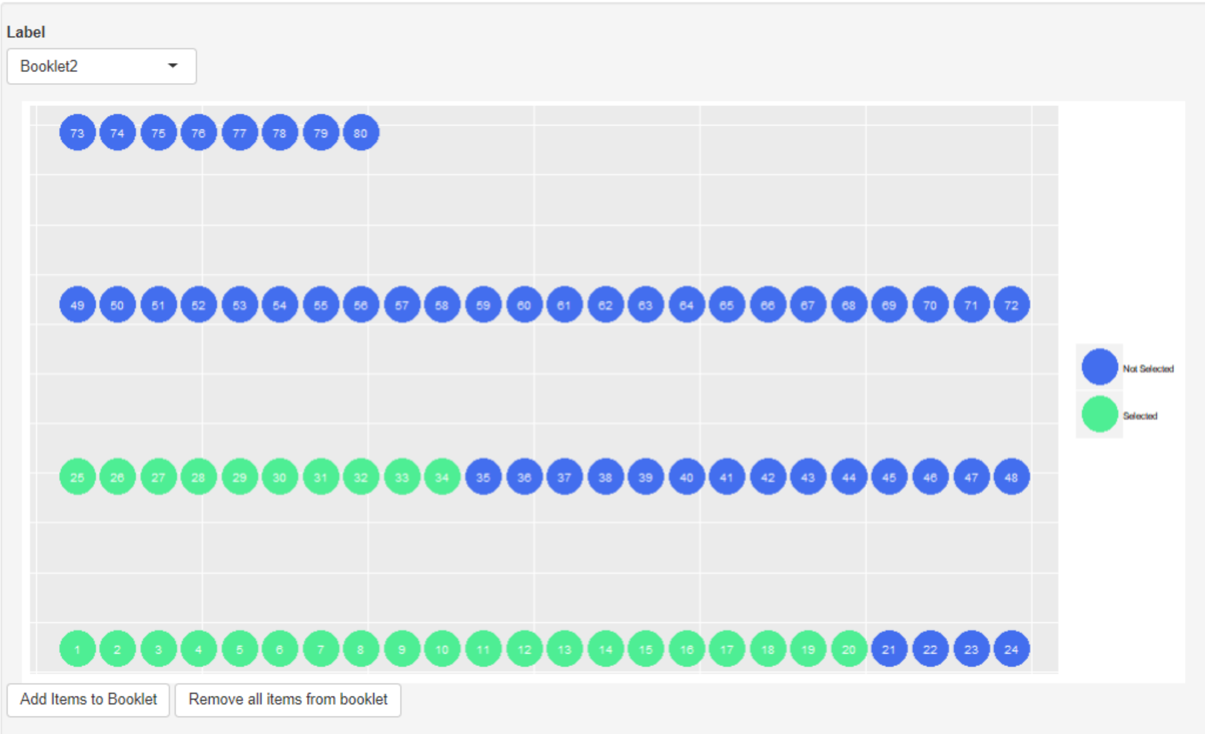
First, select the Booklet for which items will be specified under “Label”. The order in which the items are added is determined by when the button “Add items to booklet” is pressed. First select the items that are supposed to be added - by means of drag and drop -, and then press “Add Items to Booklet”. As soon as items are added to the booklet by pressing this button, the table containing columns “Avail” and “Selected” items will update, as well as the table stating information on the items updates.

5.5.3. Contents of booklets

This table automatically appears when items are added to a booklet.

All items are automatically selected but can be switched off for a specific analysis by clicking on them. In the below example we see the settings for Booklet 4, in which 15 items have been added, but 1 items has been switched off (*Item70*). Under the "Sel" column the amount of items selected is shown. There is now also the possibility to select and unselect all visible rows at once by using the buttons "+*Vis*" and "-*Vis*". The latter unselect all items that are shown below and the "+*Vis"* button selects them all. The *"Inv"* button can be used the inverse the selection made.

When items are selected through dragging, their color changes from Blue (*Unselected*) to Green (*Selected*) in the dot graph below. Once the button "Add items to booklet" is pressed, the items will remain Green and the settings are saved for this Booklet. Once a mistake is made in adding booklets or in the order of the items, all the items from the booklet can be removed by pressing "Remove all items from booklet". Be careful, this is a reset button for one booklet.



Switching between the booklets is possible by changing the Booklet name under "Label".

Example: Reversing order of items in booklet

Adding items with ID's 10-12 first and then item 8-9, will cause the booklet to consist out of

5 items and the order being 10,11,12,8,9. Items can be present in the reversed order, by selecting and adding sequentially.

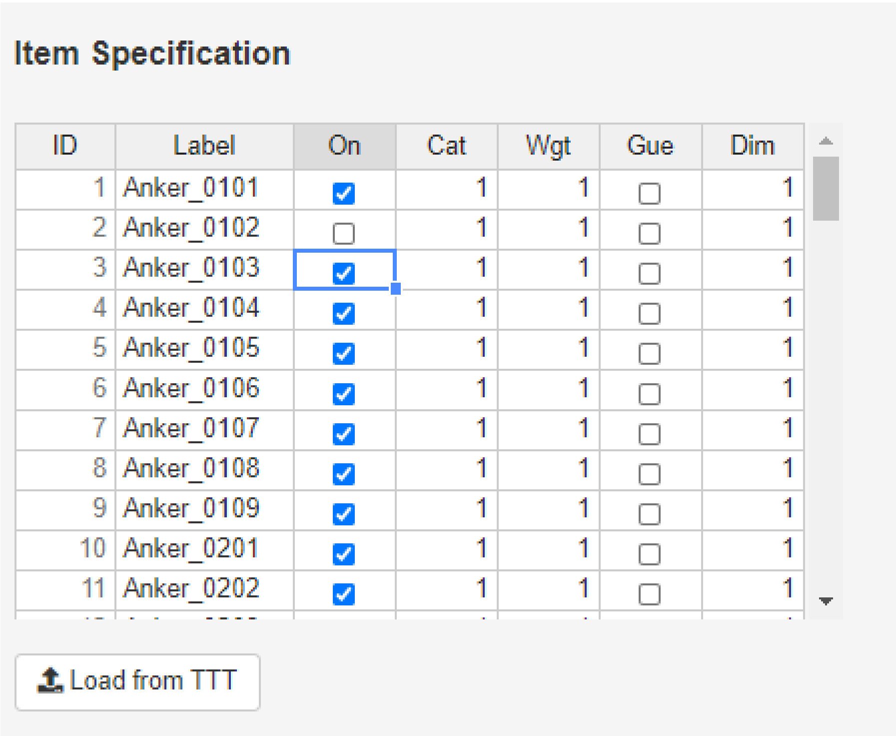
Example: Only one booklet

In the case of only one booklet, there is the option of adding all items to the booklet with just one mouse-click. If there is only one booklet, press the *"Select All"* checkbox next to "*Label*", and all items will automatically be added to the booklet.

Difference in Local and Global item setting

It is possible to (un)select items and specify the design of tests globally and locally. For the global option, the user can for example switch off an item in all booklets by using the checkboxes under "Item Specification" in the general tab. This is efficient if you want to switch off an item in all booklets.

Global Checkbox



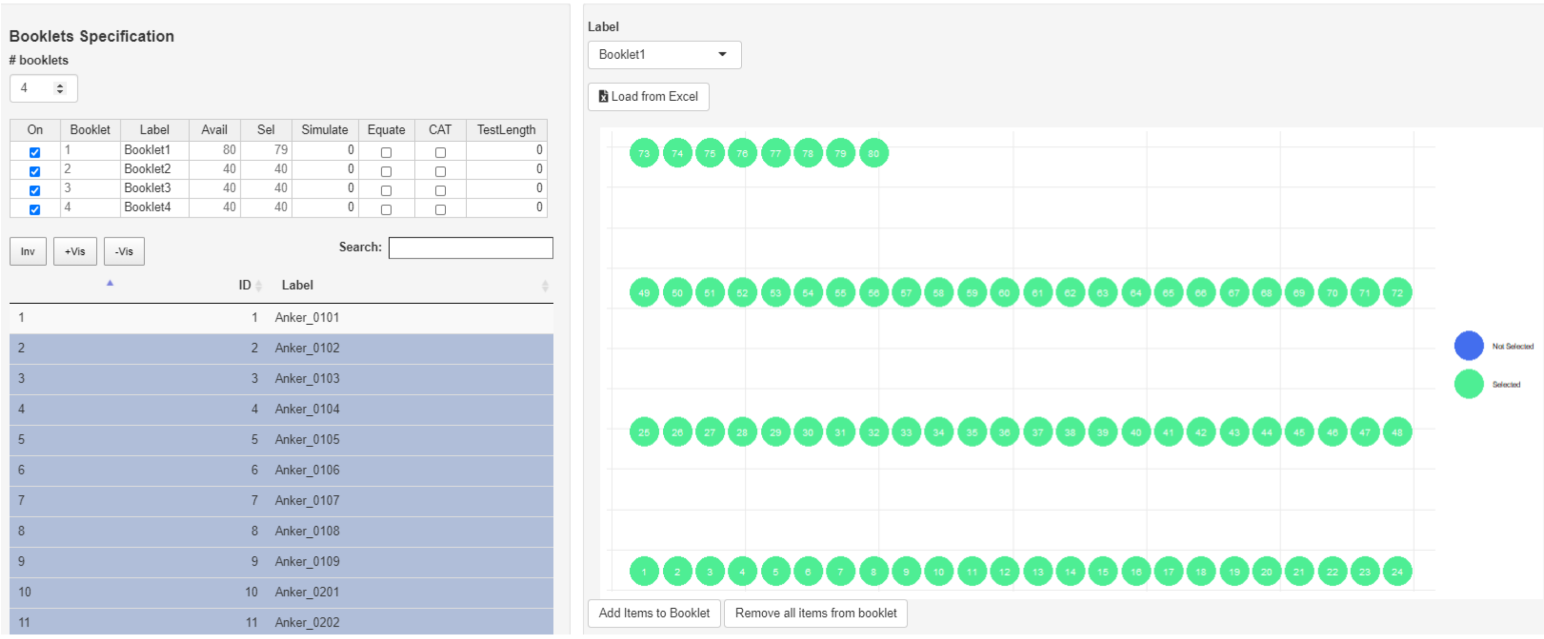
Item Specification

ID	Label	On	Cat	Wgt	Gue	Dim
1	Anker_0101	☑	1	1	☐	1
2	Anker_0102	☐	1	1	☐	1
3	Anker_0103	☑	1	1	☐	1
4	Anker_0104	☑	1	1	☐	1
5	Anker_0105	☑	1	1	☐	1
6	Anker_0106	☑	1	1	☐	1
7	Anker_0107	☑	1	1	☐	1
8	Anker_0108	☑	1	1	☐	1
9	Anker_0109	☑	1	1	☐	1
10	Anker_0201	☑	1	1	☐	1
11	Anker_0202	☑	1	1	☐	1

Load from TTT

The global checkbox will also be automatically switched on or off once the local selection of the item in all booklets is on or off.

The local checkbox is handled in the Booklets tab. Locally means "within a booklet":



In the screenshot above, item 1 has been switched off locally, only in Booklet 1, by unselecting the row of item Anker_0101 on the low left panel list of items for Booklet 1. If in all booklets item 1 was to be switched off, it would cause the “On” column on the General tabpanel of that item to be switched off as well. This checks for local booklet settings, and if all are off, the item will be switched off globally.

5.5.4. Data Simulation

It is possible to simulate data and run analysis with a customized configuration of items in a Booklet. To do this, first specify all existing booklets and then add an extra booklet. Define the desired item structure of this booklet that needs simulated data and analysis. Then, add a certain amount of simulations under the column “Simulate”, in the example below this is set on 10000 respondents for simulation. A maximum of 99999 simulation records is allowed.

Booklets Specification

\# booklets

5

On	Booklet	Label	Avail	Sel	Simulate	Equate	CAT	TestLength	Precision
☑	1	Booklet1	20	20	0	☐	☐	0	0.00
☑	2	Booklet2	25	25	0	☐	☐	0	0.00
☑	3	Booklet3	30	30	0	☐	☐	0	0.00
☑	4	Booklet4	20	20	0	☐	☐	0	0.00
☑	5	Booklet5	22	22	1000	☐	☐	0	0.00

Running the program will create the usual output with the specified data set, followed by the simulation using the results of the estimation procedure. So the simulated data do not influence the estimation, but the simulated data can be used in a subsequent run. A data set RUN.SIM1 will created using the 1PLM, and, if requested in the run definitions, a data set RUN.SIM2 will be created for the 2PLM/GPCM. These data files have the same format as the original data set. Several issues are of interest.

1. Item and/or population parameters may be fixed via the parameter file (see sections 5.7.1 and 5.10) or freely estimated.
2. A booklet can be specified that may be empty in the original data set but can still be generated. If it contains items that are not estimated in the run, they are either fixed in the parameter file or generated.
3. If one ability distribution is specified for all booklets, this distribution is also used for the simulation. In case every booklet has a unique ability distribution, new unique booklets can be either given population parameters via the parameter file, or they are given an ability distribution with a mean and variance averaged over all other booklets.

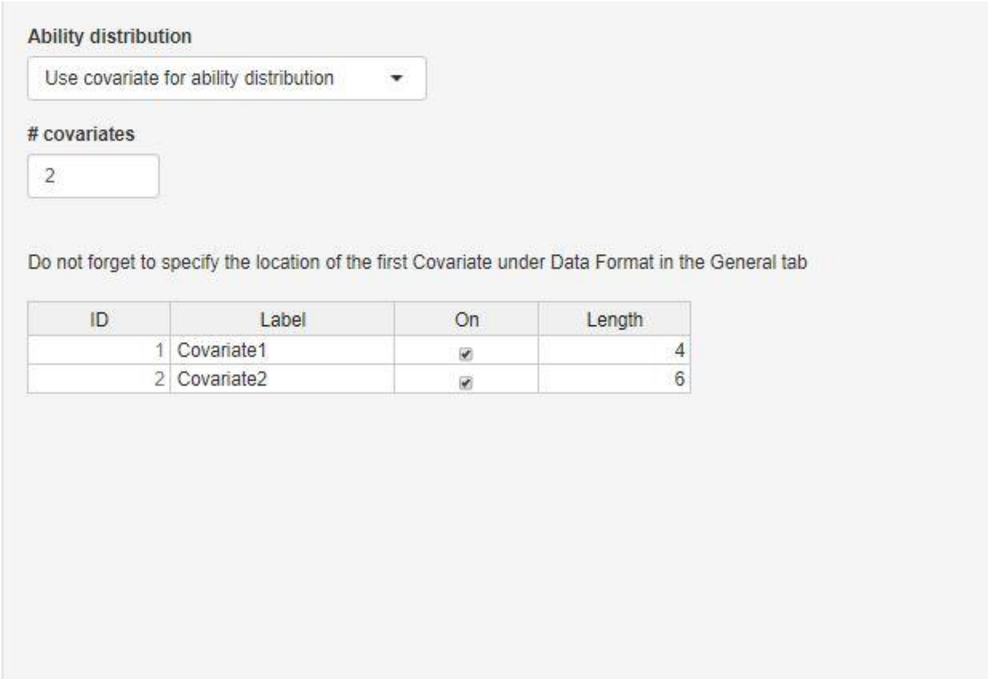
The column <**CAT**> is an indication which can be used if a CAT has to be simulated. The value 0 indicates the default and the value 1 indicates a CAT must be simulated. It is then assumed that the booklet is the item bank from which tests are drawn. Further, the parameters are usually fixed via a file RUNNAME.PAR (another name may also be used), but the parameters may also be estimated or drawn during the simulation run. For the simulation, ability parameters are drawn from the specified population parameters, that is, fixed parameters, estimated parameters or a standard normal distribution. Item are selected using the maximum likelihood principle. The ability estimates are either by WML or EAP, depending on the ability estimation procedure specified for the run, where WML has priority over EAP, and WML is also used if no estimation procedure is specified.
Either a fixed length or a variable length test can be generated.

- For a fixed length test, for every simulated respondent, a CAT of length <**TestLength**> is assembled.
- For a variable length test, items are selected until a certain precision is attained. The precision is specified in the column <**Precision**>. The precision is the ratio of the variance of the ability estimate (the squared standard error) and the ability variance in the assumed population. It is a number between 0.00 and 1.00.

If the <**TestLength**> is set to zero or to a number larger than the item bank, a variable length CAT is assembled.

5.6. The Covariates screen

In the covariates screen, one can indicate which covariates are present in the data and which ones should be used for analysis.



Field	*Description and Input*
Ability Distribution	Options: • one normal ability distribution for all booklets • every booklet a unique ability distribution • use covariates in ability distribution, with one residual distribution for all booklets
# covariates	Specify the total number of covariates present in the data

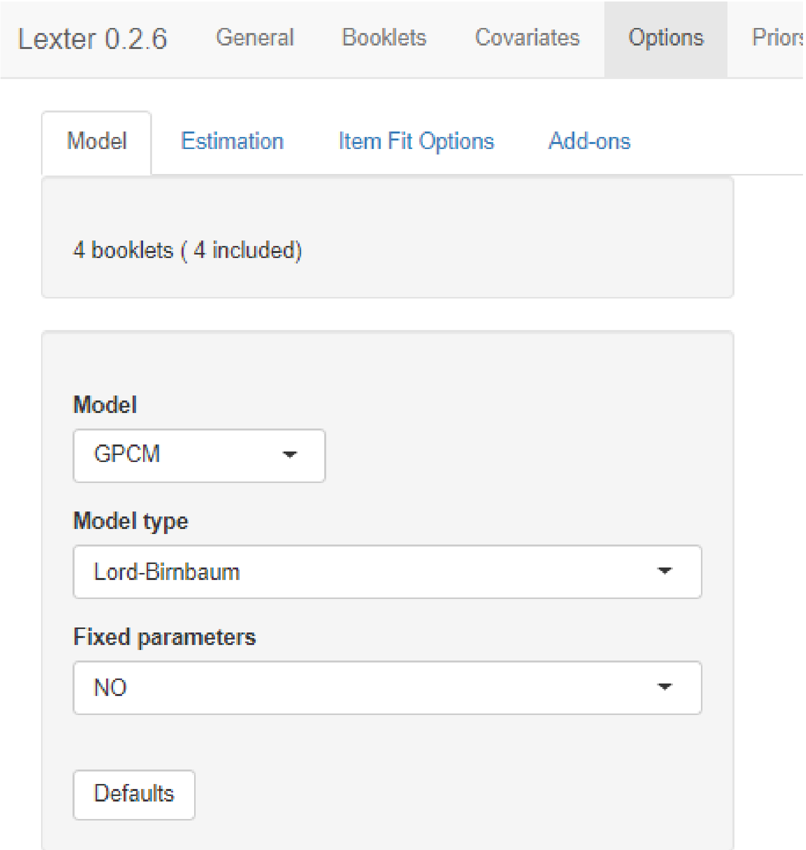
The table which appears when covariates are specified, gives room to the user to switch them off and on for the analysis and to specify the number of positions each covariate uses in the specified dataset. The maximum number of characters for the Covariate label is 20.

5.7. The Options screen

The options screen contains 4 tabs: Model, Estimation, Item Fit Options and Add-ons.

5.7.1. Options: Model

On the first tab <**Model**> the estimated model can be set with the following options:

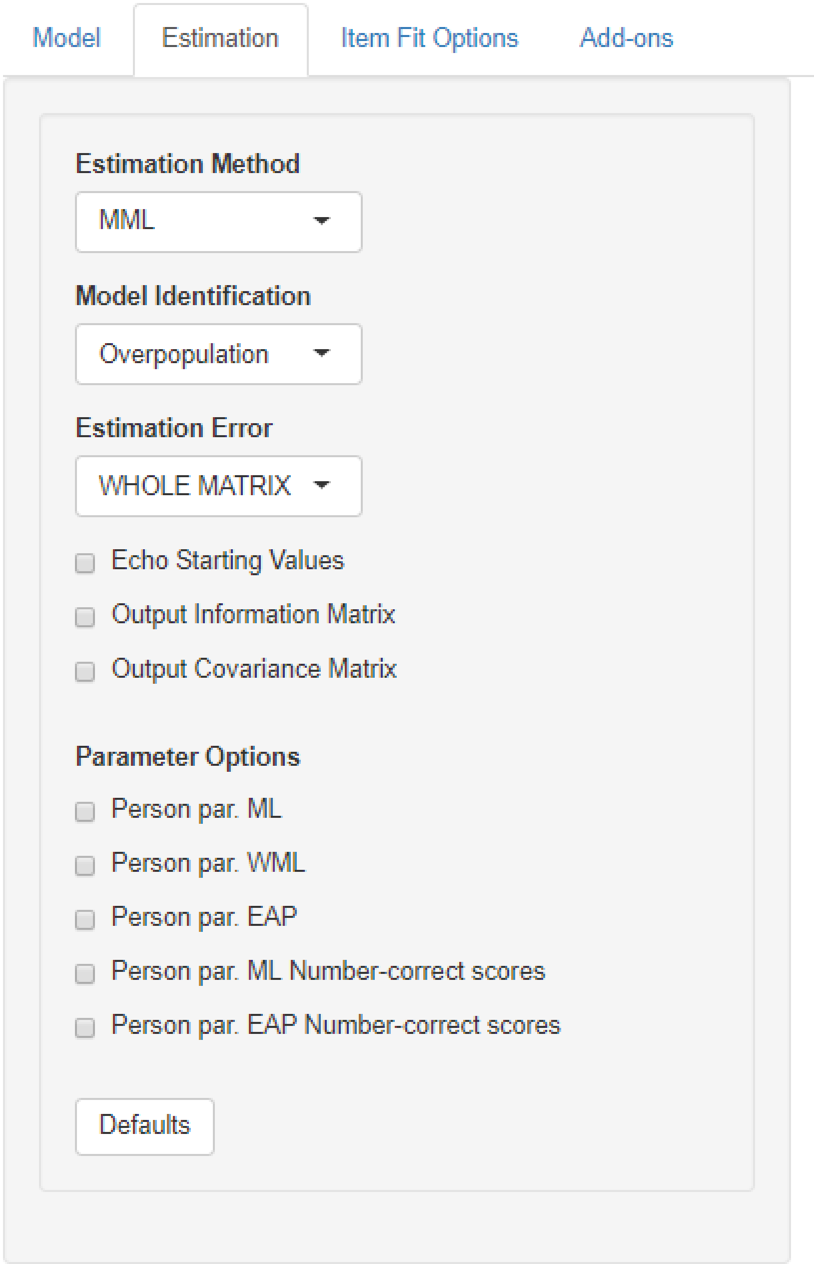


Field	Options	Remarks
Model	**GPCM**	GPCM, 2PLM in case of dichotomously scored items, 3PLM if items with guessing parameter specified in **General** screen.
	GRM	GRM
	SeqM	SEQM
	None	If NONE is selected, only basic statistics are given. This functions a good test run. None of the other options are relevant

Model Type	**Lord-Birnbaum**	Output of 2PL model
	Rasch	Output of 1PL model
Fixed parameters from file	**NO**	If fixed parameters from file = NO, nothing happens
	YES	If fixed parameters from file = YES, you are requested to specify the location of the file with the fixed item and/or population parameters and these parameters will be fixed accordingly. The parameter file has the same format as the output file RUNNAME.PAR and is described in section 5.10.

5.7.2 Options: Estimation

The Estimation tab has multiple options to set the methods for estimation, identification and options for the person parameters to be estimated. These options will be explained in the table.

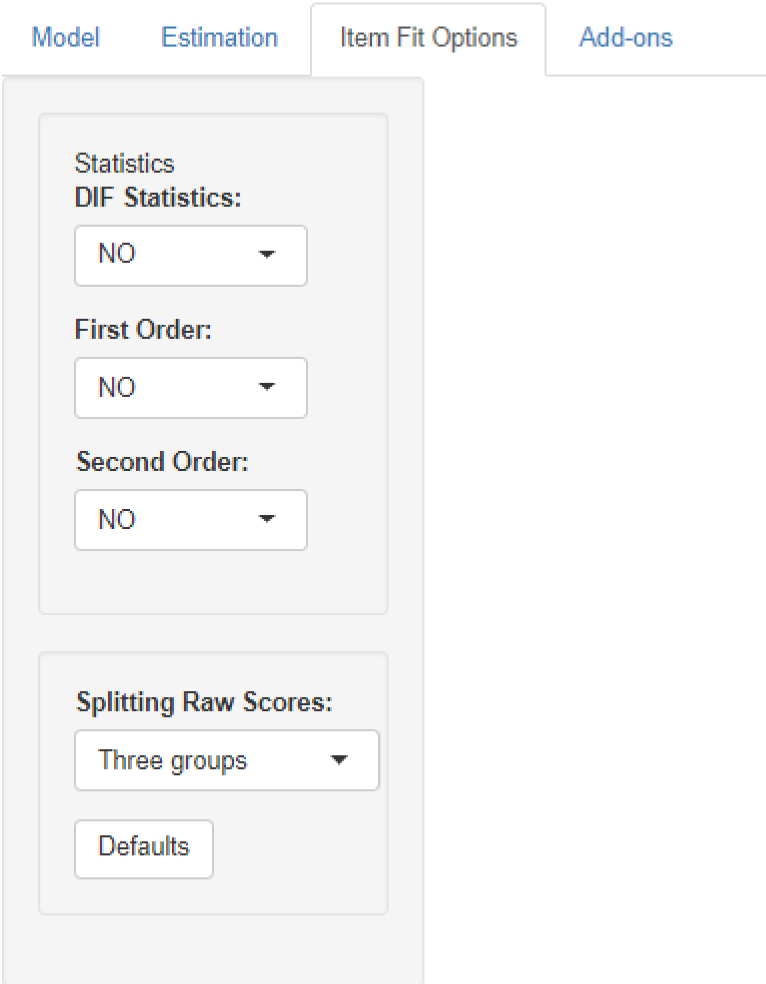


Field	Options	Remarks
Estimation	**MML**	MML estimation, starting with 1PLM/PCM and continuing with 2PLM/3PLM/GPCM if requested so in previous tab under <**Model Type**> in previous tab

	MCMC	Bayesian MCMC procedures
	CML	CML estimation
Model identification	**Overpopulation**	The sum of the means of the ability distributions is set to zero and (for 2PLM, etc.) the product of the variances is set to one.
	Last Population	The mean and variance of the ability distribution of the last booklet are set to zero and one, respectively.
	Items	
Estimation Error	**Whole matrix**	The complete information matrix is used to compute the standard errors of the parameter estimates and the weight matrix of the Lagrange multiplier fit statistics
	Diagonal only	Only the block-diagonal of the information matrix is used to compute the standard errors of the parameter estimates and the weight matrix of the Lagrange multiplier fit statistics. Each block pertains to an item.
Echo Starting Values	**Yes/No**	Echo the initial item parameter values computed taking logits of item response probabilities.
Output Information Matrix	**Yes/No**	Only if the size of the matrix is no more than 20 x 20. Output to RUNNAME.COV1 and RUNNAME.COV2.
Output Covariance Matrix	**Yes/No**	Only if the size of the matrix is no more than 20 x 20. Output to RUNNAME.COV1 and RUNNAME.COV2.
Person par. ML	**Yes/No**	ML estimation of ability plus LM person fit statistics as described in Glas and Dagohoy (2007).
Person par. WML	**Yes/No**	WML estimation of ability as described in Warm (1989).
Person par. EAP	**Yes/No**	EAP estimation of ability.

5.7.3 Options: Item Fit

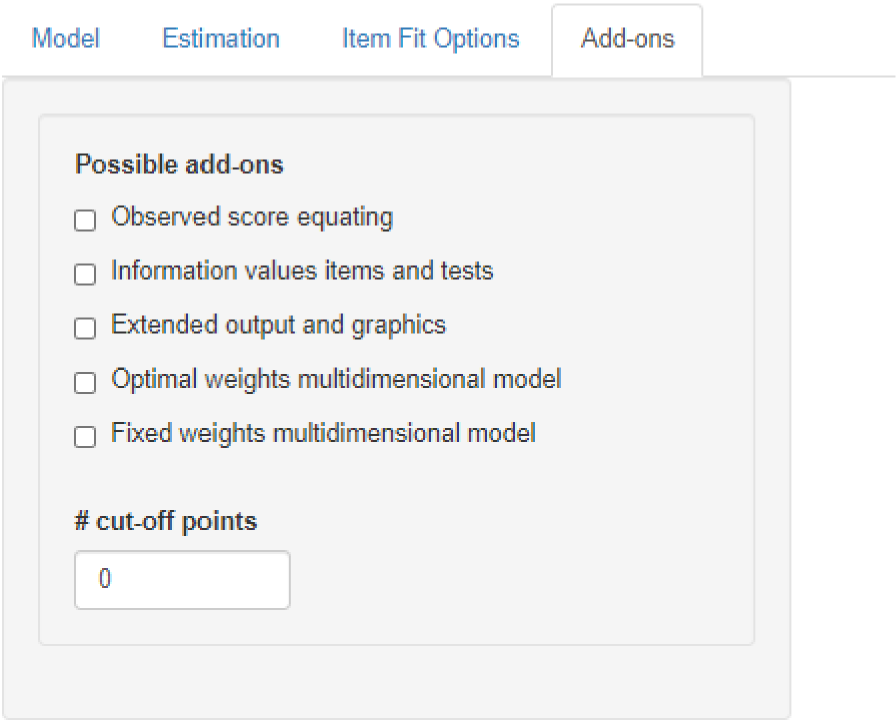
Lexter supports a number of item fit statistics. The output produced by the statistics will be commented upon in Section 5, where the output of the program is described. Here we only give an overview.



Field	Options	Remarks
Dif-statistics	**No/Booklets/Covariate**	Computation of Lagrange multiplier test statistic targeted at differential item functioning across booklets or a selected covariate. Reference, Glas (1998).
First Order	**None/R-form/Q-form**	Computation of Lagrange multiplier test statistic targeted at the form of the item response curve. Reference, Glas (1988),

		Glas (1999), Glas & Suárez-Falćon, (2003). Checking this box in combination with “Extended output and graphics” will create Figures on Observed vs Expected Response proportions.
LR-statistics	**Yes/No**	Andersen’s likelihood ratio test statistic, Rasch model only. Reference: Andersen (1977).
Second Order	**None/R-form/Q-form**	Computation of Lagrange multiplier test statistic targeted at local independence. Reference, Glas (1988), Glas (1999), Glas & Suárez-Falćon, (2003).
Splitting	**Raw Scores**	First order statistics based on subgroups formed using their raw scores, default is 3 groups. The allowed range is between 2 and 6.

5.7.4. Options: Add-ons

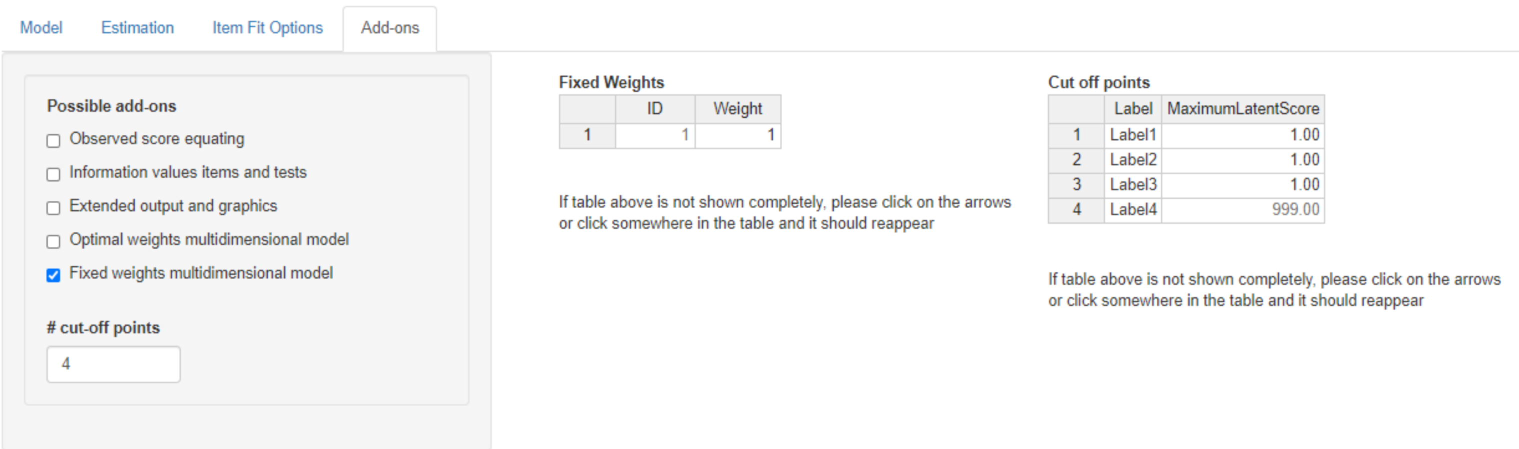


Options	Remarks
Observed score equating	For every combination of a booklet and a population, the frequency distribution of observed scores is computed. This supports observed score equating (see, Zeng and Kolen, 1995)
Information values items and tests	Computes item and test information values.
Extended output and graphics	When the checkbox “Extended output and graphics” is being checked, a separate graphics module is called to display plots of first order item fit information and/or information curves, if either of these is requested. This graphics module is explained in detail in section 5.7.5.
Optimal weights multidimensional model	Estimates a linear combination of dimensions of a multidimensional model that optimizes the reliability of the thus weighted ability estimate.

Fixed weights multidimensional model	When the option “Fixed weights multidimensional model” is checked, a fixed weights table will be shown as below. In this table it is possible to manually add preferred weights per dimension (the total number of dimensions can be specified under the General tab).
#cut-off points	The same holds for the cut-off points. If you want to request information on the scores regarding a certain latent-score, one can specify these cut-off points here manually and give them a label. Since these are cut-off points, and there are more categories than cut-off points, the final cut-off point is fixed at 999. Please be advised that these scores must be typed in increasing order (lowest cut-off point first).

5.7.5 Extended output & graphics

It is possible to generate graphics in ShinyLexter. The tab Options contains the option <**Extended output and graphics**>.



If the user checks this box in combination with the checkbox <**Information values item and tests**> and/or the checkbox of <**First order statistics**>, this causes the graphical module of ShinyLexter to automatically generate and save graphs. These .PNG files will be saved in a folder in the specified Working Directory called <runname-GRAPHICS>, among with the datafiles these graphs are based upon. Since the datafiles used to generate the graphs are available as well, the user could also create their own graphs with the raw data. The datafiles are described below. The graphs that can be automatically generated are Item Information Curves, Test Standard Error curves and Observed vs Expected Response Patterns graphs. If these graphics are asked for, the program will create and save these graphs in a folder in the specified Working Directory called <**runname-GRAPHICS**>.

Overview of actions

1) Checkbox: Information values items and tests

If this checkbox is on, the program generates *.IN1 datafiles for every item in the run for Rasch and *.IN2 datafiles for every item in the run for 2PL (if Lord-Birnbaum modeling is chosen). It also generates one datafile for each model containing Test Information, these files are called "RUNNAME.IN1" for Rasch and/or "RUNNAME.IN2" for 2PL. These files provide the information required for the graphics module, but they can also be used to create plots with other applications, such as Excel.

2) Option: First order statistics = YES

If First order statistics is set to the value "YES", the program generates a datafile called "RUNNAME.LM1" for Rasch and "RUNNAME.LM2" for 2PL.

3) Checkbox: Extended output and graphics

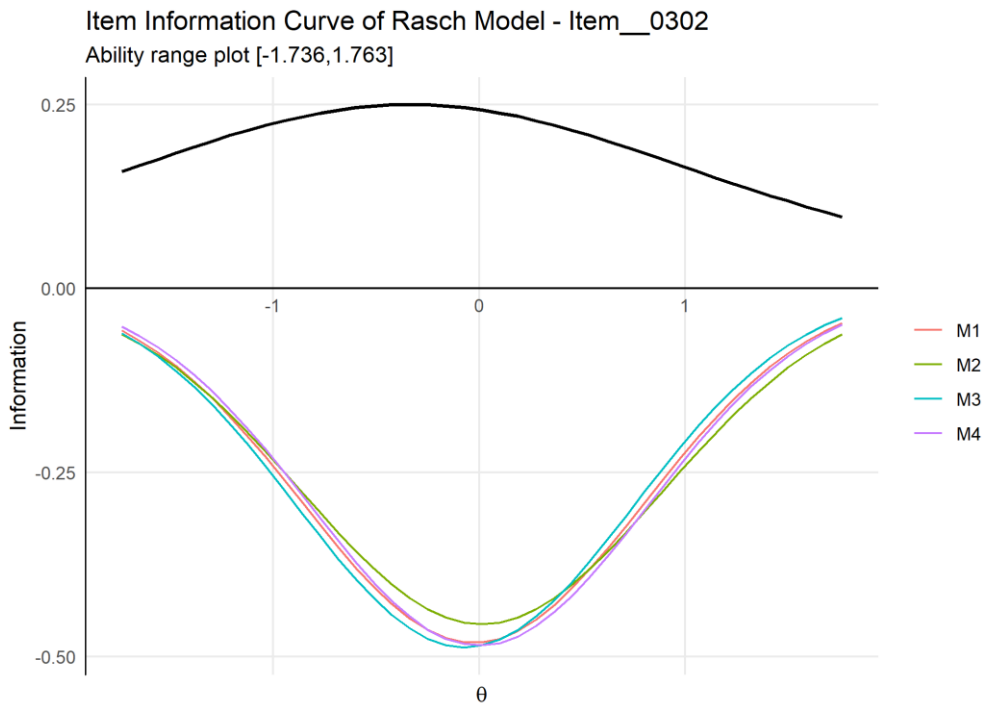
If this checkbox is on, the program checks which of the above two options 1 or 2 are on and will automatically start creating and saving graphs in the background with help of the created datafiles.

All the files described above can be found in the folder called <**runname-GRAPHICS**> in the Working Directory. It is also possible that you do not get any graphs when the third box of <**Extended output and graphics**> is not checked, you will then only see the raw datafiles of (1) and (2) in the <**runname-GRAPHICS**> folder. This would save time, as saving the graphs could require extra calculation time.

After Lexter has finished calculating and saving, one can go to the tab called "Output" and if specific graphs are requested, the graphs are shown in the ShinyLexter app. Another possibility is to view the graphs locally in the <runname-GRAPHICS> folder.

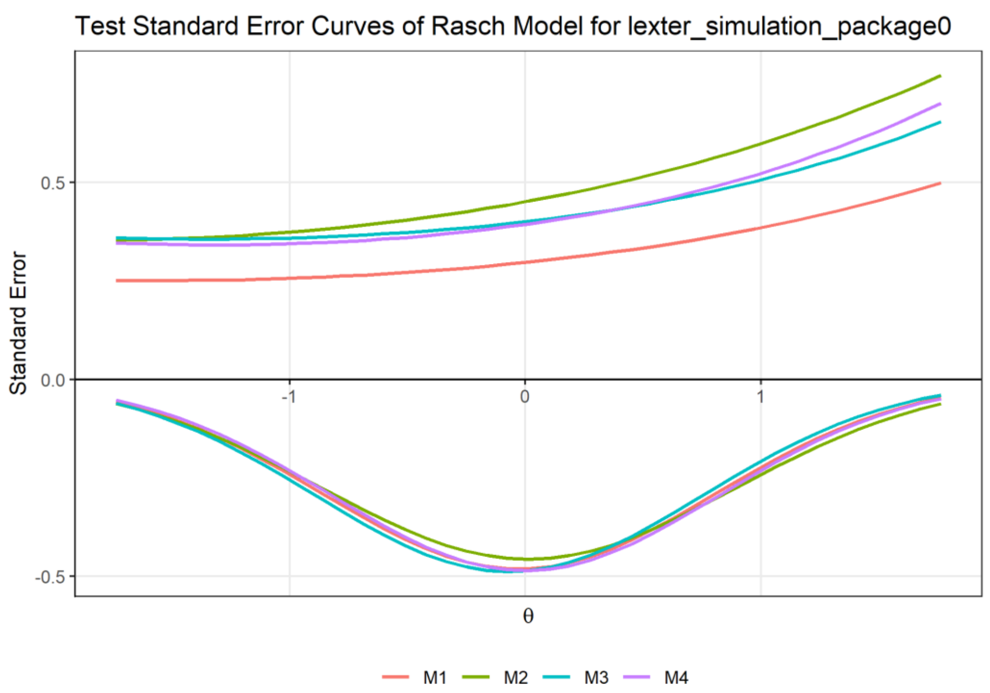
There are three different types of graphs. The Item Information Curves per item, the Test Standard error curves and Observed vs Expected Response Patterns graphs per item. Three examples of each type of graph are shown below.

Example of graph type 1: Item Information Curve



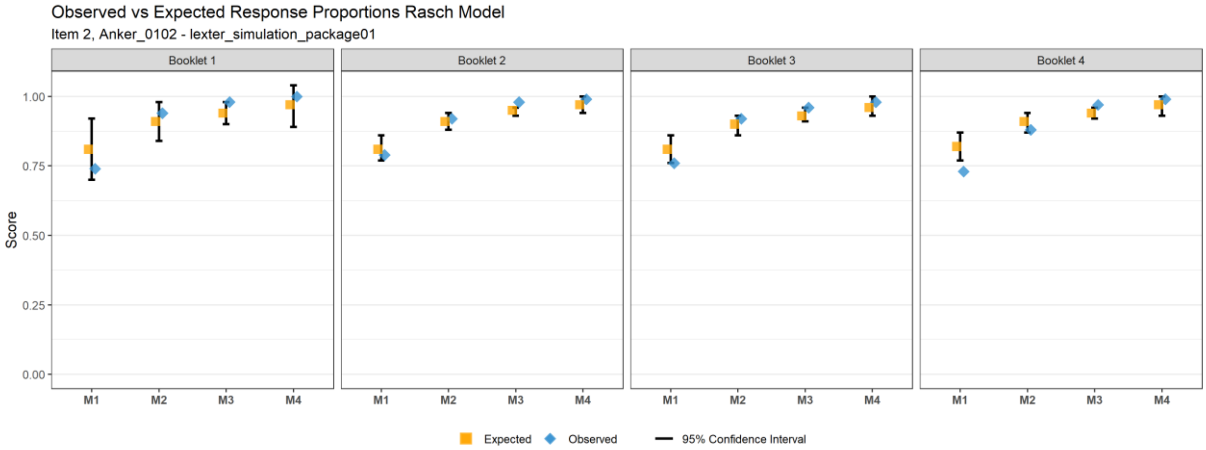
Above the x-axis we notice the item information and below the x-axis an inverted theta ability distribution is plotted per Booklet or for the whole ability distribution, depending on the choice of Ability distribution. In the plot above, the different colors indicate the ability distributions for four different booklets. The scale of the x-axis is variable based on the input from the *IN1 en *IN2 files

Example of graph type 2: Test Standard Error Curve



The Test's standard error is plotted in graph 2, with again the inverted theta distributions per booklet below the x-axis.

Example of graph type 3: Observed vs Expected Response Patterns as fit statistics

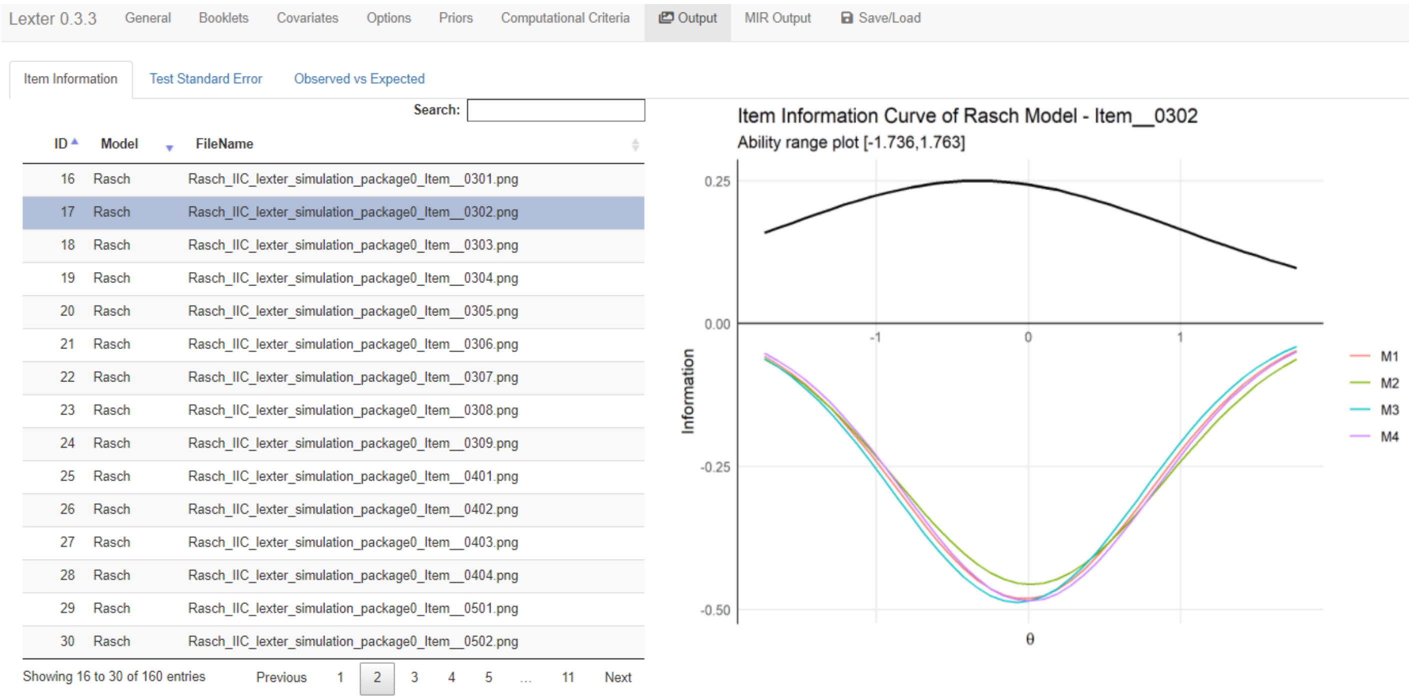


Depending on the model choice, plots for only Rasch or for both Rasch and 2PL are generated.

Below, it can be seen how to inspect the plots in the ShinyLexter app. Go to the Output tab to inspect the requested types of graphs for the current run. Keep in mind that this tab will only start showing the graphs, once they have all been created, in larger runs this could take some time. It is possible to sort on model and itemID.

When all plots are generated, the Output tab will show a list of all the newly generated plots. In RShiny, the scaling can be a little bit off. The PNG files of the graphics can also be inspected within the ShinyLexter app or within the "**runname-GRAPHICS**" folder in the working directory.

Example of inspecting plots in the Output tab of ShinyLexter

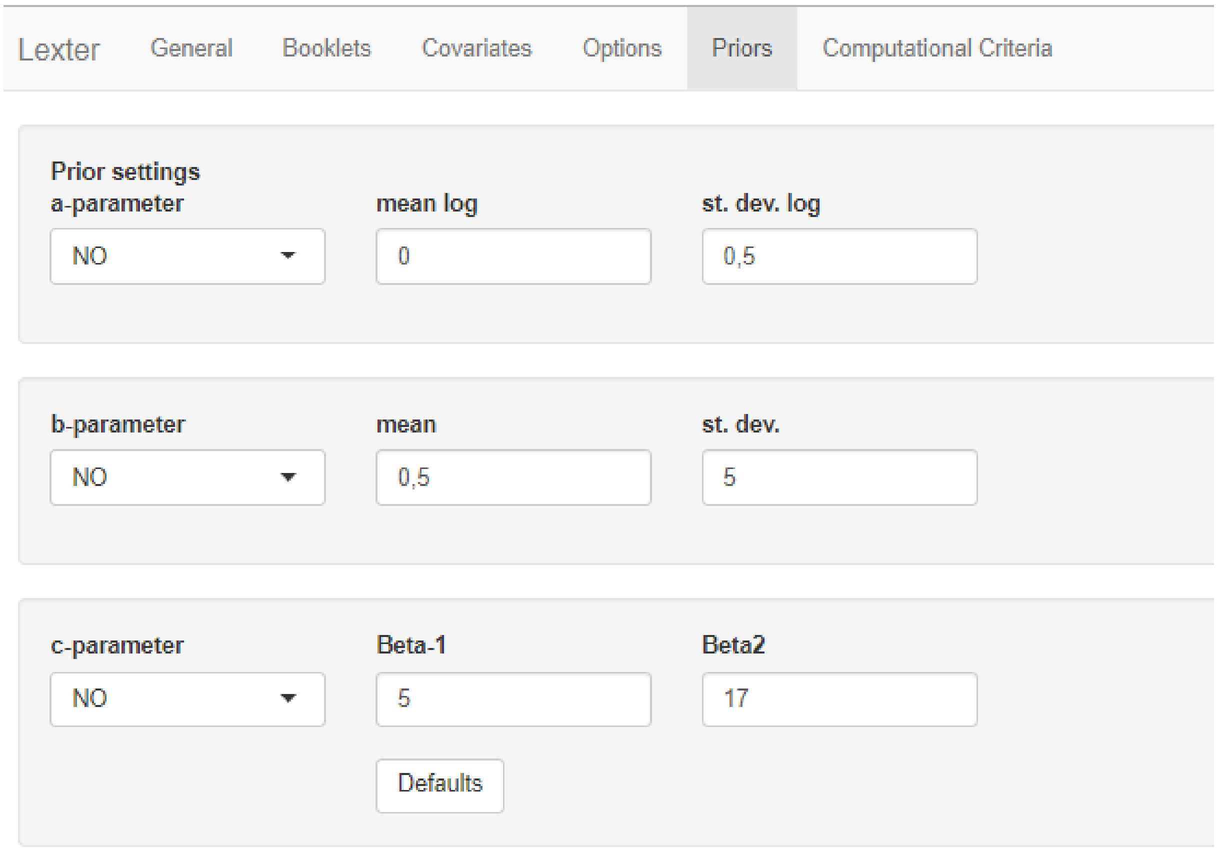


The search bar can be used to search for a specific item or model.

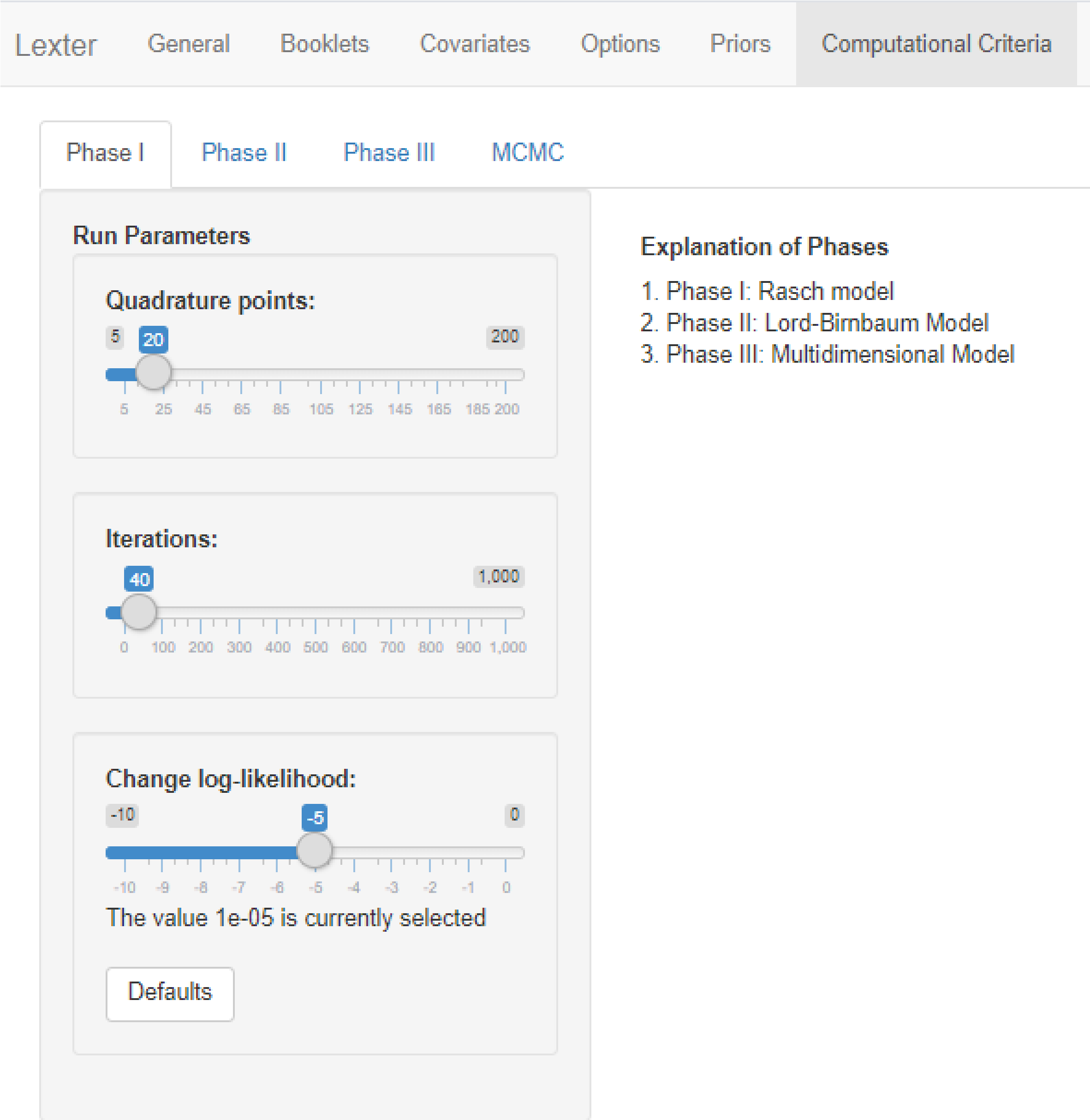
5.8. The priors tab

In the priors tab, the priors for the a, b and c-parameters can be set. The default is no priors are used (‘NO’), but these options can be changed into using a ‘FIXED’ or “FLOATING” prior.

MML estimation combined with priors is known as Bayesian modal estimation. For information on this approach, refer to Mislevy (1986)



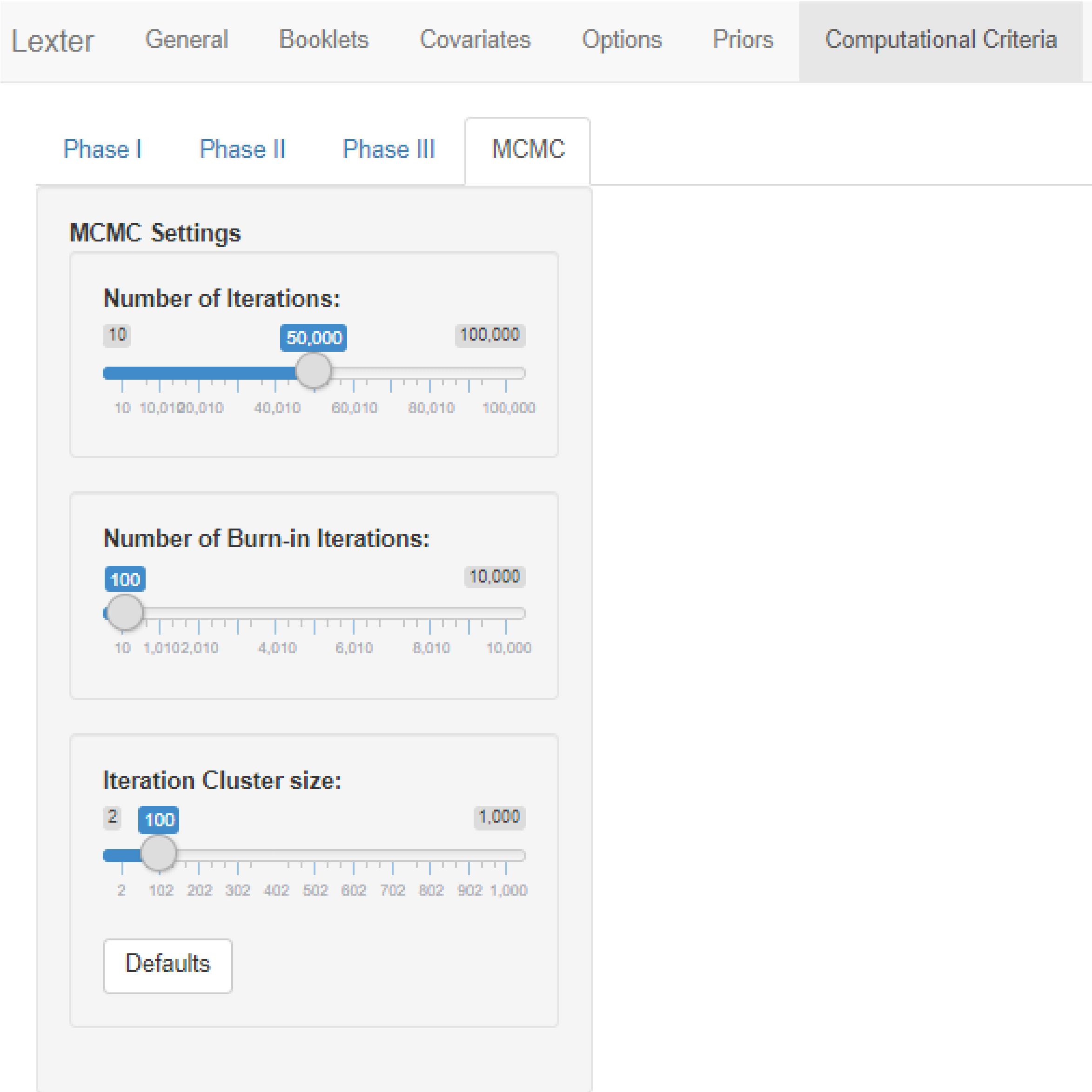
5.9. The Computational Criteria tab



Here the number of quadrature points for the computation of integrals needed for MML, maximum numbers of iterations and convergence criteria for the tree estimation phases: Phase I through III: I = Rasch model, II = Lord-Birnbaum model, III = Multidimensional IRT model can be adjusted.

If the program is used to compute ML and WML estimates for a new data set, the number of iterations should be set equal to zero (also see, **Options/Initial est./File**).

The next tab is MCMC, where the MCMC iterations, a possible burn-in period and a cluster size can be defined for the MCMC estimation. MCMC is run when this is chosen under the Options/Estimation/Estimation method or if the number of dimensions is greater than 5.



5.10. Fixing item parameters: The file RUNNAME.PAR

In every estimation run, a file RUNNAME.PAR containing the estimates of the item and population parameters is generated. An example is given below, for a test containing 6 items; the first three are dichotomously scores, thither three items have 5, 4, and 4 response categories, that is $M_4 = 4$, $M_5 = 3$, and $M_6 = 3$. Every record contains the item label, M_i , a string of technical information (item weight and category weights), the discrimination parameter, the guessing parameter and the item category parameters (as transformed parameters, see description of the output in the next section). The first 6 records contain information from estimating the 1PLM/PCM. Record 7 gives the estimates of the mean and variances of the ability distributions. These records are distinguished from the records pertaining to the item parameters by zero's following the label of the ability distribution. The mean and standard deviation follow, divided by the number 0.0000. So, in the example, the mean is 0.2500, and the standard deviation is 1.2300.

After the string of zeros, analogous information is repeated for the 2PLM/GPCM. The file ends with another string of zeros.

```
BL00045             1 10100000000000    1.0000    0.0000   -0.9280
BL00099             1 10100000000000    1.0000    0.0000   -2.5260
BL00100             1 10100000000000    1.0000    0.0000   -1.1510
BL00101             3 10123000000000    1.0000    0.0000   -0.1170   -0.7140   -1.1950
BL00160             2 10120000000000    1.0000    0.0000   -0.7940   -0.8030
BL00524             2 10120000000000    1.0000    0.0000   -0.6700   -0.3280
Overall Ability     0 00000000000000    0.2500    0.0000    1.2300
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00BL00045           1 10100000000000    0.8030    0.0000   -0.9100
BL00099             1 10100000000000    1.3780    0.0000   -2.9490
BL00100             1 10100000000000    1.0000    0.0000   -1.1510
BL00101             3 10123000000000    0.8430    0.0000   -0.7050   -1.5570   -1.1950
BL00160             2 10120000000000    1.2890    0.0000   -0.9570   -0.7730
BL00524             2 10120000000000    1.1010    0.0000   -0.6870   -0.3344
Overall Ability     0 00000000000000    0.0000    0.0000    1.0000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00
```

If several item parameters need to be fixed at the current values in a subsequent run, create a file with the records of the relevant items to be fixed. This file can then be entered in ShinyLexter app when fixing items is requested.

6. The Output

The main output is written to a file **RUNNAME.MIR**. The file can be viewed by choosing the **Output** option in the top row followed by the option **View**. The output is written to a text file which can be accessed using most general purpose editors, such as Notepad or Word. The estimated parameters are written to an output file RUNNAME,PAR, which is described in the previous section. An analogous parameter file can also be used as input for fixing item and/or population parameters.

Information on person parameters is written to

RUNNAME.WRM1	WML estimation of ability 1PLM/PCM.
RUNNAME.WRM2	WML estimation of ability 2PLM/GPCM
RUNNAME.PRS1	ML estimation of ability plus LM person fit statistics 1PLM/PCM
RUNNAME.PRS2	ML estimation of ability plus LM person fit statistics 2PLM/GPCM
RUNNAME.EAP1	EAP estimation of ability. 1PLM/PCM
RUNNAME.EAP2	EAP estimation of ability. 2PLM/GPCM
RUNNAME.EAP3	Multidimensional EAP estimation of ability.

6.1. The file RUNNAME.MIR

First page echoes run information. This file is also shown in the ShinyLexter app on the "MIR Output" tab.

```
********************************************************************************
*                                                                              *
*  LEXTER                                          2-14-2019        21:30:26   *
*                                                                              *
********************************************************************************






                             Lexter Scaling Program
                                 Version 1.00
                               Februari 1, 2019






RUN TITLE:

RUN NAME: DIA_REKENEN_FIXED



RUN SPECIFICATION:
         NUMBER OF ITEMS IN DESIGN      :  56
         NUMBER OF TESTS IN DESIGN      :   4
         NUMBER OF MARGINALS IN DESIGN :   1
         NUMBER OF DIMENSIONS           :   1

         ESTIMATION PROCEDURE :  GPCM  Lord-Birnbaum model  MML
         CONFIDENCE INTERVALS COMPUTED USING COMPLETE INFORMATION MATRIX
         MODEL FIT EVALUATED USING:
             R1-STATISTIC
             NUMBER OF SCORE LEVELS   3
             DIF COMPUTED USING   R1-STATISTIC
             ML ESTIMATES OF ABILITY COMPUTED
             WML ESTIMATES OF ABILITY COMPUTED
             EAP ESTIMATES OF ABILITY COMPUTED
             EAP-ESTIMATES PERSON PARAMETERS TOTAL SCORES
             NUMBER OF SCORING CATEGORIES      3
             ECHO STARTING VALUES
             EXTENDED OUTPUT REQUESTED
         PRIOR SPECS A B C  :  0  0  0   0.00   5.00   0.50   5.00   5.00  17.00
         NUMBER OF QUADRATURE POINTS:  20  20   5
         NUMBER OF ITERATIONS       :  40  40  20
         STOP-CRITERIA              :  0.1000E-04  0.1000E-04  0.1000E-04
================================================================================
```

Statistics on response frequencies and the item administration design.

```
INPUT DATA
```

ITEM STATISTICS

ITEM	COMPLETE ITEM LABEL	CAT	WEIGHT	TOTAL	SCORE	P-VALUE	DESIGN
1	RD514054	0	0	2407	1021	0.4242	1111
1	RD514054	1	1	2407	1386	0.5758	1111
2	RD514218	0	0	2407	785	0.3261	1111
2	RD514218	1	1	2407	1622	0.6739	1111
3	RD514574	0	0	2407	478	0.1986	1111
3	RD514574	1	1	2407	1929	0.8014	1111
4	RD514590	0	0	2407	861	0.3577	1111
4	RD514590	1	1	2407	1546	0.6423	1111
. . . .							
50	c307228	0	0	754	170	0.2255	1111
50	c307228	1	1	754	584	0.7745	1111
51	c307237	0	0	754	191	0.2533	1111
51	c307237	1	1	754	563	0.7467	1111
52	c307258	0	0	754	90	0.1194	1111
52	c307258	1	1	754	664	0.8806	1111
53	c307282	0	0	754	425	0.5637	1111
53	c307282	1	1	754	329	0.4363	1111
54	c307411	0	0	754	298	0.3952	1111
54	c307411	1	1	754	456	0.6048	1111
55	c307414	0	0	754	173	0.2294	1111
55	c307414	1	1	754	581	0.7706	1111
56	c307423	0	0	754	399	0.5292	1111
56	c307423	1	1	754	355	0.4708	1111

Below, item labels are truncated

GROUP STATISTICS

BOOKLET	COMPLETE BOOKLET LABEL	MARGINAL	#ITEM	#PERSONS	MEAN SCORE
1	Booklet1	OVERALL PROFICIENCY	56	681	21.31
2	Booklet2	OVERALL PROFICIENCY	56	915	31.41
3	Booklet3	OVERALL PROFICIENCY	56	73	24.38
4	Booklet4	OVERALL PROFICIENCY	56	738	35.28

Below booklet and distribution labels are truncated

The program proceeds with the estimation of the 1PLM/PCM and then with the 2PLM/3PLM/GPCM estimation. In the output, the former models are referred to as Rasch-Type models, the latter as Lord-type models. First the iteration history is displayed.

```
MML ITERATION HISTORY
 -------------------------------------------------------------------------------
      1   426.39431 RASCH-MODEL  time: min.   0 | sec.   1.12     -205880.355
      2    10.58604 RASCH-MODEL  time: min.   0 | sec.   1.14     -205869.769
      3     1.45607 RASCH-MODEL  time: min.   0 | sec.   1.13     -205868.313
      4     0.34339 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.969
      5     0.09939 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.870
      6     0.03089 RASCH-MODEL  time: min.   0 | sec.   1.15     -205867.839
      7     0.01013 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.829
      8     0.00351 RASCH-MODEL  time: min.   0 | sec.   1.12     -205867.825
      9     0.00131 RASCH-MODEL  time: min.   0 | sec.   1.12     -205867.824
     10     0.00053 RASCH-MODEL  time: min.   0 | sec.   1.12     -205867.824
     11     0.00023 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.823
     12     0.00011 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.823
     13     0.00006 RASCH-MODEL  time: min.   0 | sec.   1.15     -205867.823
     14     0.00003 RASCH-MODEL  time: min.   0 | sec.   1.12     -205867.823
     15     0.00002 RASCH-MODEL  time: min.   0 | sec.   1.12     -205867.823
     16     0.00001 RASCH-MODEL  time: min.   0 | sec.   1.13     -205867.823
 -------------------------------------------------------------------------------
```

The next display shows the estimates of the item parameters and their standard errors. Fixed item parameters are marked with an asterisk.

```
    1      RD514054 B*      1    -0.253     0.000
    2      RD514218 B*      1    -1.064     0.000
    3      RD514574 B*      1    -1.519     0.000
    4      RD514590 B*      1    -0.532     0.000
    5      RD514823 B*      1    -1.969     0.000
    6      RD515026 B*      1    -2.071     0.000
    7      RD515046 B*      1    -0.858     0.000
    8      RD515062 B*      1    -1.711     0.000
    9      RD515108 B*      1    -1.827     0.000
   10      RD515148 B*      1    -0.973     0.000
   11      RD515449 B*      1    -1.878     0.000
   12      RD515490 B*      1    -1.648     0.000
. . . . . . . .
   47      c307066  B       1    -2.183     0.086
   48      c307099  B       1    -0.296     0.085
   49      c307195  B       1    -1.152     0.083
   50      c307228  B       1    -2.460     0.094
   51      c307237  B       1    -2.291     0.100
   52      c307258  B       1    -3.294     0.119
   53      c307282  B       1    -0.795     0.079
   54      c307411  B       1    -1.558     0.085
   55      c307414  B       1    -2.435     0.095
   56      c307423  B       1    -0.950     0.079
 -------------------------------------------------------------------------------
```

The next display gives an example of the estimates of the population parameters.
If estimates of the person parameters are requested, also the means of these distributions are displayed. Finally, estimates of the global reliability of the booklets are given. The second column gives the variance of the EAP estimate of theta, the third column gives the expected error variance, the fourth column the total variance of theta and the last column gives an estimate of the total variance of theta.

```
ESTIMATION OF POPULATION PARAMETERS
 --------------------------------------------------------------------------------
 POPULATION : Marg2000
 --------------------------------------------------------------------------------
 MEAN     : -0.550 STANDARD DEVIATION     :  0.655
 SE(MEAN) :  0.025 SE(STANDARD DEVIATION) :  0.015
 --------------------------------------------------------------------------------
 POPULATION : Marg2003
 --------------------------------------------------------------------------------
 MEAN     :  0.005 STANDARD DEVIATION     :  0.692
 SE(MEAN) :  0.026 SE(STANDARD DEVIATION) :  0.016
 --------------------------------------------------------------------------------
 POPULATION : Marg2006
 --------------------------------------------------------------------------------
 MEAN     : -0.341 STANDARD DEVIATION     :  1.221
 SE(MEAN) :  0.035 SE(STANDARD DEVIATION) :  0.026
 --------------------------------------------------------------------------------
 POPULATION : Marg2009
 --------------------------------------------------------------------------------
 MEAN     :  0.000 STANDARD DEVIATION     :  0.677
 SE(MEAN) :  0.000 SE(STANDARD DEVIATION) :  0.017
 --------------------------------------------------------------------------------
 LOG-LIKELIHOOD     -92959.397
 --------------------------------------------------------------------------------


  ---------------------------------------------------------
  BOOKLET MARG   VAR(E(8|X)) E(VAR(8|X))  VAR(8)      REL
  ---------------------------------------------------------
    1       1     0.279        0.151     0.429     0.649
    2       2     0.336        0.143     0.479     0.702
    3       3     1.080        0.411     1.491     0.724
    4       4     0.289        0.170     0.459     0.630
  ----------------------------------------------------------
```

If requested, the next table fives the maximum likelihood ability estimates and their standard errors. Further, under the label “ERROR PERCENTAGE” it gives estimates of the estimated percentages miss-classifications, that is, the estimated percentage of respondents wrongly passing or failing the test. The last column gives the local reliability given a total score.

ML Estimates of Abilities conditional on total scores

BOOKLET : Test1

SCORE	FREQUENCY	CUM.	THETA	SE(THETA)	ERROR PERCENTAGE PASSING	ERROR PERCENTAGE FAILING	REL
1	0	0.000	-4.694	1.052	0.000	0.001	0.000
2	0	0.000	-3.903	0.769	0.000	0.002	0.092
3	0	0.000	-3.413	0.641	0.000	0.004	0.368
4	1	0.001	-3.052	0.565	0.001	0.006	0.510
5	2	0.002	-2.764	0.512	0.001	0.009	0.597
. . . .							
51	6	0.993	2.135	0.380	0.032	0.007	0.778
52	6	0.996	2.286	0.398	0.025	0.005	0.757
53	4	0.998	2.452	0.419	0.019	0.003	0.730
54	2	0.999	2.640	0.447	0.014	0.001	0.693
55	1	1.000	2.855	0.483	0.010	0.000	0.642
56	0	1.000	3.111	0.532	0.006	0.000	0.565

Analogous information is also given if EAP estimates are requested. An example is given below.

EAP Estimates of Abilities conditional on total scores

BOOKLET : Test1 POPULATION : Marg1

SCORE	FREQUENCY	CUM.	EAP	POST_VAR	ERROR PERCENTAGE PASSING	ERROR PERCENTAGE FAILING	REL
0	0	0.000	-2.928	0.449	0.000	0.001	0.691
1	0	0.000	-2.736	0.429	0.000	0.002	0.718
2	0	0.000	-2.560	0.411	0.000	0.004	0.741
3	0	0.000	-2.397	0.396	0.000	0.006	0.758
4	1	0.001	-2.245	0.384	0.001	0.010	0.773
5	2	0.002	-2.103	0.370	0.002	0.014	0.790
. . . .							
51	6	0.993	1.812	0.330	0.039	0.008	0.832
52	6	0.996	1.923	0.333	0.031	0.006	0.830
53	4	0.998	2.033	0.333	0.023	0.004	0.830
54	2	0.999	2.146	0.341	0.017	0.002	0.822
55	1	1.000	2.267	0.356	0.013	0.001	0.805
56	0	1.000	2.400	0.371	0.009	0.000	0.788
57	1	1.000	2.542	0.383	0.007	0.000	0.774

Next, if requested, information regarding model fit is given, using Lagrange multiplier tests. The first test focusses on the extent to which the frequency distribution is represented. An example is given below. Scores are pooled to obtain frequencies above 20. The first LM test reported above uses the complete matrix of weights, the second one is a diagonal approximation. The test is repeated for every booklet in the design.

```
Lagrange multipliers ability distribution for RASCH-TYPE MODEL
 ================================================================================
  Score Range Frequency Expected
  ------------------------------------------
    0    8       44       47.48
    9    9       28       35.30
   10   10       39       55.01
   11   11       93       81.06
   12   12      109      112.42
   13   13      156      147.34
   14   14      203      183.53
   15   15      233      217.82
   16   16      262      245.66
   17   17      245      261.74
   18   18      256      261.49
   19   19      236      242.87
   20   20      201      207.74
   21   21      151      161.79
   22   22       99      112.63
   23   23       66       67.83
   24   24       43       33.26
   25   26       25       14.02
  ------------------------------------------
      LM    df  Prob    Approx  df  Prob
  ------------------------------------------
     36.77  17  0.00     24.70  17  0.10
  ------------------------------------------
```

.

Two DIF-tests are presented. The first panel below displays a test for the constancy of the parameters of a certain item parameter against the same parameters in all other booklets. This test is repeated for all booklets. The second panel displays a test which tests the constancy of item parameters across all booklets.

In the example below, the focal group is booklet 1, and the reference group consists of all other booklets. The columns labeled Obs and Exp give the average observed and posterior expected item scores in the focal and reference group, respectively. The column labeled Abs Dif. Gives the absolute difference between the two. The column labeled LM gives the value of the LM statistic, the column labeled df gives the degrees of freedom and the column labeled Prob gives the significance probability. Due to the large sample size, all tests are significant. Therefore, the column the absolute differences are more informative with respect to model violations here.

For more information refer to Glas (1988, 1998, 1999) and Glas & Suárez-Falćon, (2003).

Lagrange tests DIF for RASCH-TYPE-MODEL for Booklet 1

	Item	LM	df	Prob	Focal-Group Obs	Focal-Group Exp	Reference Obs	Reference Exp	Abs. Dif.
1	Item1	7.83	1	0.01	0.97	0.96	0.96	0.97	0.01
2	Item2	0.27	1	0.60	0.91	0.91	0.93	0.93	0.00
3	Item3	46.18	1	0.00	0.97	0.95	0.95	0.96	0.01
5	Item5	295.30	1	0.00	0.65	0.52	0.57	0.62	0.09
6	Item6	1580.14	1	0.00	0.40	0.69	0.86	0.76	0.19
7	Item7	220.18	1	0.00	0.47	0.36	0.43	0.46	0.07
8	Item8	347.82	1	0.00	0.69	0.55	0.60	0.64	0.09
9	Item9	116.21	1	0.00	0.66	0.57	0.63	0.66	0.05
10	Item10	706.01	1	0.00	0.98	0.90	0.89	0.92	0.05
12	Item12	44.22	1	0.00	0.98	0.97	0.97	0.97	0.01
13	Item13	30.37	1	0.00	0.78	0.82	0.87	0.86	0.02
18	Item18	2995.19	3	0.00	1.55	2.14	2.82	2.51	0.45
19	Item19	147.61	3	0.00	2.35	2.22	2.41	2.47	0.09
20	Item20	449.10	3	0.00	1.35	1.56	2.01	1.91	0.16
21	Item21	303.87	3	0.00	1.62	1.45	1.63	1.72	0.13
23	Item23	458.96	3	0.00	1.59	1.35	1.49	1.57	0.16

Lagrange tests DIF over all groups for RASCH-TYPE-MODEL

	Item	LM	df	Prob	Abs.Dif.
1	Item1	12.02	3	0.01	0.01
2	Item2	43.18	3	0.00	0.01
3	Item3	48.89	3	0.00	0.01
4	Item4	17.31	2	0.00	0.01
5	Item5	311.09	3	0.00	0.07
6	Item6	2092.59	3	0.00	0.15
7	Item7	255.18	3	0.00	0.06
8	Item8	378.54	3	0.00	0.07
9	Item9	178.40	3	0.00	0.05
10	Item10	711.79	3	0.00	0.04
12	Item12	62.03	3	0.00	0.01
13	Item13	39.67	3	0.00	0.02
14	Item14	12.70	1	0.00	0.01
18	Item18	2801.92	2	0.00	0.40
19	Item19	156.63	2	0.00	0.09
20	Item20	595.76	2	0.00	0.16
21	Item21	278.46	2	0.00	0.12
23	Item23	751.42	3	0.00	0.18

The test for the item characteristic curves generally follows the same lines as the tests for DIF. Only here, observed and posterior expected are computed using a partitioning of respondents according to their score level (i.e., the score level computed without the item targeted). Three score levels are formed. The sample sizes within the score levels are displayed below in the three last columns. Again, due to the sample size, the absolute difference between observed and expected average score are more informative then the outcomes of the statistics.

For more information refer to Glas (1988, 1998, 1999), and Glas & Suárez-Falćon, (2003).

Lagrange multipliers tracelines for RASCH-TYPE MODEL

Booklet : 1

				Groups: 1		2		3		Abs.	1	2	3
Item		LM	df	Prob	Obs.	Exp.	Obs.	Exp.	Obs.	Exp.	Dif.	Size	Size	Size
1	Item1	10.47	2	0.01	0.94	0.94	0.98	0.96	0.99	0.98	0.01	869	763	825
2	Item2	1.16	2	0.56	0.85	0.86	0.91	0.91	0.97	0.95	0.01	837	781	834
3	Item3	32.15	2	0.00	0.95	0.92	0.97	0.95	0.99	0.97	0.02	867	766	824
5	Item5	149.74	2	0.00	0.46	0.39	0.69	0.52	0.80	0.65	0.13	766	795	882
6	Item6	1196.21	2	0.00	0.18	0.57	0.36	0.68	0.60	0.79	0.30	690	805	956
7	Item7	48.74	2	0.00	0.29	0.24	0.42	0.34	0.65	0.48	0.10	705	801	930
8	Item8	151.59	2	0.00	0.55	0.42	0.67	0.55	0.83	0.68	0.14	774	781	884
9	Item9	67.73	2	0.00	0.53	0.44	0.65	0.57	0.77	0.70	0.08	770	788	886
10	Item10	648.53	2	0.00	0.97	0.85	0.99	0.91	0.98	0.94	0.08	863	765	822
12	Item12	20.44	2	0.00	0.96	0.95	0.98	0.97	1.00	0.98	0.02	868	765	822
13	Item13	58.10	2	0.00	0.63	0.73	0.81	0.82	0.91	0.89	0.05	810	798	846
18	Item18	843.24	2	0.00	1.32	1.63	1.44	2.14	1.85	2.55	0.57	727	836	910
19	Item19	185.41	2	0.00	2.19	1.91	2.32	2.20	2.49	2.48	0.14	723	794	957
20	Item20	381.33	2	0.00	0.88	1.19	1.29	1.48	1.73	1.88	0.22	664	845	958
21	Item21	132.53	2	0.00	1.31	1.17	1.61	1.42	1.91	1.71	0.17	745	843	879
23	Item23	218.82	2	0.00	1.30	1.08	1.56	1.32	1.88	1.61	0.24	745	830	870

Also the test for local independence generally follows the same lines as the tests for DIF. The tests targets the observed average score on an item conditional on the possible values obtained on the other items. For instance, in the display below, the first row pertains to the scores obtained on item 2 conditional on the possible scores on item 1. Both items are dichotomous. So 0.72 is the average observed score on item 2 for students scoring zero on item 1 and 0.91 is the average observed score on item 2 for students scoring a one on item 1. The last item, item 23 is evaluated conditional on the response on item 21. Both items have 4 response categories. So the average scores on item 23 are 1.17, 1.50, 1.68, and 1.76 for students scoring 0, 1, 2 and 3 on item 21, respectively.

Again, due to the sample size, the absolute difference between observed and expected average score are more informative then the outcomes of the statistics.

For more information refer to Glas (1988, 1998, 1999), Glas & Suárez-Falćon, (2003) and Glas and Verhelst (1989, 1995).

Lagrange multipliers local dependence for RASCH-TYPE MODEL

Booklet : 1 respondents : 2489

Item 1		Item 2		LM	df	Prob	Obs.	Exp.	Obs.	Exp.				
2	Item2	1	Item1	7.35	1	0.01	0.72	0.85	0.91	0.91				
3	Item3	2	Item2	2.40	1	0.12	0.89	0.92	0.98	0.95				
5	Item5	3	Item3	11.19	1	0.00	0.61	0.43	0.66	0.53				
6	Item6	5	Item5	627.88	1	0.00	0.24	0.63	0.48	0.73				
7	Item7	6	Item6	82.63	1	0.00	0.41	0.31	0.56	0.43				
8	Item8	7	Item7	3.19	1	0.07	0.48	0.50	0.92	0.61				
9	Item9	8	Item8	9.16	1	0.00	0.45	0.50	0.75	0.61				
10	Item10	9	Item9	163.34	1	0.00	0.96	0.88	0.99	0.91				
12	Item12	10	Item10	3.62	1	0.06	0.86	0.95	0.98	0.97				
13	Item13	12	Item12	6.25	1	0.01	0.52	0.70	0.79	0.82				
18	Item18	13	Item13	138.21	1	0.00	1.33	1.79	1.62	2.24				
19	Item19	18	Item18	78.69	3	0.00	2.08	2.01	2.24	2.16	2.45	2.28	2.59	2.40
20	Item20	19	Item19	192.85	3	0.00	0.70	1.18	1.21	1.36	1.23	1.47	1.49	1.68
21	Item21	20	Item20	199.30	3	0.00	1.22	1.15	1.57	1.39	1.76	1.59	2.02	1.74
23	Item23	21	Item21	397.43	3	0.00	1.17	1.03	1.50	1.23	1.68	1.43	1.76	1.59

All output tables are repeated for the Birnbaum models (2PLM, 3PLM, GPCM). For instance, the table with parameter estimates looks as follows. The category bounds parameters refer to parameterization in the second section. The transformed parameters are available because they lead to simpler computations.

```
MML-PARAMETER ESTIMATION BIRNBAUM-MODEL
 --------------------------------------------------------------------------------
                              TRADITIONAL          TRANSFORMED
 ITEM     LABEL   PAR     CAT  ESTIMATE      SE     ESTIMATE     SE
 --------------------------------------------------------------------------------
    1     RD514054 A*      0      1.446     0.000
    1     RD514054 B*      1     -0.196     0.000     -0.283     0.000
    2     RD514218 A*      0      1.507     0.000
    2     RD514218 B*      1     -0.796     0.000     -1.200     0.000
    3     RD514574 A*      0      1.204     0.000
    3     RD514574 B*      1     -1.282     0.000     -1.544     0.000
    4     RD514590 A*      0      1.058     0.000
    4     RD514590 B*      1     -0.483     0.000     -0.511     0.000
    5     RD514823 A*      0      1.059     0.000
    5     RD514823 B*      1     -1.803     0.000     -1.909     0.000
. . . . . .
   53     c307282  A       0      0.971     0.166
   53     c307282  B       1     -0.689     0.180     -0.669     0.169
   54     c307411  A       0      1.564     0.194
   54     c307411  B       1     -1.289     0.087     -2.015     0.214
   55     c307414  A       0      1.364     0.182
   55     c307414  B       1     -1.986     0.124     -2.709     0.230
   56     c307423  A       0      1.078     0.167
   56     c307423  B       1     -0.853     0.149     -0.920     0.174

--------------------------------------------------------------------------------
```

One of the differences between the 1PLM and other models is that the number-correct score is no longer a sufficient statistic for the ability parameter. Ml- WML- and EAP-estimates of the ability parameters can be obtained in separate files described below. However, in some cases, practitioners also want EAP estimates of abilities given number-correct scores under the 2PLM, 3PLM and GPCM. Such estimates are provided in the table displayed below.

EAP Estimates of Abilities conditional on total scores

BOOKLET : Test1 POPULATION : Marg1

Score	Frequency	Cum.	EAP Theta E(E(8|x)|t)	VAR(E(8|x)|t)	E(VAR(8|x)|t)	Error Percentage Passing	Failing	Reliability
0	0	0.000	0.000	0.000	0.000	0.000	0.477	1.000
1	0	0.000	0.000	0.000	0.000	0.000	0.477	1.000
2	0	0.000	0.000	0.000	0.000	0.000	0.477	1.000
3	0	0.000	0.000	0.000	0.000	0.000	0.477	1.000
4	1	0.001	-2.503	0.000	0.188	0.000	0.032	0.765
5	2	0.002	-2.378	0.003	0.178	0.001	0.039	0.775
6	4	0.004	-2.177	0.002	0.160	0.002	0.056	0.798
. . . .								
39	57	0.824	0.887	0.005	0.092	0.147	0.296	0.879
40	48	0.848	0.978	0.003	0.082	0.128	0.241	0.894
41	49	0.872	1.045	0.005	0.081	0.131	0.224	0.893
42	44	0.894	1.127	0.004	0.088	0.127	0.188	0.885
43	48	0.918	1.228	0.007	0.102	0.131	0.181	0.864
44	29	0.932	1.307	0.005	0.111	0.113	0.110	0.855
. . . .								
51	6	0.993	1.974	0.006	0.137	0.045	0.010	0.822
52	6	0.996	2.164	0.006	0.146	0.032	0.007	0.811
53	4	0.998	2.288	0.005	0.150	0.024	0.004	0.806
54	2	0.999	2.391	0.001	0.156	0.018	0.001	0.804
55	1	1.000	2.395	0.000	0.156	0.018	0.000	0.805
56	0	1.000	0.000	0.000	0.000	0.332	0.000	1.000
57	1	1.000	2.921	0.000	0.203	0.006	0.000	0.746

Also for the multidimensional model, most of the information displayed above is provided. There are some minor changes however. For instance, below are the MML estimates of the item parameters for the multidimensional model. The entry under the label ‘CAT’ in the row for the discrimination parameter (row labeled A) gives the dimension to which the item related. So item 1 relates to dimension 1, item 22 relates to dimension 2, and item 38 relates to dimension 3.

```
MML-PARAMETER ESTIMATION MULTIDIMENSIONAL IRT LORD-TYPE
 --------------------------------------------------------------------------------
                         A=DIM
                               TRADITIONAL         TRANSFORMED
 ITEM     LABEL   PAR     CAT  ESTIMATE     SE     ESTIMATE     SE
 --------------------------------------------------------------------------------
    1     Item1    A       1     0.992     0.118
    1     Item1    B       1    -1.592     0.108    -1.580     0.108
. . . . . . . . .

   22     Item22   A       2     1.739     0.174
   22     Item22   B       1    -1.027     0.144    -1.786     0.144
. . . . . . . . .
   38     Item38   A       3     1.588     0.106
   38     Item38   B       1     0.691     0.082     1.098     0.082

 --------------------------------------------------------------------------------
```

The MML estimates of the population parameters are given in the next display. The mean and variances of Group 2 are constrained to zero and one to identify the model.

```
ESTIMATION OF POPULATION PARAMETERS
 --------------------------------------------------------------------------------
 POPULATION : Group 1
 --------------------------------------------------------------------------------
 --------------------------------------------------------------------------------
   dim    mean   covariance (estimated)
 --------------------------------------------------------------------------------
    1   0.042   2.163   0.540   0.589
    2  -0.155   0.540   1.375   0.762
    3  -0.107   0.589   0.762   1.671
 --------------------------------------------------------------------------------
   dim se(mean)  se(covariance)
 --------------------------------------------------------------------------------
    1   0.058   0.087
    2   0.034   0.081   0.033
    3   0.037   0.088   0.048   0.039
 --------------------------------------------------------------------------------
   dim   correlation matrix
 --------------------------------------------------------------------------------
    1   1.000   0.313   0.310
    2   0.313   1.000   0.503
    3   0.310   0.503   1.000
 --------------------------------------------------------------------------------
POPULATION : Group 2
 --------------------------------------------------------------------------------
 --------------------------------------------------------------------------------
   dim   mean   covariance (constrained estimate)
```

```
--------------------------------------------------------------------------------
    1   0.000   1.000   0.230   0.187
    2   0.000   0.230   1.000   0.391
    3   0.000   0.187   0.391   1.000
--------------------------------------------------------------------------------
  dim   correlation matrix
--------------------------------------------------------------------------------
    1   1.000   0.230   0.187
    2   0.230   1.000   0.391
    3   0.187   0.391   1.000
--------------------------------------------------------------------------------
LOG-LIKELIHOOD    -260924.437
--------------------------------------------------------------------------------
```

In certain applications, the ability dimensions are weighted to produce an overall estimate of ability. Option 14 In the file **RUNNAME.INST** enables this weighting. In the example of Section 6.12, the weights for 3 dimensions are set to unity. The program will rescale then to sum up to one, and will compute the global reliability of the compound score. Next, the program will also compute the weights optimizing the global reliability. This gives the following output.

```
A PRIORI FIXED VALUES
   0.3333333   0.3333333   0.3333333
 RELIABILITY        0.95136

OPTIMAL WEIGTS
   0.1490649   0.2539844   0.5659247
 RELIABILITY        0.95898
```

6.2. The file RUNNAME.WRM1 and RUNNAME.WRM2

The record for the first 10 respondents are displayed. The column labeled THETA gives the WML estimates, the column labeled SE gives the standard error. The column labeled PLAUS gives a ‘pseudo’-plausible value, that is, random draw from a normal approximation of the estimate. A ‘proper’ plausible value is supplied under the **EAP** option.
Person IDs are displayed if present in the input file. IV is a person number, IB is a booklet number. PROB is the proportion correct.

WEIGHTED ML ESTIMATES OF ABILITY

IV	IB	PERSID	SCORE	MAXIMUM	PROP	THETA	SE	PLAUS
1	1	-99	20.000	26.000	0.769	0.082	0.516	-0.040
2	1	-99	22.000	26.000	0.846	0.656	0.588	1.772
3	1	-99	18.000	26.000	0.692	-0.392	0.475	0.183
4	1	-99	22.000	26.000	0.846	0.656	0.588	0.012
5	1	-99	24.000	26.000	0.923	1.464	0.756	1.390
6	1	-99	18.000	26.000	0.692	-0.392	0.475	-1.207
7	1	-99	18.000	26.000	0.692	-0.392	0.475	-1.656
8	1	-99	21.000	26.000	0.808	0.352	0.546	-0.336
9	1	-99	15.000	26.000	0.577	-0.997	0.446	-1.126
10	1	-99	25.000	26.000	0.962	2.097	0.972	2.469

6.3. The file RUNNAME.PRS1 and RUNNAME.PRS2

Most columns are analogous to the columns in the previous files. The columns labeled LMB and LMD give LM person fit statistics as described in Glas and Dagohoy (2007). LMB is based on a partition of the response pattern according to the partition defined in the **Tests** screen under **Pf**. Clicking on **Pf** creates the default partition. The partition can be edited. The columns labeled EB and UD give the diagonal approximations of the statistics. The columns labeled P give the significance proportions.

ML ESTIMATES OF ABILITY AND PERSON FIT STATISTICS

IV	IB	PERSID	THETA	SE(TH)	UB	P	LMB	P	UD	P	LMD	P	PLAUS
1	1	-99	0.132	0.518	0.137	0.711	0.471	0.493	0.020	0.887	0.057	0.812	-0.491
2	1	-99	0.743	0.595	0.470	0.493	1.656	0.198	0.279	0.597	1.142	0.285	0.955
3	1	-99	-0.359	0.476	0.011	0.918	0.037	0.847	0.156	0.693	0.662	0.416	0.487
4	1	-99	0.743	0.595	0.519	0.471	1.829	0.176	0.259	0.611	1.383	0.240	0.996
5	1	-99	1.651	0.783	0.188	0.664	0.723	0.395	0.218	0.641	1.709	0.191	1.737
6	1	-99	-0.359	0.476	0.011	0.918	0.037	0.847	0.071	0.790	0.429	0.512	-0.700
7	1	-99	-0.359	0.476	0.011	0.918	0.037	0.847	0.011	0.915	0.038	0.846	-0.503
8	1	-99	0.417	0.550	0.202	0.653	0.697	0.404	0.051	0.821	0.177	0.674	0.498
9	1	-99	-0.989	0.446	1.070	0.301	3.880	0.049	0.709	0.400	1.734	0.188	-1.450
10	1	-99	2.460	1.058	0.075	0.784	0.314	0.575	0.148	0.700	1.132	0.287	1.402

6.4. The file RUNNAME.EAP1 and RUNNAME.EAP2

The column labeled EAP(T) gives the posterior expectation, the column labeled POST(T) gives the posterior standard deviation. The column labeled PLAUS gives a plausible value, that is, random draw from the posterior distribution.

```
 EAP ESTIMATES OF ABILITY
========================================================================
     IV  IB   PERSID   SCORE  MAXIMUM     PROP    EAP(T)  POST(T)   PLAUS
========================================================================
      1   1      -99  20.000   26.000    0.769   -0.106    0.397  -0.174
      2   1      -99  22.000   26.000    0.846    0.221    0.412   0.277
      3   1      -99  18.000   26.000    0.692   -0.411    0.385  -0.071
      4   1      -99  22.000   26.000    0.846    0.221    0.412   0.135
      5   1      -99  24.000   26.000    0.923    0.577    0.432   0.819
      6   1      -99  18.000   26.000    0.692   -0.411    0.385  -0.572
      7   1      -99  18.000   26.000    0.692   -0.411    0.385  -0.352
      8   1      -99  21.000   26.000    0.808    0.055    0.404   0.225
      9   1      -99  15.000   26.000    0.577   -0.843    0.374  -0.976
     10   1      -99  25.000   26.000    0.962    0.768    0.443   0.588
```

7.References

Albert, J.H. (1992). Bayesian estimation of normal ogive item response functions using Gibbs sampling. *Journal of Educational Statistics*, *17*, 251-269.

Andersen, E.B. (1977). Sufficient statistics and latent trait models. *Psychometrika*, *42*, 69-81.

Béguin, A.A., & Glas, C.A.W. (2001). MCMC estimation and some fit analysis of multidimensional IRT models. *Psychometrika, 66,* 541-562.

Birnbaum, A. (1968). Some latent trait models. In F.M. Lord & M.R. Novick (Eds.), *Statistical theories of mental test scores.* Reading, MA: Addison-Wesley.

Glas, C.A.W. (1988). The derivation of some tests for the Rasch model from the multinomial distribution. *Psychometrika*, *53*, 525-546.

Glas, C.A.W. (1998) Detection of differential item functioning using Lagrange multiplier tests. *Statistica Sinica, 8*, 647-662.

Glas, C.A.W. (1999). Modification indices for the 2-pl and the nominal response model. *Psychometrika*, *64*, 273-294.

Glas, C.A.W. (2019). Reliability Issues in High-Stakes Educational Tests. In B.P. Veldkamp & C. Sluijter (eds.). *Theoretical and Practical Advances in Computer-based Educational Measurement* (pp. 213-230). Springer International Publishing. https://link.springer.com/book/10.1007%2F978-3-030-18480-3.

Glas, C.A.W., and Verhelst, N.D. (1989). Extensions of the partial credit model. *Psychometrika, 54,* 635-659.

Glas, C.A.W., & Verhelst, N.D. (1995). Tests of fit for polytomous Rasch models. In G.H. Fischer & I.W. Molenaar (Eds.), *Rasch models: foundations, recent developments and applications.* (pp.325-352). New York, NJ: Springer.

Glas, C.A.W., & Suárez-Falćon, J.C. (2003). A comparison of item-fit statistics for the three-parameter logistic model. *Applied Psychological Measurement*, *27*, 87-106.

Mellenbergh, G.J. (1995). Conceptual notes on models for discrete polytomous item responses. *Applied Psychological Measurement*, *19*, 91-100.

Mislevy, R.J. (1986). Bayes modal estimation in item response models. *Psychometrika, 51,* 177-195.

Muraki, E. (1992). A generalized partial credit model: application of an EM algorithm. *Applied Psychological Measurement, 16,* 159- 176.

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests.* Copenhagen: Danish Institute for Educational Research.

Samejima, F. (1969). Estimation of latent ability using a pattern of graded scores. *Psychometrika*, *Monograph Supplement, No. 12.*

Tutz, G. (1990). Sequential item response models with an ordered response. *British Journal of Mathematical and Statistical Psychology*, *43*, 39-55.

Verhelst, N.D., Glas, C.A.W., & de Vries, H.H. (1997). A steps model to analyze partial credit. In: W.J. van der Linden and R.K. Hambleton (Eds.), *Handbook of modern item response theory*. (pp. 123-138). New York, NJ: Springer.

Warm, T.A. (1989). Weighted likelihood estimation of ability in item response theory. *Psychometrika*, *54*, 427-450.

Zeng, L., & Kolen, M.J. (1995). An alternative approach for IRT observed- score equating of number correct scores. Applied Psychological Measurement, 19, 231-240.

Appendix A. Structure of the RUNNAME.TTT file

The file **RUNNAME.TTT** is the file generated by the "Lexter" app in Shiny, to be able to run the modules **Lexter.EXE.** These modules can also be run without help of the ShinyLexter app, by using the command prompt. This is done by entering the module's name without an extension and the jobname. For instance, entering **LEXTER RUNNAME** will result in running Lexter executable.

Before describing its contents, we will first given an example of the file **RUNNAME.TTT.**

```
    2   1  80 4 0 0 1 1 1   3 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0
 0 D:/LexterFiles
 0 0 0    0.0000    0.5000    0.5000    5.0000    5.0000   17.0000
FIG_1B


 D:/LexterFiles/FIG_1B.DAT
(T1,I2,T3,80I1,T1,I1)                                                          0
  50  50
   1   1
   1   1
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
  41  42  43  44  45  46  47  48  49  50
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
   1   1   1   1   1   1   1   1   1   1
  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50
  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70
  71  72  73  74  75  76  77  78  79  80
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
   1   1   1   1   1   1   1   1   1   1
Booklet1              1000 1     20   0.000
Booklet2                00 0      0
Item1               1 1 0 1
Item2               1 1 0 1
Item3               1 1 0 1
Item4               1 1 0 1
Item5               1 1 0 1
Item6               1 1 0 1
Item7               1 1 0 1
Item8               1 1 0 1
Item9               1 1 0 1
Item10              1 1 0 1
Item11              1 1 0 1
Item12              1 1 0 1
Item13              1 1 0 1
Item14              1 1 0 1
Item15              1 1 0 1
Item16              1 1 0 1
Item17              1 1 0 1
Item18              1 1 0 1
Item19              1 1 0 1
Item20              1 1 0 1
Item21              1 1 0 1
Item22              1 1 0 1
Item23              1 1 0 1
Item24              1 1 0 1
```

```
Item25              1 1 0 1
Item26              1 1 0 1
Item27              1 1 0 1
Item28              1 1 0 1
Item29              1 1 0 1
Item30              1 1 0 1
Item31              1 1 0 1
Item32              1 1 0 1
Item33              1 1 0 1
Item34              1 1 0 1
Item35              1 1 0 1
Item36              1 1 0 1
Item37              1 1 0 1
Item38              1 1 0 1
Item39              1 1 0 1
Item40              1 1 0 1
Item41              1 1 0 1
Item42              1 1 0 1
Item43              1 1 0 1
Item44              1 1 0 1
Item45              1 1 0 1
Item46              1 1 0 1
Item47              1 1 0 1
Item48              1 1 0 1
Item49              1 1 0 1
Item50              1 1 0 1
Item51              1 1 0 1
Item52              1 1 0 1
Item53              1 1 0 1
Item54              1 1 0 1
Item55              1 1 0 1
Item56              1 1 0 1
Item57              1 1 0 1
Item58              1 1 0 1
Item59              1 1 0 1
Item60              1 1 0 1
Item61              1 1 0 1
Item62              1 1 0 1
Item63              1 1 0 1
Item64              1 1 0 1
Item65              1 1 0 1
Item66              1 1 0 1
Item67              1 1 0 1
Item68              1 1 0 1
Item69              1 1 0 1
Item70              1 1 0 1
Item71              1 1 0 1
Item72              1 1 0 1
Item73              1 1 0 1
Item74              1 1 0 1
Item75              1 1 0 1
Item76              1 1 0 1
Item77              1 1 0 1
Item78              1 1 0 1
Item79              1 1 0 1
Item80              1 1 0 1
Covariate1         1 1
  20  40       0.00001
  20  40       0.00001
   5  20       0.00001
```

Record 1.

Fortran code defining positions of variables in RUNNAME.TTT:

```
      READ(5,801) NTOETS0,IEXTEND,KTOT0,ISCHAT,IERROR,IGLOBAL,IML,IWML,
     + IEAP,IGROUPS,IFIRST,NORMPOP,IISCORE,ISECOND,INFVAL,IDIF,ECHO,
     + IINFO,ICOV,IML_TOT,EAP_TOT,OPTI_W,FIX_W,NDIM,IFIX,NCOV0,CUT_OFF,
     + IRUN,ITTT
801 FORMAT(3I4,6I2,I4,100I2)
```

	Variable	Meaning
1 (4)	NTOETS	Number of booklets in the data file
2 (4)	IEXTEND	Indicator for extended output (1 = Yes)
3 (4)	KTOT	Total number of items in the data file
4 (2)	ISCHAT	Model codes (19 codes, see table below)
5 (2)	IERROR	Computation of standard errors (0 =whole matrix, 1 = diagonal)
6 (2)	IGLOBAL	Model identification (0 = overpopulation, 1 = last population, 2 =fixed item and/or population parameters)
7 (2)	IML	ML estimates theta (1 = Yes)
8 (2)	IWML	WML estimates theta (1 = Yes)
9 (2)	IEAP	EAP estimates theta (1 = Yes)
10 (4)	IGROUPS	Indicator of scoregroups (the numer of groups from 2-6, default =3)
11 (2)	IFIRST	First order statistics (1 = Yes)
12 (2)	MARG	Ability distribution (0= one ability distribution for all booklets, 1 = an ability distribution for each booklet)
13 (2)	ISCORE	Add-on: Observed score equating (1 = Yes)
14 (2)	ISECOND	Second order statistic (1 = Yes)
15 (2)	INFVAL	Add-on: Information Values (1 = Yes)
16 (2)	IDIF	DIF Statistics (0 = NO, 1 = Booklets, 2 = COVARIATE)
17 (2)	ECHO	Echo starting values (1 = Yes)
18 (2)	INFO	Output Information matrix (1 = Yes)
19 (2)	COV	Output Covariance matrix parameter estimates(1 = Yes)
20 (2)	IML_TOT	Person pars. ML, Number-correct scores (1 = Yes)
21 (2)	EAP_TOT	Person pars. EAP, Number-correct scores (1 = Yes)
22 (2)	OPTI_W	Optimal Weights (1 = Yes)
23 (2)	FIX_W	Fixed weights (1 = Yes)
24 (2)	NDIM	Number of dimensions (0 <= NDIM <= 10), with 0 in table "Dim"means items loading on all dimensions
25 (2)	IFIX	Fixed item parameters (1 = Yes)
26 (2)	NCOV	Number of covariates
27 (2)	CUT_OFF	Number of cut-off points
28(2)	IRUN	Indicator stating whether run-button was clicked (1) or the save TTT button (0)
29(2)	ITTT	Indicator of having a Manual Fortran Format filled in (1) or not (0)

Available model codes or combinations as in record 1, position 4:

Code	Model	Parameters	Estimation
0	No model	None/any	None/any
1	GPCM	1PL	MML
2	GPCM	1PL	MCMC
3	GPCM	1PL	CML
4	GPCM	2PL	MML
5	GPCM	2PL	MCMC
6	GPCM	2PL	CML
7	--	--	--
8	GRM	1PL	MCMC
9	--	--	--
10	--	--	--
11	GRM	2PL	MCMC
12	--	--	--
13	SeqM	1PL	MML
14	SeqM	1PL	MCMC
15	SeqM	1PL	CML
16	SeqM	2PL	MML
17	SeqM	2PL	MCMC
18	SeqM	2PL	CML

Record 2. Indication (I2) = Person ID present yes/no, Yes=1) + the specified working directory path

Record 3. Three lines indicating the type of priors (0=no, 1=fixed, 2=floating) and the 2 values of each prior

Record 4. Run Name (spaces are automatically changed into underscores)

Record 5. If item parameter file fixed item parameters = TRUE, then the file location, else this record is blank

Record 6. Comment for output

Record 7. Address of the Data file

Record 8. Format data file (first entry position booklet number, length of booklet nr, if present the Person ID and it's length,then first position data and length of responses, then only if present the first position of the covariates and their lengths (can vary for each covariate)). Position 201 contains a flag equal to 0 if the format is entered using the position buttons in the Shiny app, or equal to 1 if a Manual Fortran Format was specified in the Shiny app. The value should be equal to the value of the variable ITTT on the first line of the TTT file. The value of ITTT overrides the value in Record 8.

Record 9. (KTOETS(booklet): Number of items in booklet), booklet = 1,…,NTOETS

Record 10. (Ability distribution of booklet), booklet = 1,…,NTOETS (Now fixed on 1, because no marginal ability distributions exist anymore)

Record 11. Indication if booklet is switched on or off for each booklet (booklet on (=1) or off (=0), booklet = 1,…,NTOETS)

As of Record 11, the length of the records differ depending on the settings

Record 12. Records for every booklet.
For every booklet, the included item numbers, then for every booklet a switch (0=item off, 1 = item on) based on the product of the global off/on switch (in 'General') and local off/on switch (in 'Booklets')

Record 13: For every booklet: booklet labels (20 positions), the number of Simulations (5 positions), whether Equating errors should be computed for the booklet (1=yes, 0=no), flag for CAT data (1 position on position 28, 1=yes, 0=no), CAT test length individual test taker (3 positions starting on position 34), and CAT Precision (8 positions starting on position 37)

Record 14: For every item: Item label, Number of response categories, Scoring weight, Guessing parameter (1=Yes),
Dimension on which the item loads (<= NDIM)

Record 15: For every covariate: covariate labels, then for every covariate a switch (0=off 1 =on) and the Length of the data.

Record 16. Convergence criteria for the 3 phases (I-II-III), number of quadrature points, iterations & changelog

Record 17. If record1, (position 23, record 1) FIX_W, Fixed weights is on, then the weights per dimension, 3 positions per integer weight.

Record 18. If Cut-off points option (position 27, record 1) is larger than 0, and IWML=1 (position 8, record 1) then for every cutoff point, a label of the cut-off must be present (20 positions), with the cut-off boundary behind it (F10.4)

Record 19. Optional MCMC settings. First 8 positions are the number of MCMC iterations (up to 100.000), then 8 positions for the number of burn-in iterations (up to 10.000) and finally 8 positions for the iteration cluster size for testing convergence (up to 5.000)