June 2019

Graphing IRT Test Information, SEE, And Log Likelihood Function in SAS

Bo Klauth
Western Michigan University

Abstract

Statistical applications make the calculations of IRT models easier and more accurate than hand calculation. However, not all applications provide the users with some outputs they need. PROC IRT in SAS does not have an option to obtain a graph on test information, standard error of estimate (SES) and log likelihood function (LLF) for a score pattern. The purpose of this paper is to apply dichotomous score IRT theories of one-parameter (1PL), two-parameter (2PL), three-parameter (3PL), and four-parameter (4PL) models and build a macro producing outputs necessary for building a graph for test information and SEE and LLF by score pattern in SAS. The calculation method used is Maximum Likelihood (ML). The users can exploit the given outputs for other uses.

Keywords: IRT, test information, SEE, Log likelihood, Score pattern, SAS.

The SAS scripts for the macro and associated files can be found in a Github repository at `https://github.com/Boklauth/Graphing-IRT`.

1. Introduction

IRT model was introduced in the field of psychometrics for ability assessment purposes. Currently, it is used to build psychological and educational tests because IRT can improve measurement accuracy and reliability (SAS Institute Inc, 2017). As noted by (Cai et al., 2016), IRT has been used in the social and behavioral sciences in the academic settings and in high profile assessment programs, including but not limited to, the Organization for Economic Cooperation and Development (OECD) Program for International Student Assessment (PISA), the National Assessment of Educational Progress (NAEP), the National Institutes of Health's Patient Reported Outcomes Measurement Information System initiative, and China's new National Assessment of Basic Education Quality.

There are over 60 applications for calculating IRT. Some of them can perform a calculation with large datasets, including many test items and observations (Zhao and Hambleton, 2009). SAS, SPSS, Scientific Software International (Bilog / Lisrel / HLM / Parscale / Multilog, etc.), and Lahey Fortran are commercial software that can analyze IRT models (UMass, nd). In addition, fifty-seven packages related to IRT in R have been developed (Mair, 2018).

Those applications make the calculations of IRT models easier and more accurate than hand calculation. However, not all applications provide all pieces of the calculations the users need. PROC IRT in SAS does not provide test information function with a standard error of estimate. It does not have an option for the user to see the score pattern and find the maximum log likelihood function. In addition, people who wish to replicate the calculations of dichotomous IRT models in order to learn what calculation is involved may need to spend an enormous amount of time and efforts to achieve that. Having an already written code that produces table outputs for manipulation to fit other uses would be helpful.

The purpose of this paper is to apply dichotomous score IRT theories of one-parameter (1PL), two-parameter (2PL), three-parameter (3PL), and four-parameter (4PL) models and build a macro producing outputs necessary for building a graph for test information and standard error of estimate (SEE) and LLF by score pattern in SAS. The users can exploit the given outputs for other uses. For educational purposes, this macro may help students understand how they can obtain necessary parameters to calculate a probability of obtaining a correct item. It provides additional graphs in addition to what has already provided by PROC IRT. Therefore, this paper will demonstrate two main graphs for a dichotomous test in IRT for the four models: (1) test information and SEE by person location (θ) and (2) LLF by score pattern.

2. Review of Theories in IRT Models

IRT relies on several assumptions. A widely used assumption in IRT is unidimensonality. That is, a set of test items only measures one ability. Although examinees tend to be influenced by other factors when taking a test, for a set of test data to meet the unidimensional assumption sufficiently requires the test to measure a dominant component or factor that influences test performance. Another related assumption is local independence or conditional independence. It means that examinees' responses to different items are not related when examinees' abilities are partialled out or held constant, provided that the test is unidimensional. Stated another way, examinees' responses to any pair of items are statistically independent, holding the abilities influencing the test performance constant. Mathematically, this assumption means that the product of probabilities associated with the examinees' responses to the individual items is equal to the probability of a response pattern on a set of items. (Hambleton et al., 1991; de Ayala, 2009). Another assumption related to IRT models is that the true relationship among the abilities and item responses can be represented by the item characteristic function or item characteristic curve (ICC). The final assumption is that item characteristics are relevant to an examinee's performance on an item (Hambleton et al., 1991).

Four IRT models will be presented. They are 1PL, in which the Rasch model is included; 2PL; 3PL (de Ayala, 2009); and 4PL model (Barton and Lord, 1981).

2.1. The 1PL Model

In the 1PL model, it is assumed that the construct being measured, which is a latent variable conceptualized as a latent continuum, is manifested through an individual's responses to a series of items. In this model, there is only one parameter that characterizes the test items. This parameter is the item's location on the latent continuum that represents the construct (de Ayala, 2009). A formula for calculating this model is given by (Hambleton et al., 1991; de Ayala, 2009):

$$p(x_j = 1|\theta, \delta_j) = \frac{e^{\alpha(\theta-\delta_j)}}{1 + e^{\alpha(\theta-\delta_j)}} \tag{1}$$

$$p(x_j = 1|\theta, \delta_j) = \frac{1}{1 + e^{-\alpha(\theta-\delta_j)}} \tag{2}$$

where j = 1, 2, ..., L; L is the number of items in the test; $p(x_j = 1|\theta, \delta_j)$ is the probability that a randomly chosen examinee with ability θ answers item j correctly; α is the steep of item characteristic curve of test items; δ_j is the item j's difficulty parameter or item j's location; and e is a transcendental number whose value is 2.718 (correct to three decimals).

De Ayala showed the way to work with the exponent term (2009). Notice that the exponent term $(\theta - \delta_j)$ has a multiplier of α. Then the exponent term $\alpha(\theta - \delta_j)$ can be simplified further: $\alpha(\theta - \delta_j) = \alpha\theta - \alpha\delta_j$

$$= \alpha\theta + (-\alpha\delta_j) \tag{3}$$

Let $\beta_j = -\alpha\delta_j$. Then,

$$\delta_j = -\frac{\beta_j}{\alpha} \tag{4}$$

Equation 2 becomes $\beta_j + \alpha\theta$, which is in a linear form; β_j is an intercept related to item location (δ_j); α is the slope; and θ, person location, functions as a predictor and may range from negative infinity to positive infinity (de Ayala, 2009).

The Rasch model, which is a special kind of 1PL model, has a constant α for each item. The difference is that, in the Rasch model, α is equal to 1; but in the 1PL model, α has a different value than 1.

2.2. The 2PL Model

In this model, α is allowed to vary across test items j. Thus, a constant α in the 1PL model has become α_j to denote its variations across test items. Thus, the probability of getting an item correct is expressed in the equation below.

$$p(x_j = 1|\theta, \delta_j) = \frac{e^{\alpha_j(\theta-\delta_j)}}{1 + e^{\alpha_j(\theta-\delta_j)}} \tag{5}$$

Where α_j is a steep for test item j.

2.3. The 3PL Model

In testing, some individuals with a low ability may be expected to respond correctly to some of the test items because of guessing, especially if the test uses multiple choice or true/false response format. Thus, these items' response functions have a non-zero lower asymptote. The 3PL model has been developed to address this issue (de Ayala, 2009).

If a person responds to an item correctly according to his or her ability, the probability of getting the item correct can be calculated using Equation 5 in the 2PL model, and let the probability be p_j. Let c_j be the person′s probability of getting an item j correct based on chance or guessing alone. When a person has a lower ability on the trait continuum, the probability of getting item j correct is approaching zero. However, with chance or guessing, the probability of getting item j correct can be represented by $c_j(1-p_j)$. By combining these two probabilities, we obtain a probability of a correct response with the following equation:

$$p(x_j = 1|\theta, \alpha_j, \delta_j, c_j) = p_j + c_j(1 - p_j) \tag{6}$$

where p_j is the probability of getting item j correct in the 2PL model; c_j is the probability of getting item j correct by chance or guessing or is also called a “pseudo guessing” or “pseudo chance parameter” (de Ayala, 2009, p. 124).

By rearranging Equation 6, $p(x_j = 1|\theta, \alpha_j, \delta_j, c_j) = c_j + (1 - c_j)p_j$. Simplifying p_j, we will get:

$$p(x_j = 1|\theta, \alpha_j, \delta_j, c_j) = c_j + (1 - c_j)\frac{e^{\alpha_j(\theta-\delta_j)}}{1 + e^{\alpha_j(\theta-\delta_j)}} \tag{7}$$

2.4. The 4PL Model

Out of the concern that the 3PL model might arbitrarily estimate high-ability students, this model has been developed to account for the situations in which a high-ability student may make a mistake and obtain a wrong answer on an easy item. Thus, an upper asymptote can have a value lower than 1 for this student (Barton and Lord, 1981). To estimate the probability of getting an item correct in this situation, (Barton and Lord, 1981) used the following equation:

$$p(x_j = 1|\theta, \alpha_j, \delta_j, c_j, d_j) = c_j + (d_j - c_j)\frac{e^{\alpha_j(\theta-\delta_j)}}{1 + e^{\alpha_j(\theta-\delta_j)}} \tag{8}$$

Where, d_j is the upper asymptote or the probability of getting an item correct due to clerical error (Barton and Lord, 1981) or inattentive level (Magis, 2013).

3. Methods

3.1. SAS Environment for Code Development

The macro was developed using SAS 9.4, which was running on personal computer with Windows 10 Home Edition, 8 GB of RAM, and a CPU speed of 2.70 GHz. For SAS system requirements, please see “System Requirements for SAS 9.4 Foundation for Microsoft Windows for x64”(SAS Institute Inc, 2019). It is recommended that SAS default result features are maintained to see the outputs. Users may enable the output delivery in HTML format.

3.2. Data

The raw scores of eight dichotomous test items with 170 out of 256 score patterns were generated. Each pattern was given different frequencies, and the total number of observations in the dataset was 1,024. There were no missing data. The complete SAS script to generate the raw score can be found in the Appendix section.

3.3. Calculation Methods

Two main goals of the calculation were aimed to generate (1) a graph for test information by SEE and (2) a graph for log likelihood function by score pattern. To achieve these two goals, I used PROC IRT with Maximum Likelihood method to generate a table output, which includes intercept (β_j), slope (α_j), guessing (c_j), and ceiling (d_j). One parameter missing from the SAS output is the item location or item difficulty. According to Equation 3, item difficulty $\delta_j = \frac{-\beta_j}{\alpha}$. If α varies across item j, for 2PL, 3PL, and 4PL models, then the formula for obtaining item difficulty should be: $\delta_j = \frac{-\beta_j}{\alpha_j}$. These parameters were then used to calculate a probability of getting an item correct, item information, total test information, and SEE. In calculating the probability of obtaining a correct answer on an item, the person's ability θ was selected to range from -4 to +4 with an increment of 0.01. The range is 8. Within the range, the number of ability points is $\frac{8}{0.01} + 1 = 801$. The results included test information, SEE, and θ to produce a graph.

To create a graph for LLF by score pattern, I utilized the parameters used in calculating a probability of getting an item correct to calculate the log likelihood function for each person and score pattern. For one observation in the dataset, the p_j calculation will utilize 801 values of trait ability, creating 801 new observations. The dataset contains 1,024 observations and the calculation will create 1,024 times 801 observations, equaling 817,224.

To ease in computational programming, I used Equation 8 to apply to all the four parameter logistic models. In other words, for computational purposes Equation 8 (the 4PL model) can be used as a generic equation for the Equations 1 (the 1PL model), 5 (the 2PL model), and 7 (the 3PL model) to calculate the probability of getting item j correct, although it is not theoretically appropriate to do so. The reason is that PROC IRT will yield a value of 0 for c_j (guessing) and a value of 1 for d_j (inattentiveness) for the 1PL and 2PL models. Thus, the equation $p_j = c_j + (d_j - c_j)\frac{e^{\alpha_j(\theta-\delta_j)}}{1+e^{\alpha_j(\theta-\delta_j)}}$ will be equal to

$$p_j = 0 + (1+0)\frac{e^{\alpha_j(\theta-\delta_j)}}{1+e^{\alpha_j(\theta-\delta_j)}} \tag{9}$$

$$p_j = \frac{e^{\alpha_j(\theta-\delta_j)}}{1+e^{\alpha_j(\theta-\delta_j)}} \tag{10}$$

When it comes to the 3PL model, PROCT IRT will yield a non-zero value for c_j and 1 for d_j. Thus, $p_j = c_j + (d_j - c_j)\frac{e^{\alpha_j(\theta-\delta_j)}}{1+e^{\alpha_j(\theta-\delta_j)}}$ will become the 3PL model's equation: $p_j = c_j + (1 - c_j)\frac{e^{\alpha_j(\theta-\delta_j)}}{1+e^{\alpha_j(\theta-\delta_j)}}$. The macro script for this calculation can be found in the Appendix.

4. Results and Outputs

After running PROC IRT, the macro extracts parameters such as c_j (guessing), d_j (inattentiveness), intercept, and slope (α). Then, it calculates item difficulty (δ_j). For the 1PL and 2PL models, for each item d = 1 and c = 0 (see Tables 1 and 2). The slopes for the 1PL

Table 1: Guessing, Inattentiveness, Intercept, Slope, and Item Difficulty Parameters 1PL)

Obs	_VAR1_	d	c	intercept	slope	item_diff
1	item1	1	0	0.92234	1.55506	-0.59312
2	item2	1	0	2.15162	1.55506	-1.38363
3	item3	1	0	1.03965	1.55506	-0.66856
4	item4	1	0	2.39170	1.55506	-1.53802
5	item5	1	0	0.60033	1.55506	-0.38605
6	item6	1	0	1.81508	1.55506	-1.16721
7	item7	1	0	1.27008	1.55506	-0.81674
8	item8	1	0	2.10815	1.55506	-1.35567

Table 2: Guessing, Inattentiveness, Intercept, Slope, and Item Difficulty Parameters 2PL)

Obs	_VAR1_	d	c	intercept	slope	item_diff
1	item1	1	0	1.03661	1.93083	-0.53687
2	item2	1	0	2.44169	1.97313	-1.23747
3	item3	1	0	0.93444	1.23633	-0.75582
4	item4	1	0	2.11584	1.14910	-1.84130
5	item5	1	0	0.68076	1.95684	-0.34788
6	item6	1	0	2.02465	1.91273	-1.05851
7	item7	1	0	1.21382	1.42249	-0.85331
8	item8	1	0	1.83369	1.10109	-1.66533

model are constant, but the slopes for the 2PL model vary across items. For the 3PL model, d = 1, but c for each item no longer equals zero (see Table 3). For the 4PL model, d and c values vary among items (see Table 4). Like the 2PL model, the slopes for these models are not constant.

The macro also produced a frequency table of total score (Table 5) and of score pattern (Table 6).

4.1. Test Information and Standard Error of Estimate (SEE)

In classical test theory (CTT), the true score is the expected observed score with the propensity distribution of a person on a given measurement. The difference between the observed score (X) and true score (T) is commonly referred to as the error of measurement (E). Here, E and X are random variables, and T is a constant, the expected value of observed score X (Lord and Novick, 2008). The standard error of measurement is assumed to be constant

Table 3: Guessing, Inattentiveness, Intercept, Slope, and Item Difficulty Parameters (3PL)

Obs	_VAR1_	d	c	intercept	slope	item_diff
1	item1	1	0.00000	1.00670	1.90723	-0.52783
2	item2	1	0.19776	2.25070	2.37428	-0.94795
3	item3	1	0.18973	0.55412	1.57508	-0.35180
4	item4	1	0.61262	0.84480	2.61777	-0.32272
5	item5	1	0.00000	0.65739	1.94637	-0.33775
6	item6	1	0.30318	1.74389	3.13917	-0.55552
7	item7	1	0.00000	1.20925	1.44167	-0.83878
8	item8	1	0.56142	0.62273	2.41088	-0.25830

Table 4: Guessing, Inattentiveness, Intercept, Slope, and Item Difficulty Parameters (4PL)

Obs	_VAR1_	d	c	intercept	slope	item_diff
1	item1	0.95167	0.00000	1.39593	2.30010	-0.60690
2	item2	1.00000	0.21972	2.20742	2.40131	-0.91926
3	item3	1.00000	0.21246	0.49952	1.65805	-0.30127
4	item4	1.00000	0.61824	0.81918	2.67975	-0.30569
5	item5	1.00000	0.00000	0.66416	1.96263	-0.33840
6	item6	0.99028	0.32667	2.02535	3.79947	-0.53306
7	item7	0.96829	0.00000	1.44933	1.63574	-0.88604
8	item8	1.00000	0.58492	0.49364	2.74400	-0.17990

Table 5: Frequency of Total Score

Obs	total	Frequency	PERCENT
1	1	40	3.9063
2	2	43	4.1992
3	3	81	7.9102
4	4	77	7.5195
5	5	129	12.5977
6	6	162	15.8203
7	7	186	18.1641
8	8	306	29.8828

Table 6: Frequency of Score Pattern

Obs	score_pattern	Frequency	PERCENT
1	00000001	10	0.97656
2	00000010	3	0.29297
3	00000011	1	0.09766
4	00000100	6	0.58594
5	00000101	3	0.29297
6	00000111	1	0.09766
7	00001001	2	0.19531
8	00001011	1	0.09766
9	00010000	13	1.26953
10	00010001	7	0.68359
...	....	...	...
170	11111111	306	29.8828

for all persons for a test. With some important distinctions in the underlying philosophies and statistical calculations, item response theory (IRT) extends the theories further (Wang and Osterlind, 2013). The standard error of estimate in IRT for the whole test or instrument varies with the trait or ability level θ. The confidence intervals also vary (Fan and Sun, 2013). As De Ayala (2009) and Magis (2013) demonstrated, in IRT SEE for 1PL, 2PL, 3PL, and 4PL models are calculated differently. Therefore, they have different shapes (see Figs. 1, 2, 3, and 4 in the Results section). For 1 and 2PL models, the test information can be calculated using the following formula:

$$I(\theta) = \sum_{j=1}^{L} \alpha_j^2 p_j (1 - p_j) \tag{11}$$

For the 1PL model, α_j is the same for every item; however, for the rest of the models α_j may vary across test items. For the 3PL model, the test information formula (de Ayala, 2009) is involved with the pseudo guessing parameter (c_j, lower asymptote parameter):

$$I(\theta) = \sum_{j=1}^{L} \alpha_j^2 \frac{(p_j - c_j)^2 (1 - p_j)}{(1 - c_j)^2 p_j} \tag{12}$$

The test information for the 4PL model needs to include the pseudo guessing parameter (c_j) and the probability of getting an item correct due to inattentiveness (d_j, upper asymptote parameter). It can be expressed with the following equation (Magis, 2013):

$$I(\theta) = \sum_{j=1}^{L} \alpha_j^2 \frac{(p_j - c_j)^2 (d_j - p_j)^2}{(d_j - c_j)^2 p_j (1 - p_j)} \tag{13}$$

SEE for every model follows the same generic formula SEE $=\sqrt{\frac{1}{I(\theta)}}$. However, the test

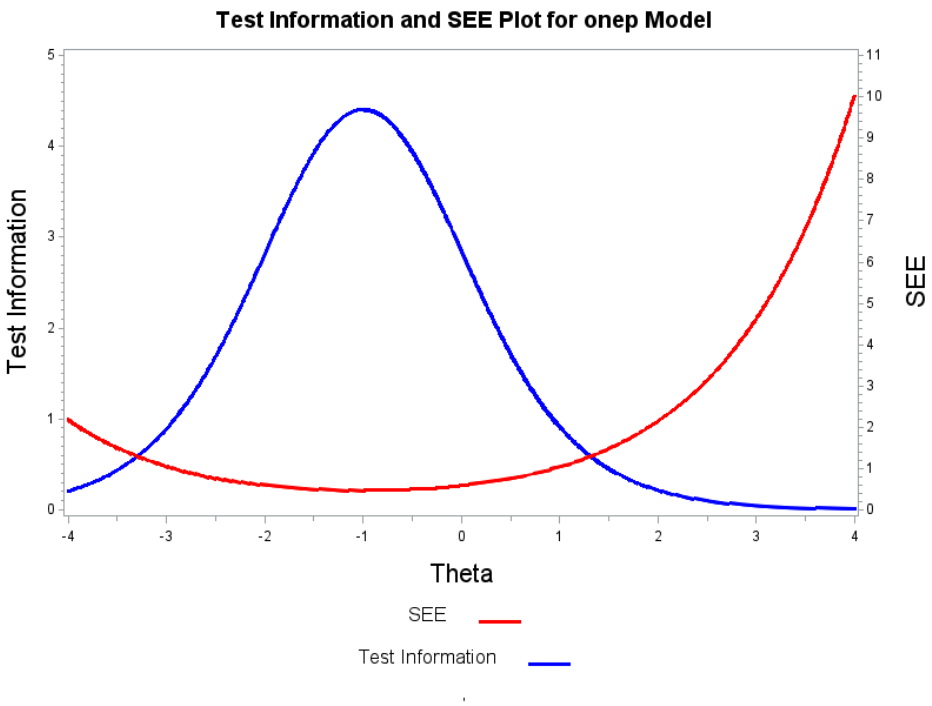

Figure 1: Test Information and SEE Plot for 1PL Model.

information needs to be replaced by a corresponding parameter logistic model as described above (de Ayala, 2009; Magis, 2013).

4.2. The Log-Likelihood Function (LLF)

In estimating person and item parameters, there are two approaches: joint maximum likelihood estimation (JMLE) and marginal maximum likelihood estimation (MMLE) (de Ayala, 2009). This method maximizes the joint likelihood function of both persons and items in order to simultaneously estimate both person and item parameters. There are two issues using this method to estimate both person and item parameters. One issue is related to estimating the item parameters (structural parameters) that are associated with the person parameters (incidental parameters). This may requires a large sample size, which leads to having more person parameters to be estimated. Still, this approach has been found to create biased estimates. The second issue is related to efficiency in the process of calculation. When some items are identified as misfits, the instrument may need to be recalibrated to remove the bad items out. Consequently, there is a need to re-estimate the person and item perameters (de Ayala, 2009).

It is known that estimating item and person parameters separately improves the theoretical accuracy for some instruments (de Ayala, 2009). MMLE estimates item parameters first. After obtaining item parameter estimates and achieving model-data fit, person parameters can be estimated, using a maximum likelihood estimation (MLE), expected a posteriori (EAP, also known as Bayes Mean Estimate), or maximum a posteriori (MAP, also known as Bayes Model Estimate) (de Ayala, 2009).

Below, I used the formula in MMLE for estimating person parameters after having the item

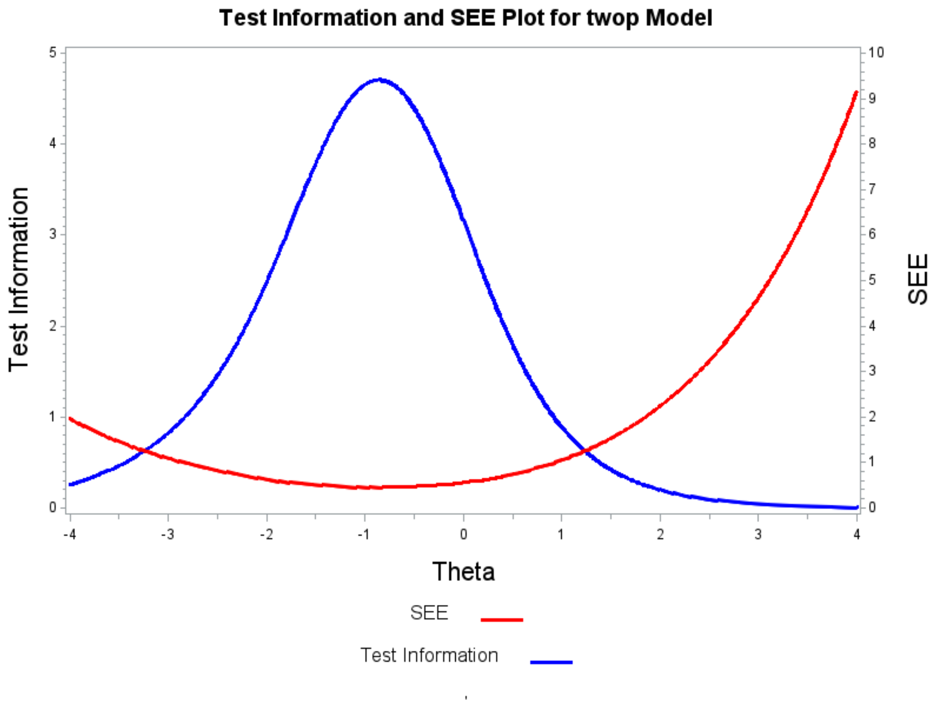

Figure 2: Test Information and SEE Plot for 2PL Model.

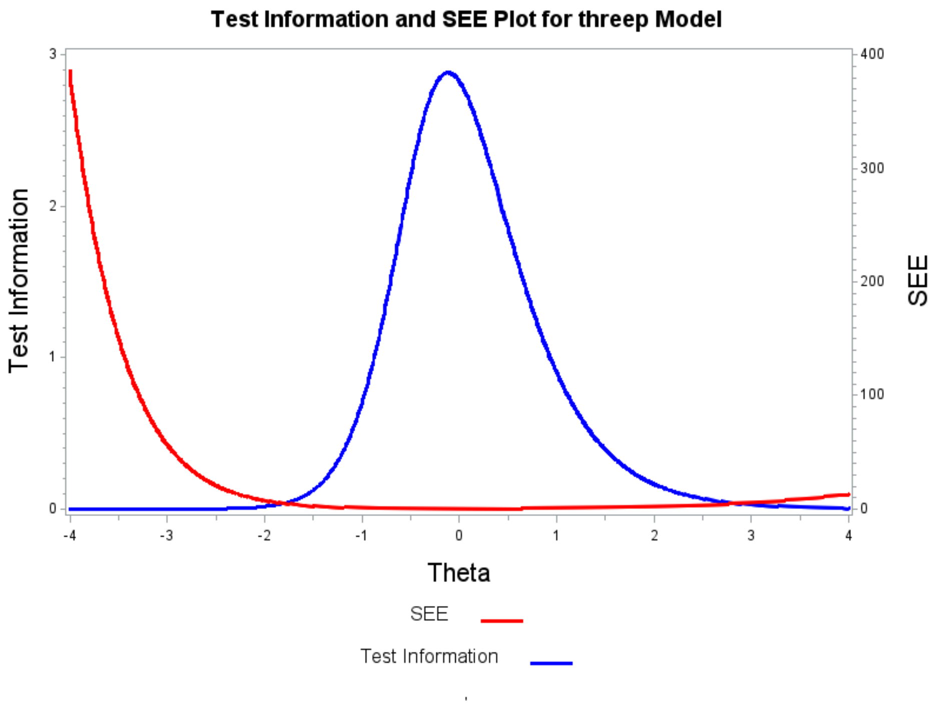

Figure 3: Test Information and SEE Plot for 3PL Model.

parameters. A log likelihood function is a key to estimating a person's location. It is a way

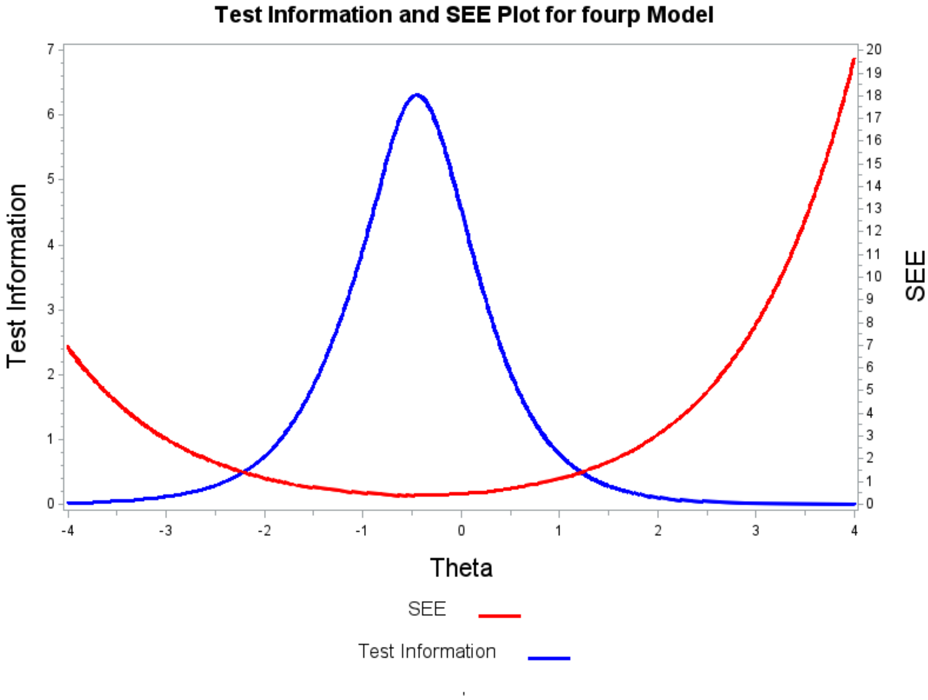

Figure 4: Test Information and SEE Plot for 4PL Model.

to determine which person location on the trait continuum (from -4 to +4) has the highest likelihood of producing a certain score pattern. This can be approached by using the formula for each model to calculate the probability of getting a score for all items. de Ayala (2009) provided the following formula:

$$p_L = \prod_{j=1}^{L} (p_j)^{x_{ij}} (1 - p_j)^{1 - x_{ij}}, \tag{14}$$

where p_L is the probability of getting a score pattern with an L length of a test; x_{ij} is person i's response to the item. Equation 12 is equivalent to another equation using the log likelihood function [LLF](de Ayala, 2009), where

$$LLF = \sum_{j=1}^{L} [x_{ij} ln(p_j) + (1 - x_{ij}) ln(1 - p_j)] \tag{15}$$

The macro will use both formulae. To find the person location for a score pattern is to find the maximum LLF associated with that score pattern. For example, a score pattern "01010101" may have multiple LLFs associated with multiple theta values. Only the maximum LLF is selected to find its associated theta value. Please see Figs. 5, 6, 7, and 8.

5. How to Use irtTheta and Limitations

Two SAS files accompany this paper. One file named "irtDataSimulation.sas," is a SAS script that creates a sample dataset. The other file is the macro file named "irtThetaMacro.sas." The following steps describe how to use the files.

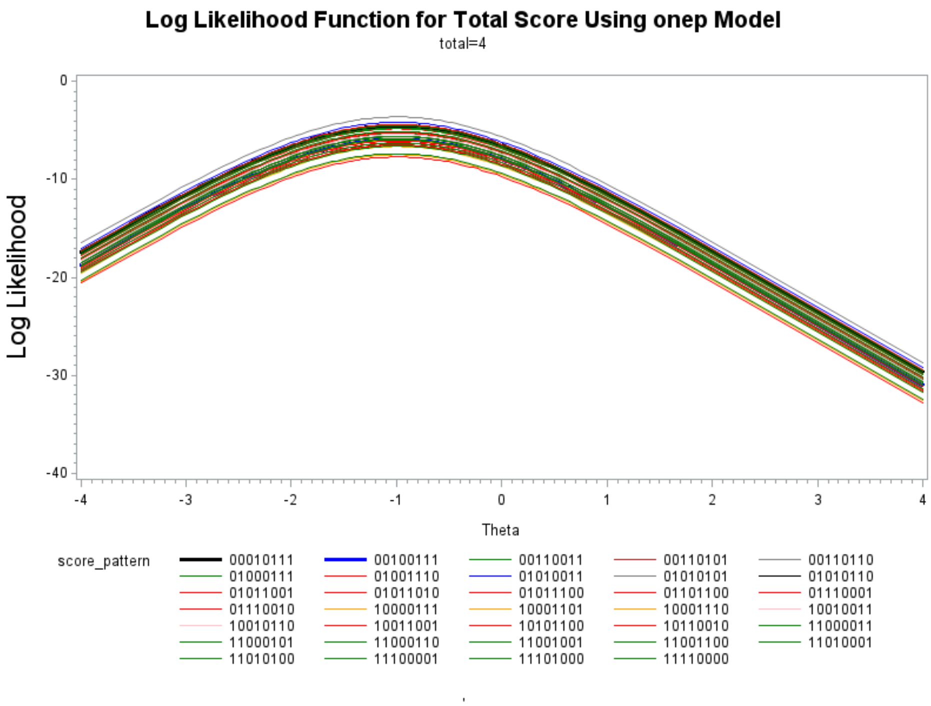

Figure 5: Log-Likelihood Function and Score Pattern for Total Score = 4 (1PL Model)

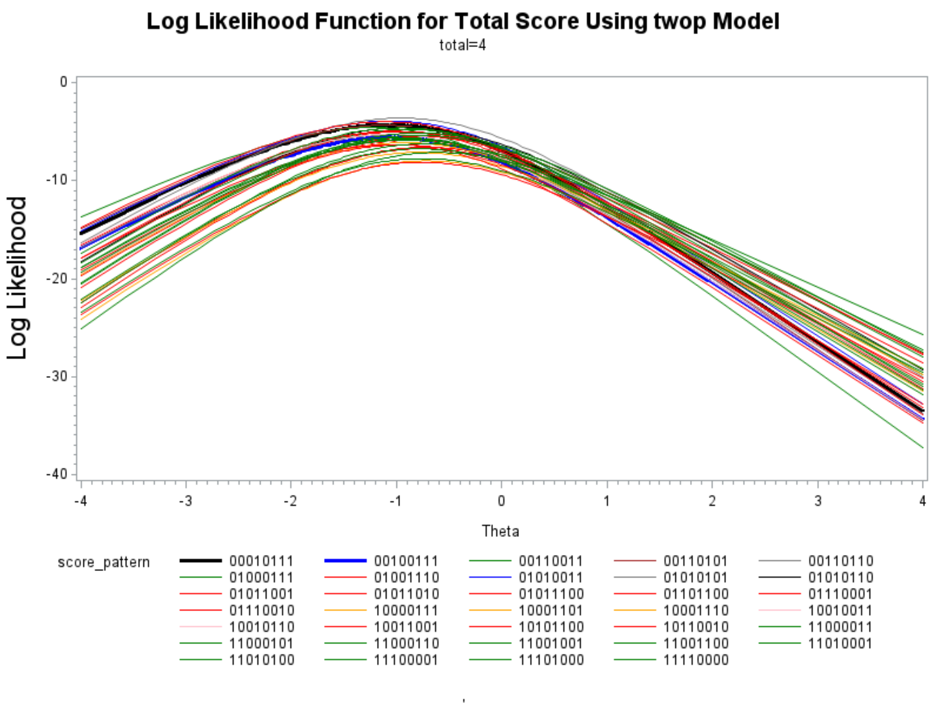

Figure 6: Log-Likelihood Function and Score Pattern for Total Score = 4 (2PL Model)

1. Getting a sample dataset: open the file named “irtDataSimulation.sas” and run the whole

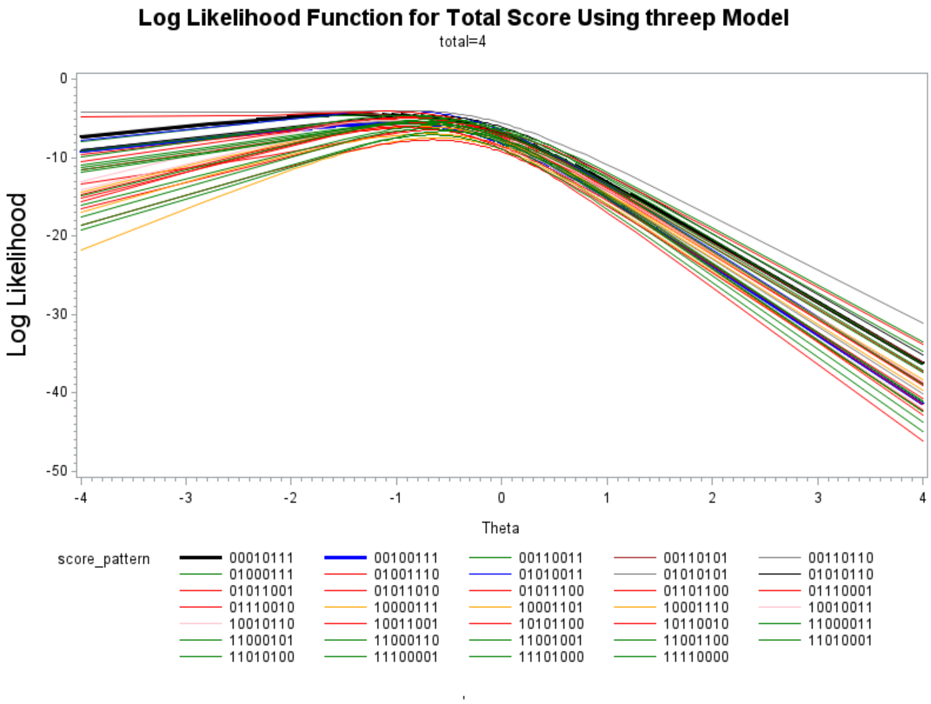

Figure 7: Log-Likelihood Function and Score Pattern for Total Score = 4 (3PL Model)

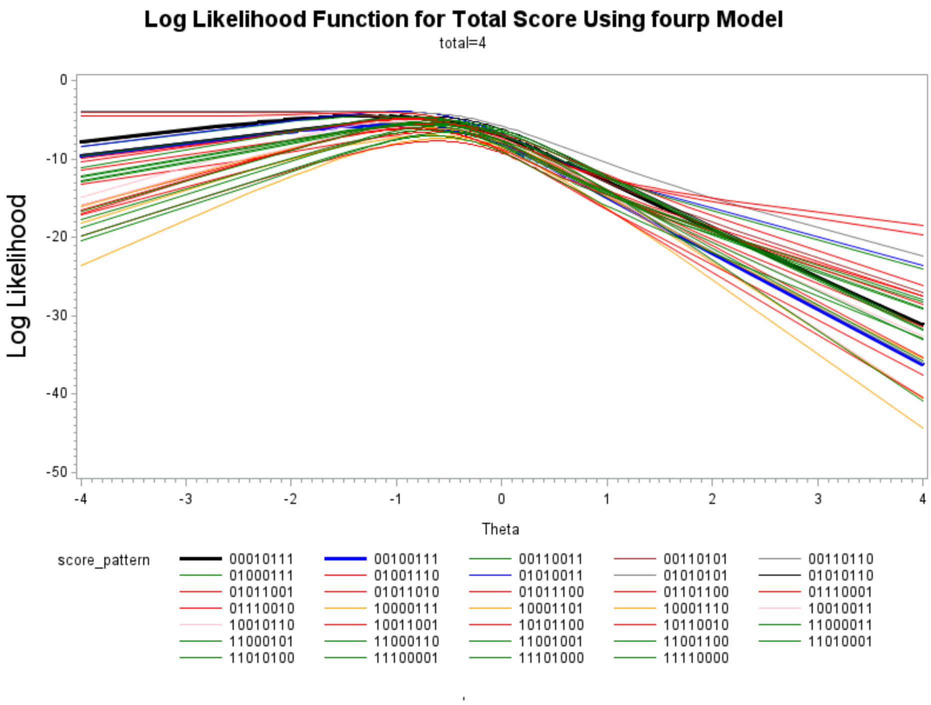

Figure 8: Log-Likelihood Function and Score Pattern for Total Score = 4 (4PL Model)

script. A SAS dataset named “SampleData2”will be generated and stored in the “Work” library.

2. Download the macro file and save it in a folder.

3. Open a new SAS editor and include the path to the macro. For example, "%include 'C:\Users\Dell\irtThetaMacro.sas';"

4. Fill out appropriate variable names for the macro. For example, %irtTheta(ds =, model =, start_var =, end_var =, nitems =, method =, ods_graphics = off), where ds = [enter your dataset name]; model = [enter your model]; the models available are onep, twop, threep, and fourp; start_var = [the name of your first test item in your dataset]; end_var = [the name of your last test item in your dataset]; nitems = [the total number of test items]; method = ML ; ods_graphics = [off/on]; selecting "on"will produce item characteristic curves using PROC IRT.For example: %irtTheta(ds = SampleData2, model = twop, start_var = item1, end_var = item8, nitems = 8, method=ML, ods_graphics = off);

5.1. SAS File Produced by the irtTheta

This macro produces four additional important file outputs that are not provided by PROC IRT. First, "Iteminfo_see"is a table output used to produce a graph on test information and SEE. Second, "Graph_loginfo"is a table output used a build a graph on score pattern based on log likelihood function. Third, the file "Source"contains all parameters associated with each item and person. Finally, the file "Comparing_theta"contains theta calculated by the macro and by PROC IRT.

5.2. Limitations

There are three limitations of the macro. First, the method of estimation used is ML (SCOREMETHOD = ML). When ML is used to estimate the intercept and slope in PROC IRT, it yields identical results to the estimates calculated by the macro. Other methods (EAP and MAP) are out of the scope of this paper. In PROC IRT, these methods may not yield the identical results to that yielded by the macro.

The second limitation is the difference θ values at the ends of the continuum (-4 and +4) produced by the irtTheta macro and PROC IRT. When $\theta = -4$ or $\theta = 4$, SAS tends to have values beyond -4 or 4. If the macro has person location θ = -4, SAS will estimate the person location _Factor1 <-4; when the macro has the person location θ = 4, SAS will estimate person location _Factor1 >4.

Finally, this macro has not been equipped with error handing methods. So, it does not give any warnings about a failed calculation under a model. Not all datasets can be run using "threep"(3PL) and "fourp"(4PL). That is because the iterations cannot be converged. When convergence is not met, it may be a sign that the dataset does not fit the "threep"or "fourp"model. When convergence is not met, PROC IRT will fail, and the macro will fail. Another instance is that when a dataset is performed by using PROC IRT, yielding incomplete data. In this situation, the irtTheta macro will also fail. It is likely that when the dataset can be calculated using the "threep"or "fourp"model, it is going to work with "onep"and "twop"models.

6. Summary and discussion

As PROC IRT does not show the log-likelihood function and the probability of getting an item correct. It is time consuming to write code from scratch to get those estimates. Having a macro that can produce the outputs may help the user. This paper has cited existing theories and applied them to building a SAS macro using PROC IRT. This macro has provided users with some outputs that PROC IRT does not. The outputs can be used to build a graph for test information by SEE, a graph for maximum log likelihood function by score pattern. The macro has also provided another file that contains parameters and probabilities for each item j and person i in the dataset. Another file provides a comparison on θ estimates by the macro and PROC IRT. The results are identical across trait continuum, except the ends of its range (-4 and +4).

Acknowledgments

I am grateful to Dr. Brooks Applegate, who helped debug some errors of the graphs and answered my questions related to IRT.

References

Barton, M. A. and Lord, F. M. (1981). An upper asymptote for the three-parameter logistic item-response model. *ETS Research Report Series*, 1981(1):i–8.

Cai, L., Choi, K., Hansen, M., and Harrell, L. (2016). Item Response Theory. *Annual Review of Statistics and Its Application*, 3(1):297–321.

de Ayala, R. J. (2009). *The theory and practice of item response theory.* The Guilford Press, New York, NY.

Fan, X. and Sun, S. (2013). Item Response Theory. In Teo, T., editor, *Handbook of Quantitative Methods for Educational Research.* SensePublishers, Rotterdam.

Hambleton, R. K. and Swaminathan, H. (1985). *Item response theory: principles and applications / Ronald K. Hambleton, Hariharan Swaminathan.* Evaluation in education and human services. Kluwer-Nijhoff Pub, Distributors for North America, Kluwer Boston.

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of Item Response Theory.* Sage Publications, Newbury Park, CA.

Lord, F. M. and Novick, M. R. (2008). *Statistical theories of mental test scores Frederic M. Lord and Melvin R. Novick ; with contributions by Allan Birnbaum.* Addison-Wesley series in behavioral sciences. Quantitative methods. Information Age Pub, Inc, Charlotte, N.C.].

Magis, D. (2013). A Note on the Item Information Function of the Four-Parameter Logistic Model. *Applied Psychological Measurement*, 37(4):304–315.

Mair, P. (2018). CRAN task view: Psychometric models and methods.

SAS Institute Inc (2017). Sas/stat 14.3 user's guide: the irt procedure.

SAS Institute Inc (2019). System requirements for SAS 9.4 foundation for microsoft windows for x64.

UMass, A. (n.d.). UMassAmherst: Research, Educational Measurement, and Psychometrics (REMP).

Wang, Z. and Osterlind, S. J. (2013). Classical test theory. In Teo, T., editor, *Handbook of Quantitative Methods for Educational Research*. Sense Publishers, Rotterdam.

Zhao, Y. and Hambleton, R. (2009). Software for IRT analyses: descriptions and features. Technical report, University of Massachusetts Amherst.

A. Appendix

A.1. A Script for Creating Raw Score for an Eight-Item Dichotomous Test

File name: irtDataSimulation.sas

```
/*
irtDataSimulation creates raw scores of a binary eight-item test.
Copyright (C) 2019  Bo Klauth

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
*/
******************************;
* SIMULATE 4PL IRT DATA;
* Step 1: Run the data step below;

data b0 ;
input id item1-item8 freq;
datalines;
1 0 0 0 0 0 0 0 1 10
2 0 0 0 0 0 0 1 0 3
3 0 0 0 0 0 0 1 1 1
4 0 0 0 0 0 1 0 0 6
5 0 0 0 0 0 1 0 1 3
6 0 0 0 0 0 1 1 1 1
7 0 0 0 0 1 0 0 1 2
8 0 0 0 0 1 0 1 1 1
9 0 0 0 1 0 0 0 0 13
10 0 0 0 1 0 0 0 1 7
11 0 0 0 1 0 0 1 0 3
12 0 0 0 1 0 0 1 1 1
13 0 0 0 1 0 1 0 0 5
14 0 0 0 1 0 1 0 1 9
15 0 0 0 1 0 1 1 0 1
16 0 0 0 1 0 1 1 1 3
17 0 0 0 1 1 0 0 0 1
18 0 0 0 1 1 0 1 0 1
```

```
19 0 0 0 1 1 1 0 0 1
20 0 0 0 1 1 1 1 1 3
21 0 0 1 0 0 0 0 0 2
22 0 0 1 0 0 0 0 1 2
23 0 0 1 0 0 0 1 1 3
24 0 0 1 0 0 1 0 1 3
25 0 0 1 0 0 1 1 0 1
26 0 0 1 0 0 1 1 1 1
27 0 0 1 0 1 0 0 0 1
28 0 0 1 1 0 0 0 1 10
29 0 0 1 1 0 0 1 0 3
30 0 0 1 1 0 0 1 1 6
31 0 0 1 1 0 1 0 0 2
32 0 0 1 1 0 1 0 1 1
33 0 0 1 1 0 1 1 0 3
34 0 0 1 1 0 1 1 1 6
35 0 0 1 1 1 1 0 1 1
36 0 0 1 1 1 1 1 1 6
37 0 1 0 0 0 0 0 0 5
38 0 1 0 0 0 0 0 1 5
39 0 1 0 0 0 0 1 0 1
40 0 1 0 0 0 0 1 1 2
41 0 1 0 0 0 1 0 0 5
42 0 1 0 0 0 1 0 1 1
43 0 1 0 0 0 1 1 0 3
44 0 1 0 0 0 1 1 1 2
45 0 1 0 0 1 0 0 0 1
46 0 1 0 0 1 0 0 1 1
47 0 1 0 0 1 0 1 0 1
48 0 1 0 0 1 1 1 0 1
49 0 1 0 0 1 1 1 1 3
50 0 1 0 1 0 0 0 0 4
51 0 1 0 1 0 0 0 1 11
52 0 1 0 1 0 0 1 0 1
53 0 1 0 1 0 0 1 1 4
54 0 1 0 1 0 1 0 0 2
55 0 1 0 1 0 1 0 1 10
56 0 1 0 1 0 1 1 0 1
57 0 1 0 1 0 1 1 1 11
58 0 1 0 1 1 0 0 0 1
59 0 1 0 1 1 0 0 1 2
60 0 1 0 1 1 0 1 0 1
61 0 1 0 1 1 0 1 1 3
62 0 1 0 1 1 1 0 0 1
63 0 1 0 1 1 1 0 1 3
64 0 1 0 1 1 1 1 0 4
65 0 1 0 1 1 1 1 1 8
```

66	0	1	1	0	0	0	0	0	1
67	0	1	1	0	0	0	0	1	1
68	0	1	1	0	0	0	1	0	1
69	0	1	1	0	0	1	0	0	1
70	0	1	1	0	0	1	1	1	4
71	0	1	1	0	1	0	0	0	1
72	0	1	1	0	1	0	1	1	1
73	0	1	1	0	1	1	0	0	1
74	0	1	1	0	1	1	0	1	2
75	0	1	1	0	1	1	1	1	5
76	0	1	1	1	0	0	0	0	1
77	0	1	1	1	0	0	0	1	8
78	0	1	1	1	0	0	1	0	3
79	0	1	1	1	0	0	1	1	10
80	0	1	1	1	0	1	0	1	11
81	0	1	1	1	0	1	1	0	3
82	0	1	1	1	0	1	1	1	29
83	0	1	1	1	1	0	0	1	2
84	0	1	1	1	1	0	1	0	1
85	0	1	1	1	1	0	1	1	3
86	0	1	1	1	1	1	0	0	1
87	0	1	1	1	1	1	0	1	4
88	0	1	1	1	1	1	1	0	1
89	0	1	1	1	1	1	1	1	36
90	1	0	0	0	0	0	0	0	1
91	1	0	0	0	0	1	1	1	1
92	1	0	0	0	1	0	0	1	1
93	1	0	0	0	1	0	1	0	2
94	1	0	0	0	1	1	0	0	1
95	1	0	0	0	1	1	0	1	1
96	1	0	0	0	1	1	1	0	1
97	1	0	0	1	0	0	0	0	1
98	1	0	0	1	0	0	0	1	1
99	1	0	0	1	0	0	1	0	2
100	1	0	0	1	0	0	1	1	1
101	1	0	0	1	0	1	0	0	1
102	1	0	0	1	0	1	1	0	2
103	1	0	0	1	0	1	1	1	1
104	1	0	0	1	1	0	0	0	1
105	1	0	0	1	1	0	0	1	2
106	1	0	0	1	1	1	1	0	1
107	1	0	0	1	1	1	1	1	2
108	1	0	1	0	0	0	1	0	2
109	1	0	1	0	0	1	0	0	1
110	1	0	1	0	1	1	0	0	1
111	1	0	1	0	1	1	0	1	1
112	1	0	1	0	1	1	1	1	3

```
113 1 0 1 1 0 0 0 0 2
114 1 0 1 1 0 0 1 0 1
115 1 0 1 1 0 0 1 1 1
116 1 0 1 1 0 1 0 1 2
117 1 0 1 1 0 1 1 0 1
118 1 0 1 1 0 1 1 1 6
119 1 0 1 1 1 0 0 1 1
120 1 0 1 1 1 0 1 0 1
121 1 0 1 1 1 1 0 0 1
122 1 0 1 1 1 1 0 1 3
123 1 0 1 1 1 1 1 0 2
124 1 0 1 1 1 1 1 1 4
125 1 1 0 0 0 0 1 1 1
126 1 1 0 0 0 1 0 1 3
127 1 1 0 0 0 1 1 0 3
128 1 1 0 0 1 0 0 0 2
129 1 1 0 0 1 0 0 1 1
130 1 1 0 0 1 1 0 0 1
131 1 1 0 0 1 1 0 1 4
132 1 1 0 0 1 1 1 0 2
133 1 1 0 0 1 1 1 1 3
134 1 1 0 1 0 0 0 1 3
135 1 1 0 1 0 0 1 1 5
136 1 1 0 1 0 1 0 0 1
137 1 1 0 1 0 1 0 1 10
138 1 1 0 1 0 1 1 0 4
139 1 1 0 1 0 1 1 1 19
140 1 1 0 1 1 0 0 1 5
141 1 1 0 1 1 0 1 0 2
142 1 1 0 1 1 0 1 1 4
143 1 1 0 1 1 1 0 0 7
144 1 1 0 1 1 1 0 1 11
145 1 1 0 1 1 1 1 0 8
146 1 1 0 1 1 1 1 1 41
147 1 1 1 0 0 0 0 1 3
148 1 1 1 0 0 0 1 1 2
149 1 1 1 0 0 1 1 0 3
150 1 1 1 0 0 1 1 1 5
151 1 1 1 0 1 0 0 0 2
152 1 1 1 0 1 0 1 1 1
153 1 1 1 0 1 1 0 1 4
154 1 1 1 0 1 1 1 0 3
155 1 1 1 0 1 1 1 1 8
156 1 1 1 1 0 0 0 0 1
157 1 1 1 1 0 0 0 1 1
158 1 1 1 1 0 0 1 0 2
159 1 1 1 1 0 0 1 1 6
```

```
160 1 1 1 1 0 1 0 1 12
161 1 1 1 1 0 1 1 0 9
162 1 1 1 1 0 1 1 1 39
163 1 1 1 1 1 0 0 0 3
164 1 1 1 1 1 0 0 1 3
165 1 1 1 1 1 0 1 0 1
166 1 1 1 1 1 0 1 1 9
167 1 1 1 1 1 1 0 0 1
168 1 1 1 1 1 1 0 1 35
169 1 1 1 1 1 1 1 0 14
170 1 1 1 1 1 1 1 1 306
run;

* Step 2: Run the macro code below ;

%Macro sim();
proc sql;
select max(freq)as score_pattern_maxfreq into: nmax
from b0;
run;

%do i = 0 %to &nmax;
%let j = %eval(&i+1);
data b&j;
set b&i;
freq=freq-1;
if freq=<0 then delete;
run;
proc append base=b0 data=b&j;
run;
%end;
* create a new dataset from oldone;
data sampleData2(drop=freq);
set b0;
id=_N_;
run;

%let max =&nmax+1;
%do z = 0 %to &max;
proc datasets lib=work nolist;
delete  b&z;
quit;
run;
%end;
%mend sim;
%sim();
```

A.2. A Script for the irtTheta Macro

File name: irtThetaMacro.sas

```
/*
irtTheta creates a graph for dichotomous IRT test information
and standard error of estimate (SEE) by person location (theta);
and a graph for log likelihood function by score pattern.
Copyright (C) 2019  Bo Klauth

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
*/
*******************************;
/*
ds= enter your dataset name
model= enter your model the models available are onep, twop, threep, and fourp.
start_var= the name of your first item
end_var=the name of your last item
nitems= the total number of items
method=ML
ods_graphics= off or on. "on" will produce item characteristic curves;
*/

*Exmaple;
/*%Theta(ds=SampleData2, model=twop, start_var=item1, end_var=item8, nitems=8,
method=ML, ods_graphics=off);*/
* Note that the method is always ML. If a different method is used,
theta locations obtained by the macro will be different from SAS.;


%macro irtTheta (ds, model, start_var, end_var, nitems, method, ods_graphics);

* data step to get the composite score;
data temp;
set &ds;
total = sum (of &start_var--&end_var);
new_id=_n_;
```

```
run;

%let dataset=temp;

PROC SQL;
select nobs INTO: nobs
from sashelp.vtable
where libname="WORK" and memname="TEMP"
;
quit;

* Using proc irt to get the output;
ods graphics &ods_graphics;
proc irt data=&dataset itemstat itemfit
scoremethod=&method out=theta outmodel=model_output
plots=(ICC(unpack) IIC(unpack) TIC);
var &start_var--&end_var;
model &start_var--&end_var
/resfunc=&model;
title1 "&model IRT";
run;
ods graphics off;

* Transforming data to get ceiling(d), guessing(c), intercept, and slope;
data want(keep=_SUBTYP_ _NAME_ _VAR1_ _EST_ _STDERR_);
set model_output;
run;
data ceiling (drop = _name_ rename=(_est_=d));
set want;
if _SUBTYP_="CEILING" & _VAR1_^="";
run;

data guessing (drop = _name_ rename=(_est_=c));
set want;
if _SUBTYP_="GUESSING" & _VAR1_^="";
run;

data intercept (drop = _name_ rename=(_est_=intercept));
set want;
if _SUBTYP_="INTERCEPT";
run;

data slope (drop=_name_ rename=(_est_=slope));
set want;
if _SUBTYP_="SLOPE" & _VAR1_^="";
run;
```

```
* Using the intercept and slope to find the item difficulty/item location;
data item_diff;
merge ceiling guessing intercept slope;
item_diff=-intercept/slope;
run;
data item_diff (drop=_subtyp_ _STDERR_);
set item_diff;
run;

* Printing all parameter estimates;
proc print data=item_diff;
label _var1_ ="Item" item_diff="p" slope="Slope" intercept="Intercept";
title "Guessing, Item Difficult and Item Discrimination Estimates for &model Model";
footnote '';
run;


* Transposing data;
proc transpose data=ceiling out=tceiling prefix=d;
var d;
run;

proc transpose data=guessing out=tguessing prefix=c;
var c;
run;

proc transpose data=slope out=tslope prefix=slope;
var slope;
run;

proc transpose data=item_diff out=titem_diff prefix=p;
var item_diff;
run;

data slope_p;
merge tceiling tguessing tslope titem_diff;
run;

*--------------------------------;
*This is to calculate total test information and SEE;
/*
Item information only concerns with the probability of getting each item correct
from -4 to +4. It does not involve score pattern.
And the thus, thus n = (range/increment of i) + 1= 801
*/
data Iteminfo_SEE (rename=(i=Theta));
set slope_p;
```

```
do i = -4 to 4 by 0.01;
array d[&nitems] d1-d&nitems;
array c[&nitems] c1-c&nitems;
array alpha[&nitems]slope1-slope&nitems;
array item_diff[&nitems] p1-p&nitems;
array pj[&nitems] pj1-pj&nitems;
array iteminfo[&nitems]iteminfo1-iteminfo&nitems;
do j=1 to &nitems;
pj[j]=c[j]+(d[j]-c[j])*1/(1+exp(-alpha[j]*(i-item_diff[j])));

* item info and SEE for onep to twop models;
%if &model=onep or &model=twop %then %do;
iteminfo[j]=(alpha[j]**2)*(pj[j])*(1-pj[j]);
%end;

*item info and see for threep model;
%if &model=threep %then %do;
iteminfo[j]=(alpha[j]**2)*((pj[j]-c[j])**2)*(1-pj[j])/((1-c[j])**2)*(pj[j]);
%end;

*item info and see for fourp model;
%if &model=fourp %then %do;
iteminfo[j]=((alpha[j]**2)*((pj[j]-c[j])**2)*(d[j]-pj[j])**2)/
(((d[j]-c[j])**2)*(pj[j])*(1-pj[j]));
%end;

iteminfototal=sum( of iteminfo1-iteminfo&nitems);
SEE=sqrt(1/iteminfototal);
drop j;
end;
output;
end;

*--------------------------------;

*build Graph SEE and test info;
axis1 label=(a=90 h=2 'Test Information');
axis2 label=(h=2 'Theta');
Axis3 label=(a=90 h=2 'SEE');

symbol1 w = 2 v=none color=blue interpol=join;
symbol2 w = 2 v=none color=red interpol=join;
legend1 label =(height=1.5 'Test Information') value =('');
legend2 label=(height=1.5 'SEE') value=('');

* graphing item information by SEE;
proc gplot data=Iteminfo_SEE;
```

```
title "Test Information and SEE Plot for &model Model";
plot1 iteminfototal*Theta/hminor=1 vaxis=axis1 haxis=axis2 legend=legend1;
plot2 SEE*Theta /vaxis=axis3 legend=legend2;
run;
quit;


* preparing data steps for calcualting irt score;
proc means data=theta mean std noprint;
var total;
output out= z;
run;


data z(keep=_stat_ total);
set z;
if _stat_="N" or _stat_="MIN" or _stat_="MAX" then delete;
run;

proc transpose data=z out=z_tr;
id _stat_;
run;


data z_tr_obs (drop=_name_ i);
set z_tr;
do i = 1 to &nobs;
output;
end;
run;

data IRT_score;
merge theta z_tr_obs;
run;
* calculating IRT score;
data IRTscore;
set IRT_score;
IRT_score=MEAN + _Factor1 * STD;
new_id = _n_;
run;

*Obtaining slopes and item difficulties and prepare them for all the observations;

data itemdiff_slope (keep=d1-d&nitems c1-c&nitems slope1-slope&nitems p1-p&nitems);
set iteminfo_see;
new_id=_n_;
if new_id=1;
```

```
do i=1 to &nobs;
output;
end;
run;
* Merge tables for items and slopes and item difficulties togeter;
data wantxxx;
merge &dataset itemdiff_slope;
run;
* CALCULATING PERSON PROBABILITY OF EACH ITEM;
/*
Calculating probability of each item for every observation from person locations
ranged from -i to +i;
* So each probability (pj1 to pjn) is the probability of getting an item correct;
* This is to prepare for calculating the pattern probability
or the log-likelihood function.
*/
data wantxx (rename=(i=Theta));
set wantxxx;
do i = -4 to 4 by 0.01;
array dj[&nitems] d1-d&nitems;
array cj[&nitems] c1-c&nitems;
array slope[&nitems]slope1-slope&nitems;/*alpha*/
array p[&nitems]p1-p&nitems;
array pj[&nitems]pj1-pj&nitems;
do j = 1 to &nitems;
/*
adding dj cj in the pj calculation to make a generic calculation
for all four models
*/
pj[j]=cj[j]+(dj[j]-cj[j])*1/(1+exp(-slope[j]*(i-p[j])));
drop j;
end;
output;
end;
run;

* CALCULATING PROBABILITY OF A SCORE PATTERN OR LLF OF A SCORE PATTERN;
data wantx;
set wantxx;
/*array for test items*/
array items[&nitems]&start_var--&end_var;

/* array for probability of getting an item correct*/
array pj[&nitems]pj1-pj&nitems;

/* array for probability of item according the actual response*/
array phi[&nitems] phi1-phi&nitems;
```

```
/*arrya for log likelihood of each item based on the actual response*/
array Ln[&nitems] Ln1-Ln&nitems;
do K= 1 to &nitems;

if items[K]=1 then phi[k]=pj[K];
if items[K]=0 then phi[k]=1-pj[K];

/* Total probabilty for an examinee = multiplication of probability of each item,
which is equivalent to using geomean*/
total_phi=geomean( of phi1-phi&nitems)**n(of phi1-phi&nitems);
Ln[K]=log(phi[K]);
Total_Ln=sum (of Ln1-Ln&nitems);
/*checking phi (probability) and log calculation*/
Exp=exp(total_Ln);
drop k;
end;
run;

*Creating score pattern;
data wantx1;
set wantx;
length score_pattern $&nitems;
array score &start_var--&end_var;
do over score;
if not missing(score) then score_pattern=cats(score_pattern, score);
else score_pattern=catt(score_pattern, score);
pattern_length=length(score_pattern); /*checking the length of the pattern*/
end;
run;

data AcrossContinuum_Parameters;
set wantx1;
run;

* Creating frequency for score pattern;
*-----------;

*checking frequency of total score and response pattern;
* Converting simulating frequency to dataset frequency;
proc freq data=Acrosscontinuum_parameters noprint;
Tables total /out =simTotalScore;
run;

proc freq data=Acrosscontinuum_parameters noprint;
Tables score_pattern /out =simScorePattern;
run;
```

```
data OriTotalScore;
set simTotalScore;
Frequency=COUNT/((8/0.01)+1);
label total="Total Score";
run;

data OriScorePattern;
set simScorePattern;
Frequency=COUNT/((8/0.01)+1);
label score_pattern="Score Pattern";
run;

proc print data=oriTotalScore;
var total Frequency percent;
label total="Total Score";
title "Frequency and Percentage of Total Composite Score";
run;

proc print data=oriScorePattern;
var score_pattern Frequency percent;
title "Frequency and Percentage of Score Pattern";
run;


*creating data for log likelihood function graphs;
proc sql;
create table graph_loginfo as
select distinct(score_pattern),total, theta, total_ln
from wantx1;
quit;

proc sort data=graph_loginfo;
by total;
run;

*build log likelihood function (LLF) and score pattern;

axis1 label=(a=90 h=2 'Log Likelihood');
axis2 label=(h=2 'Theta');

symbol1 color=black interpol=join r=0;
symbol2 color=blue interpol=join r=0;
symbol3 color=green interpol=join r=0;
symbol4 color=brown interpol=join r=0;
symbol5 color=gray interpol=join r=0;
symbol6 color=green interpol=join r=0;
```

```
symbol7 color=red interpol=join r=0;
symbol8 color=blue interpol=join r=0;
symbol9 color=gray interpol=join r=0;
symbol10 color=black interpol=join r=0;
symbol11 color=red interpol=join r=0;
symbol12 color=red interpol=join r=5;
symbol13 color=orange interpol=join r=3;
symbol14 color=pink interpol=join r=2;
symbol15 color=red interpol=join r=3;
symbol16 color=green interpol=join r=13;
symbol17 color=blue interpol=join r=13;
symbol18 color=blue interpol=join r=5;
symbol19 color=red interpol=join r=13;
symbol20 color=black interpol=join r=5;
symbol21 color=blue interpol=join r=10;
symbol22 color=black interpol=join r=1000;



proc gplot data=graph_loginfo;
plot Total_Ln*Theta=score_pattern/ vaxis=axis1 haxis= -4 to 4;
by total;
/*where total^=0;*/
title "Log Likelihood Function for Total Score Using &model Model";
run;
quit;

*-----------;

* Sort the data set by id;
proc sort data=wantx1;
by new_id;
run;


* Finding max of log likelihood;
proc sort data=wantx1;
by new_id Total_Ln;
run;

* Select only records with maximum likelihood;
proc sql;
create table max_ln as
select *,max(total_ln) as max_tln
from wantx1
group by new_id
having max(total_ln)=total_ln;
```

```
quit;

* working on here;
proc sort data=irt_score;
by new_id;
run;

data _Factor1(keep=_Factor1);
set irt_score;
run;

Data Dataset_Parameters;
merge max_ln _Factor1;
run;

Data Comparing_theta(keep=new_id total theta _Factor1 total_ln total_phi exp);
set Dataset_Parameters;
run;

* Printing 45 observations to compare the estimates of the macro and SAS;
proc print data=Comparing_theta (obs=45);
title "Comparing Theta for &model  (printed 45 observations)";
var new_id total theta _factor1;
run;

* deleting unused datasets;
proc datasets lib=work nolist;
delete  want wantx wantx1;
/*                delete graph_loginfo;*/
delete wantxx;
delete wantxxx;
delete z z_tr z_tr_obs;
delete guessing intercept;
delete item_diff;
delete slope;
delete slope_p;
delete itemdiff_slope;
delete tguessing titem_diff;
delete tslope ceiling tceiling;
delete max_ln Temp _Factor1 Theta irt_score;
quit;
run;

%Mend IRTTheta;
```

Affiliation:

Bo Klauth
Department of Educational Leadership, Technology and Research
Western Michigan University
1903 W Michigan Ave Kalamazoo, MI 49008
E-mail: bklauth@gmail.com

Western Michigan University
College of Education and Human Development
Department of Educational Leadership, Research and Techology

June 2019