

Software Corner:**RKward: IRT analyses and person scoring with ltm**

Aaron Olaf Batty
abatty@sfc.keio.ac.jp
Keio University
Lancaster University

In *SRB 16(2)*, I introduced the ever-improving, free and open-source RKward GUI for the free and open-source R statistical package (Batty, 2012). In this issue, I will detail how to conduct IRT analyses from RKward's drop-down interface using `ltm`, a flexible R package for unidimensional item response models. This allows users to get many of the important features of commercial software packages such as Xcalibur and IRTPro for free, and with a user-friendly interface.

Updates

Since that article, the RKward development team has been hard at work improving the package for Windows and Macintosh users, and the following changes have since been made to installation practices:

- The RKward project page has been much improved, and can be found at the following address:

<http://rkward.sourceforge.net/>

- On Windows, the installation process is now simply a matter of decompressing a .zip file and running the .exe application within the created folder. The folder can be saved in your Program Files or Program Files (x86) folder, with a shortcut on the Start menu/screen if you so desire, but it does not necessarily need to be. You can store it anywhere and run it from anywhere on your hard disk.
- On the Mac, the installation method is to first install the standard R package, then install the RKward GUI. When downloading the RKward package, make note of the file name, as it specifies what version of R you should install. Previously, RKward installed its own instance of R, but this resulted in a bloated installation that was difficult to maintain. R can be downloaded from the following mirror in Japan:

<http://cran.md.tsukuba.ac.jp/>

In the year since I first introduced RKward in these pages, it has become my main statistical package. It features SPSS' ease of use and the power, flexibility, and price of R. Despite the fact that I have a personal copy of SPSS, and site licenses through both of my institutions, IBM's licensing controls are simply too onerous to justify using it when there is a free alternative.

In this issue, I will explain how to use the `ltm` IRT package via RKward to fit Rasch and other IRT models; plot item characteristic curves; check item, person, and model-data fits; compare models for selection; and output person ability estimates.

Notes:

- *I will be demonstrating with the Macintosh version of RKward, but the Windows version is identical, since the software itself runs in a KDE environment. The only real difference is that the*

menubar for the Windows version is in the RKWard window, whereas on the Mac it is at the top of the screen.

- *Sometimes it is necessary to click the “Refresh Output” button on the output display for the results of an analysis to appear.*
- *This is not intended to be a comprehensive guide to `ltm`. Much more information is available in the package reference manual (Rizopoulos, 2013).*

The `ltm` package

The `ltm`—which stands for “latent trait models”—package is a set of functions for R, developed by Dimitris Rizopoulos (Rizopoulos, 2006). The package offers a wide array of testing and IRT functions, including 1PL, 2PL and 3PL dichotomous models, Rasch and Generalized Partial Credit Models and the Linear Logistic Test Model, which is similar to Linacre’s many-facet Rasch model. Many of these models can be run via RKWard’s GUI, and parameters can further be constrained to fixed values specified by the user via the command line. For a full overview of functions, see the project page at the URL below, or in the reference manual (Rizopoulos, 2013):

<http://rwiki.sciviews.org/doku.php?id=packages:cran:ltm>

The `ltm` package is not installed by default with RKWard; however, the first time you attempt to use it, RKWard will walk you through choosing a download mirror (there are three in Japan), after which you merely have to click “Apply” and then “OK.” On the first use, or the first use with a workspace, some messages will display at the bottom of the output telling you that it has been installed and/or loaded for the analysis.

Preparing your data

As explained in my last column, data need to be imported to RKWard in either delimited text or SPSS `.sav` format. Your data for IRT analyses must conform to the following:

- Numeric responses, either binary for dichotomous datasets, or integer data for polytomous.
- No person IDs. Item responses only.
- (Optional) Item names in the header row, which can be imported as variable names.

As a general rule, when importing datasets to RKWard, I prefer to name them “[descriptive name].data”, e.g., “Listening.data”, rather than keeping the default “my.csv.data”, etc., that RKWard enters automatically. Data tables can also be renamed in the Workspace pane by right-clicking and selecting “Rename.”

Fitting IRT models

A wide array of IRT models, both dichotomous and polytomous, can be fitted; the process for each, however, is the same.

1. Go to the following menu entry:

Analysis → Item Response Theory → Dichotomous data → *[choose the model you want]*

2. A window such as that shown in Figure 1 appears. Select your test data from the list on the left and click the green triangle to select it for analysis.

3. (Optional, but recommended) Change the name of the output at the bottom to reflect the dataset. This is especially important if you have multiple datasets in your workspace. See Figure 1 for an example.
4. Click “Submit.”

The parameter estimates will be displayed in a table in the output. They can then be exported or copy/pasted into another program (e.g., Excel) for saving. In addition, an item estimates object will be added to the workspace pane, which will be used for further analyses, some of which will be explained later in this article.

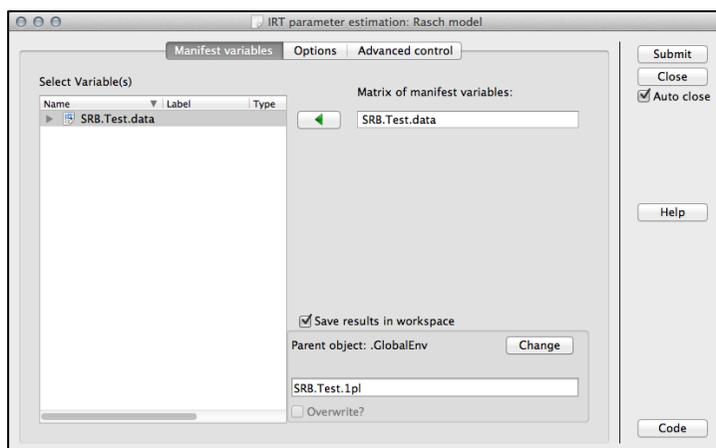


Figure 1. Selecting the data for IRT model fitting and defining the output name

Fitting a traditional Rasch model

It is important to note that the default Rasch model fitted by `1tm` is perhaps better described as a one-parameter IRT (1PL) model, as the discrimination parameter (or “slope”) is estimated separately for each data set, rather than assumed to be 1 as it is in programs such as `Winsteps` (Linacre, 2012). Such a model retains the item invariance that is the main “selling point” of the traditional Rasch model, but also allows better fit with test data which have higher or lower average item discrimination, which can allow for a more realistic picture of the test information function (test reliability by ability levels). For a graphical representation of the difference, see Figure 2.

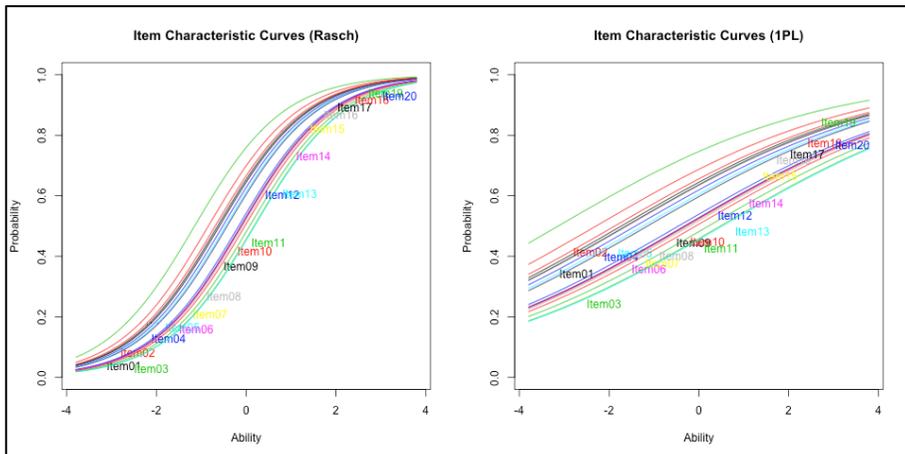


Figure 2. Comparison of Rasch and 1PL item characteristic curves of the same data

If you would prefer to fit a constrained Rasch model, there are two extra steps.

Preparing a constraint matrix

To constrain the model with a discrimination parameter of 1, an initial step is necessary (see Figure 3).

1. Click the “R Console” button in the lower left of the Rkward window.
2. Enter the following R command:

```
[Some descriptive name] <- cbind([number of items in your test, plus one], 1)
e.g., for a twenty-item test:
TwentyItemConstraint <- cbind(21, 1)
```

3. Hit enter/return.

This will create a new object in your Workspace pane (click the vertical “Workspace” button in the upper left-hand side of the Rkward window to view it) with whatever name you’ve assigned it (“TwentyItemConstraint” in my example), which is a matrix for setting the discrimination estimate to 1.

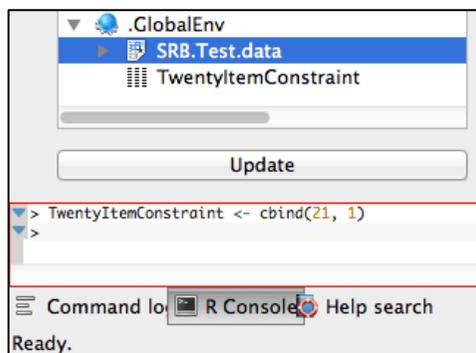


Figure 3. R command for creating a constraint matrix, and its display in the Workspace pane

Fitting the Rasch model

With the constraint matrix created, fitting the Rasch model is simple:

1. Go to the following menu entry:
 Analysis → Item Response Theory → Dichotomous data → Rasch model fit
2. Follow steps 2 and 3 from the “Fitting IRT Models” section above.
3. Click the “Options” tab. The window shown in Figure 4 will appear.
4. Select your constraint matrix and click the green triangle as shown in Figure 4.
5. Click “Submit.”

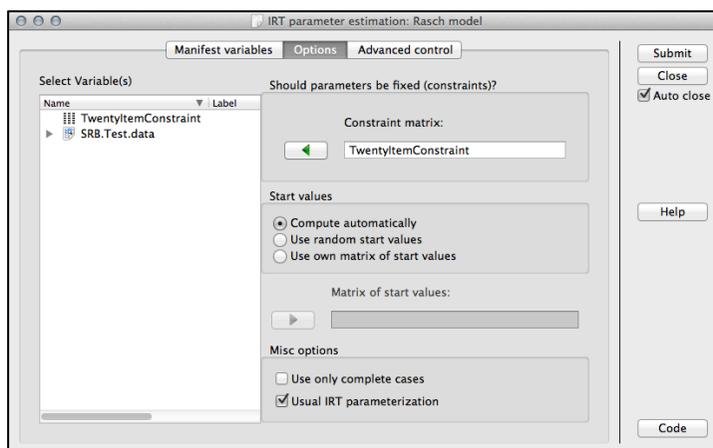


Figure 4. Selecting the constraint matrix to constraint discrimination estimates to 1.

Plotting IRT item characteristic curves (ICCs)

One of the most intuitive methods of interpreting item behavior is by viewing ICC graphs (see Figure 2 for an example). Ltm can create these *once you have already estimated a model and the estimate output object is available in your workspace pane*.

ICC graph settings are accessible from the following menu:

Plots → Item Response Theory → *[choose the model you want]*

Once again, dichotomous and polytomous data each have their own submenu. Choose the appropriate plot for the model you have already estimated and follow the instructions below:

1. Select the appropriate ICC plot type from the submenus. The window shown in Figure 5 opens.
2. Select the estimates object created when you estimated your IRT model from the list on the left and click the green triangle.
3. (Optional) Click the “Plot type” tab and select the kind of plot you want. You can also specify subsets of items here. The default is to create a graph with all the ICCs displayed, but other options are available as well.
4. (Optional) Further options are available in the “Options” tab, especially by pressing the “Generic plot options” button, where you can define the title, subtitle, and other such plot characteristics.
5. Click “Submit.”

Your selected plot will display in the output. Consult my previous article for instructions on how to export these.

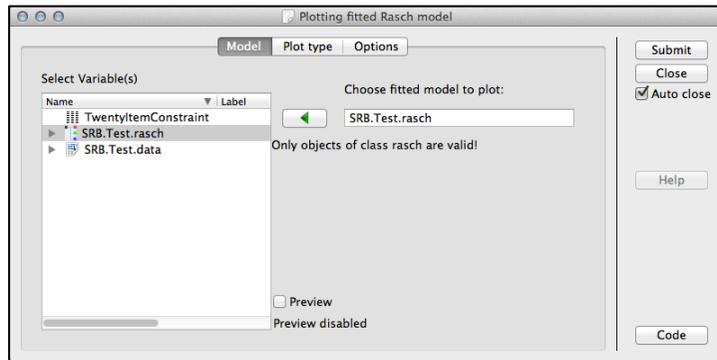


Figure 5. Selecting the model output for plotting.

Checking and comparing fit

The ltm package provides a wide variety of fit statistics, both for items and persons, and for entire models. For items, persons, and Rasch models, fit can be determined via chi-square tests. It is also possible to compare fits between two nested models—for example, a 2PL model versus a Rasch model, in order to determine if the additional parameter substantially improves fit.

Item and person fit

Item and person fits statistics are provided as chi-square statistics comparing the observed responses to those predicted by the model. These tests are available from the following menu:

Analysis → Item Response Theory → Tests → [select either item-fit or person-fit statistics]

The process is as follows:

1. Select the estimate output from your model estimation (e.g., “SRB.Test.rasch”) and click the button with the green triangle.
2. Click “Submit.”

Fit statistics are provided as a table in the output, listing item names, chi-square statistics, and p-values. A significant chi-square statistic indicates that the item or person responses differ significantly from the model-expected responses, i.e., the item or person *misfits* the model.

Model-data fit for a Rasch/1PL model estimation

For one-parameter models, a chi-square test for the fit of the entire model is available from the following menu:

Analysis → Item Response Theory → Tests → Goodness of fit (Rasch)

The process is the same as the other analyses discussed in this article. The estimation output object is selected, and the button with the green triangle is clicked, followed by the “Submit” button.

NOTE: It may take a long time to complete the analysis. Be sure to look at the R engine status indicator in the lower right of the RKWard window. If it is red, the engine is still working.

The result of the chi-square test is displayed in the output. Once again, a significant p-value indicates that the model-expected and observed values are significantly different, hence the fit is *poor*. A non-significant chi-square indicates good model-data fit.

Comparing model-data fit

When choosing an IRT model, it is important to choose the most parsimonious model that fits the data (Kang & Cohen, 2007). Item response models with additional parameters will almost always have an at least negligibly better fit than models with fewer parameters, but the more complex model may simply overfit the data and provide information that will not be generalizable to another sample. Luckily, `ltm` offers two methods of model comparison to aid in selection.

Rasch/1PL models

If you want to compare two one-parameter (i.e., Rasch or 1PL) models to determine which has superior fit with the data, the easiest method is to simply run the Rasch goodness of fit test discussed above for each and compare the chi-square fit statistics.

Anova command for use with other models

`ltm` includes a convenient command for comparing the fits of two nested models, but it does not, unfortunately, have a graphical interface in RKWard. Luckily, though, RKWard allows the user to use a standard R console as well. To use the `ltm` `anova` command, do the following:

1. Click the “R Console” button on the bottom left of the RKWard window (see Figure 6).
2. Type the following at the R prompt:

```
anova([the output object for the simpler model], [the object for the more complex model])
```

e.g., to compare a Rasch model to a 2PL model (see Figure 6):

```
anova(SRB.Test.rasch, SRB.Test.2pl)
```

3. Hit enter/return.

The result is displayed in the console (see Figure 6). It will also eventually appear in the output pane when another command is issued through the GUI.

The `anova` command compares the first object, the null, to the second, the alternative, using a number of statistics for nested models, including the Akaike Information Criterion (AIC) the Bayesian Information Criterion (BIC) and a Likelihood Ratio Test (For more information on model fit comparison, see Kang & Cohen, 2007.). In contrast to the chi-square statistics discussed above a significant p-value for the likelihood ratio test indicates that the alternative model fit is *superior* to that of the more parsimonious model.

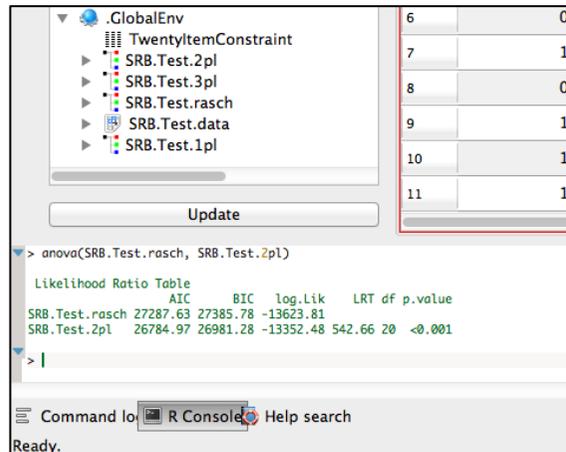


Figure 6. *l*tm anova command and output in the R Console.

Estimating and exporting person ability estimates

*l*tm includes several methods of estimating person ability (see Rizopoulos, 2006, 2013 for details), but they are all accessed with the `factor.score` command. In this section I will use the default empirical Bayes estimation method. Unfortunately, there is no GUI for person scoring in RKWard as of this writing.

Generating ability estimates

To estimate ability scores, do the following:

- In the R Console at the bottom of the RKWard window, type the following (see Figure 7):


```
[name of the person score object you want] <- factor.scores([name of the model estimation output you want to score])
```

 e.g.:


```
RaschScores <- factor.scores(SRB.Test.rasch)
```
- A new object is added to the workspace pane (called “RaschScores” in this example). Click the expansion triangle next to it and locate the data table inside called “score.dat”.
- Double-click `score.dat` to open it in the main RKWard pane. The person measures and their SEs are listed in the last two columns of data, labeled “z1” and “se.z1”. See Figure 7 for an example.

	21	22	23	24
Name	Obs	Exp	z1	se.z1
1	1	0.0026003...	-1.161681...	0.4406188...
2	1	0.0017028...	-0.972560...	0.4297000...
3	1	0.0036762...	-0.972560...	0.4297000...
4	1	0.0006891...	-0.443653...	0.4138387...
5	1	0.0001811...	-0.443653...	0.4138387...
6	1	0.0046583...	-1.361997...	0.4551329...
7	1	0.0032965...	-1.161681...	0.4406188...
8	1	0.0004490...	-0.972560...	0.4297000...
9	1	0.0002734...	0.0727773...	0.4194810...

Figure 7. `factor.score` command, output object, and score data.

Exporting ability estimates

Although I discussed exporting *outputs* in my last RKWard article, to be able to use the ability measures elsewhere (e.g., in Excel), they need to be exported. To export tabular data to a file:

1. Go to the following menu:
File → Export → Export tabular data
2. The window shown in Figure 8 opens. Select the `score.dat` table and click the button with the green triangle.
3. Under “Name of the file,” click the folder button and navigate to where you would like to save the file, give it a name, and click “OK.”
4. (Optional) Because the first column simply contains the case numbers with no heading, the header row will be off by one column if you export with the default settings. To remedy this, click on the “Columns and Rows” tab and select “No names” under “Name of rows” in the middle of the window.
5. Click “Submit.”

The file that is created is simply a tab-delimited text file that can be opened in Excel, etc. It may be necessary to start your chosen software first and direct it to open the file, or to right-click the file and select which application to use. The delimiter can be changed (e.g., to a comma) under the “Options” tab before clicking “Submit.”

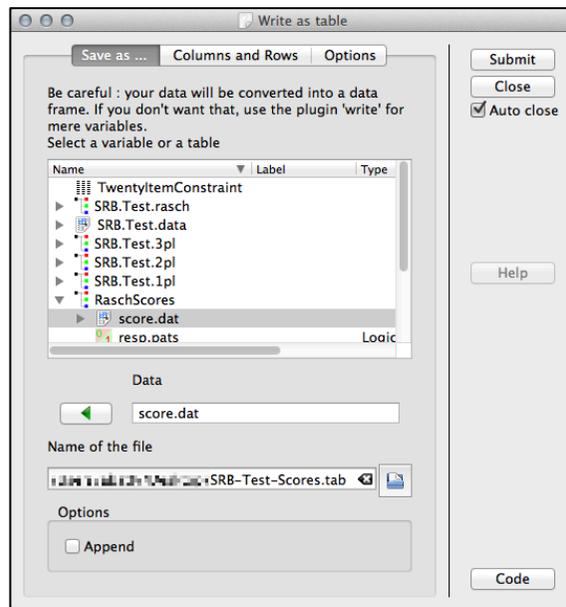


Figure 8. Export dialog window for tabular data

Conclusion

I have only introduced some of the basic functions available in ltm in RKWard, and even more functionality is available on the command line. Although commercial IRT software is often easier to use, and sometimes creates more versatile plots and outputs, you may find that ltm, especially when paired with the RKWard GUI, offers features that meet your testing or research needs at no cost whatsoever.

References

- Batty, A. O. (2012). Software corner: RKWard: Installation and use. *Shiken Research Bulletin*, 16(2), 21–29. Retrieved from <http://teval.jalt.org/sites/teval.jalt.org/files/SRB-16-2-Batty-SoftCorner.pdf>
- Kang, T., & Cohen, A. S. (2007). IRT model selection methods for dichotomous items. *Applied Psychological Measurement*, 31(4), 331–358. doi:10.1177/0146621606292213
- Linacre, J. M. (2012). *Winsteps*®. Beaverton, Oregon: Winsteps.com. Retrieved from <http://www.winsteps.com/>
- Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, 17(5), 1–25. Retrieved from <http://www.jwalkonline.org/docs/Grad%20Classes/Spring%2008/modmeas/homework/ltm.pdf>
- Rizopoulos, D. (2013, February 15). ltm reference manual. Retrieved from <http://cran.r-project.org/web/packages/ltm/ltm.pdf>